PRESIDENCY UNIVERSITY

(Established under the Presidency University Act, 2013 of the Karnataka Act 41 of 2013)

Department of Computer Science & Engineering

Network Programming Lab

CSE 257

NAME: Sai Ram. K

ID NO: 20181CSE0621

SEC: 6-CSE-10

COURSE CODE: CSE 257

Academic Year 2020 - 2021

Module – 1

DOS Commands

1.

20181CSE0621

MODULE - 1

DOS COMMANDS

1] PING Command :- To check whether your internet connections work you can use command prompt to test your connection to a certain website or internet location. To check connectivity with GOOGLE we type "ping www.google.com".

- Type c:\> ping x.x.x.x .

By default ping sends 4 ICMP packets each of 32 bytes. The response packets are called ICMP echo reply packets.

- Type c:\> ping x.x.x.x -t

-t switch will continue to send packets to the destination until user stops by pressing ctrl+C.

```
Command Prompt

C:\Users\ram10>ping 192.168.0.103

Pinging 192.168.0.103 with 32 bytes of data:
Reply from 192.168.0.103: bytes=32 time<1ms TTL=128
Reply from 192.168.0.103: bytes=32 time<1ms TTL=128
Reply from 192.168.0.103: bytes=32 time<1ms TTL=128
Reply from 192.168.0.103: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.0.103:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\ram10>
```

2.

2) IPCONFIG Command : Displays full TCP/IP configuration of all network adapters installed in your system.

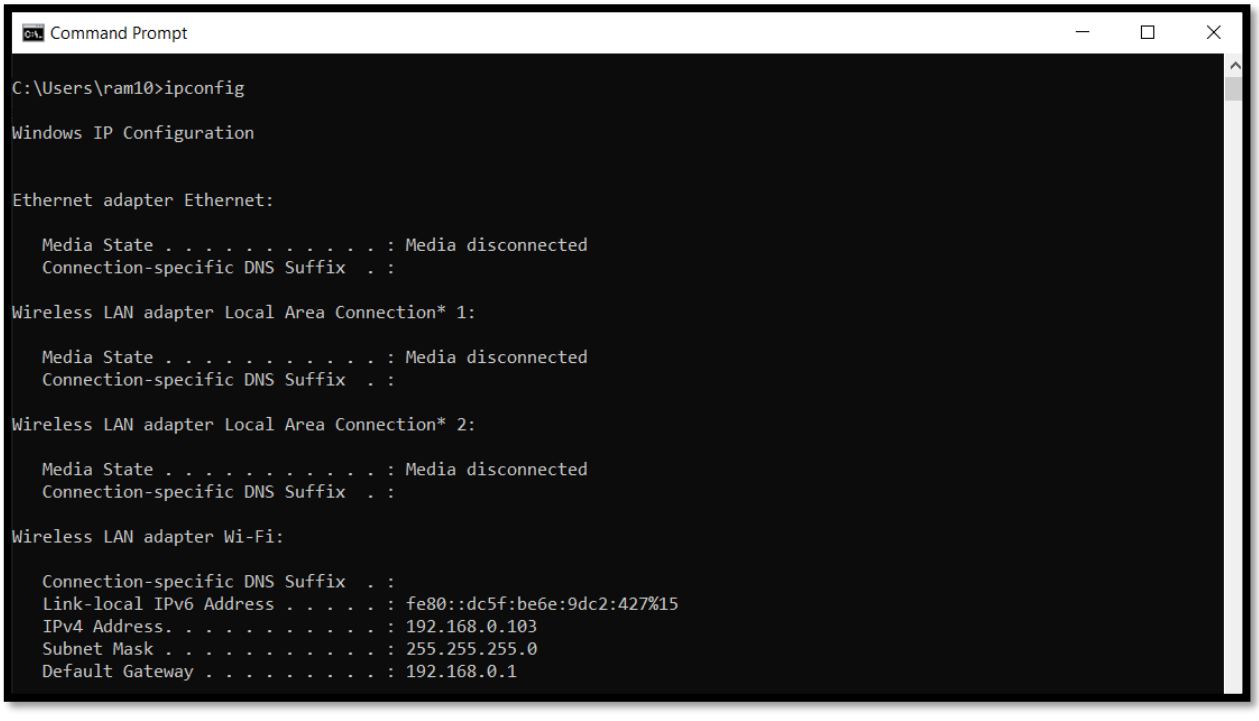
Type c:\>ipconfig

If you add /all switch to the command you can get a whole new level of details. The DNS, MAC and other information about each network component.

- Ipconfig has a number of switches, the most common:-
  - ipconfig /all : displays more information about the network setup on your system including the MAC address.

1

  - ipconfig /release : release the current IP address
  - ipconfig /renew : renew IP address
  - ipconfig /? : shows help
  - ipconfig /flushdns : flush the dns cache.



3.

3] Tracert Command :- It tells you the path a packet takes from your computer to the destination. It will list all the routers from which a packet passes until it reaches destination.

c:\> tracert google.com

```
C:\Users\ram10>tracert google.com

Tracing route to google.com [142.250.76.46]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  192.168.0.1
  2     2 ms     1 ms     2 ms  10.232.0.1
  3     2 ms     *        *     broadband.actcorp.in [202.83.20.43]
  4     2 ms     2 ms     3 ms  14.141.145.5.static-Bangalore.vsnl.net.in [14.141.145.5]
  5     9 ms    11 ms     9 ms  ^C
C:\Users\ram10>
C:\Users\ram10>
```

4.

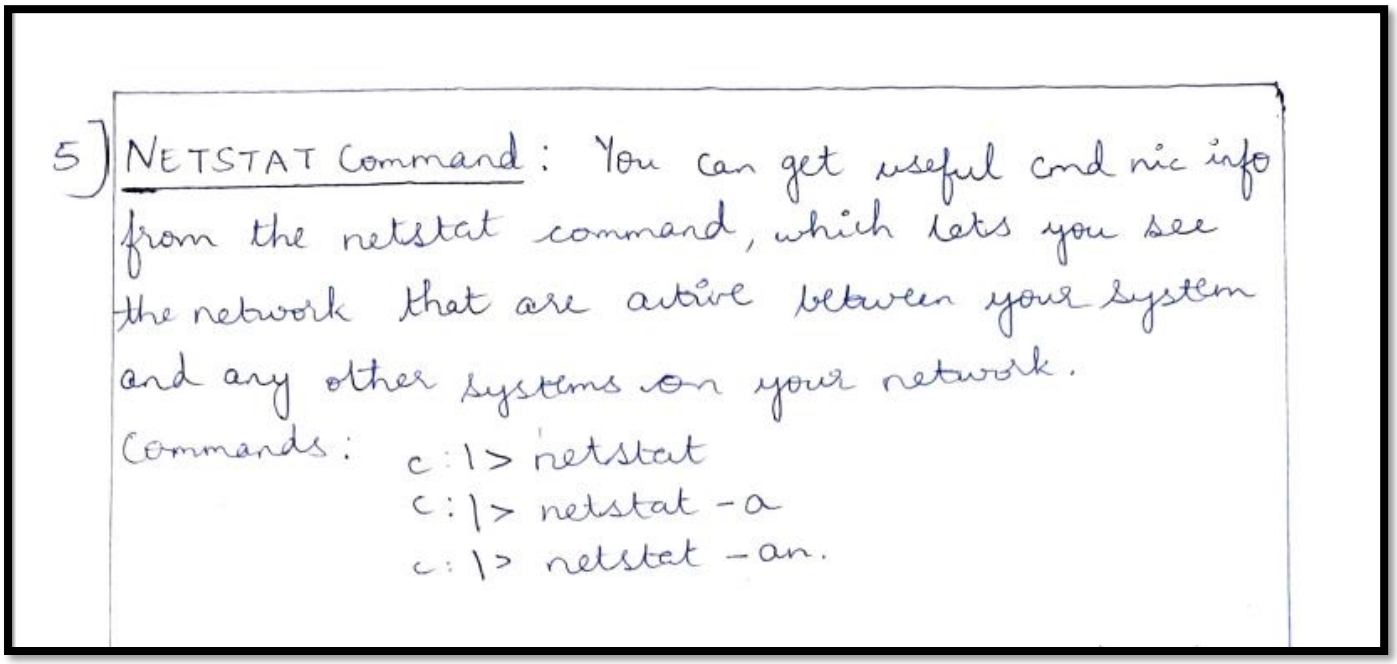
4] NS lookup command :

Displays the default DNS server information.

command :- c:\> nslookup


Command Prompt - nslookup
C:\Users\ram10>nslookup
Default Server: broadband.actcorp.in
Address: 49.205.72.130
>

5.


5) NETSTAT Command: You can get useful cmd nic info from the netstat command, which lets you see the network that are active between your system and any other systems on your network.
Commands:
c:\> netstat
c:\> netstat -a
c:\> netstat -an.

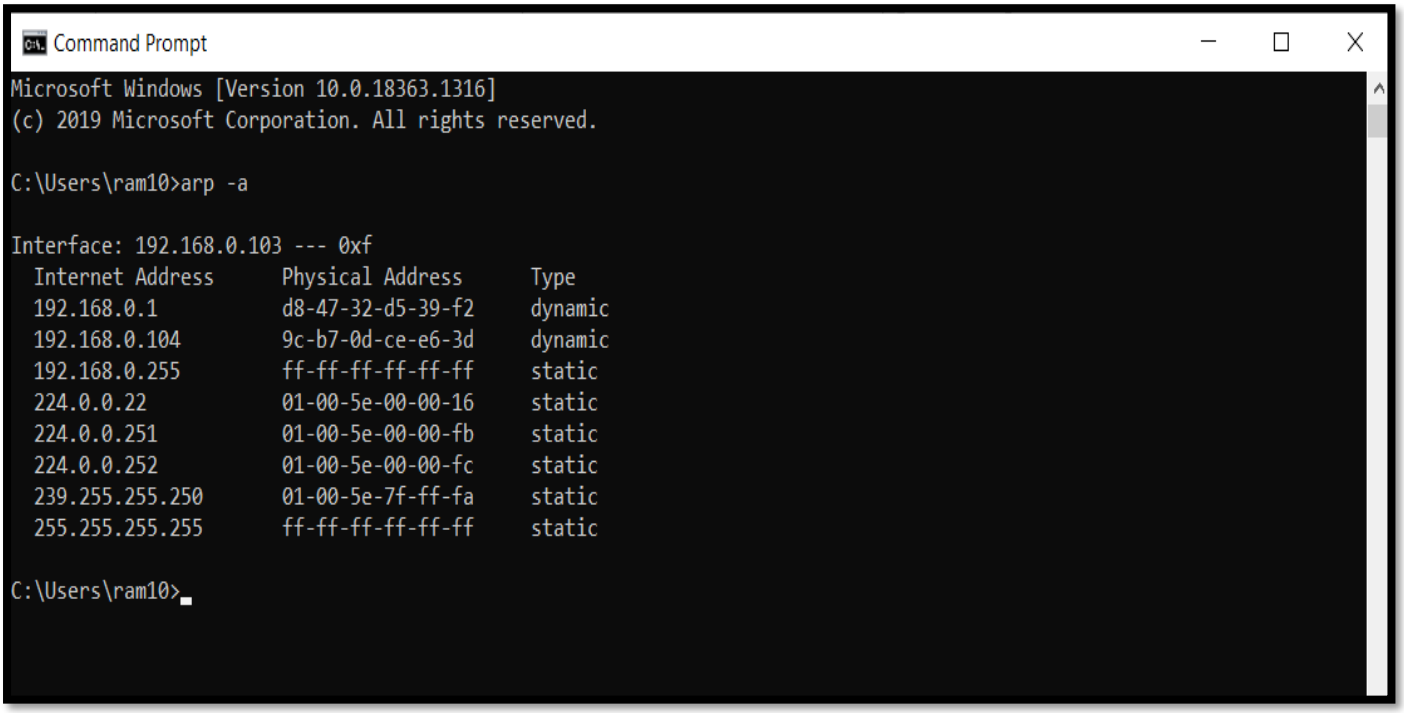
```
C:\Users\ram10>
C:\Users\ram10>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    192.168.0.103:49674    52.139.250.253:https   ESTABLISHED
  TCP    192.168.0.103:49694    broadband:https        ESTABLISHED
  TCP    192.168.0.103:49695    broadband:https        ESTABLISHED
  TCP    192.168.0.103:49696    broadband:https        ESTABLISHED
  TCP    192.168.0.103:49701    broadband:http         ESTABLISHED
  TCP    192.168.0.103:49702    broadband:http         CLOSE_WAIT
  TCP    192.168.0.103:49703    broadband:http         ESTABLISHED
  TCP    192.168.0.103:49704    broadband:http         ESTABLISHED
  TCP    192.168.0.103:49705    52.139.250.253:https   ESTABLISHED
  TCP    192.168.0.103:49709    broadband:http         ESTABLISHED
  TCP    192.168.0.103:49730    broadband:https        ESTABLISHED
  TCP    192.168.0.103:50223    broadband:https        ESTABLISHED
^C
C:\Users\ram10>
```

```
C:\Users\ram10>netstat -n

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    192.168.0.103:49674    52.139.250.253:443     ESTABLISHED
  TCP    192.168.0.103:49694    106.51.145.136:443     ESTABLISHED
  TCP    192.168.0.103:49695    106.51.145.136:443     ESTABLISHED
  TCP    192.168.0.103:49696    106.51.145.136:443     ESTABLISHED
  TCP    192.168.0.103:49701    106.51.146.30:80       ESTABLISHED
  TCP    192.168.0.103:49702    106.51.146.30:80       CLOSE_WAIT
  TCP    192.168.0.103:49703    106.51.146.30:80       ESTABLISHED
  TCP    192.168.0.103:49704    106.51.146.30:80       ESTABLISHED
  TCP    192.168.0.103:49705    52.139.250.253:443     ESTABLISHED
  TCP    192.168.0.103:49709    106.51.146.30:80       ESTABLISHED
  TCP    192.168.0.103:49730    106.51.145.136:443     ESTABLISHED
  TCP    192.168.0.103:50223    106.51.144.8:443       ESTABLISHED
  TCP    192.168.0.103:50305    52.114.14.231:443      ESTABLISHED
  TCP    192.168.0.103:50329    52.114.40.55:443       ESTABLISHED
  TCP    192.168.0.103:50362    52.111.252.0:443       ESTABLISHED
  TCP    192.168.0.103:50364    52.114.6.215:443       ESTABLISHED
  TCP    192.168.0.103:50375    52.114.133.60:443      ESTABLISHED
  TCP    192.168.0.103:50876    20.44.232.74:443       TIME_WAIT
  TCP    192.168.0.103:50880    20.44.232.74:443       ESTABLISHED
  TCP    192.168.0.103:50881    52.109.124.51:443      TIME_WAIT
  TCP    192.168.0.103:50882    52.114.32.111:443      ESTABLISHED
  TCP    192.168.0.103:50883    52.113.194.132:443     ESTABLISHED
  TCP    192.168.0.103:50884    52.114.158.91:443      ESTABLISHED
```

```
C:\Users\ram10>netstat -an

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:808            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:5040           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:5357           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:7680           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49664          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49665          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49666          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49667          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49668          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49670          0.0.0.0:0              LISTENING
  TCP    192.168.0.103:139      0.0.0.0:0              LISTENING
  TCP    192.168.0.103:49674    52.139.250.253:443     ESTABLISHED
  TCP    192.168.0.103:49694    106.51.145.136:443     ESTABLISHED
  TCP    192.168.0.103:49695    106.51.145.136:443     ESTABLISHED
  TCP    192.168.0.103:49696    106.51.145.136:443     ESTABLISHED
  TCP    192.168.0.103:49701    106.51.146.30:80       ESTABLISHED
  TCP    192.168.0.103:49702    106.51.146.30:80       CLOSE_WAIT
  TCP    192.168.0.103:49703    106.51.146.30:80       ESTABLISHED
  TCP    192.168.0.103:49704    106.51.146.30:80       ESTABLISHED
  TCP    192.168.0.103:49705    52.139.250.253:443     ESTABLISHED
  TCP    192.168.0.103:49709    106.51.146.30:80       ESTABLISHED
  TCP    192.168.0.103:49730    106.51.145.136:443     ESTABLISHED
  TCP    192.168.0.103:50223    106.51.144.8:443       ESTABLISHED
  TCP    192.168.0.103:50305    52.114.14.231:443      ESTABLISHED
  TCP    192.168.0.103:50329    52.114.40.55:443       ESTABLISHED
  TCP    192.168.0.103:50362    52.111.252.0:443       ESTABLISHED
  TCP    192.168.0.103:50364    52.114.6.215:443       ESTABLISHED
  TCP    192.168.0.103:50375    52.114.133.60:443      ESTABLISHED
  TCP    192.168.0.103:50880    20.44.232.74:443       ESTABLISHED
  TCP    192.168.0.103:50881    52.109.124.51:443      TIME_WAIT
  TCP    192.168.0.103:50882    52.114.32.111:443      ESTABLISHED
  TCP    192.168.0.103:50883    52.113.194.132:443     ESTABLISHED
  TCP    192.168.0.103:50884    52.114.158.91:443      ESTABLISHED
  TCP    [::]:135               [::]:0                 LISTENING
  TCP    [::]:445               [::]:0                 LISTENING
  TCP    [::]:808               [::]:0                 LISTENING
  TCP    [::]:5357              [::]:0                 LISTENING
  TCP    [::]:7680              [::]:0                 LISTENING
  TCP    [::]:49664             [::]:0                 LISTENING
```

6.

6] ARP command : ARP command corresponds to the address resolution protocol, it is easy to understand of network communications in term of IP addressing. packet delivery is ultimately dependent on the MAC address of the device's network adapter. This is where ARP comes into play. Its job is to map IP address to MAC address. It shows the contents of this cache by using the ARP -a command. If any problems communicating with one specific host, you can append the remote host's IP address to the ARP - A command.



7.

20181CSE0621

1] Nbstat -n Command :-

The nbstat -n command for example shows the NetBIOS names that are in use by a devices. The nbstat -r command shows how many netBIOS names the device has been able to resolve recently.

```
Command Prompt

C:\Users\ram10>nbtstat -n

Ethernet:
Node IpAddress: [0.0.0.0] Scope Id: []

    No names in cache

Wi-Fi:
Node IpAddress: [192.168.0.104] Scope Id: []

               NetBIOS Local Name Table

       Name               Type         Status
    ---------------------------------------------
    LAPTOP-MVFDFOVD<00>  UNIQUE      Registered
    WORKGROUP      <00>  GROUP       Registered
    LAPTOP-MVFDFOVD<20>  UNIQUE      Registered

Local Area Connection* 1:
Node IpAddress: [0.0.0.0] Scope Id: []

    No names in cache

Local Area Connection* 2:
Node IpAddress: [0.0.0.0] Scope Id: []

    No names in cache

C:\Users\ram10>
```

```
C:\Users\ram10>nbtstat -r

    NetBIOS Names Resolution and Registration Statistics
    ----------------------------------------------------

    Resolved By Broadcast     = 0
    Resolved By Name Server   = 0

    Registered By Broadcast   = 3
    Registered By Name Server = 0

C:\Users\ram10>
```

8.

8) Route command :-

Ip networks using the routing table to direct packets from one subnet to another. The windows route utility allows to view the device's routing tables. The route command is that it not only shows you the routing table, it lets you make changes. Commands such as route add, delete and route change are used for such modifications.

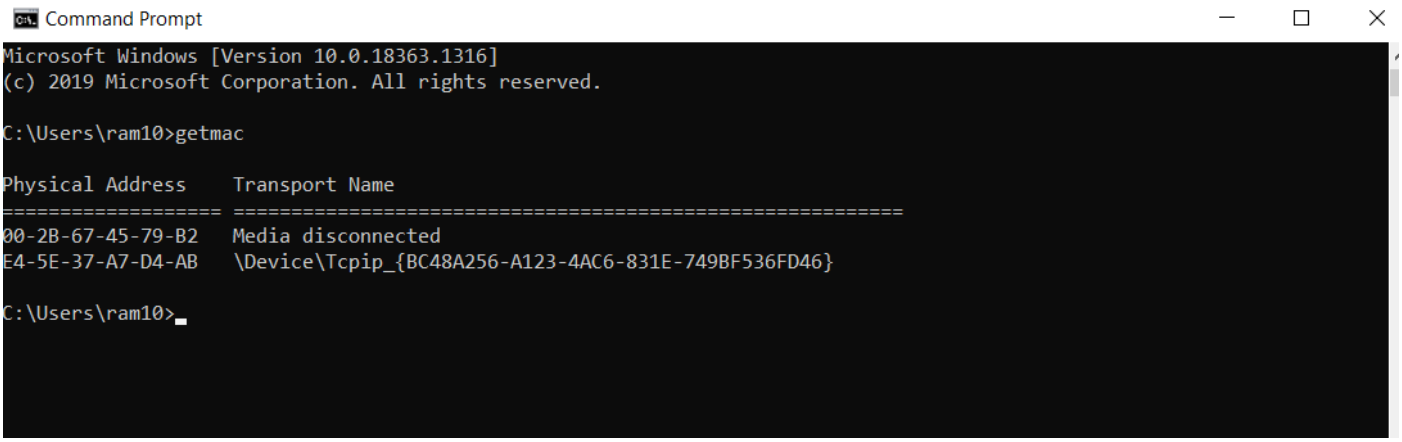
```
C:\Users\ram10>route -4

Manipulates network routing tables.

ROUTE [-f] [-p] [-4|-6] command [destination]
                  [MASK netmask]  [gateway] [METRIC metric]  [IF interface]

  -f           Clears the routing tables of all gateway entries.  If this is
               used in conjunction with one of the commands, the tables are
               cleared prior to running the command.

  -p           When used with the ADD command, makes a route persistent across
               boots of the system. By default, routes are not preserved
               when the system is restarted. Ignored for all other commands,
               which always affect the appropriate persistent routes.

  -4           Force using IPv4.

  -6           Force using IPv6.

  command      One of these:
                 PRINT     Prints  a route
                 ADD       Adds    a route
                 DELETE    Deletes a route
```

9.

20181CSE0621

9] Getmac Command :

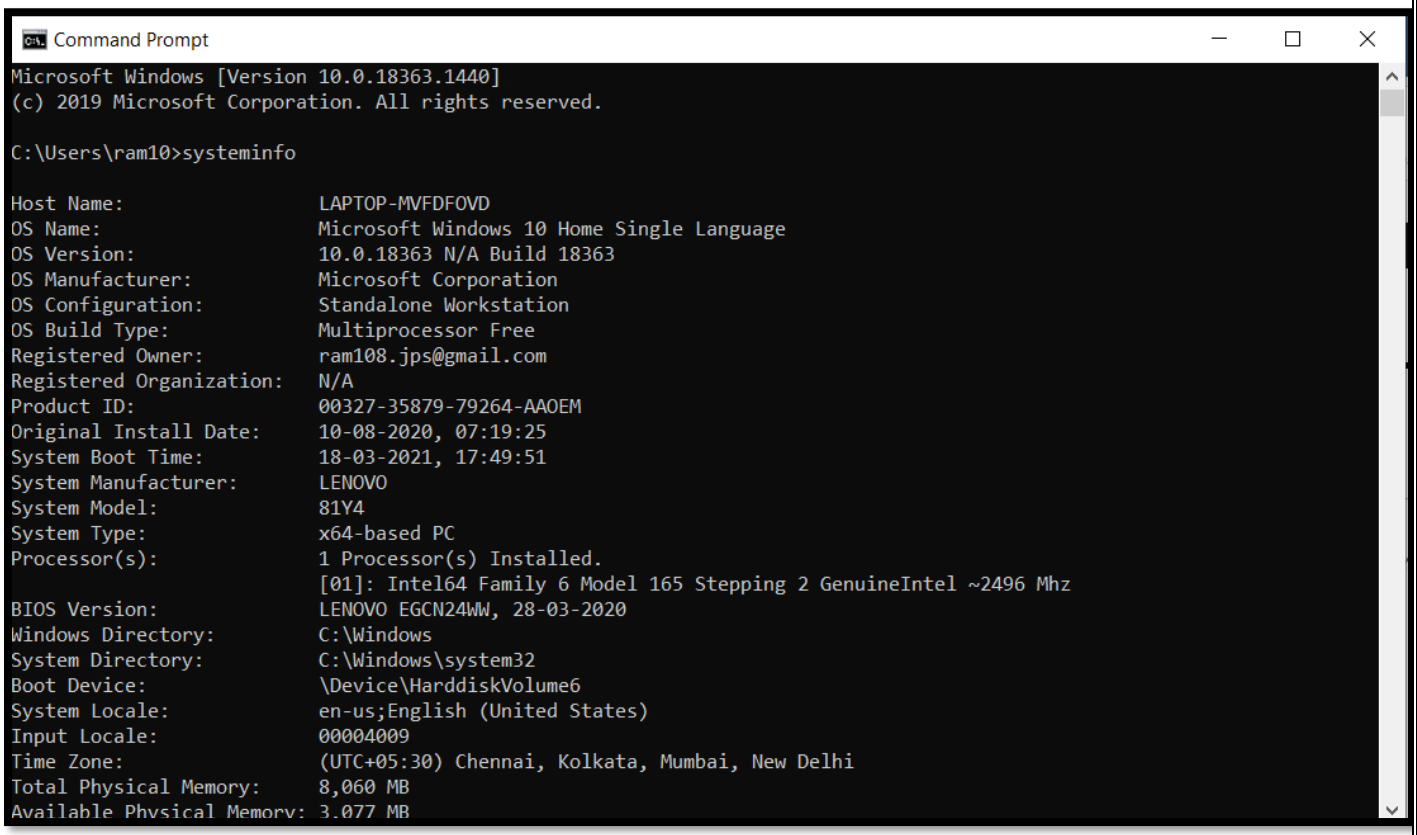
Getmac is a windows command that used to display the MAC addresses for each network adapter. One of the fastest way to obtain the MAC address is by using getmac command.



10.

10] Systeminfo Command :-

If you need to know what brand of network you have, processor details, or exact version of OS this command helps. This command polls your system & pulls the most important information & lists in a easy form to read.

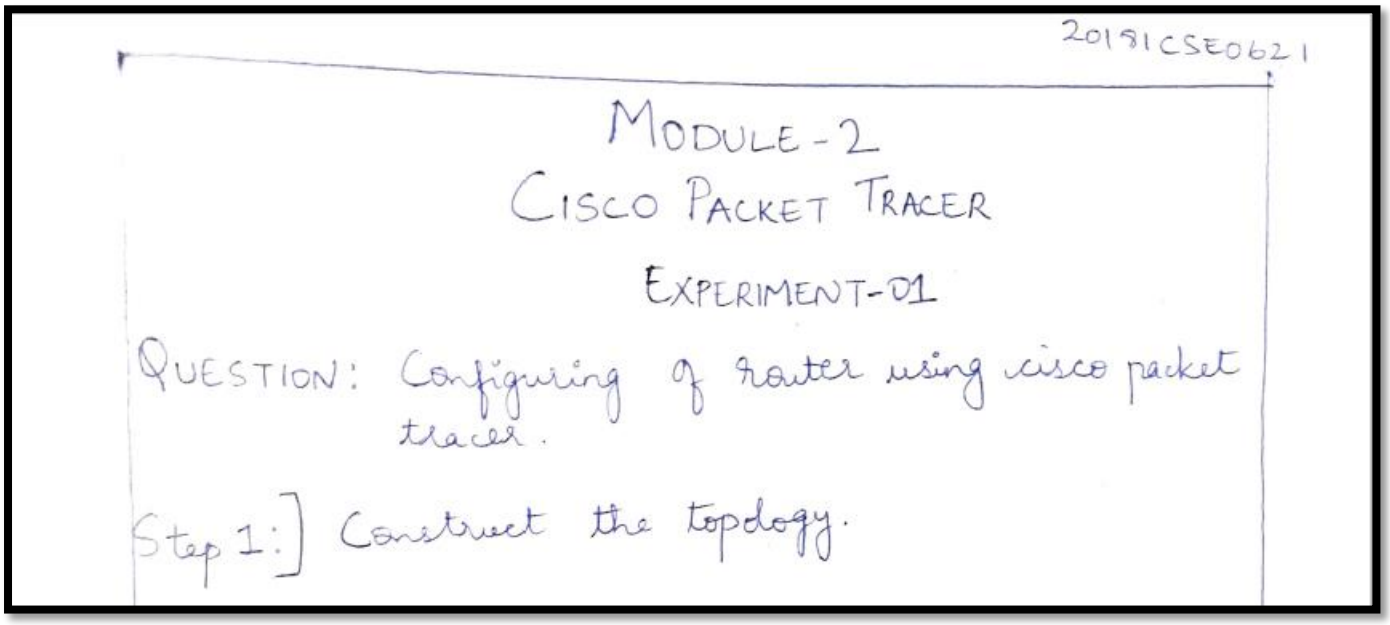

Command Prompt
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\ram10>systeminfo
Host Name: LAPTOP-MVFDFOVD
OS Name: Microsoft Windows 10 Home Single Language
OS Version: 10.0.18363 N/A Build 18363
OS Manufacturer: Microsoft Corporation
OS Configuration: Standalone Workstation
OS Build Type: Multiprocessor Free
Registered Owner: ram108.jps@gmail.com
Registered Organization: N/A
Product ID: 00327-35879-79264-AAOEM
Original Install Date: 10-08-2020, 07:19:25
System Boot Time: 18-03-2021, 17:49:51
System Manufacturer: LENOVO
System Model: 81Y4
System Type: x64-based PC
Processor(s): 1 Processor(s) Installed.
[01]: Intel64 Family 6 Model 165 Stepping 2 GenuineIntel ~2496 Mhz
BIOS Version: LENOVO EGCN24WW, 28-03-2020
Windows Directory: C:\Windows
System Directory: C:\Windows\system32
Boot Device: \Device\HarddiskVolume6
System Locale: en-us;English (United States)
Input Locale: 00004009
Time Zone: (UTC+05:30) Chennai, Kolkata, Mumbai, New Delhi
Total Physical Memory: 8,060 MB
Available Physical Memory: 3,077 MB

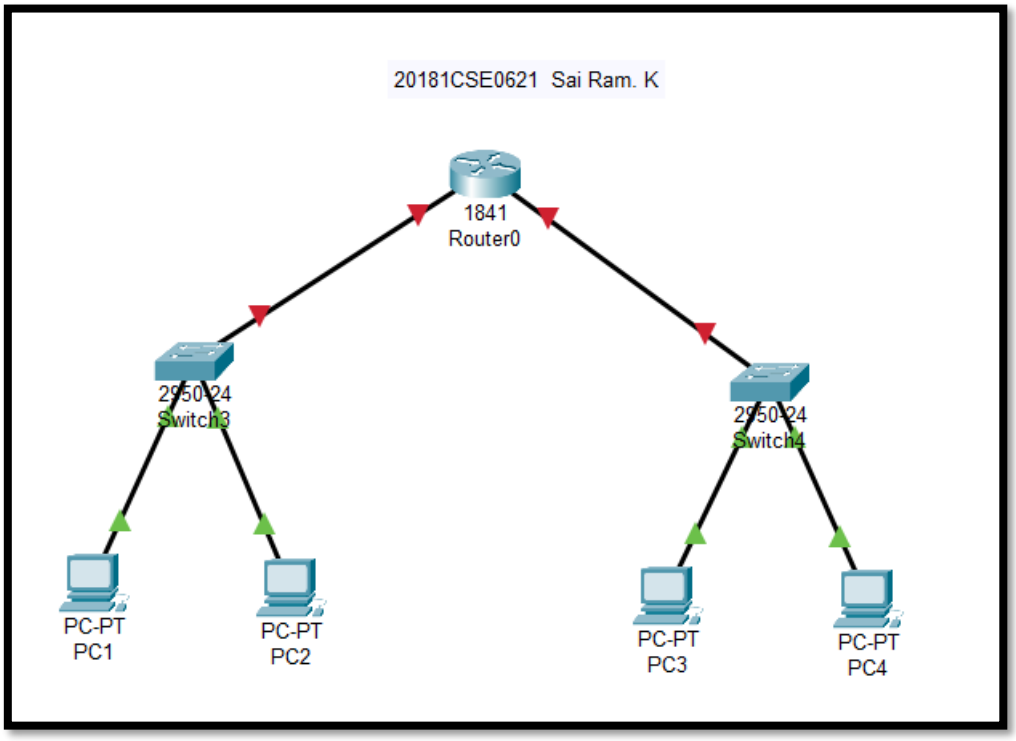
Module – 2

Cisco Packet Tracer

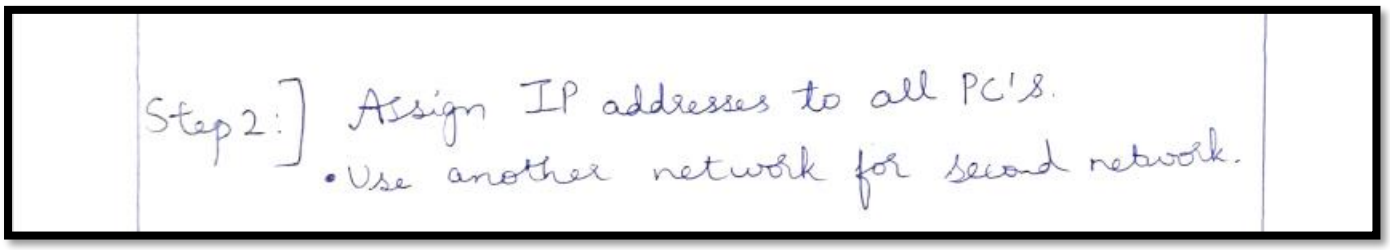
Experiment – 1

Step 1.





Step 2.



20181CSE0621 Sai Ram. K

PC1

Physical Config Desktop Programming Attributes

IP Configuration

Interface FastEthernet0

IP Configuration

DHCP Static

IPv4 Address 192.168.0.2

Subnet Mask 255.255.255.0

Default Gateway 0.0.0.0

DNS Server 0.0.0.0

IPv6 Configuration

Automatic Static

IPv6 Address /

Link Local Address FE80::201:97FF:FEEC:C0BC

Default Gateway

DNS Server

802.1X

Use 802.1X Security

Authentication MD5

Top

20181CSE0621 Sai Ram. K

PC2

Physical Config Desktop Programming Attributes

IP Configuration

Interface FastEthernet0

IP Configuration

DHCP Static

IPv4 Address 192.168.0.3

Subnet Mask 255.255.255.0

Default Gateway 0.0.0.0

DNS Server 0.0.0.0

IPv6 Configuration

Automatic Static

IPv6 Address /

Link Local Address FE80::290:CFF:FED0:9DD6

Default Gateway

DNS Server

802.1X

Use 802.1X Security

Authentication MD5

Top

20181CSE0621 Sai Ram. K

PC3

Physical Config Desktop Programming Attributes

IP Configuration

Interface FastEthernet0

IP Configuration

DHCP Static

IPv4 Address 192.168.1.2

Subnet Mask 255.255.255.0

Default Gateway 0.0.0.0

DNS Server 0.0.0.0

IPv6 Configuration

Automatic Static

IPv6 Address /

Link Local Address FE80::2D0:D3FF:FEE6:70B3

Default Gateway

DNS Server

802.1X

Use 802.1X Security

Authentication MD5

Top

20181CSE0621 Sai Ram. K

PC4

Physical Config Desktop Programming Attributes

IP Configuration

Interface FastEthernet0

IP Configuration

DHCP Static

IPv4 Address 192.168.1.3

Subnet Mask 255.255.255.0

Default Gateway 0.0.0.0

DNS Server 0.0.0.0

IPv6 Configuration

Automatic Static

IPv6 Address /

Link Local Address FE80::201:96FF:FEC2:5B02

Default Gateway

DNS Server

802.1X

Use 802.1X Security

Authentication MD5

Top

Step 3.

Step 3:] Assign the IP address for router.

- Assign the gateway address of 1st network and don't forget to turn on the port status.
- Assign the gateway address of 2nd address for FastEthernet 0/1 interface.

20181CSE0621 Sai Ram. K

PC1

Physical Config Desktop Programming Attributes

GLOBAL
Settings
Algorithm Settings
INTERFACE
FastEthernet0
Bluetooth

FastEthernet0

Port Status: On
Bandwidth: 100 Mbps 10 Mbps Auto
Duplex: Half Duplex Full Duplex Auto
MAC Address: 0001.97EC.C0BC

IP Configuration
DHCP
Static
IPv4 Address: 192.168.0.2
Subnet Mask: 255.255.255.0

IPv6 Configuration
Automatic
Static
IPv6 Address:
Link Local Address: FE80::201:97FF:FEEC:C0BC

Top

20181CSE0621 Sai Ram. K

PC2

Physical Config Desktop Programming Attributes

IP Configuration

Interface: FastEthernet0

IP Configuration
DHCP
Static
IPv4 Address: 192.168.0.3
Subnet Mask: 255.255.255.0
Default Gateway: 0.0.0.0
DNS Server: 0.0.0.0

IPv6 Configuration
Automatic
Static
IPv6 Address:
Link Local Address: FE80::290:CFF:FED0:9DD6
Default Gateway:
DNS Server:

802.1X
Use 802.1X Security
Authentication: MD5

Top

20181CSE0621 Sai Ram. K

PC3

Physical Config Desktop Programming Attributes

IP Configuration

Interface: FastEthernet0

IP Configuration
DHCP
Static
IPv4 Address: 192.168.1.2
Subnet Mask: 255.255.255.0
Default Gateway: 0.0.0.0
DNS Server: 0.0.0.0

IPv6 Configuration
Automatic
Static
IPv6 Address:
Link Local Address: FE80::2D0:D3FF:FEE6:70B3
Default Gateway:
DNS Server:

802.1X
Use 802.1X Security
Authentication: MD5

Top

20181CSE0621 Sai Ram. K

PC4

Physical Config Desktop Programming Attributes

IP Configuration

Interface: FastEthernet0

IP Configuration
DHCP
Static
IPv4 Address: 192.168.1.3
Subnet Mask: 255.255.255.0
Default Gateway: 0.0.0.0
DNS Server: 0.0.0.0

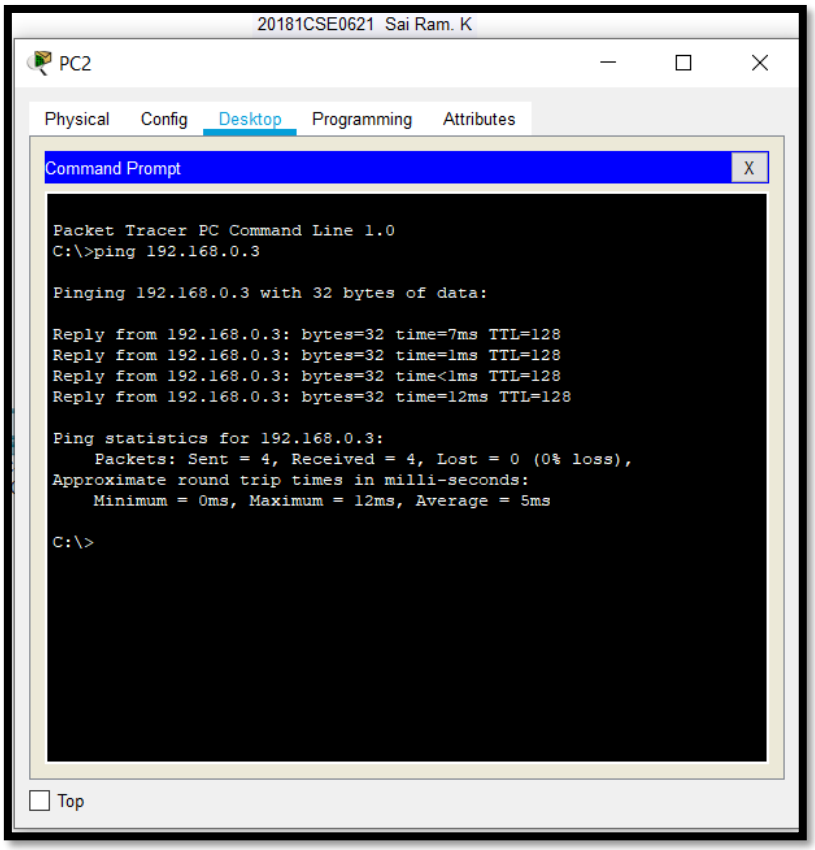
IPv6 Configuration
Automatic
Static
IPv6 Address:
Link Local Address: FE80::201:96FF:FEC2:5B02
Default Gateway:
DNS Server:

802.1X
Use 802.1X Security
Authentication: MD5

Top

Step 4.

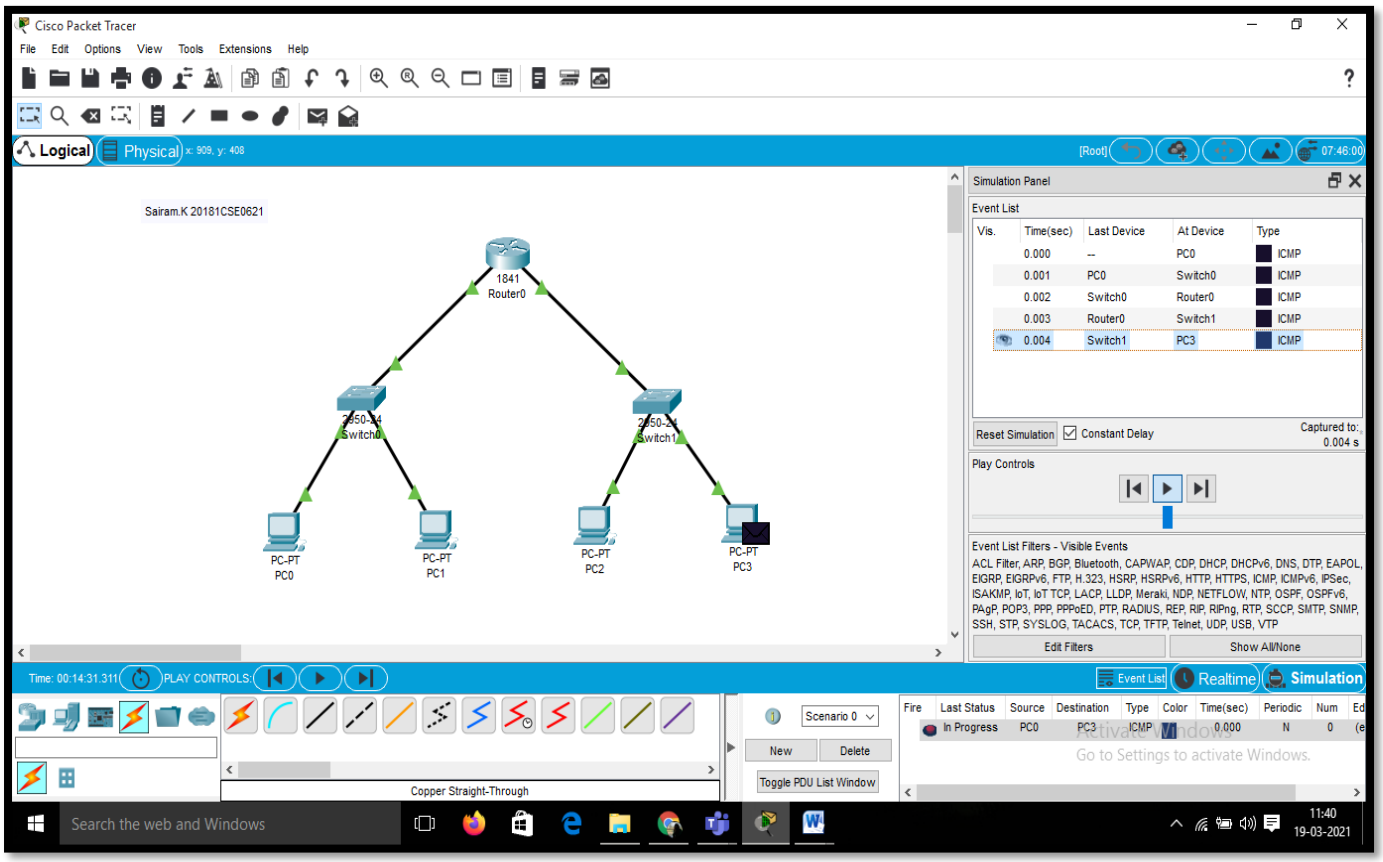
Step 4:] Check the connectivity from one network to other.
- Select any PC from 1st network go to desktop tab > Command Prompt > execute ping command for 2nd network



Step 5 & 6.

Step 5:] Send simple PDU.

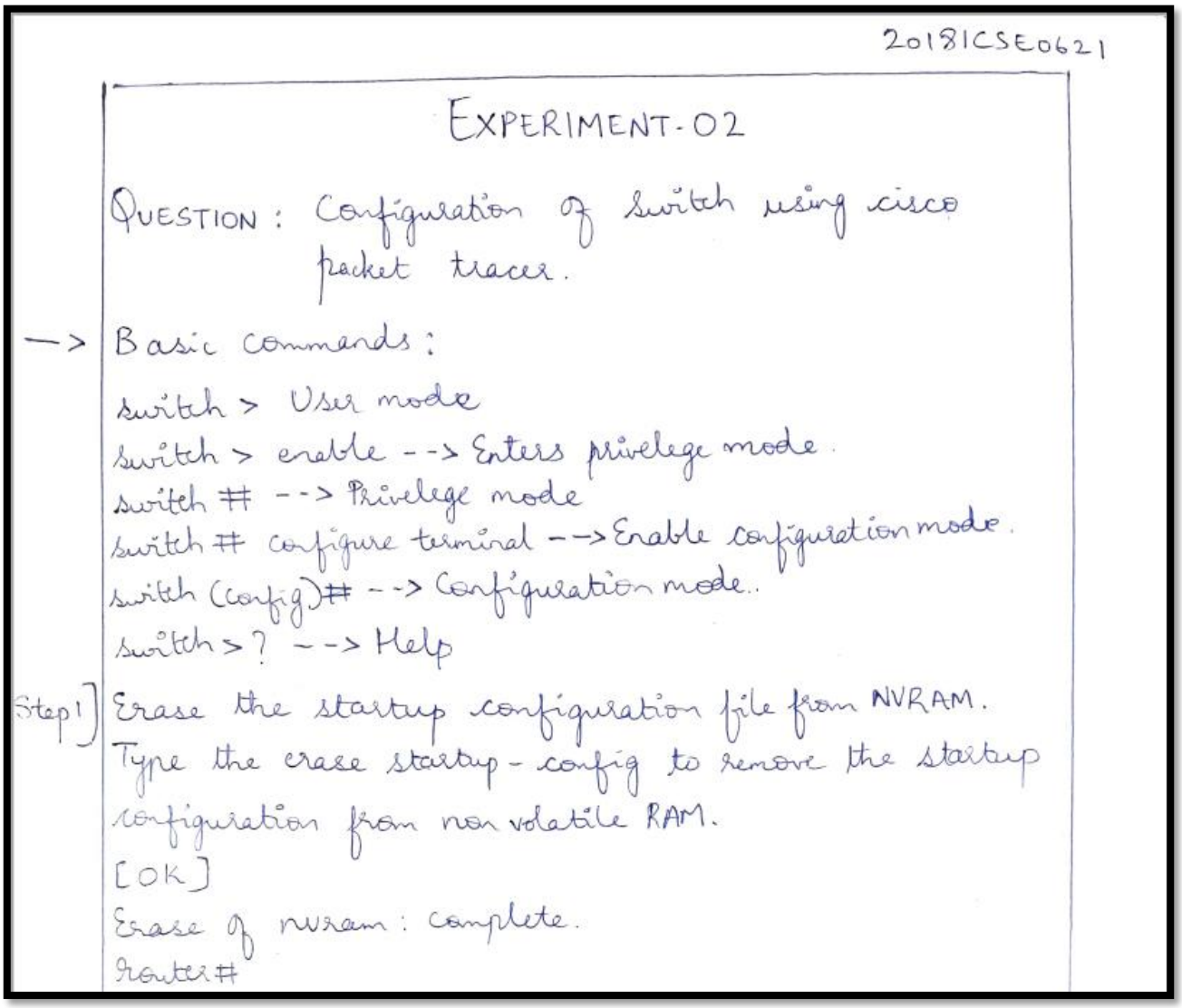
Step 6:] Check in simulation mode.


Cisco Packet Tracer
File Edit Options View Tools Extensions Help
Logical
Physical
Sairam.K 20181CSE0621
1841
Router0
2950-24
Switch0
2950-24
Switch1
PC-PT
PC0
PC-PT
PC1
PC-PT
PC2
PC-PT
PC3
Simulation Panel
Event List
Vis. Time(sec) Last Device At Device Type
0.000 -- PC0 ICMP
0.001 PC0 Switch0 ICMP
0.002 Switch0 Router0 ICMP
0.003 Router0 Switch1 ICMP
0.004 Switch1 PC3 ICMP
Reset Simulation
Constant Delay
Captured to:
0.004 s
Play Controls
Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, Meraki, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP
Edit Filters
Show All/None
Time: 00:14:31.311
PLAY CONTROLS:
Event List
Realtime
Simulation
Copper Straight-Through
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(sec) Periodic Num
In Progress PC0 PC3 ICMP 0.000 N 0
Activate Windows
Go to Settings to activate Windows.
Search the web and Windows
11:40
19-03-2021

Experiment – 2

Configuration of Switch using cisco packet tracer

Step 1.



20181CSE0621

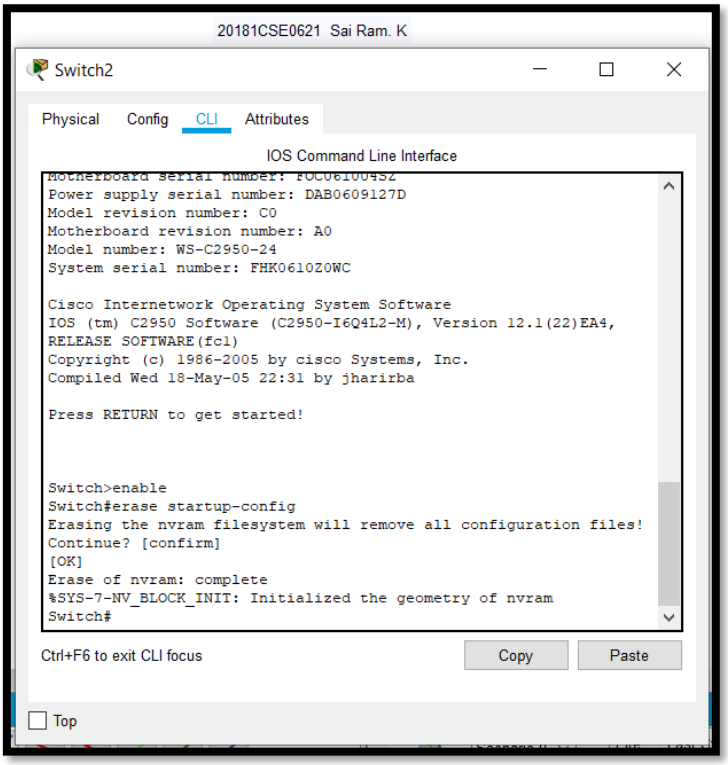
EXPERIMENT-02

QUESTION : Configuration of Switch using cisco packet tracer.

—> Basic commands :

switch > User mode
switch > enable --> Enters privelege mode.
switch # --> Privelege mode
switch # configure terminal --> Enable configuration mode.
switch (config)# --> Configuration mode..
switch > ? --> Help

Step1] Erase the startup configuration file from NVRAM.
Type the erase startup-config to remove the startup configuration from non volatile RAM.
[OK]
Erase of nvram : complete.
Router#



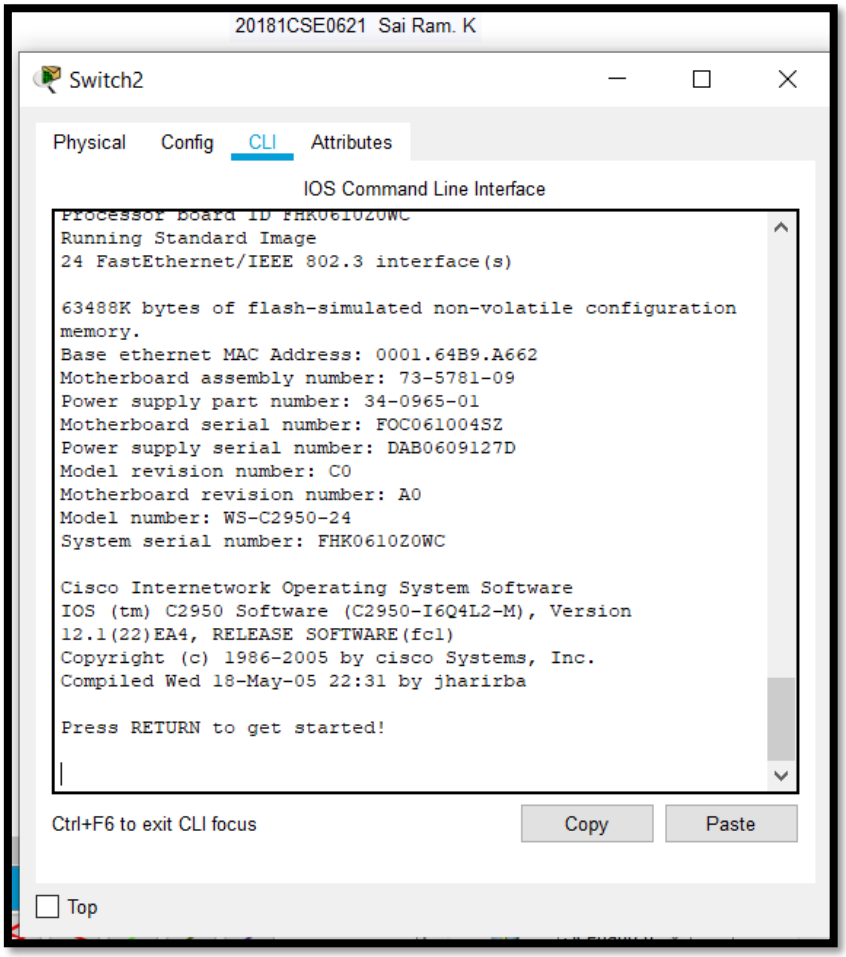
Step 2.

Step 2] Reload the switch.

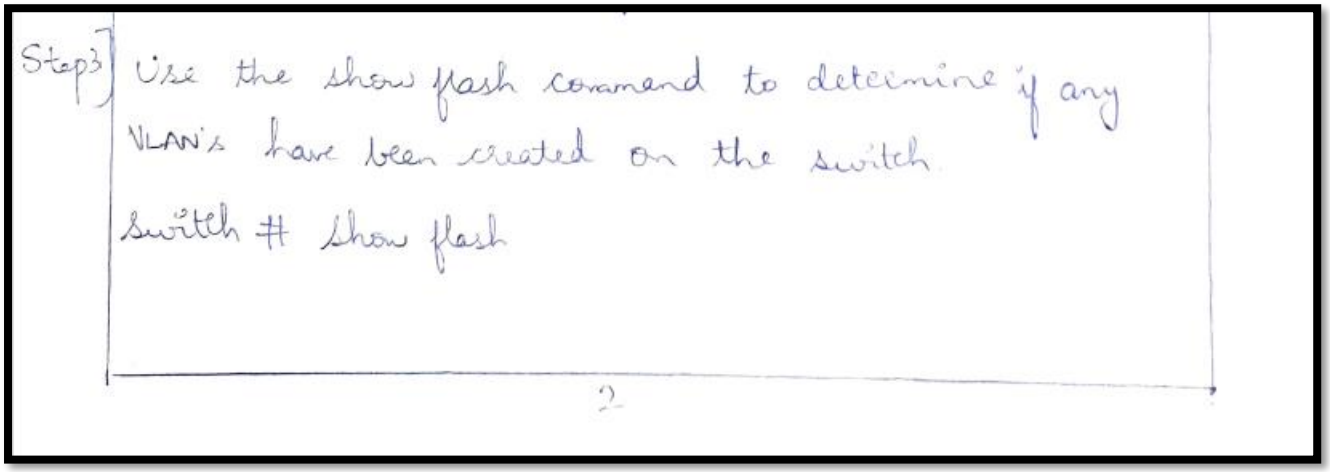
Issue the reload command to remove an old configuration from memory. When prompted to proceed with reload press enter to confirm.

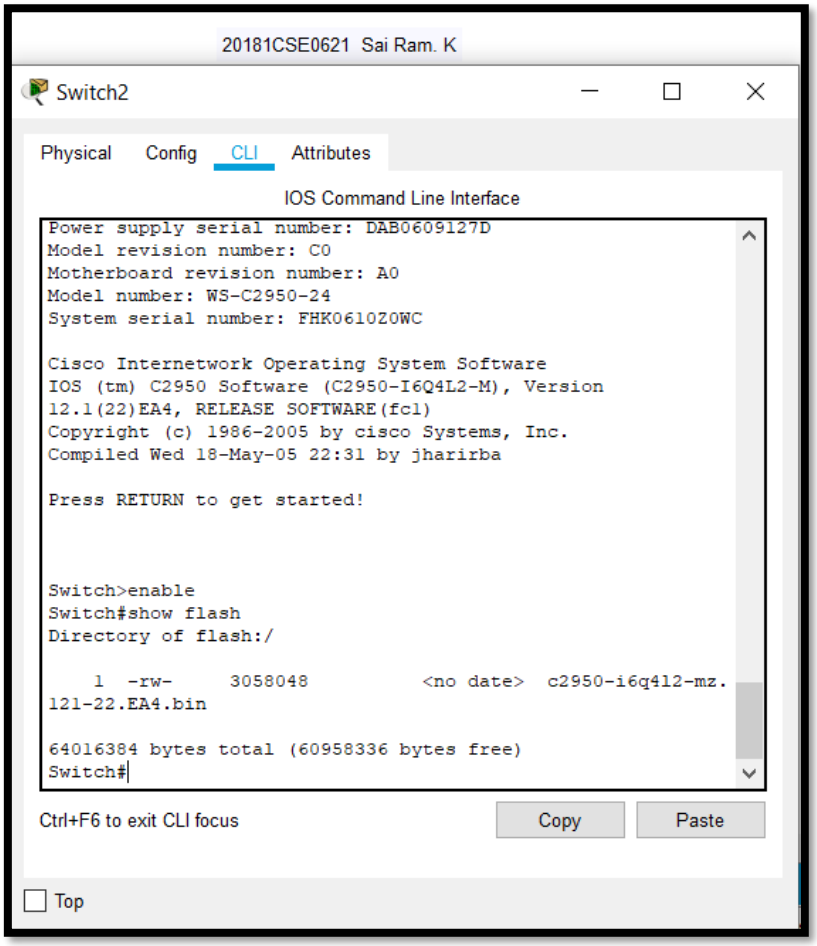
switch # reload

proceed with reload ? [confirm]

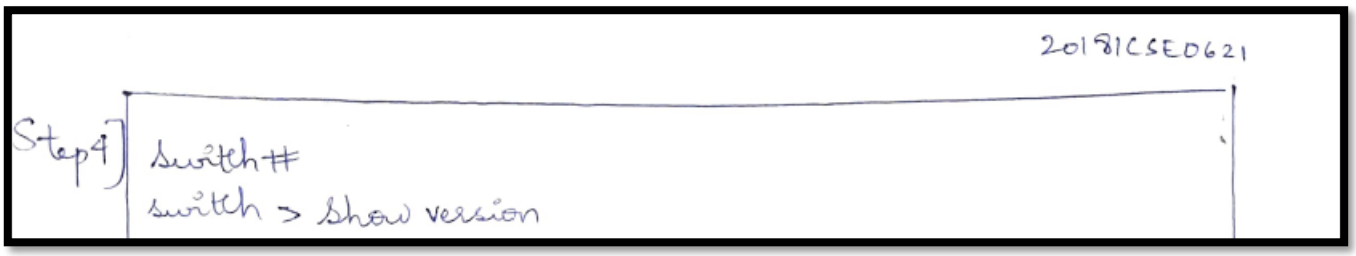

20181CSE0621 Sai Ram. K
Switch2
Physical
Config
CLI
Attributes
IOS Command Line Interface
Processor board ID FHK0610Z0WC
Running Standard Image
24 FastEthernet/IEEE 802.3 interface(s)
63488K bytes of flash-simulated non-volatile configuration memory.
Base ethernet MAC Address: 0001.64B9.A662
Motherboard assembly number: 73-5781-09
Power supply part number: 34-0965-01
Motherboard serial number: FOC061004SZ
Power supply serial number: DAB0609127D
Model revision number: C0
Motherboard revision number: A0
Model number: WS-C2950-24
System serial number: FHK0610Z0WC
Cisco Internetwork Operating System Software
IOS (tm) C2950 Software (C2950-I6Q4L2-M), Version 12.1(22)EA4, RELEASE SOFTWARE(fc1)
Copyright (c) 1986-2005 by cisco Systems, Inc.
Compiled Wed 18-May-05 22:31 by jharirba
Press RETURN to get started!
Ctrl+F6 to exit CLI focus
Copy
Paste
Top

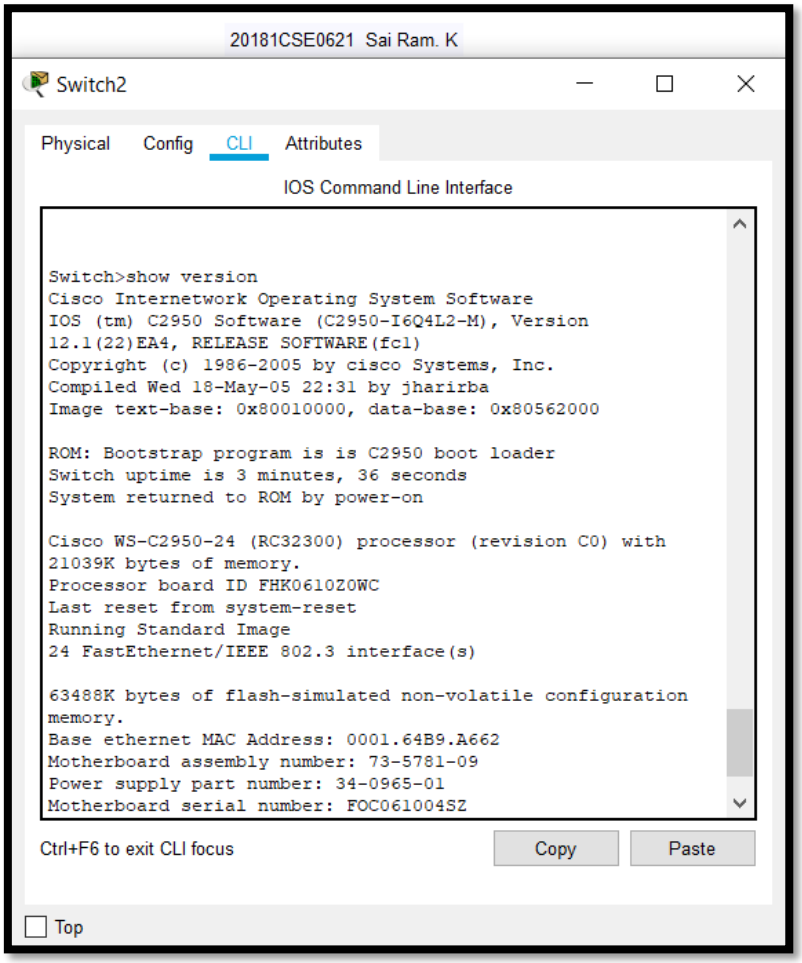
Step 3.


Step3
Use the show flash command to determine if any VLAN's have been created on the switch.
Switch # show flash
2


20181CSE0621 Sai Ram. K
Switch2
Physical
Config
CLI
Attributes
IOS Command Line Interface
Power supply serial number: DAB0609127D
Model revision number: C0
Motherboard revision number: A0
Model number: WS-C2950-24
System serial number: FHK0610Z0WC
Cisco Internetwork Operating System Software
IOS (tm) C2950 Software (C2950-I6Q4L2-M), Version
12.1(22)EA4, RELEASE SOFTWARE(fc1)
Copyright (c) 1986-2005 by cisco Systems, Inc.
Compiled Wed 18-May-05 22:31 by jharirba
Press RETURN to get started!
Switch>enable
Switch#show flash
Directory of flash:/
1 -rw- 3058048 <no date> c2950-i6q412-mz.
121-22.EA4.bin
64016384 bytes total (60958336 bytes free)
Switch#
Ctrl+F6 to exit CLI focus
Copy
Paste
Top

Step 4.


20181CSE0621
Step4
Switch#
switch > show version



Step 5.

Steps]

Configure the clock

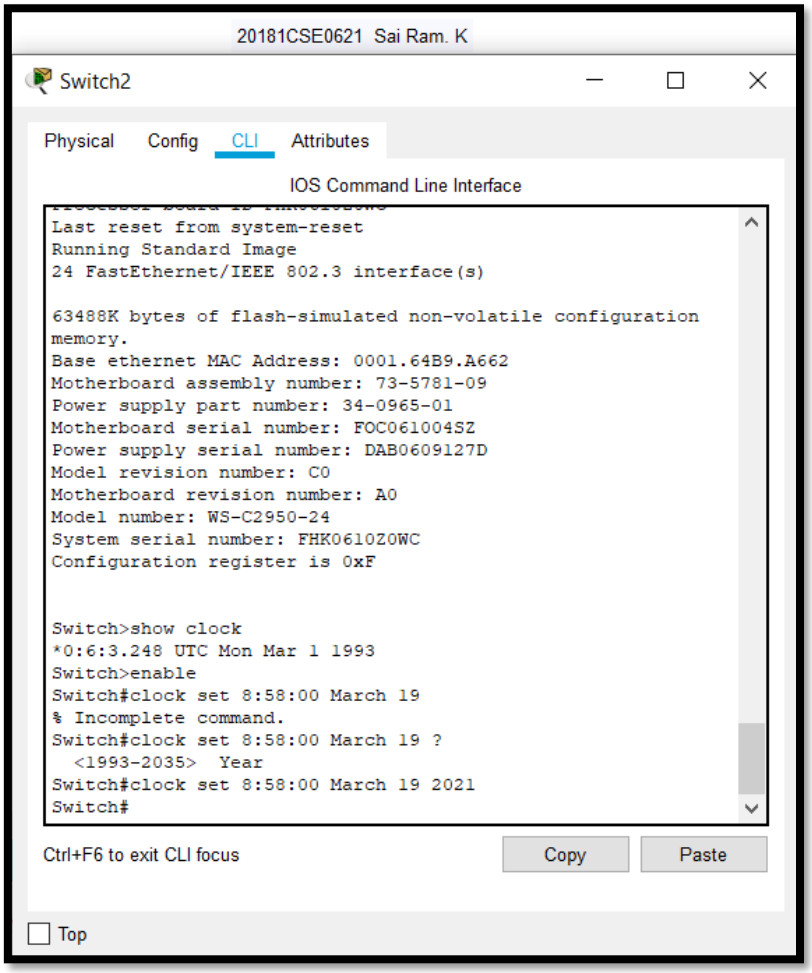
As you learn more about networking you will see that configuring the correct time on a cisco switch can be helpful when you are troubleshooting the problems.

a] Display current clock settings

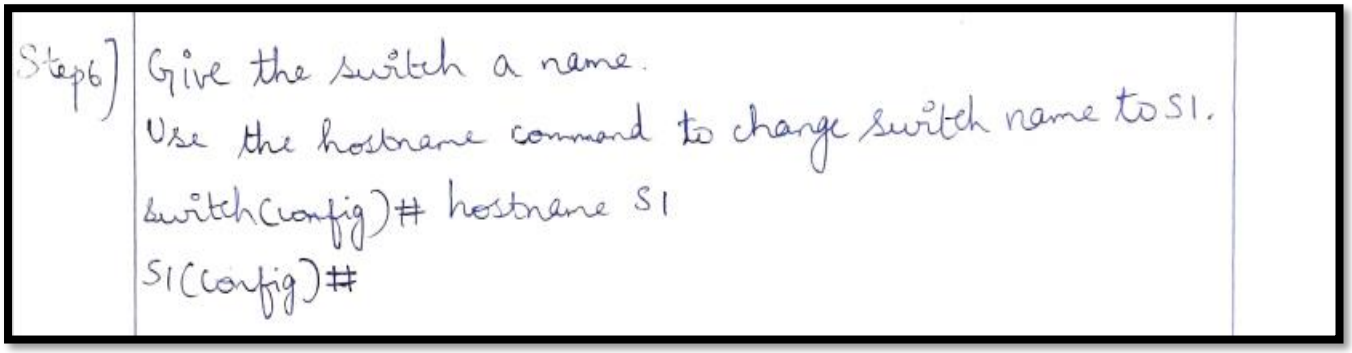
switch > show clock.

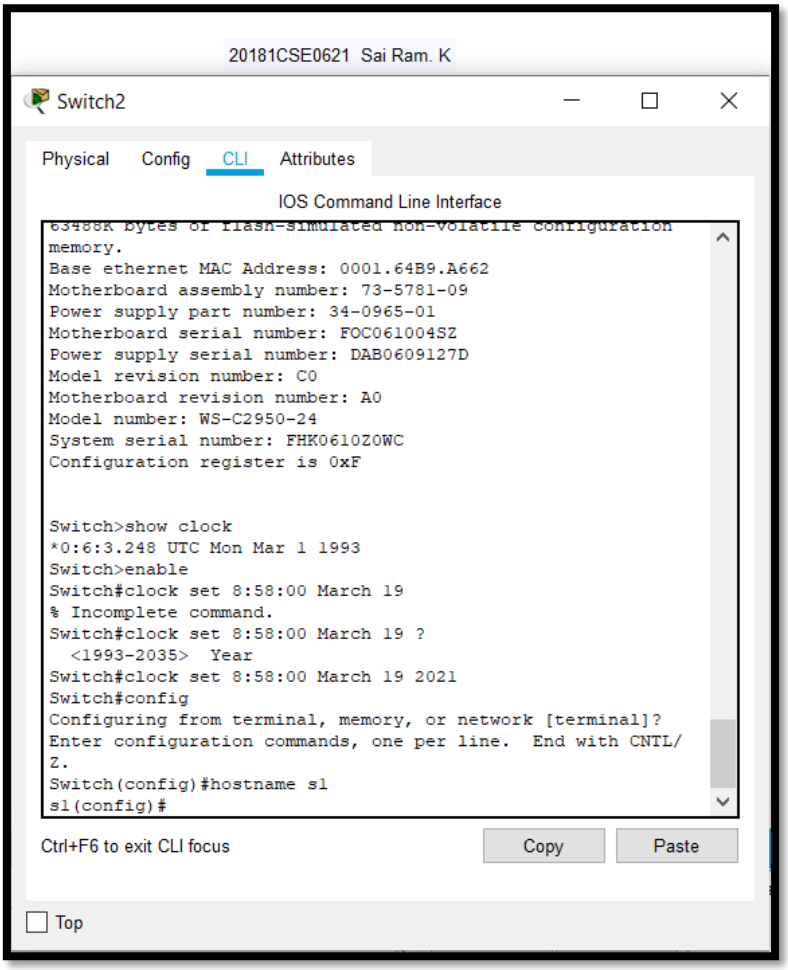
b] Configure the clock settings

switch # clock set 15:08:00

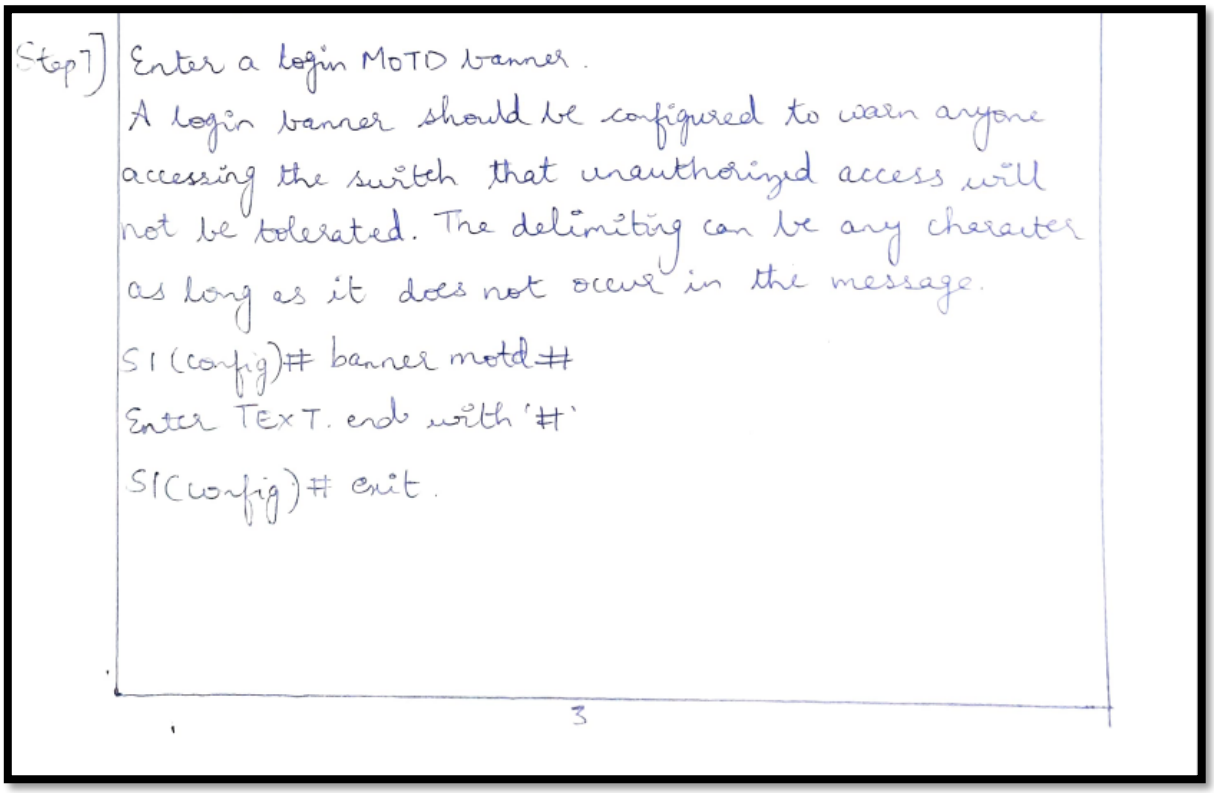

20181CSE0621 Sai Ram. K
Switch2
Physical
Config
CLI
Attributes
IOS Command Line Interface
Last reset from system-reset
Running Standard Image
24 FastEthernet/IEEE 802.3 interface(s)
63488K bytes of flash-simulated non-volatile configuration memory.
Base ethernet MAC Address: 0001.64B9.A662
Motherboard assembly number: 73-5781-09
Power supply part number: 34-0965-01
Motherboard serial number: FOC061004SZ
Power supply serial number: DAB0609127D
Model revision number: C0
Motherboard revision number: A0
Model number: WS-C2950-24
System serial number: FHK0610Z0WC
Configuration register is 0xF
Switch>show clock
*0:6:3.248 UTC Mon Mar 1 1993
Switch>enable
Switch#clock set 8:58:00 March 19
% Incomplete command.
Switch#clock set 8:58:00 March 19 ?
<1993-2035> Year
Switch#clock set 8:58:00 March 19 2021
Switch#
Ctrl+F6 to exit CLI focus
Copy
Paste
Top

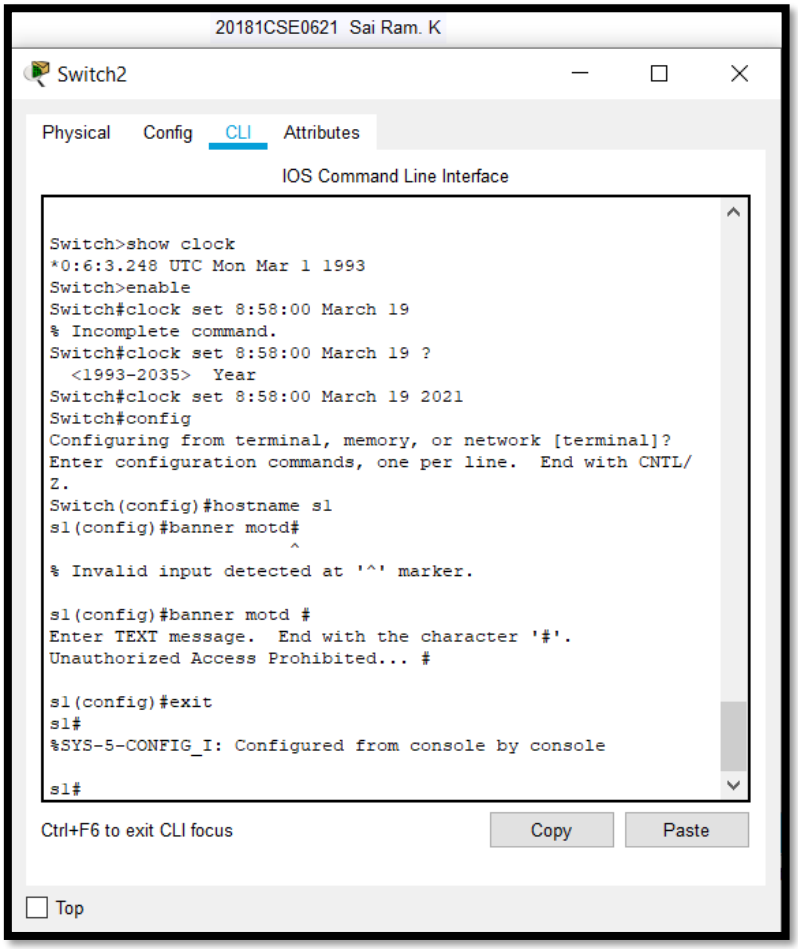
Step 6.


Step6)
Give the switch a name.
Use the hostname command to change switch name to S1.
Switch(config)# hostname S1
S1(config)#

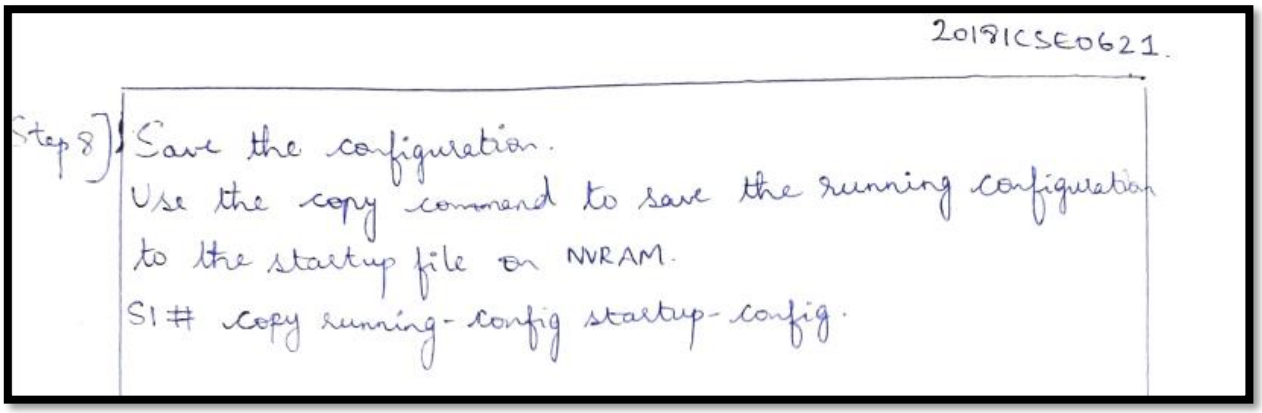

20181CSE0621 Sai Ram. K
Switch2
Physical
Config
CLI
Attributes
IOS Command Line Interface
63488K bytes of flash-simulated non-volatile configuration memory.
Base ethernet MAC Address: 0001.64B9.A662
Motherboard assembly number: 73-5781-09
Power supply part number: 34-0965-01
Motherboard serial number: FOC061004SZ
Power supply serial number: DAB0609127D
Model revision number: C0
Motherboard revision number: A0
Model number: WS-C2950-24
System serial number: FHK0610Z0WC
Configuration register is 0xF
Switch>show clock
*0:6:3.248 UTC Mon Mar 1 1993
Switch>enable
Switch#clock set 8:58:00 March 19
% Incomplete command.
Switch#clock set 8:58:00 March 19 ?
<1993-2035> Year
Switch#clock set 8:58:00 March 19 2021
Switch#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#hostname s1
s1(config)#
Ctrl+F6 to exit CLI focus
Copy
Paste
Top

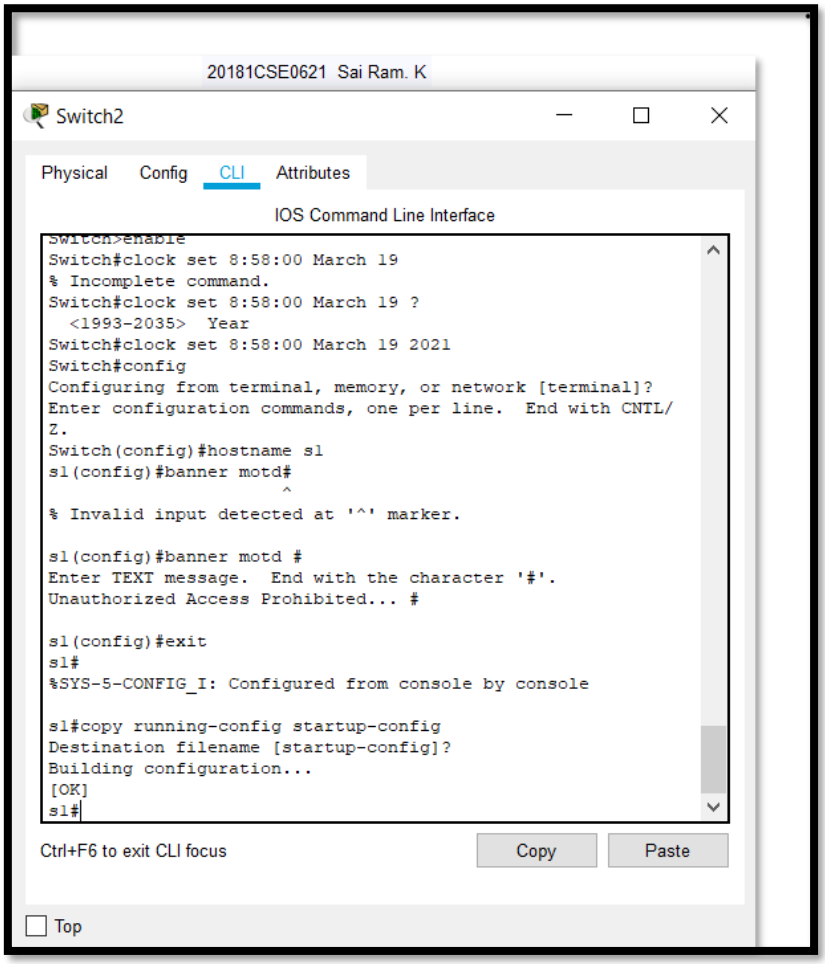
Step 7.


Step7] Enter a login MOTD banner.
A login banner should be configured to warn anyone accessing the switch that unauthorized access will not be tolerated. The delimiting can be any character as long as it does not occur in the message.
S1(config)# banner motd #
Enter TEXT. end with '#'
S1(config)# exit.
3

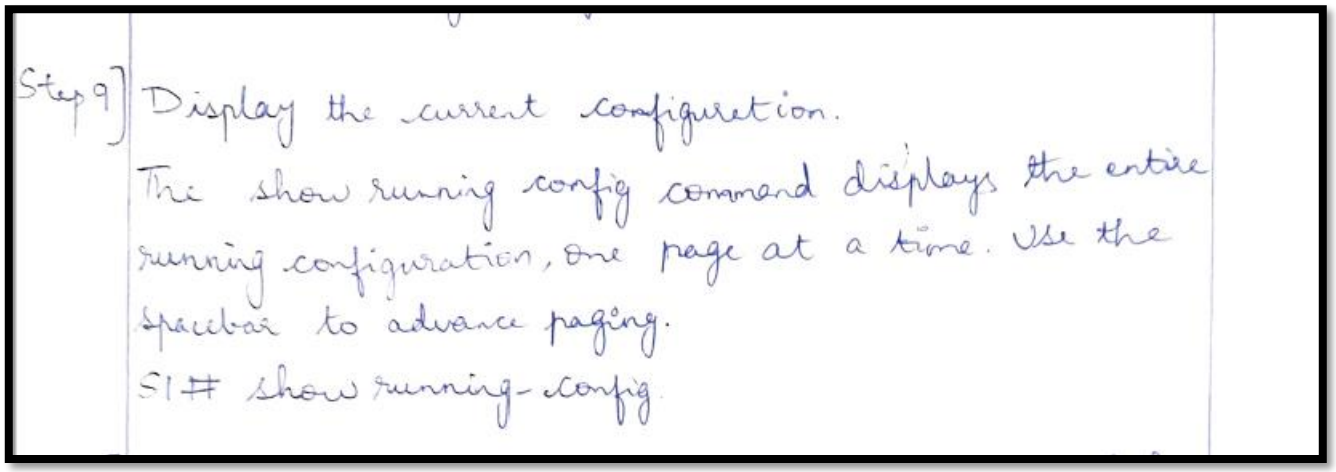

20181CSE0621 Sai Ram. K
Switch2
Physical
Config
CLI
Attributes
IOS Command Line Interface
Switch>show clock
*0:6:3.248 UTC Mon Mar 1 1993
Switch>enable
Switch#clock set 8:58:00 March 19
% Incomplete command.
Switch#clock set 8:58:00 March 19 ?
<1993-2035> Year
Switch#clock set 8:58:00 March 19 2021
Switch#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/
Z.
Switch(config)#hostname s1
s1(config)#banner motd#
^
% Invalid input detected at '^' marker.
s1(config)#banner motd #
Enter TEXT message. End with the character '#'.
Unauthorized Access Prohibited... #
s1(config)#exit
s1#
%SYS-5-CONFIG_I: Configured from console by console
s1#
Ctrl+F6 to exit CLI focus
Copy
Paste
Top

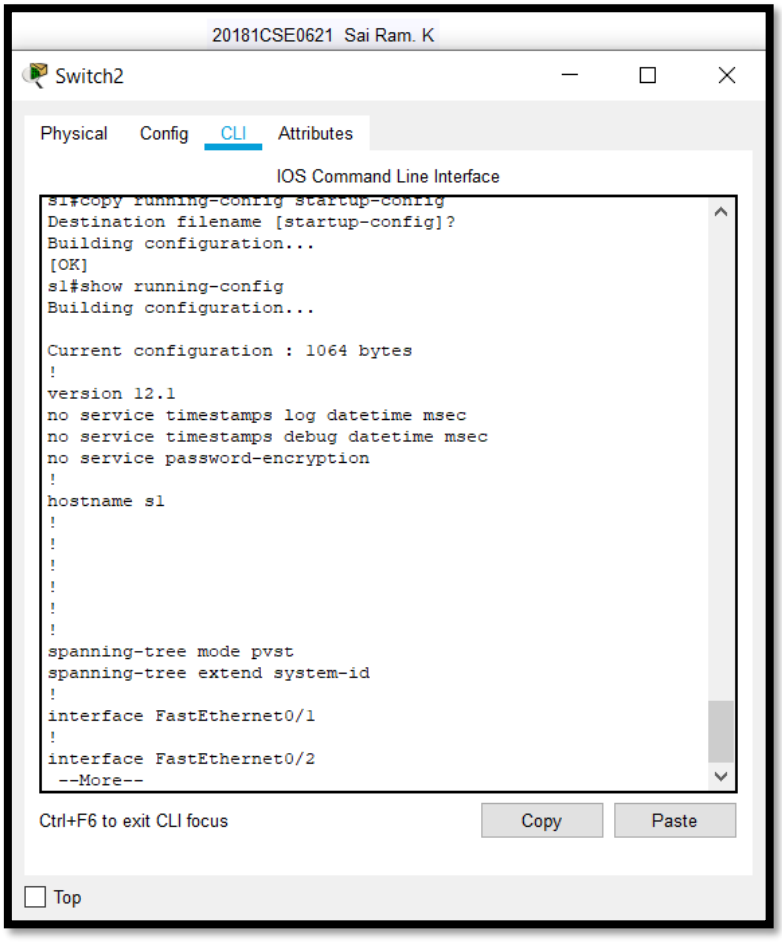
Step 8.


20181CSE0621.
Step 8]
Save the configuration.
Use the copy command to save the running configuration
to the startup file on NVRAM.
S1# copy running-config startup-config.

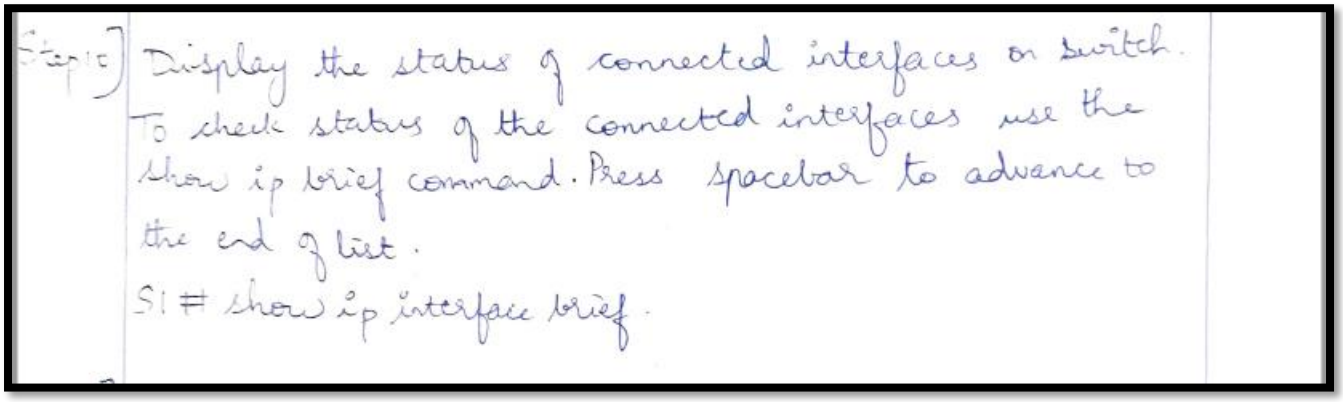

20181CSE0621 Sai Ram. K
Switch2
Physical
Config
CLI
Attributes
IOS Command Line Interface
Switch>enable
Switch#clock set 8:58:00 March 19
% Incomplete command.
Switch#clock set 8:58:00 March 19 ?
<1993-2035> Year
Switch#clock set 8:58:00 March 19 2021
Switch#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/
Z.
Switch(config)#hostname s1
s1(config)#banner motd#
^
% Invalid input detected at '^' marker.
s1(config)#banner motd #
Enter TEXT message. End with the character '#'.
Unauthorized Access Prohibited... #
s1(config)#exit
s1#
%SYS-5-CONFIG_I: Configured from console by console
s1#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
s1#
Ctrl+F6 to exit CLI focus
Copy
Paste
Top

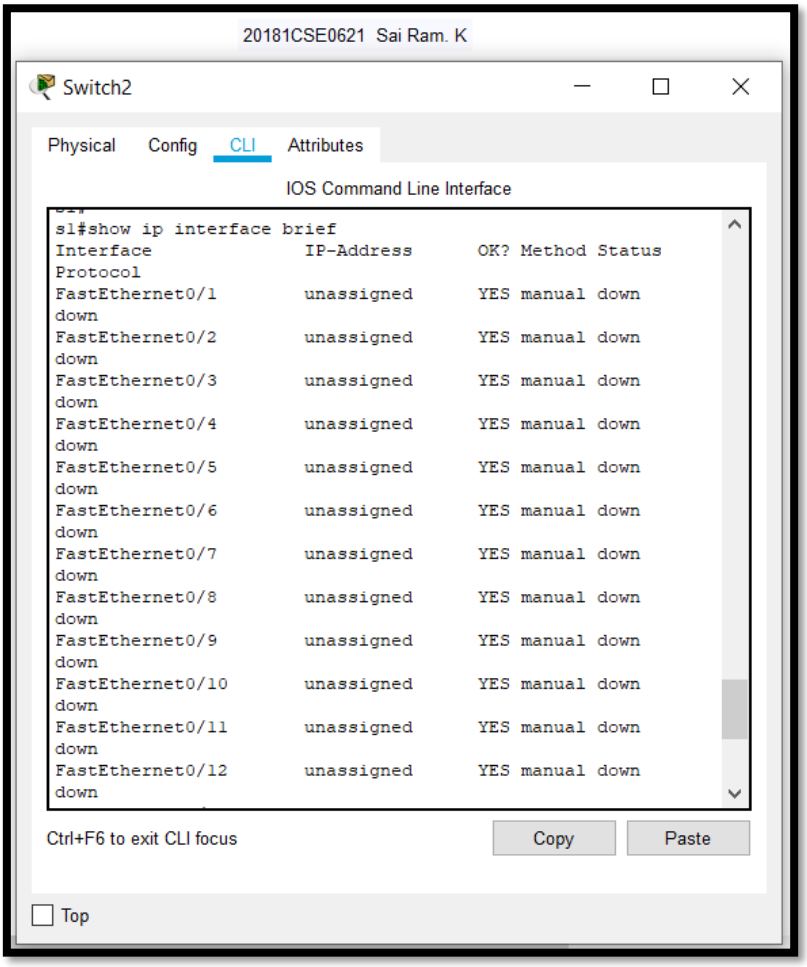
Step 9.


Step 9] Display the current configuration.
The show running config command displays the entire running configuration, one page at a time. Use the spacebar to advance paging.
S1# show running-config

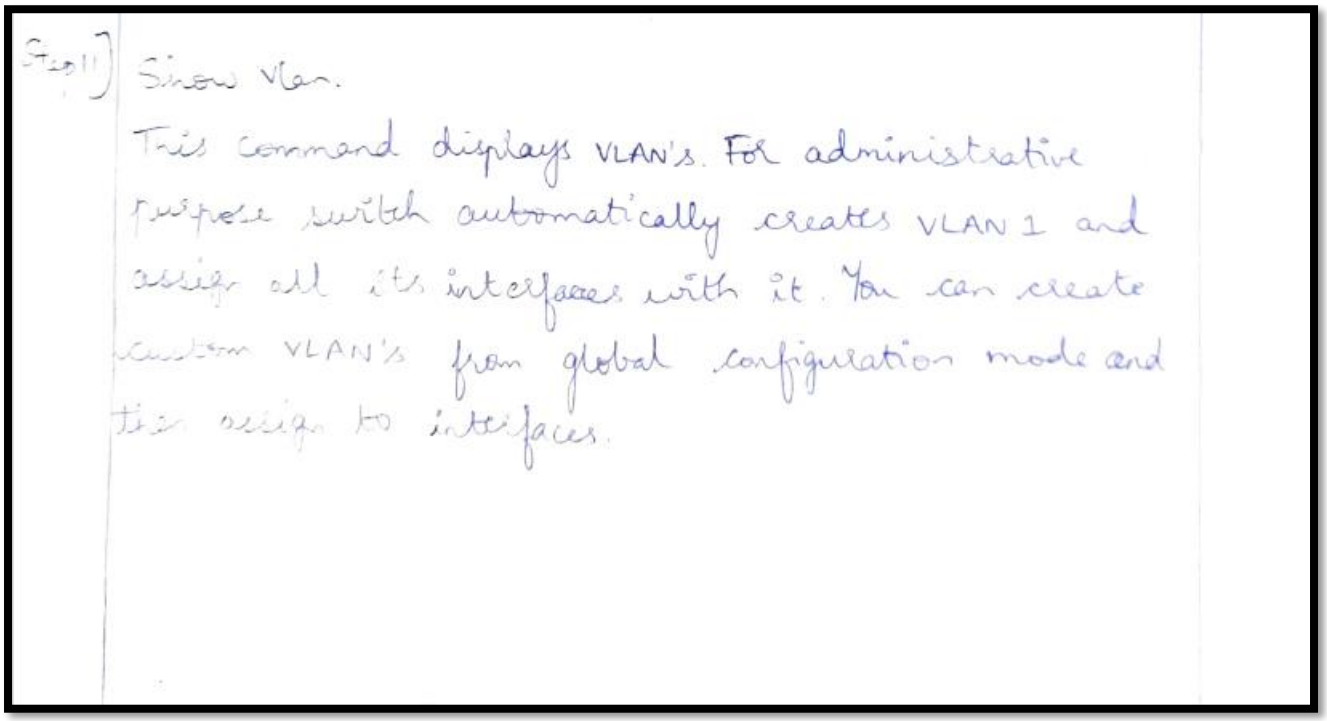

20181CSE0621 Sai Ram. K
Switch2
Physical
Config
CLI
Attributes
IOS Command Line Interface
sl#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
sl#show running-config
Building configuration...
Current configuration : 1064 bytes
!
version 12.1
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname sl
!
!
!
!
!
!
spanning-tree mode pvst
spanning-tree extend system-id
!
interface FastEthernet0/1
!
interface FastEthernet0/2
--More--
Ctrl+F6 to exit CLI focus
Copy
Paste
Top

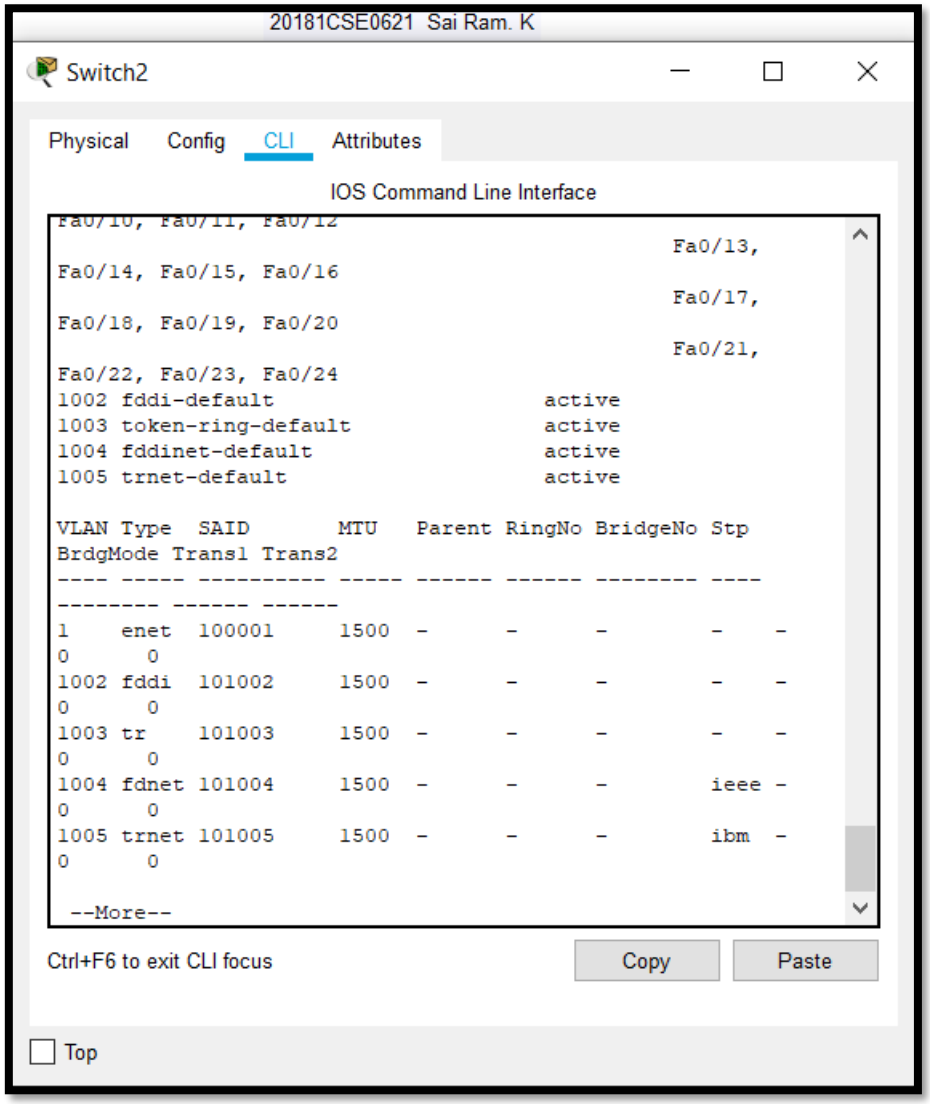
Step 10.


Step 10] Display the status of connected interfaces on switch.
To check status of the connected interfaces use the show ip brief command. Press spacebar to advance to the end of list.
S1 # show ip interface brief.


20181CSE0621 Sai Ram. K
Switch2
Physical
Config
CLI
Attributes
IOS Command Line Interface
sl#show ip interface brief
Interface IP-Address OK? Method Status
Protocol
FastEthernet0/1 unassigned YES manual down
down
FastEthernet0/2 unassigned YES manual down
down
FastEthernet0/3 unassigned YES manual down
down
FastEthernet0/4 unassigned YES manual down
down
FastEthernet0/5 unassigned YES manual down
down
FastEthernet0/6 unassigned YES manual down
down
FastEthernet0/7 unassigned YES manual down
down
FastEthernet0/8 unassigned YES manual down
down
FastEthernet0/9 unassigned YES manual down
down
FastEthernet0/10 unassigned YES manual down
down
FastEthernet0/11 unassigned YES manual down
down
FastEthernet0/12 unassigned YES manual down
down
Ctrl+F6 to exit CLI focus
Copy
Paste
Top

Step 11.

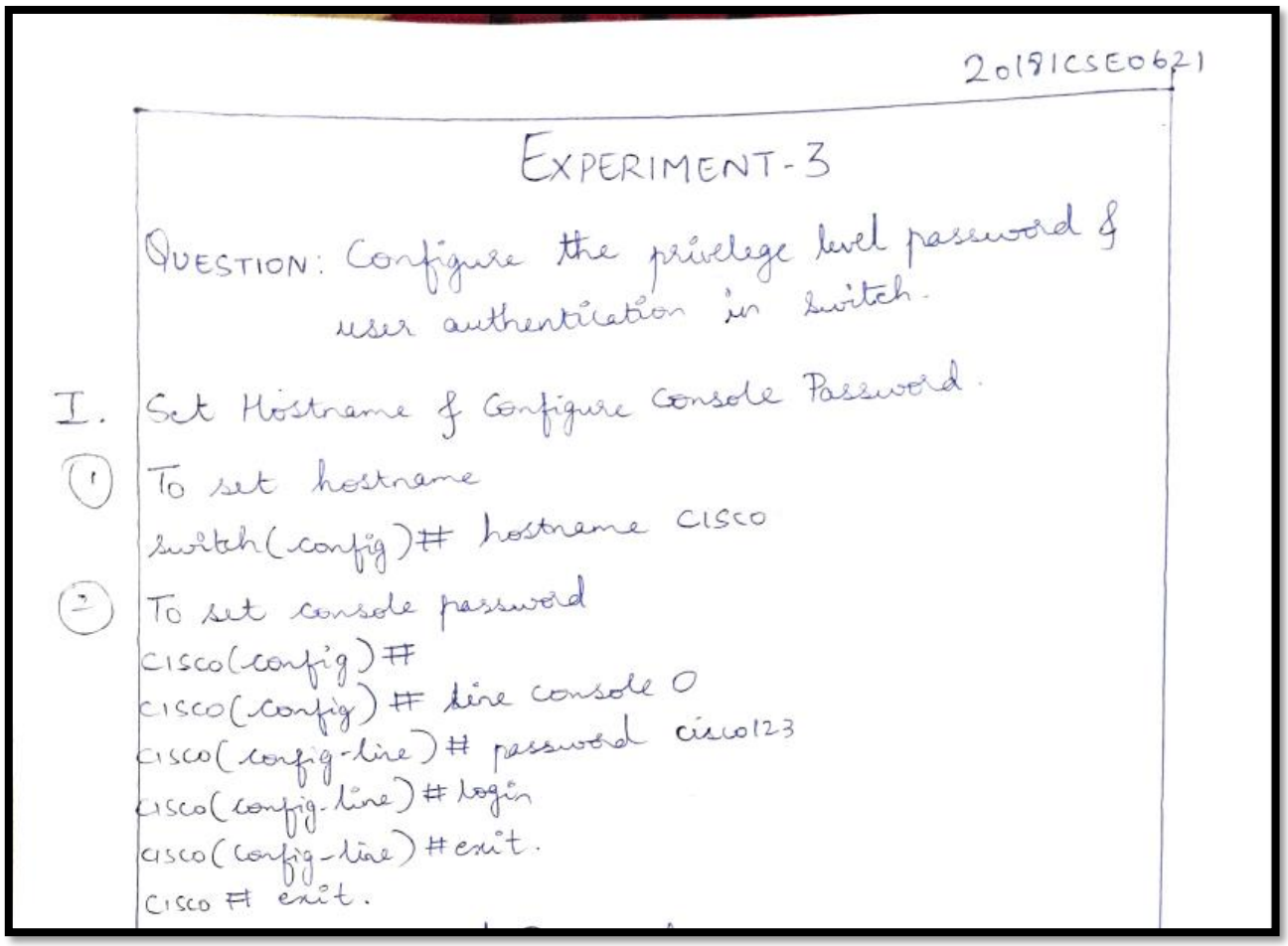

Step 11) Show Vlan.
This command displays VLAN's. For administrative purpose switch automatically creates VLAN 1 and assign all its interfaces with it. You can create custom VLAN's from global configuration mode and then assign to interfaces.


20181CSE0621 Sai Ram. K
Switch2
Physical
Config
CLI
Attributes
IOS Command Line Interface
Fa0/10, Fa0/11, Fa0/12
Fa0/13,
Fa0/14, Fa0/15, Fa0/16
Fa0/17,
Fa0/18, Fa0/19, Fa0/20
Fa0/21,
Fa0/22, Fa0/23, Fa0/24
1002 fddi-default active
1003 token-ring-default active
1004 fddinet-default active
1005 trnet-default active
VLAN Type SAID MTU Parent RingNo BridgeNo Stp
BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ----
-------- ------ ------
1 enet 100001 1500 - - - - -
0 0
1002 fddi 101002 1500 - - - - -
0 0
1003 tr 101003 1500 - - - - -
0 0
1004 fdnet 101004 1500 - - - ieee -
0 0
1005 trnet 101005 1500 - - - ibm -
0 0
--More--
Ctrl+F6 to exit CLI focus
Copy
Paste
Top

Experiment – 3

Configure the privilege level password and user authentication in switch.

Step 1.



20181CSE0621

EXPERIMENT-3

QUESTION: Configure the privilege level password & user authentication in switch.

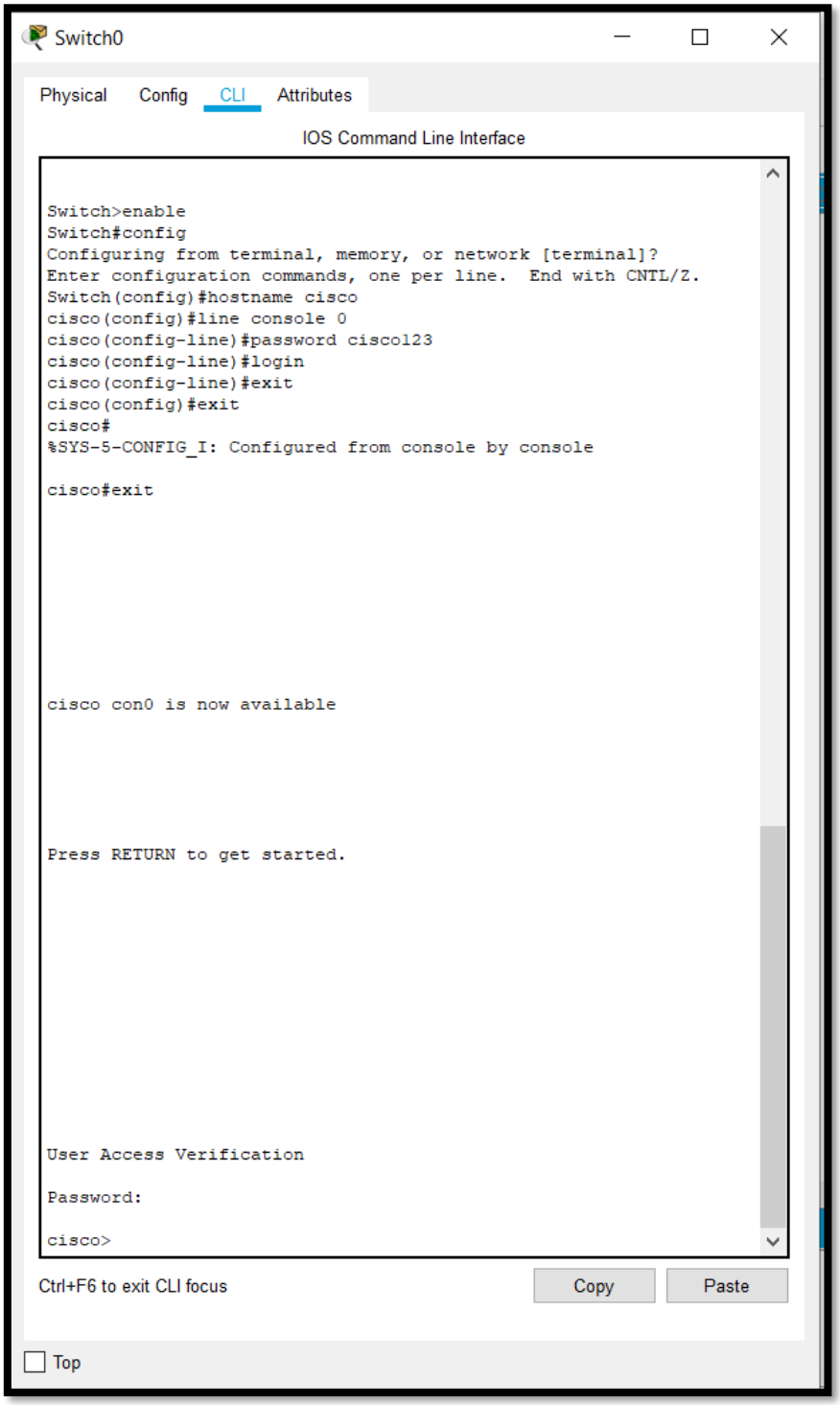
I. Set Hostname & Configure Console Password.

(1) To set hostname

```
switch(config)# hostname CISCO
```

(2) To set console password

```
CISCO(config)#
CISCO(config)# line console 0
CISCO(config-line)# password cisco123
CISCO(config-line)# login
CISCO(config-line)# exit.
CISCO # exit.
```


Switch0
Physical
Config
CLI
Attributes
IOS Command Line Interface
Switch>enable
Switch#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#hostname cisco
cisco(config)#line console 0
cisco(config-line)#password cisco123
cisco(config-line)#login
cisco(config-line)#exit
cisco(config)#exit
cisco#
%SYS-5-CONFIG_I: Configured from console by console
cisco#exit
cisco con0 is now available
Press RETURN to get started.
User Access Verification
Password:
cisco>
Ctrl+F6 to exit CLI focus
Copy
Paste
Top

Step 2.

II. Set Privelege level Password.

(1) Set a privelege password

!!! clear Text Password not encrypted (less priority)

cisco(config)# enable password muscat

!!! Encrypted password more priority

cisco(config)# enable secret nizwa.

(2) Verify the privileged password

CISCO # exit

CISCO con0 is now available

press return to get started.

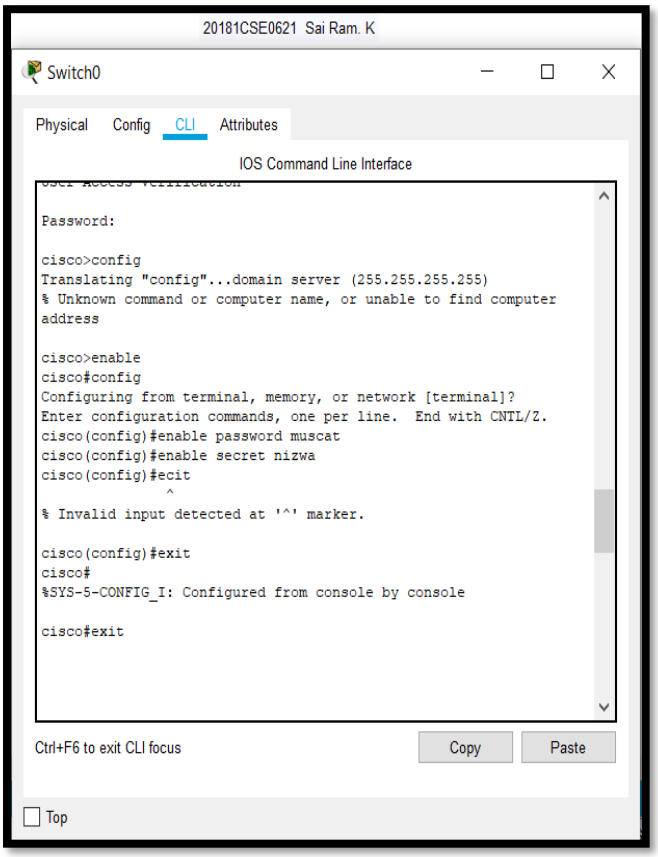
User Access Verification

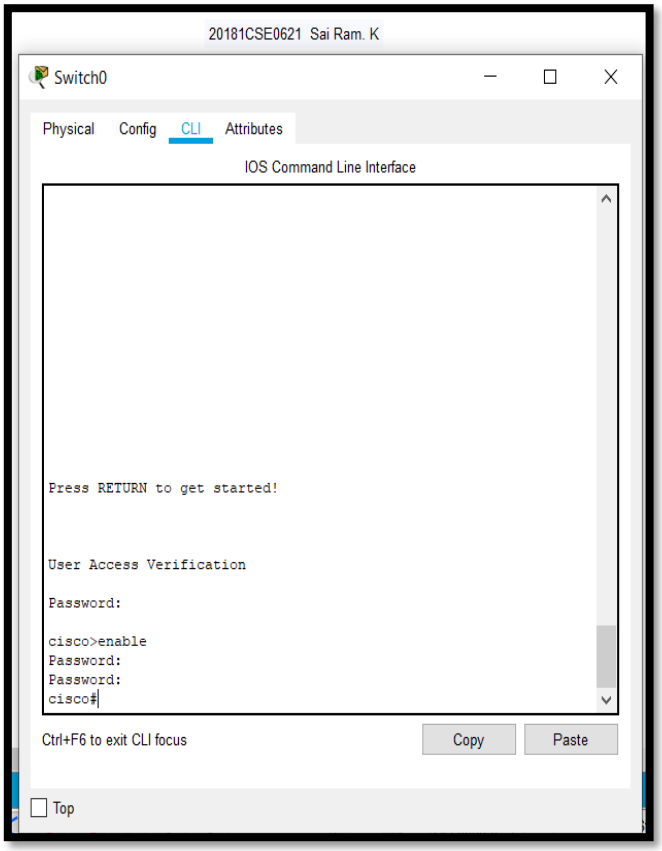
password :

Cisco > enable

password :

5





Step 3.

20181CSE0621

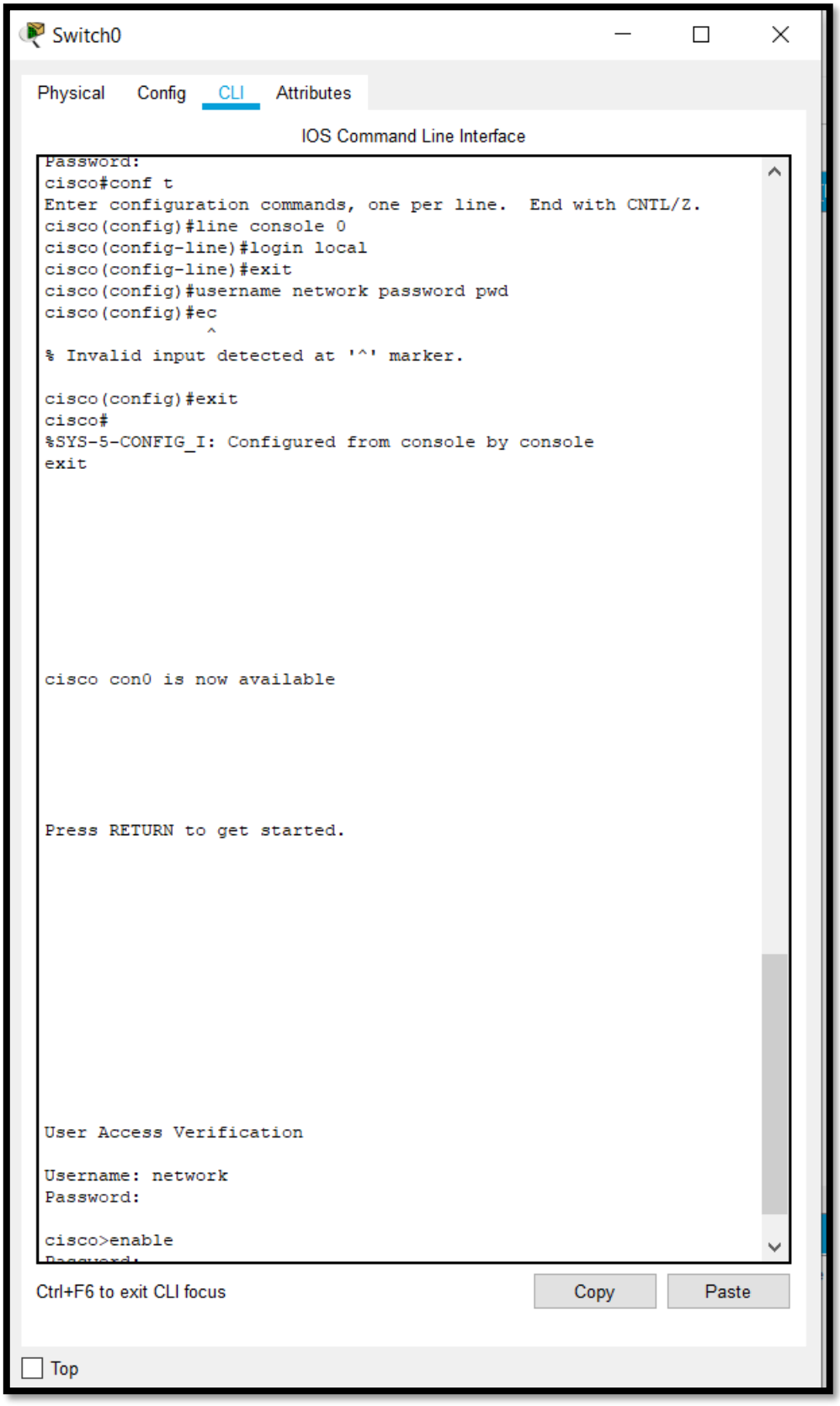
III Set User Authentication in switch.

(1) Set user Authentication

```
cisco# conf t
cisco(config)# line console 0
cisco(config)# login local
cisco(config-line)# exit
cisco(config)# username network password pwd.
```

(2) Verify the authentication.

```
cisco(config)# exit
cisco# exit
User Access Verification
Username: network
Password:
cisco> enable
Password:
```


Switch0
Physical
Config
CLI
Attributes
IOS Command Line Interface
Password:
cisco#conf t
Enter configuration commands, one per line. End with CNTL/Z.
cisco(config)#line console 0
cisco(config-line)#login local
cisco(config-line)#exit
cisco(config)#username network password pwd
cisco(config)#ec
^
% Invalid input detected at '^' marker.
cisco(config)#exit
cisco#
%SYS-5-CONFIG_I: Configured from console by console
exit
cisco con0 is now available
Press RETURN to get started.
User Access Verification
Username: network
Password:
cisco>enable
Ctrl+F6 to exit CLI focus
Copy
Paste
Top

Experiment – 4

Configure the DHCP Server and wireless router and check the connectivity

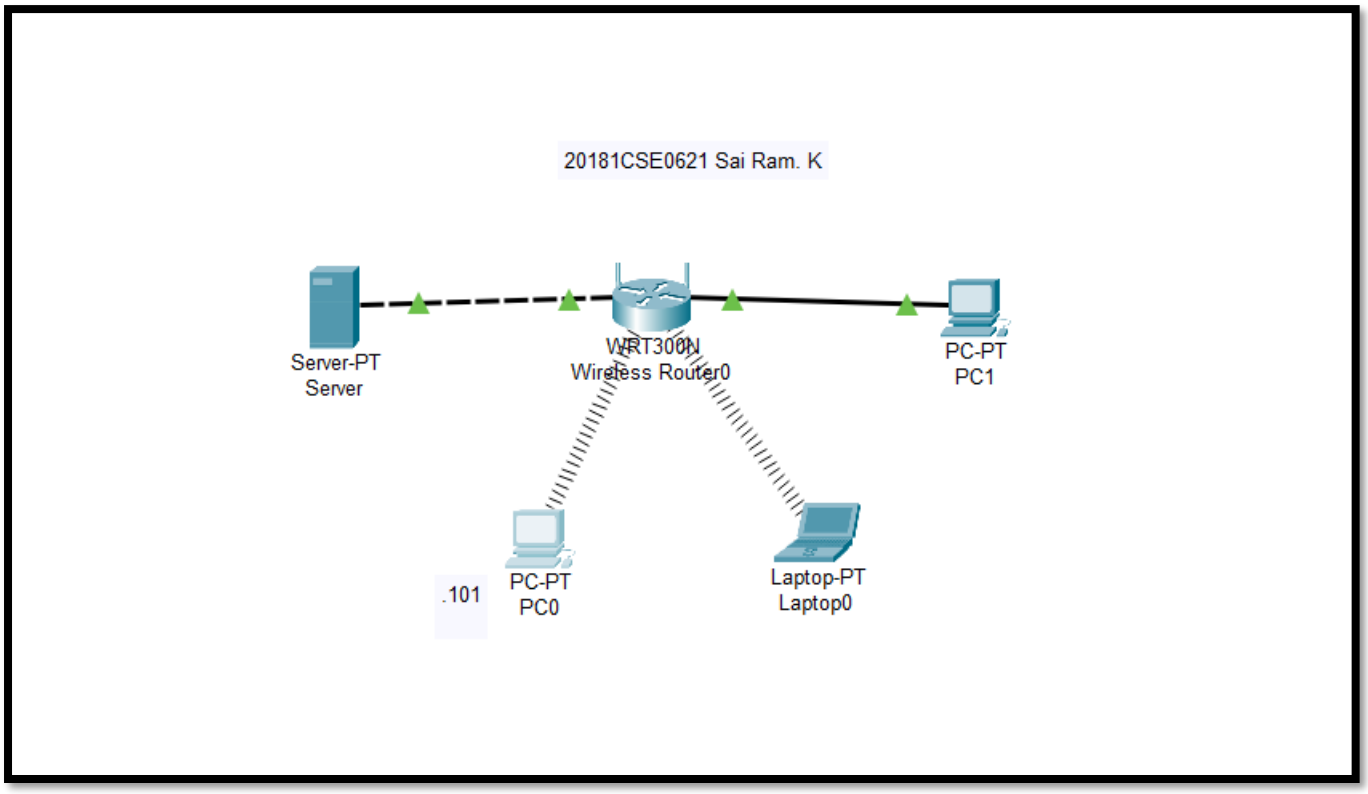
Step 1.

20181CSE0621

EXPERIMENT-04

QUESTION: Configure the DHCP server & wireless router & check connectivity.

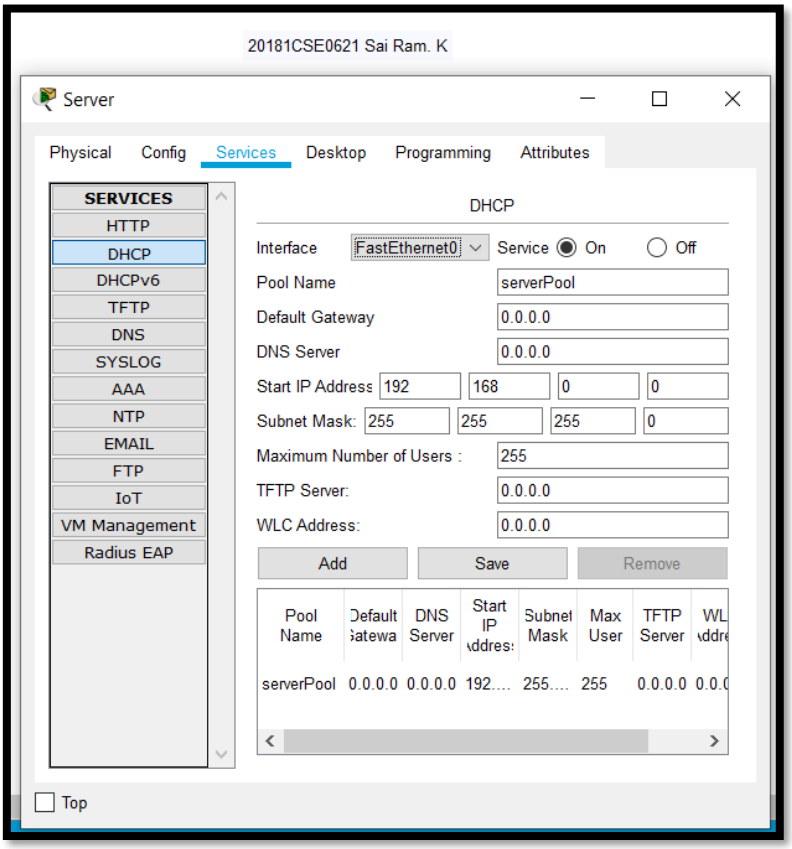
Step1] Configure the DHCP server.



Step 2 & 3.

Step2] In server configure the fast Ethernet

Step3] Configure the wireless router

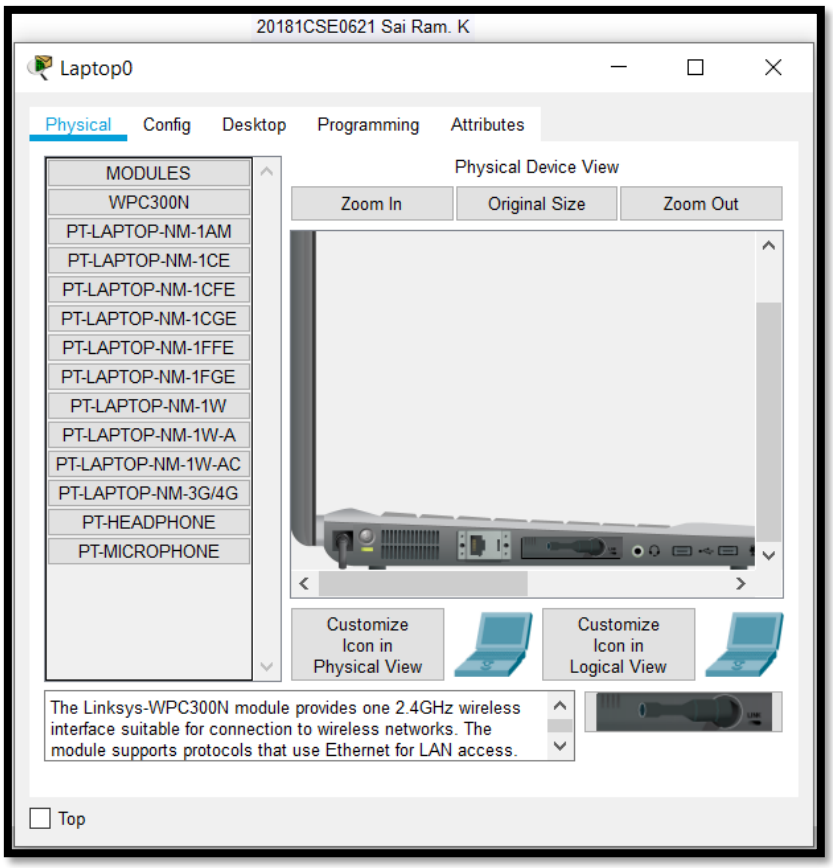


Step 4,5 & 6:

Step4] Laptop remove the wired NIC card to wireless card.

Step5] Laptop connect the wireless.

Step6] Connect laptop with wireless router.

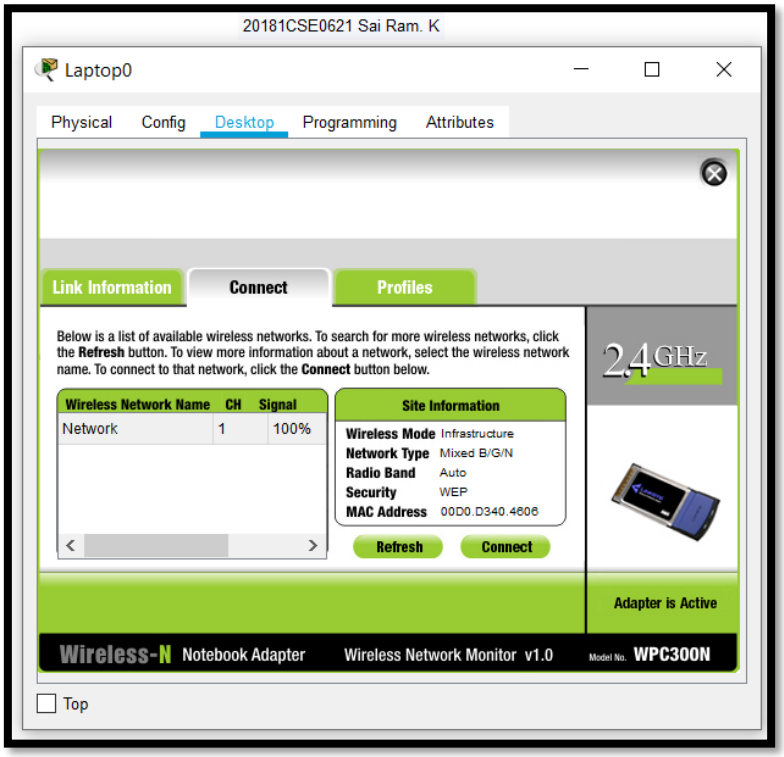


Step 7 & 8.

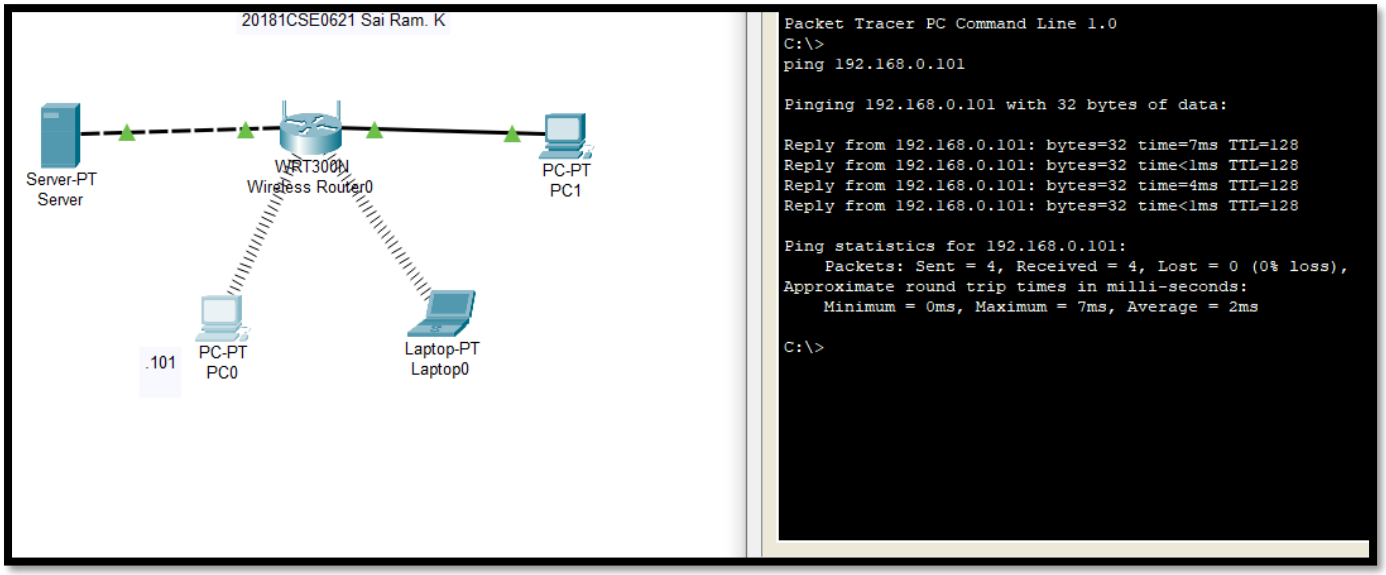
Step7] Enter the web key.

Step8] PC remove the wired NIC & inseet wireless card.

NOTE: repeat 5,6,7


20181CSE0621 Sai Ram. K
Laptop0
Physical
Config
Desktop
Programming
Attributes
Link Information
Connect
Profiles
Below is a list of available wireless networks. To search for more wireless networks, click the Refresh button. To view more information about a network, select the wireless network name. To connect to that network, click the Connect button below.
2.4GHz
Wireless Network Name
CH
Signal
Network
1
100%
Site Information
Wireless Mode Infrastructure
Network Type Mixed B/G/N
Radio Band Auto
Security WEP
MAC Address 00D0.D340.4606
Refresh
Connect
Adapter is Active
Wireless-N Notebook Adapter
Wireless Network Monitor v1.0
Model No. WPC300N
Top

Step 9 &10.


20181CSE0621 Sai Ram. K
Server-PT
Server
WRT300N
Wireless Router0
PC-PT
PC1
PC-PT
PC0
Laptop-PT
Laptop0
.101
Packet Tracer PC Command Line 1.0
C:\>
ping 192.168.0.101
Pinging 192.168.0.101 with 32 bytes of data:
Reply from 192.168.0.101: bytes=32 time=7ms TTL=128
Reply from 192.168.0.101: bytes=32 time<1ms TTL=128
Reply from 192.168.0.101: bytes=32 time=4ms TTL=128
Reply from 192.168.0.101: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.0.101:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 7ms, Average = 2ms
C:\>

Experiment – 5

Configure the telnet protocol using cisco packet tracer

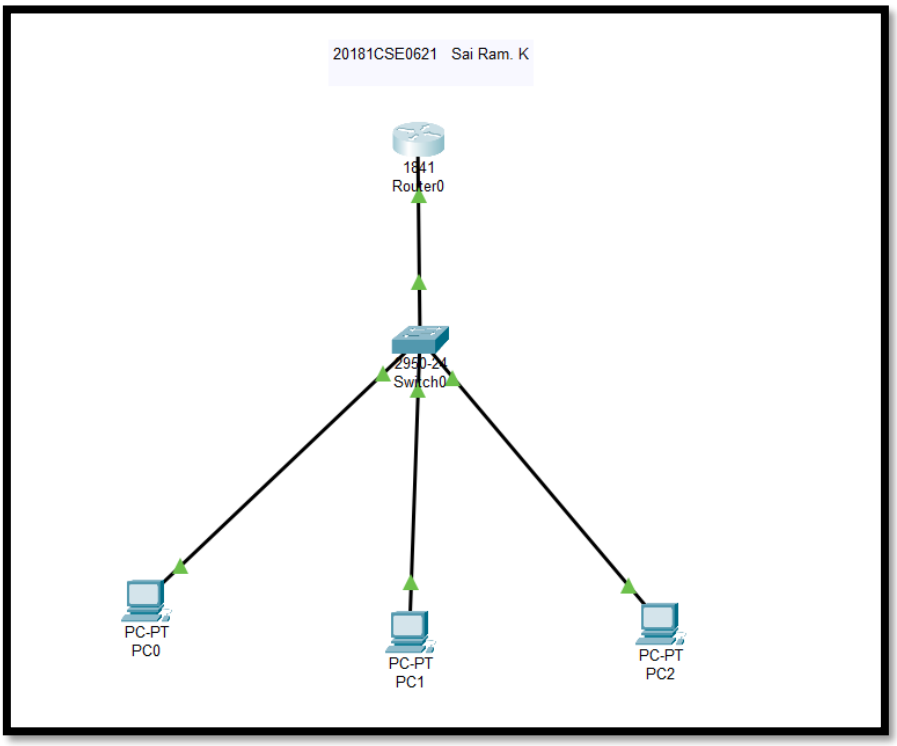
Step 1.

20181CSE0621

EXPERIMENT-05

QUESTION: Configure the Telnet protocol using Cisco packet Tracer.

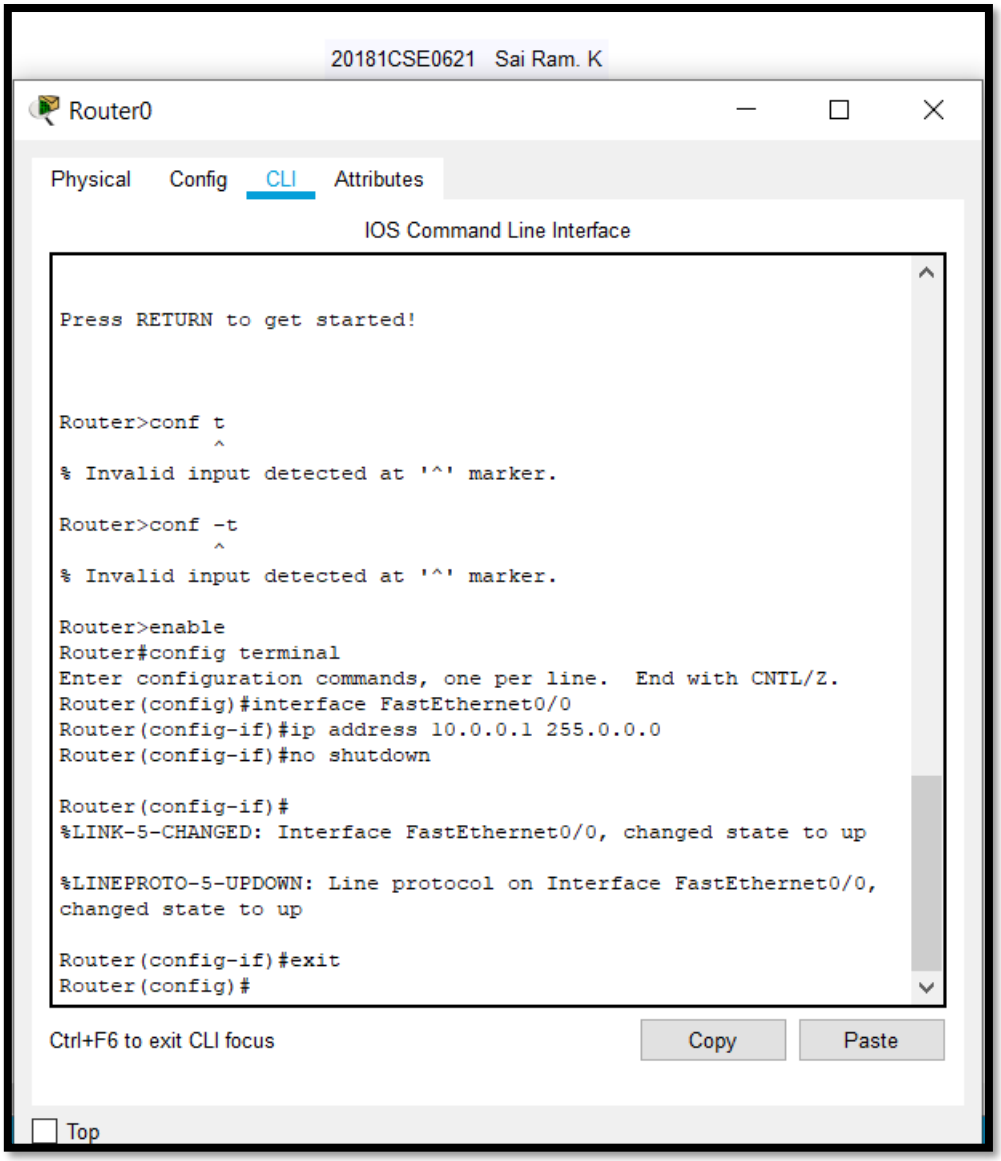
Step1] Draw a topology as shown and assign the IP addresses to all PC's.



Step 2.

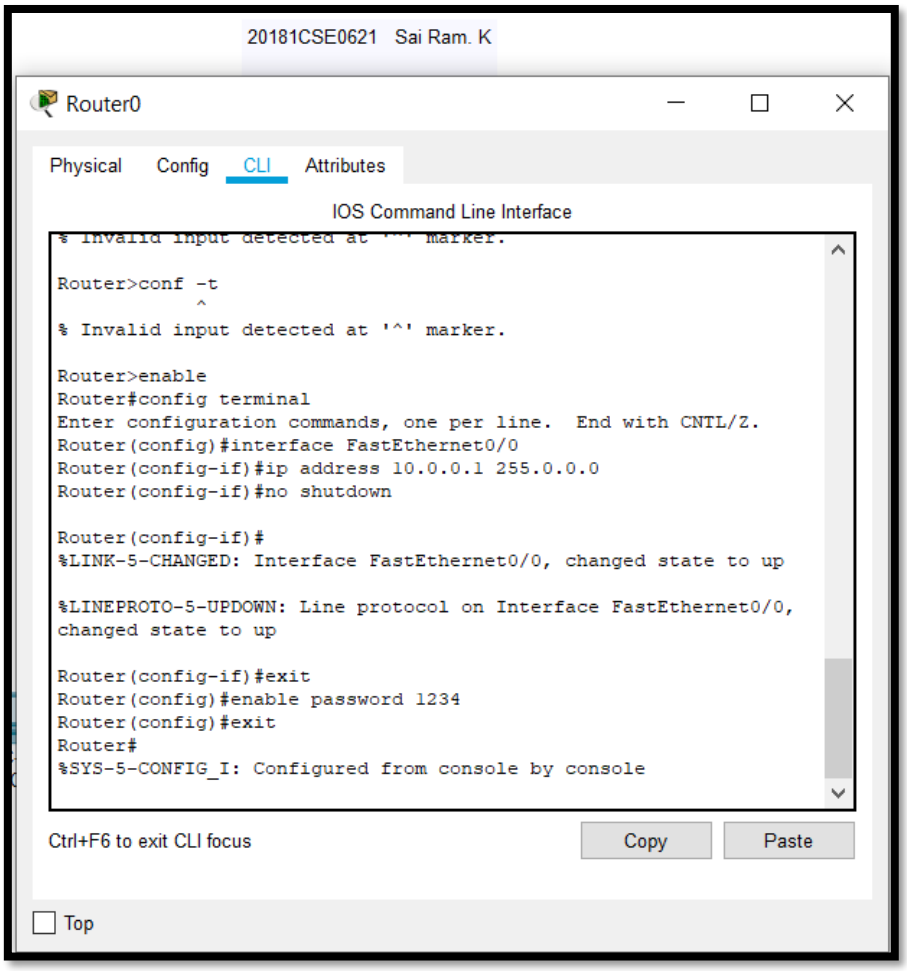
Step 2] Configure IP address to router.

```
Router(config)# interface fastEthernet 0/0.
Router(config-if)# ip address 10.0.0.1 255.0.0.0
Router(config-if)# no shutdown.
Router(config-if)# exit.
```



Step 3.

Step3] To set privelege mode password.
Click on router and go to CLI tab & type below.
Router(config)# enable password 1234
router(config)# exit.



Step 4.

Step 4] To configure Telnet

```
Router# conf t
Enter configuration commands one per line. End with ctrl+z
Router(config)# line vty 0 4
Router(config)# password cisco
Router(config-line) # login
Router(config-line) ## exit
```

8

20181CSE0621 Sai Ram. K

Router0

Physical Config CLI Attributes

IOS Command Line Interface

```
Router#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up

Router(config-if)#exit
Router(config)#enable password 1234
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#line vty 0 4
Router(config-line)#password cisco
Router(config-line)#login
Router(config-line)#exit
Router(config)#
```

Ctrl+F6 to exit CLI focus Copy Paste

Top

Step 5.

20181CSE0621.

Steps) To check Telnet configuration

Router # sh run.

20181CSE0621 Sai Ram. K

Router0

Physical Config CLI Attributes

IOS Command Line Interface

```
Router#sh run
Building configuration...

Current configuration : 598 bytes
!
version 12.4
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Router
!
!
!
enable password 1234
!
!
!
!
!
!
ip cef
no ipv6 cef
 --More--
```

Ctrl+F6 to exit CLI focus Copy Paste

Top

Step 6.

Step6] To access cisco router via telnet connection from any PC.

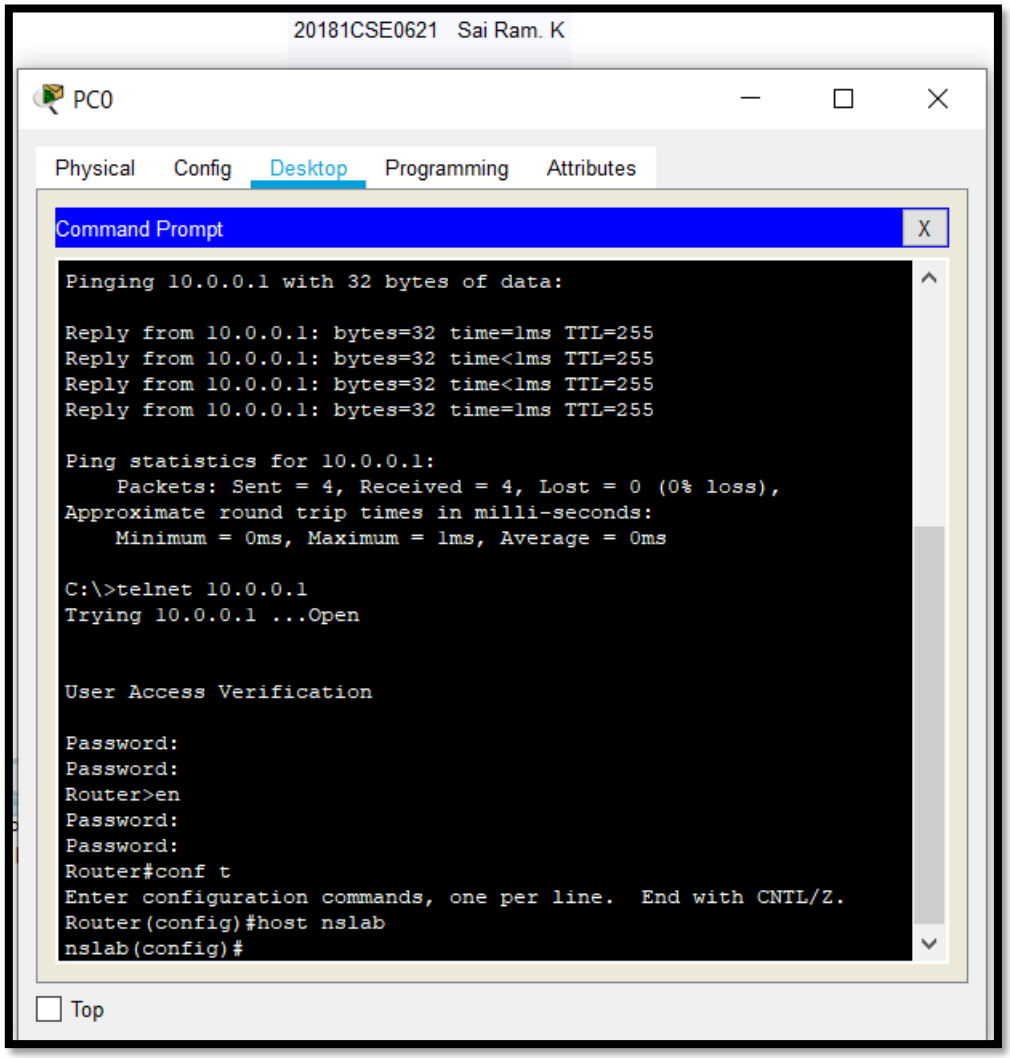
Click on any PC

Click on desktop

Select command prompt & type the following commands

PC > ping 10.0.0.1

PC > telnet 10.0.0.1



Experiment – 6

Configure Static routing using Cisco Packet Tracer.

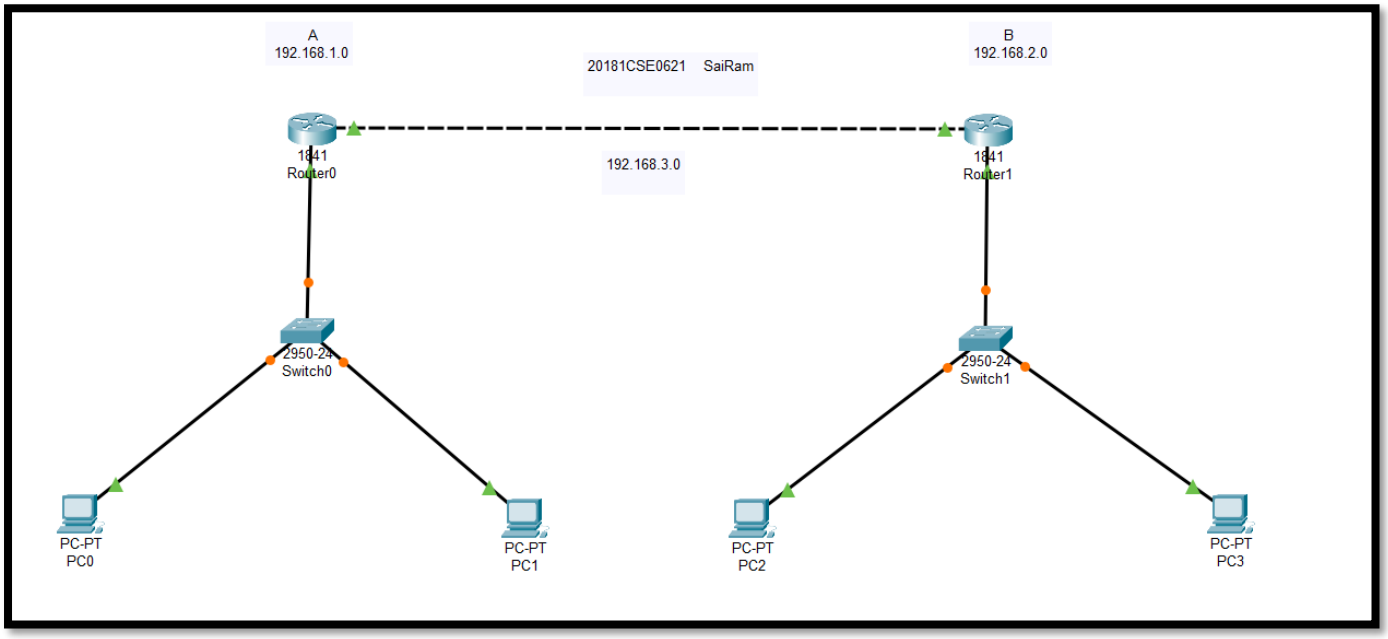
Step 1.

20181CSE0621

EXPERIMENT-06

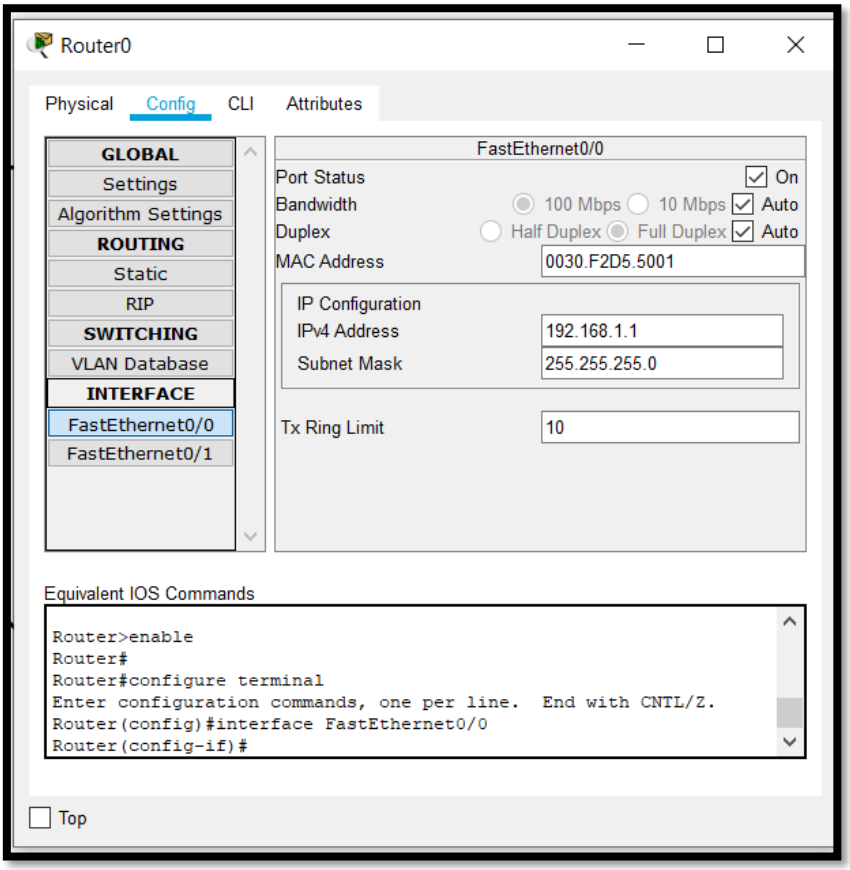
QUESTION: Configure the static routing using Cisco Packet Tracer.

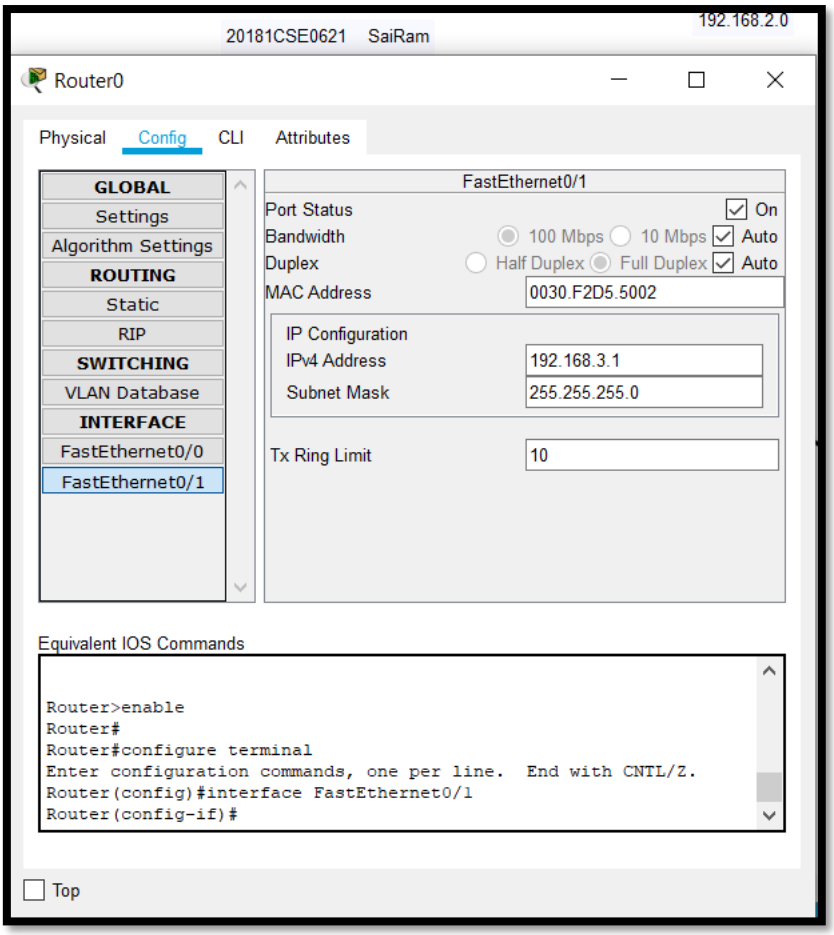
Step1] Draw the topology as shown below and assign IP addresses to all PC's.



Step 2.

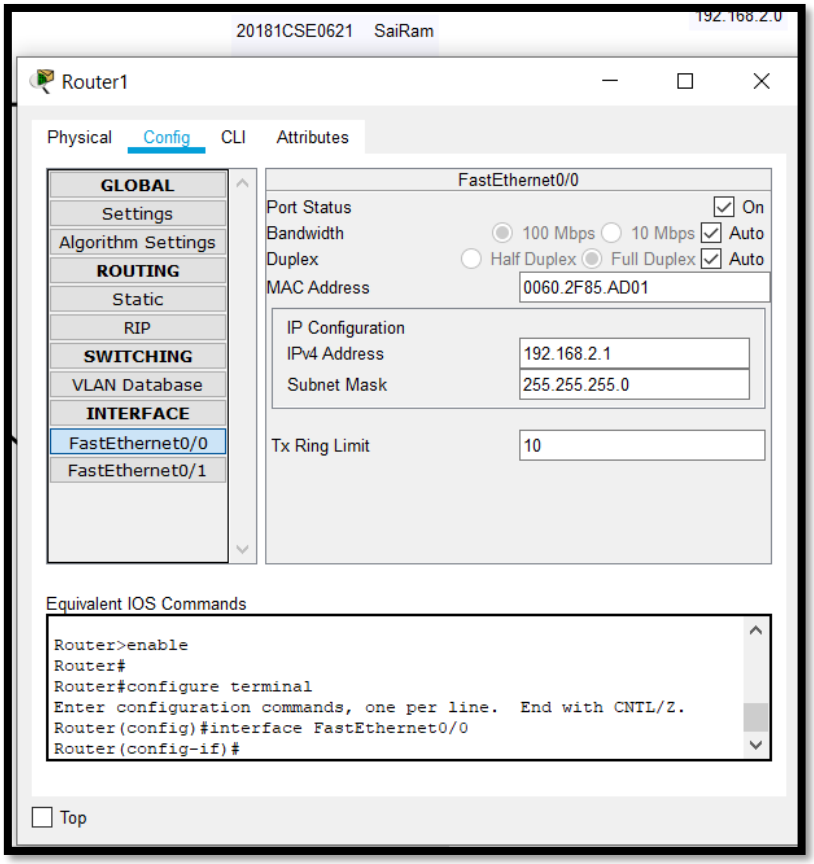
Step2] Configure IP address to router 1.
For Fast Ethernet 0/0, and
For Fast Ethernet 0/1.

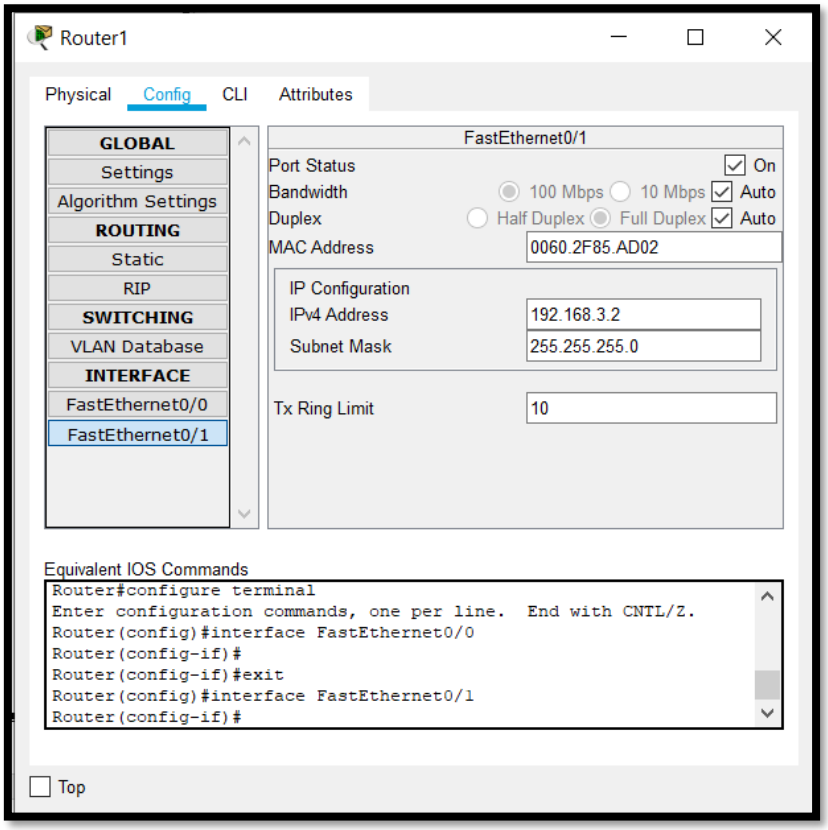

Router0
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
FastEthernet0/0
FastEthernet0/1
FastEthernet0/0
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
0030.F2D5.5001
IP Configuration
IPv4 Address
192.168.1.1
Subnet Mask
255.255.255.0
Tx Ring Limit
10
Equivalent IOS Commands
Router>enable
Router#
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#
Top


20181CSE0621 SaiRam
192.168.2.0
Router0
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
FastEthernet0/0
FastEthernet0/1
FastEthernet0/1
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
0030.F2D5.5002
IP Configuration
IPv4 Address
192.168.3.1
Subnet Mask
255.255.255.0
Tx Ring Limit
10
Equivalent IOS Commands
Router>enable
Router#
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/1
Router(config-if)#
Top

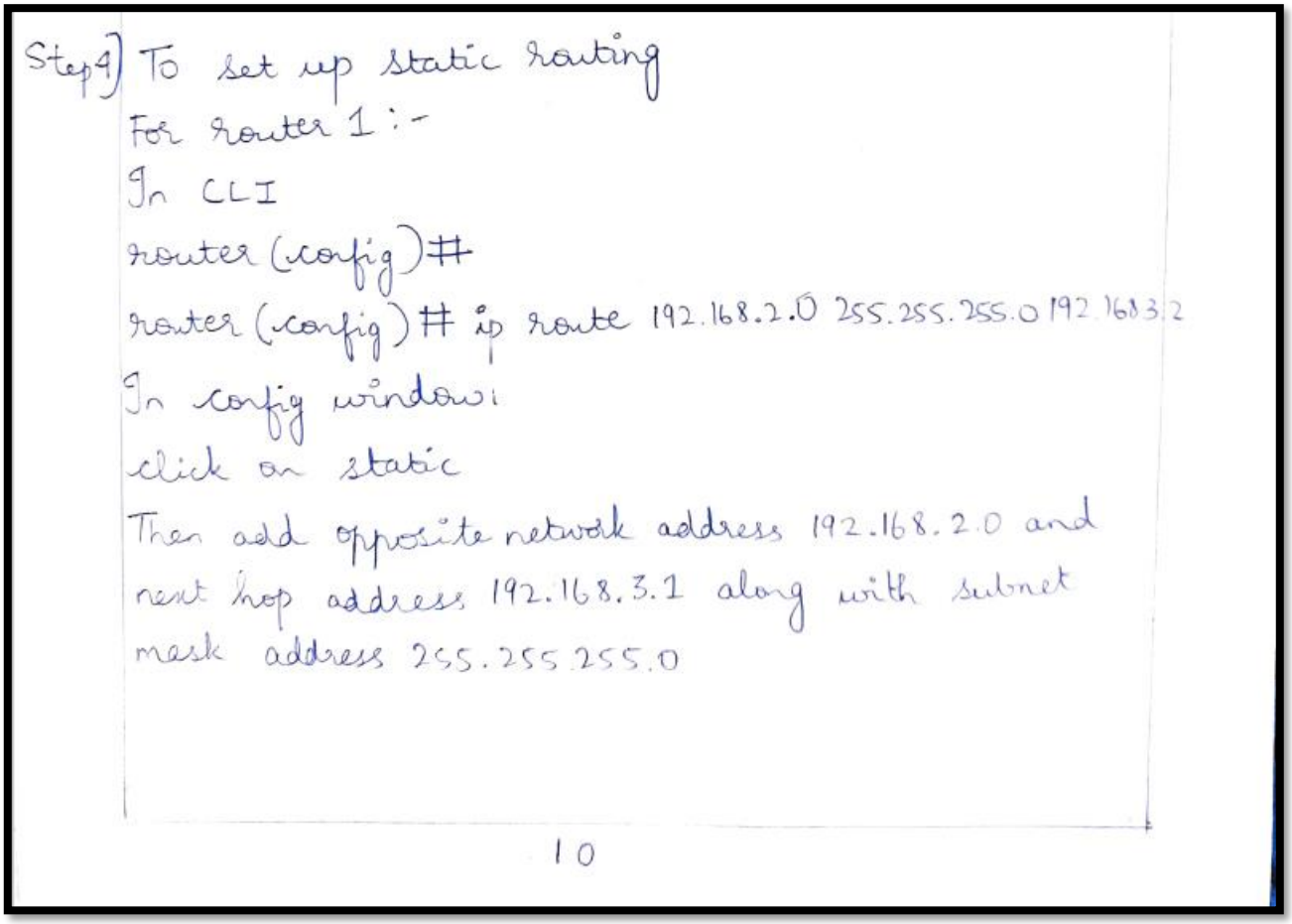
Step 3.

Step3] Configure IPaddress to router2
For Fast Ethernet 0/0, and
For Fast Ethernet 0/1.




Router1
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
FastEthernet0/0
FastEthernet0/1
FastEthernet0/1
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
0060.2F85.AD02
IP Configuration
IPv4 Address
192.168.3.2
Subnet Mask
255.255.255.0
Tx Ring Limit
10
Equivalent IOS Commands
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/1
Router(config-if)#
Top

Step 4.


Step4) To set up static routing
For router 1 :-
In CLI
router (config)#
router (config)# ip route 192.168.2.0 255.255.255.0 192.168.3.2
In config window:
click on static
Then add opposite network address 192.168.2.0 and
next hop address 192.168.3.1 along with subnet
mask address 255.255.255.0
10

20181CSE0621

For router 2 :

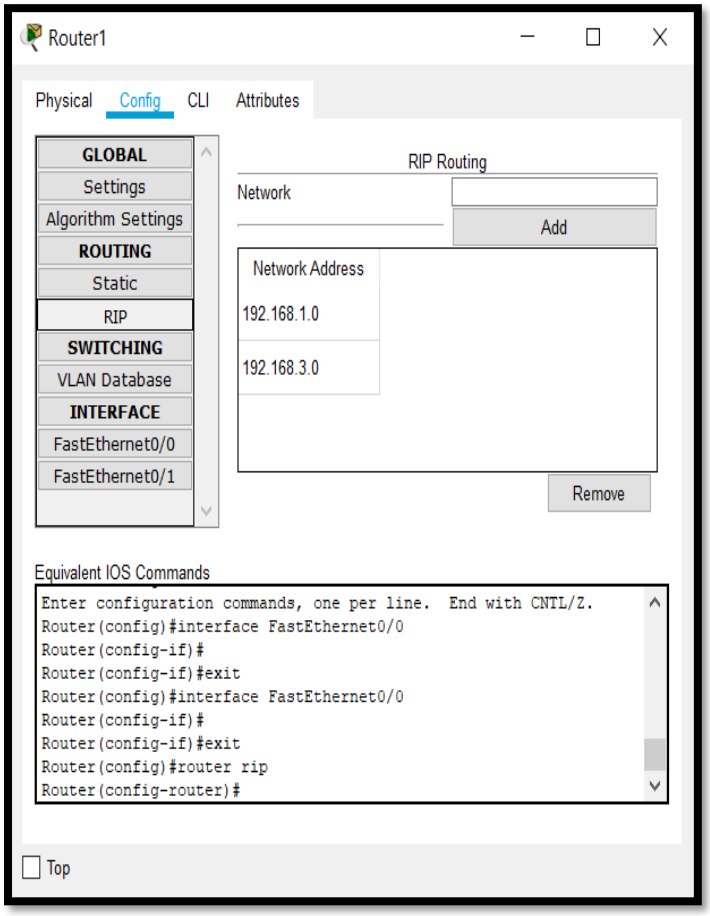
In CLI :-

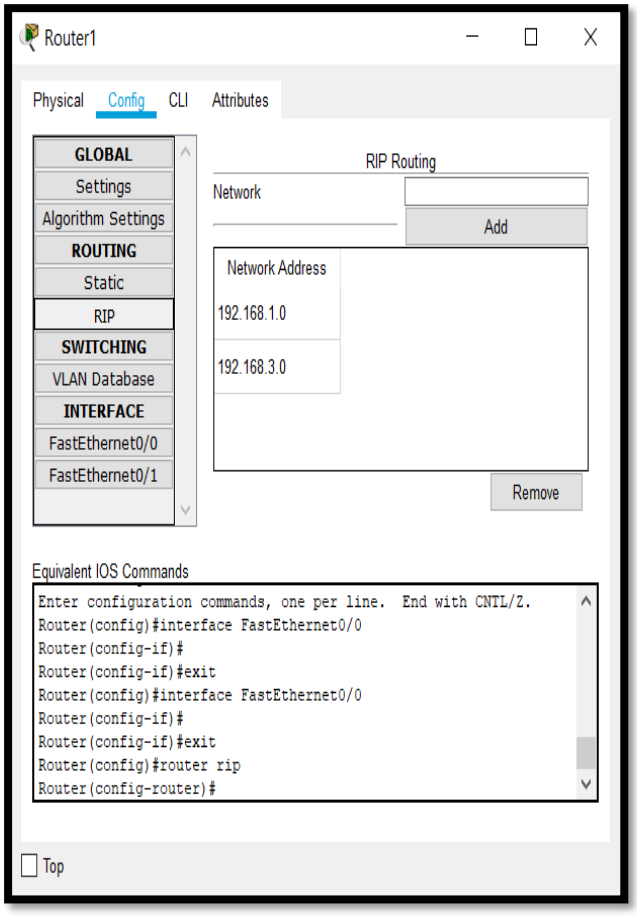
router (config)#

router (config)# ip route 192.168.1.0 255.255.255.0 192.168.3.1

In config window:

click on static . Add opposite Network address 192.168.3.0 & next hop address 192.168.3.1 along with subnet mask address 255.255.255.0 .



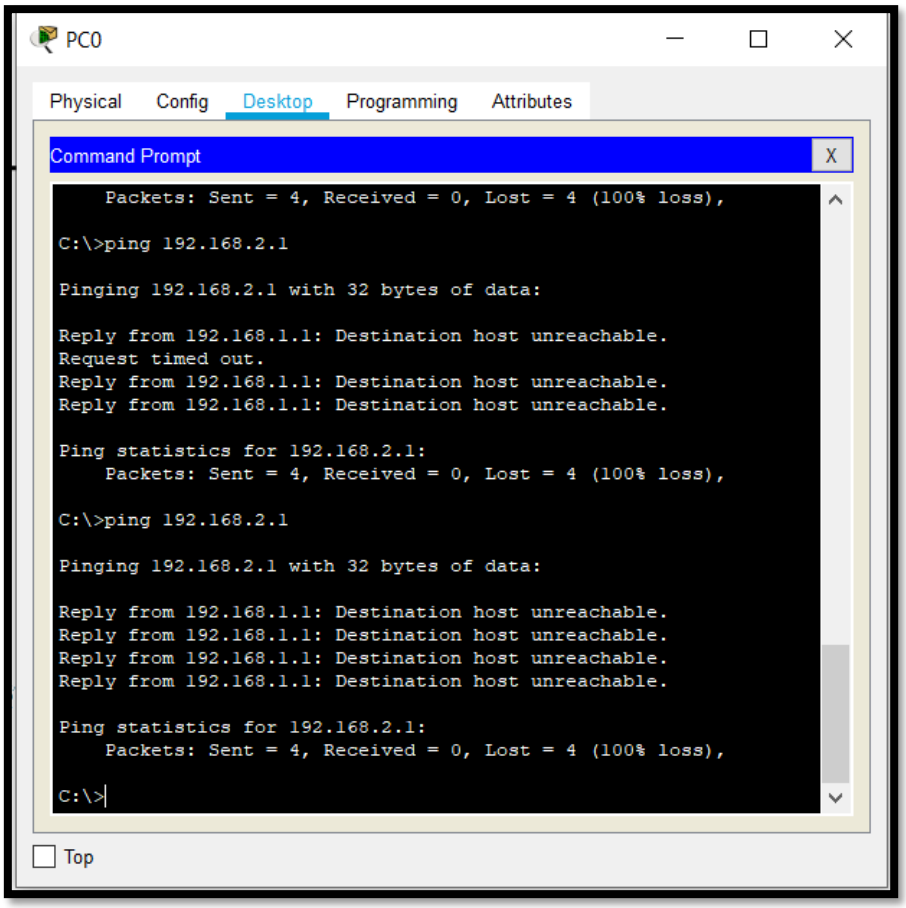


Step 5.

Step5] To check connectivity between the networks using static routing.

Click on any PC > desktop > select command prompt type the following.

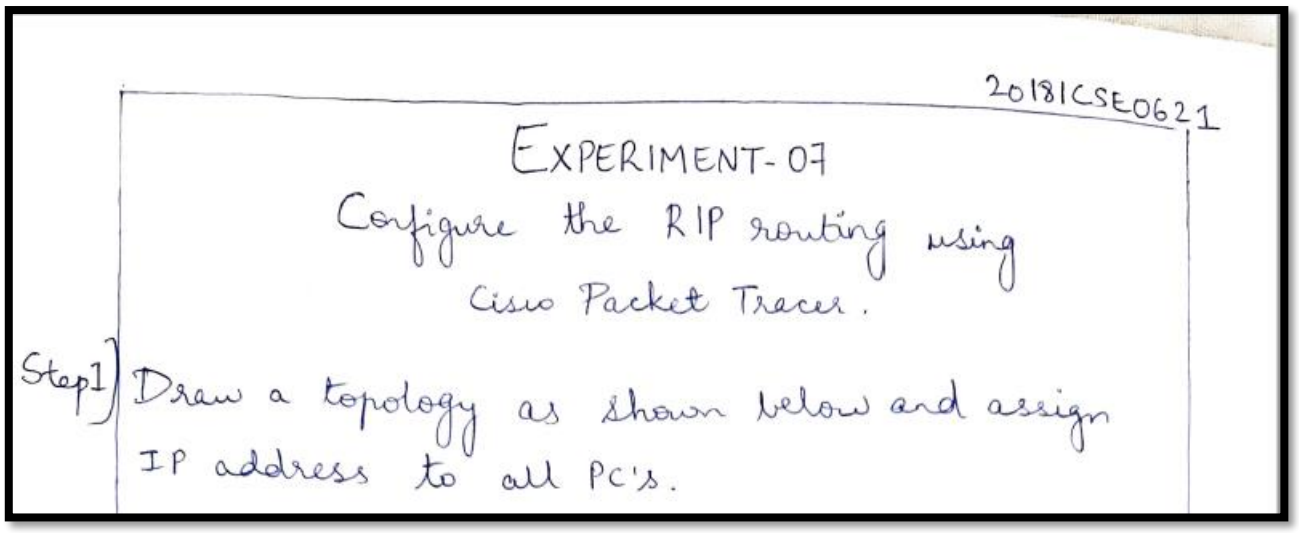
PC > ping 192.168.2.1

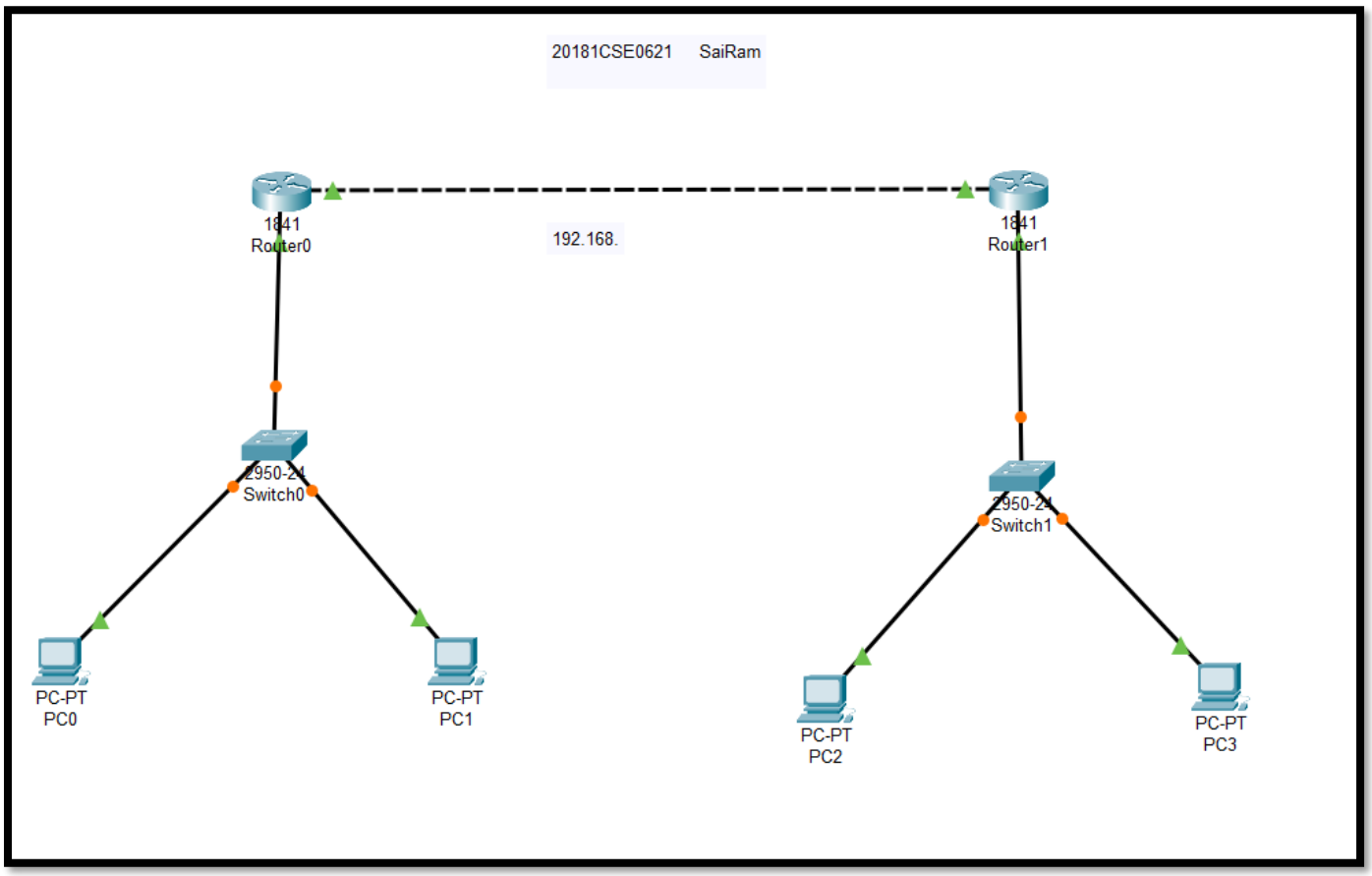


Experiment – 7

Configure RIP routing using Cisco Packet Tracer

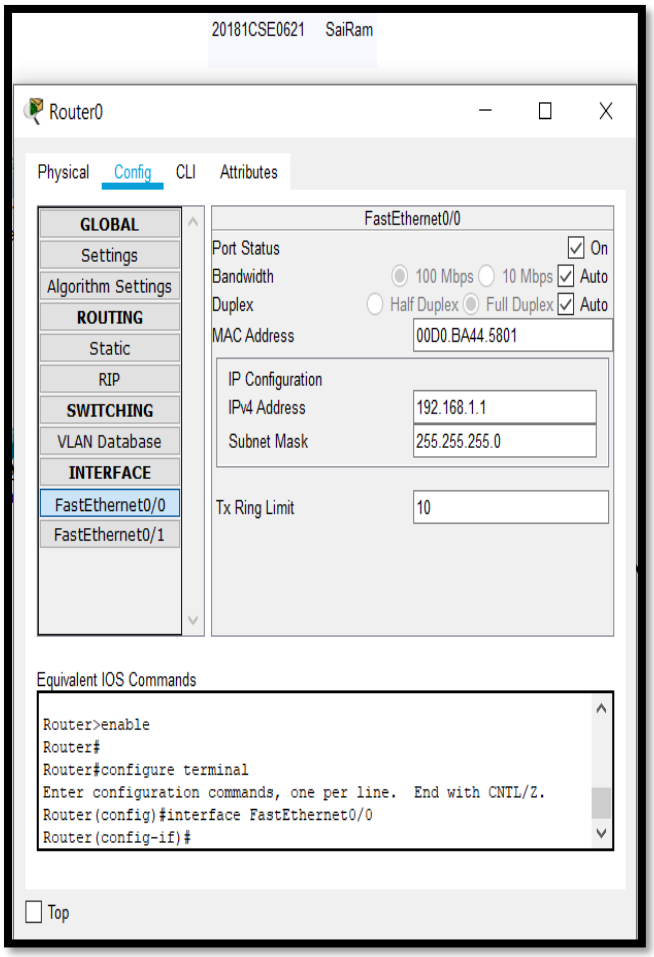
Step 1.

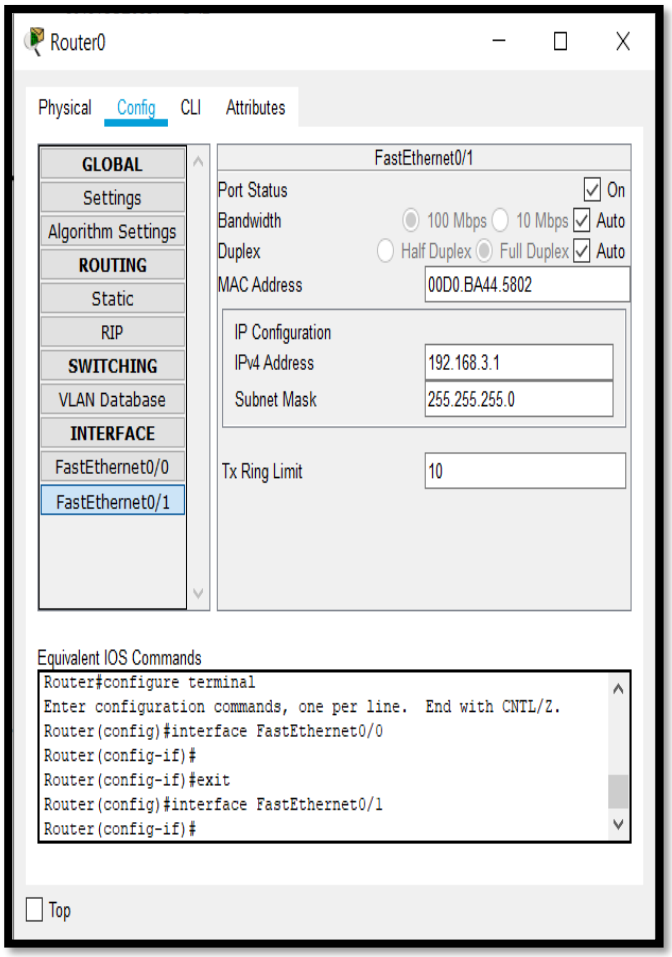




Step 2.

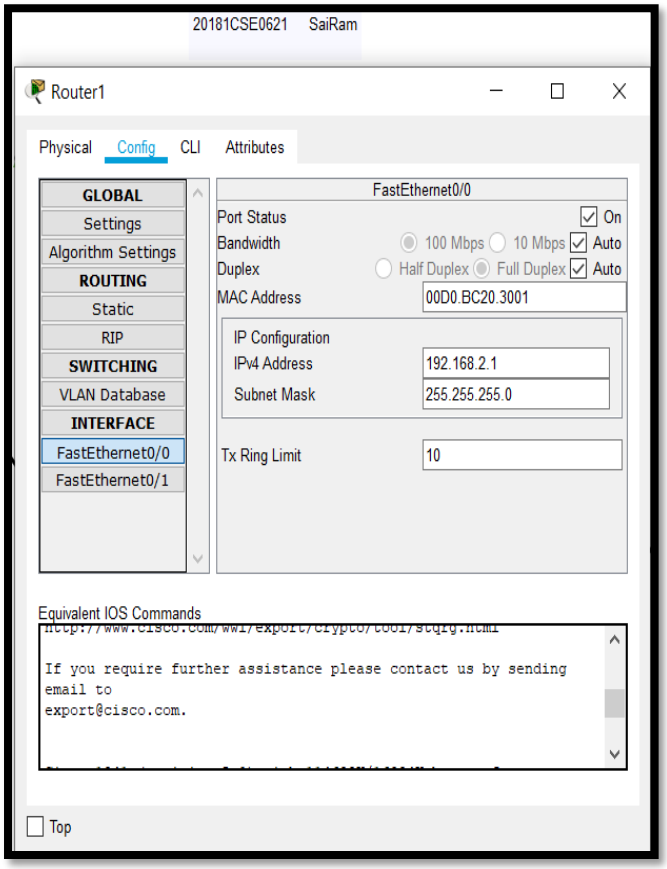
Step2] Configure IP address to router1.
For Fast Ethernet 0/0, and
For Fast Ethernet 0/1

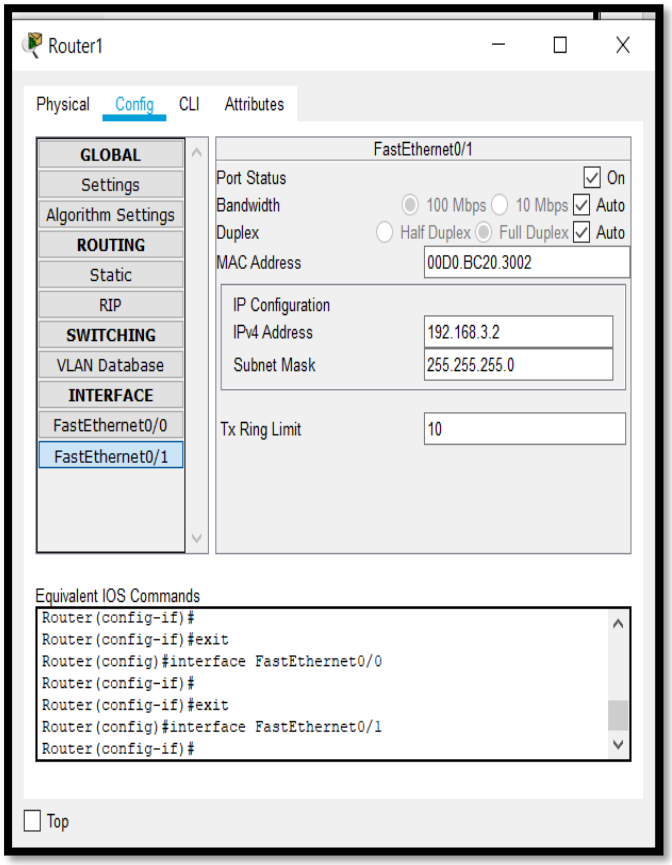




Step 3.

Step3] Configure IP address for router 2
For Fast Ethernet 0/0 and
for Fast Ethernet 0/1.





Step 4.

Step4) To set up Dynamic routing.

For router 1:

In CLI:

Router(config)# router rip

Router(config)# network 192.168.3.0

Router(config-router)# network 192.168.2.0

In config window:

click on RIP

Then add opposite Network address 192.168.2.0 and 192.168.3.0.

20181CSE0621

For router 2:

In CLI :-

Router (config) # router rip

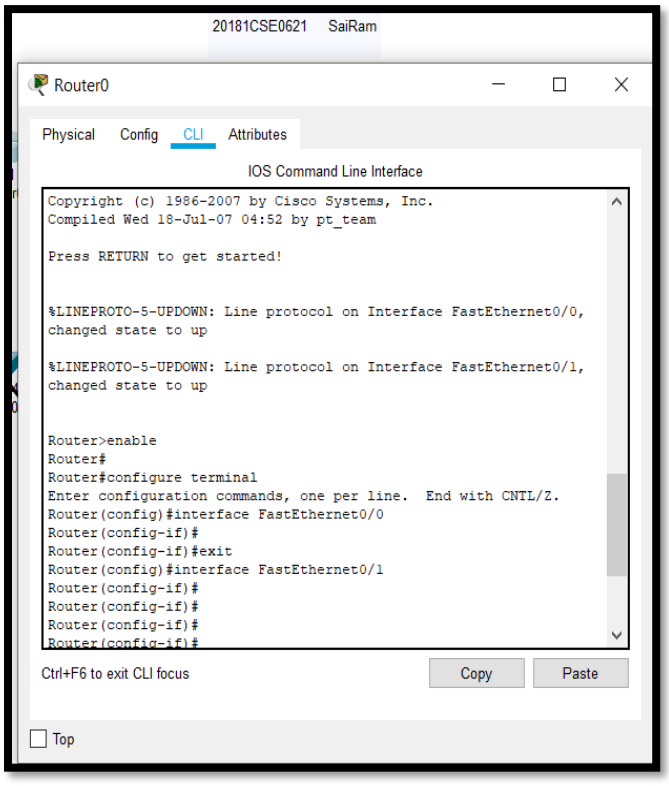
Router (config-router) # network 192.168.3.0

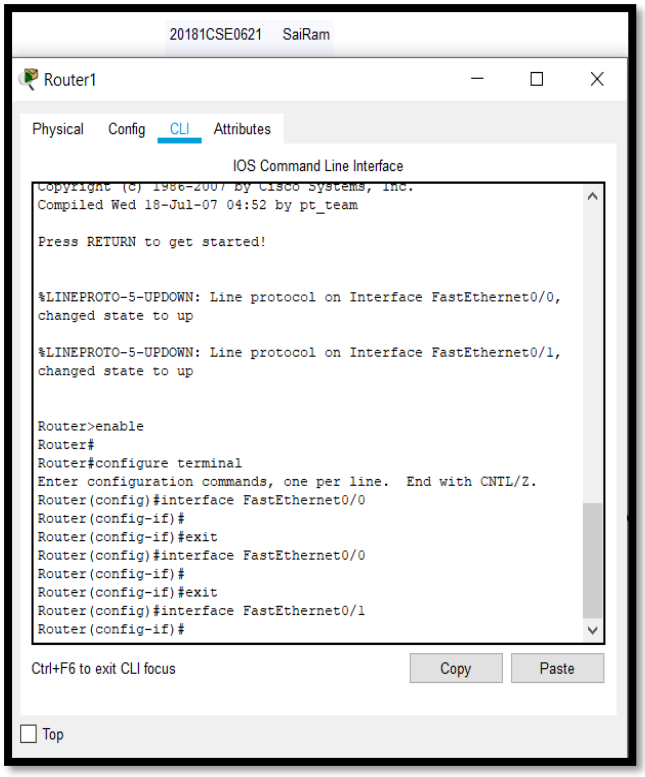
Router (config-router) # network 192.168.1.0

In config window:

Click on RIP

Then add opposite networks 192.168.1.0 & 192.168.3.0



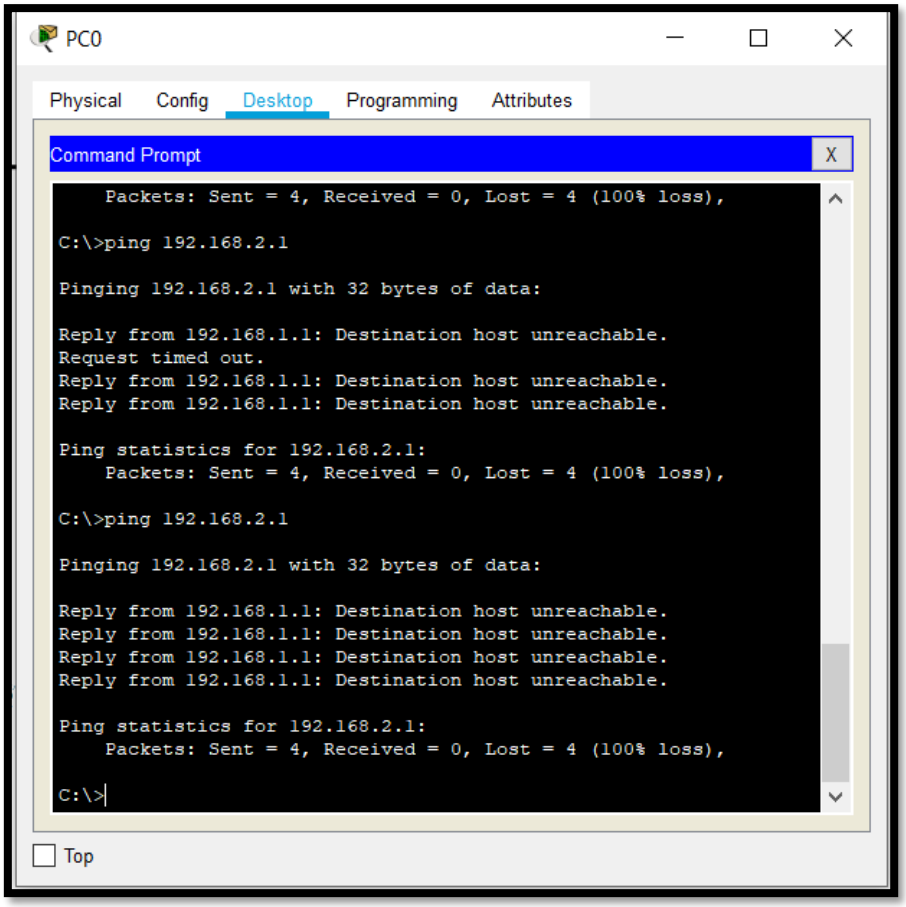


Step 5.

Steps] To check connectivity between two networks using RIP routing

click on any PC > Desktop > Select command prompt and type below commands.

PC > ping 192.168.2.1.



Experiment – 8

Configure the Static NAT using cisco packet tracer.

Step 1.

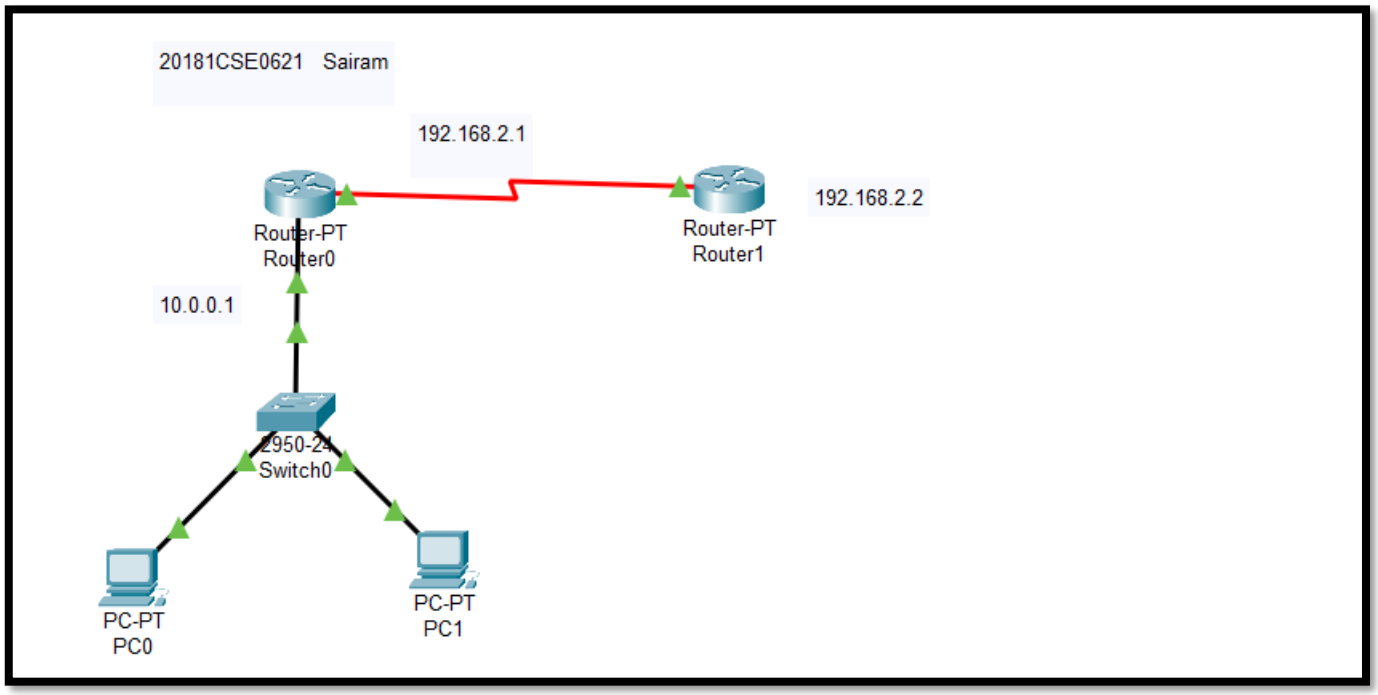
20181CSE0621

EXPERIMENT-08

QUESTION : Configure the static NAT using Cisco Packet Tracer.

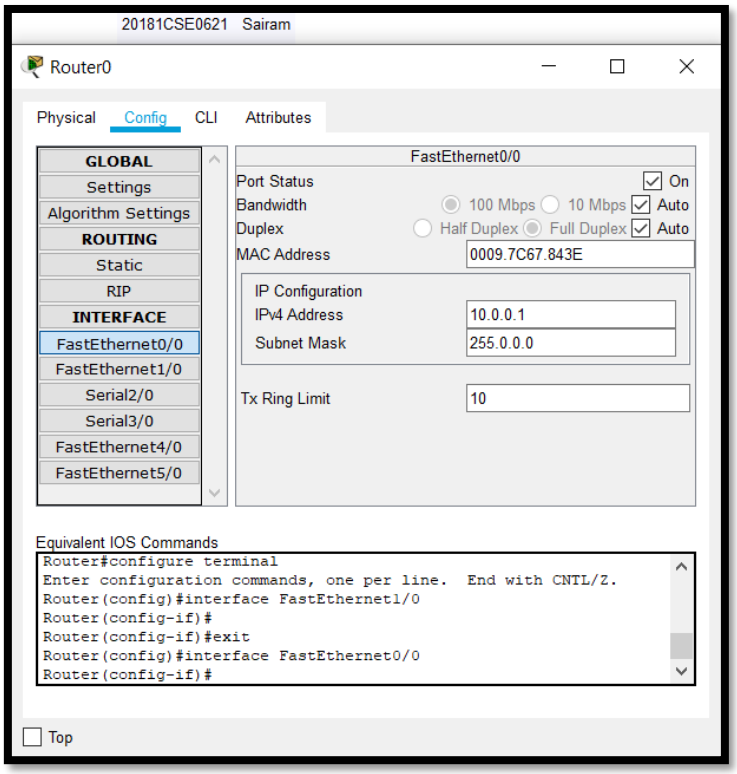
- NAT : It is a process in which one or more local IP address is translated into global IP or vice versa. It allows multiple devices to access internet through single public IP address.

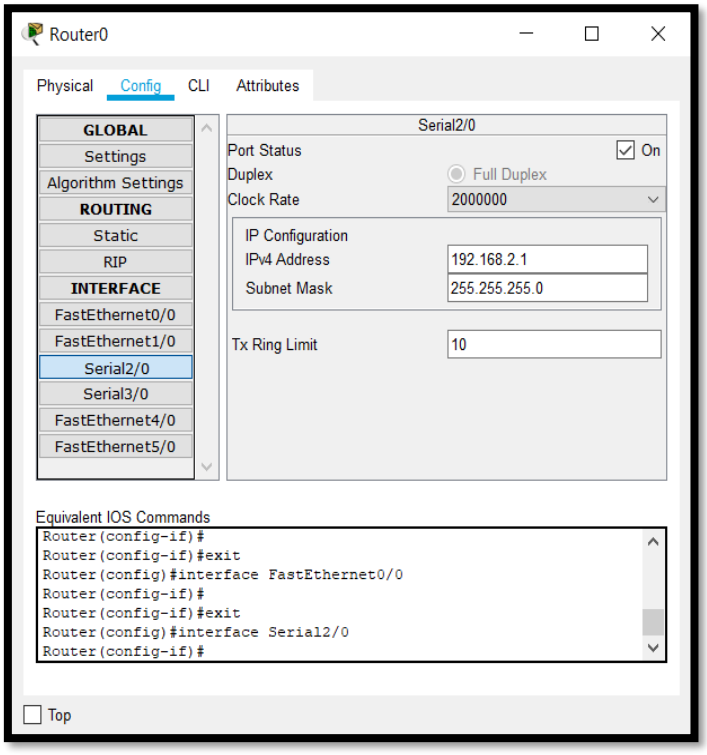
Step1] Draw a topology as shown below & assign IP addresses to all PC's.



Step 2.

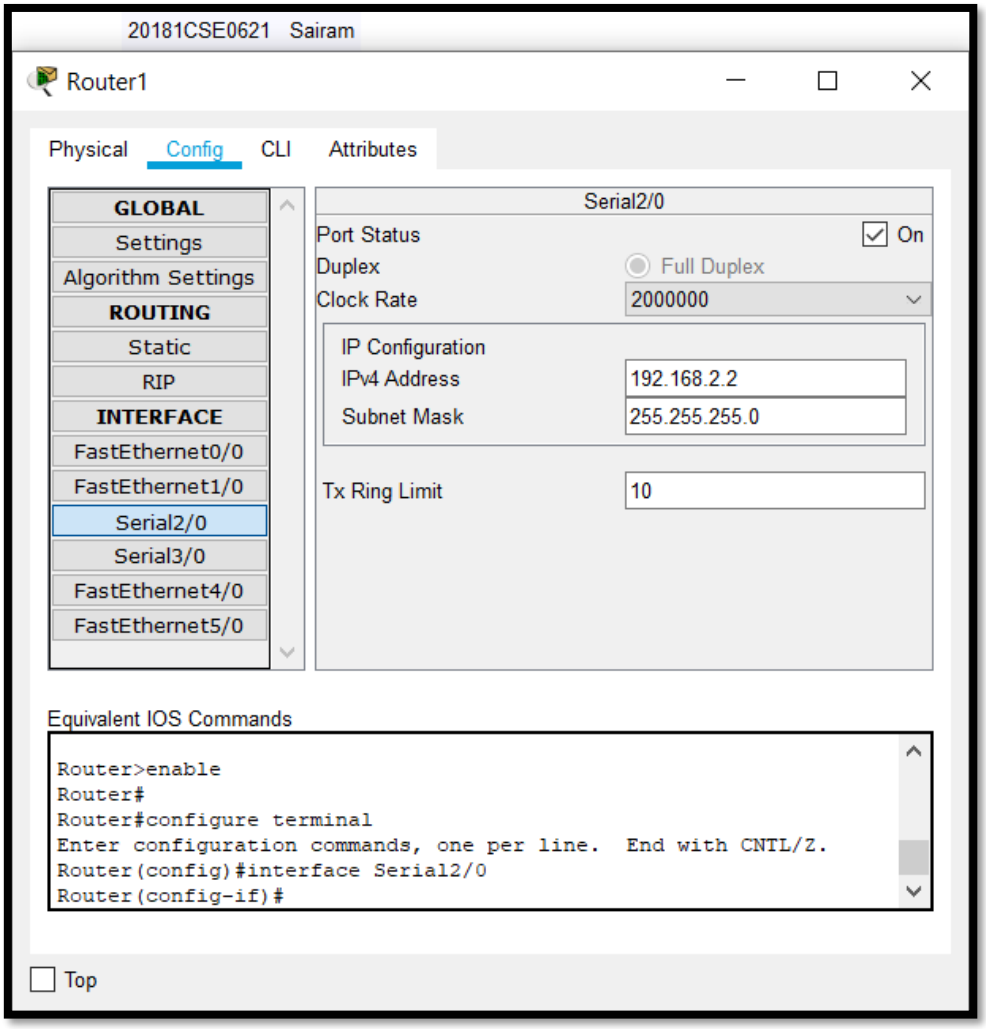
Step 2] Configure IP address to router 1
For FastEthernet 0/0
For Serial 2/0





Step 3.

Step3] Configure IP address to router2 For serial 2/0.



Step 4.

Step4] To setup static NAT

Router# sh ip nat translation

Router# config t

Router(config)# ip nat inside source static 10.0.0.2 192.168.1.3

Provide interface for NAT cable

Router(config)# int fa0/0

Router(config-if)# ip nat inside

exit

14

20181CSE0621

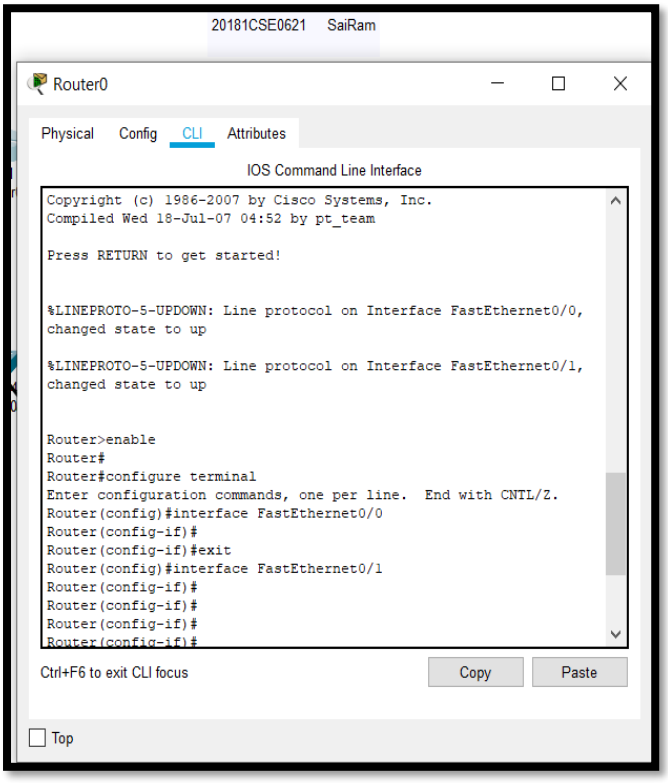
Router(config)# int serial 2/0

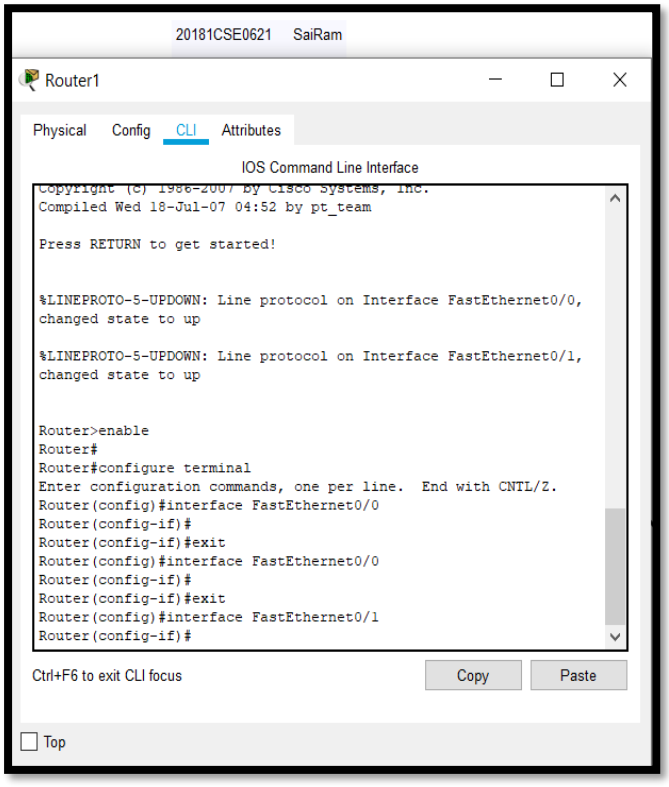
Router(config-if)# ip nat outside

exit

exit

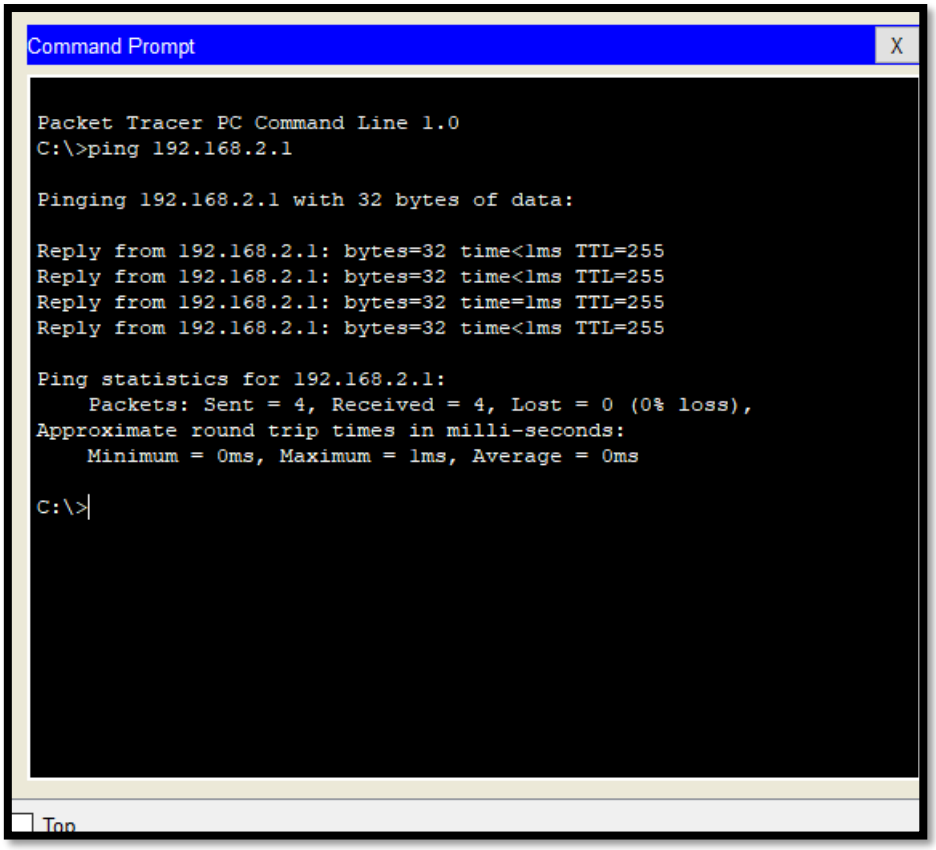
Router# sh ip nat translation





Step 5.

Steps] To check connectivity between two network
Click on any PC > Desktop > Select command prompt
PC > ping 192.168.2.1.



Experiment – 9

Configure the Dynamic NAT using cisco packet tracer.

Step 1.

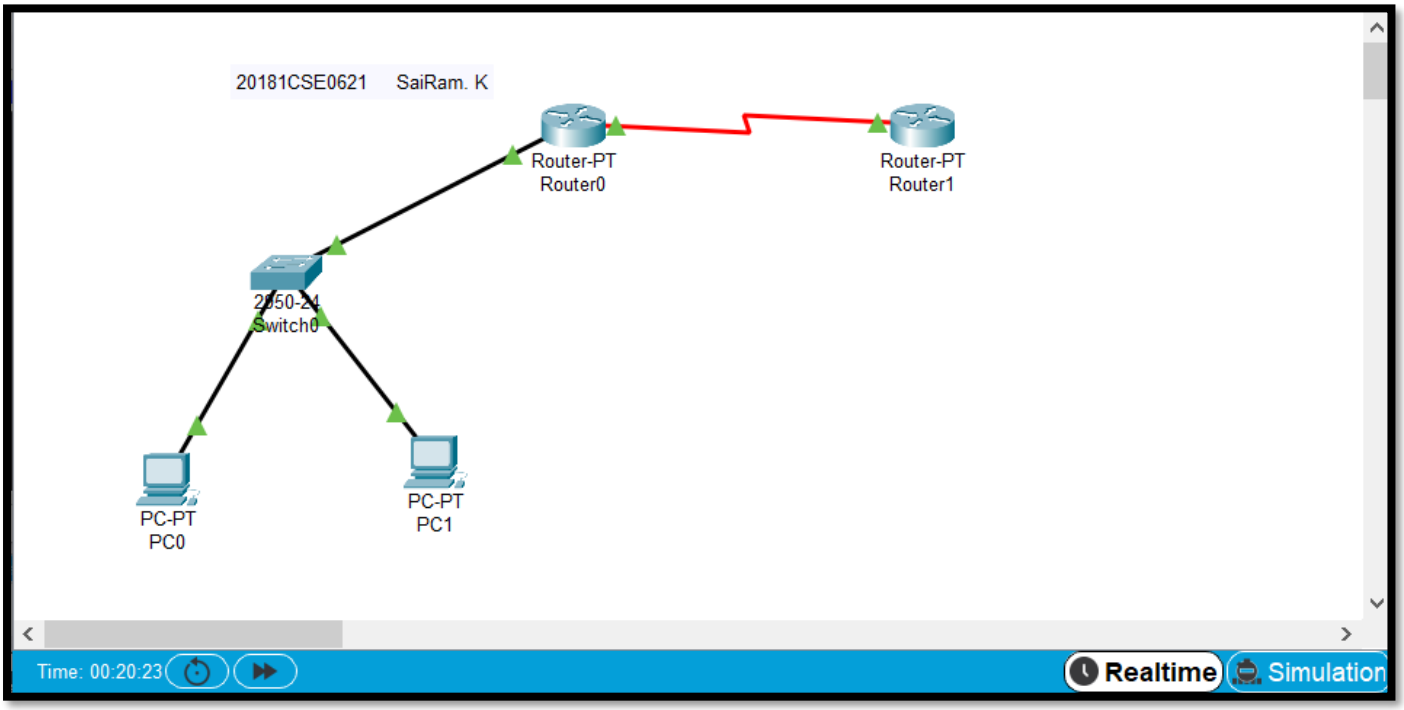
20181CSE0621

EXPERIMENT-09

QUESTION: Configure Dynamic NAT using Cisco Packet Tracer.

NAT : It is a process in which one or more local IP addresses is translated into global IP or vice versa.
It allows multiple devices to access internet through single IP address.

Step1] Draw a topology as shown below and assign IP addresses to all PC's.

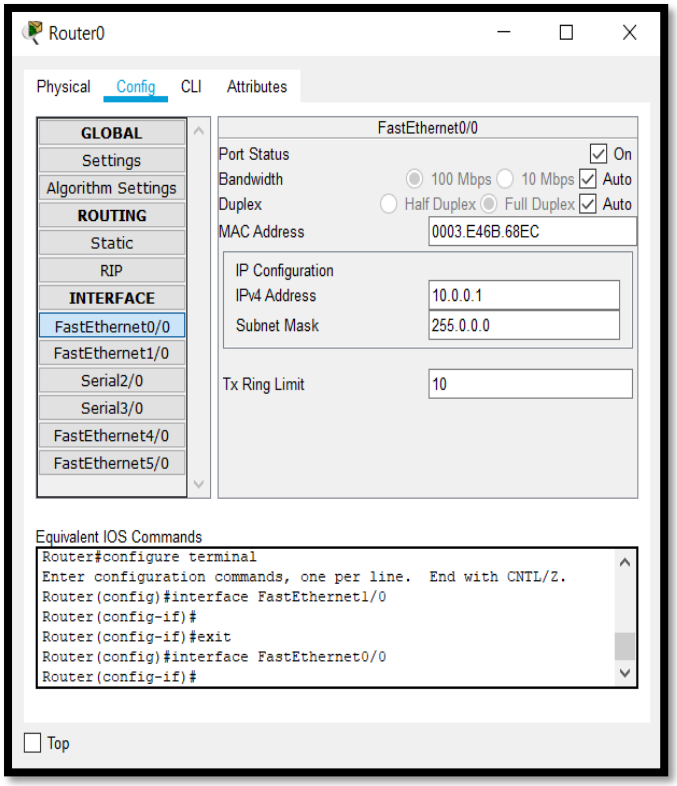


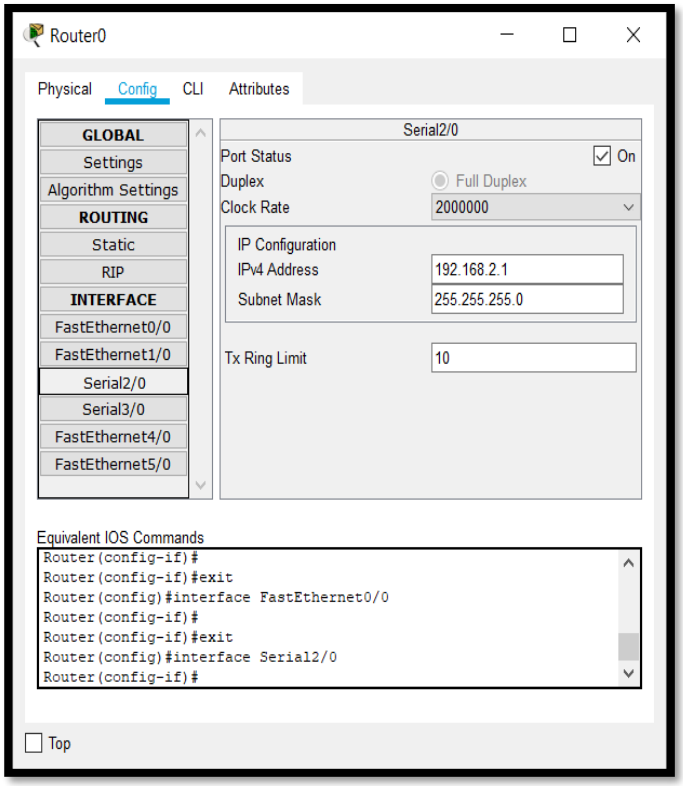
Step 2.

Step 2] Configure IP address for router 1

For Fast Ethernet 0/0.

For Serial 2/0.

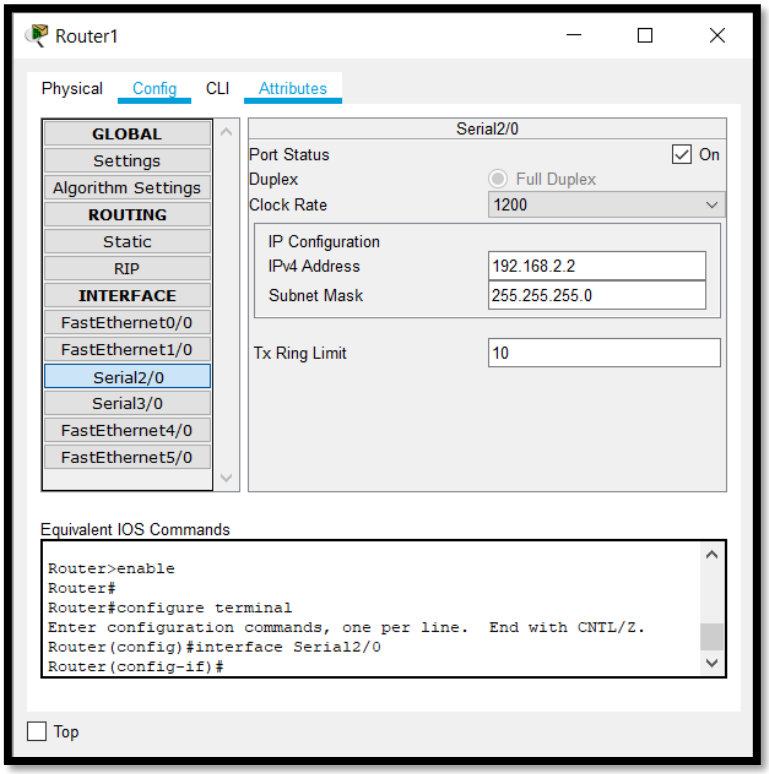




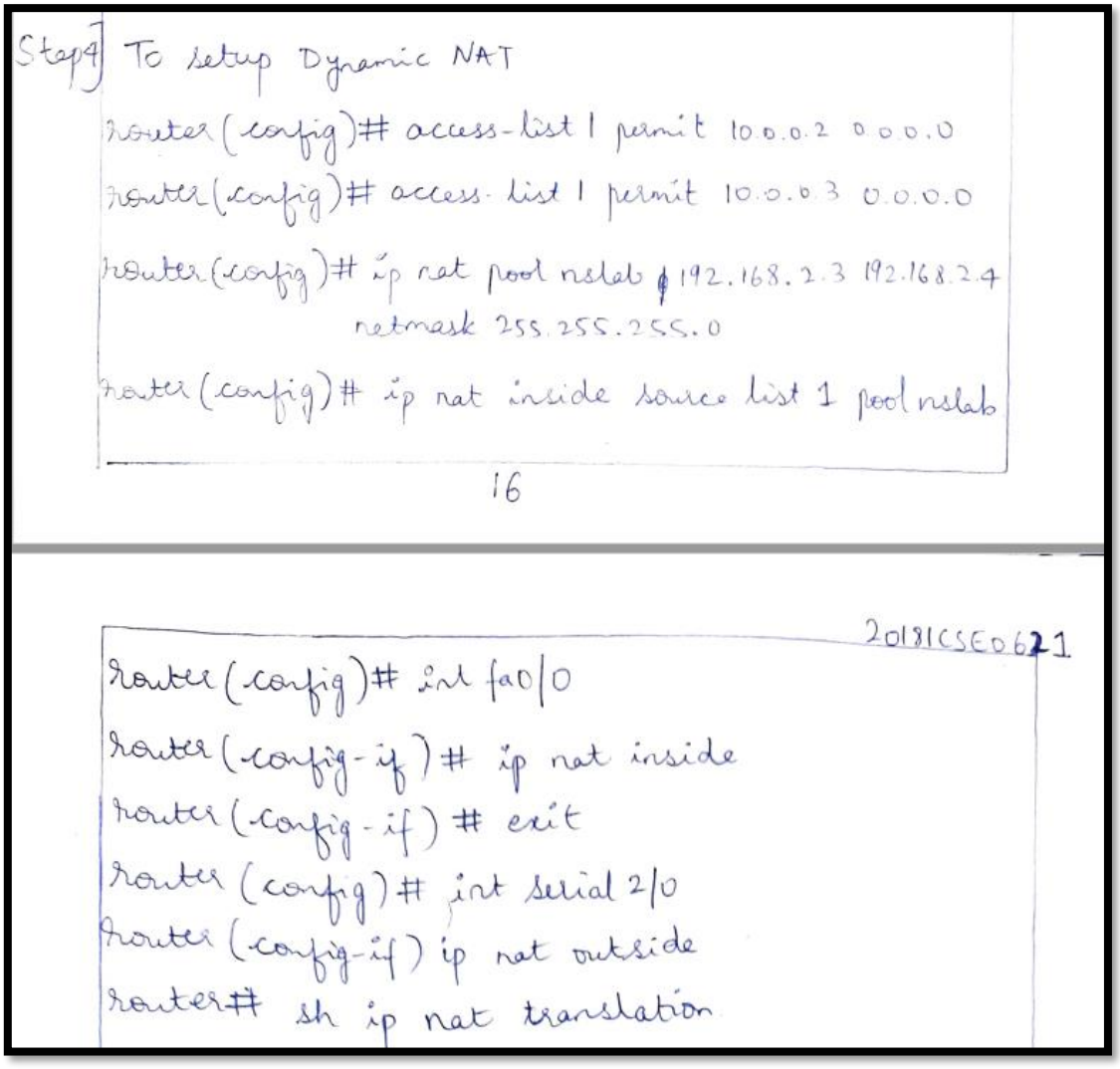
Step 3.

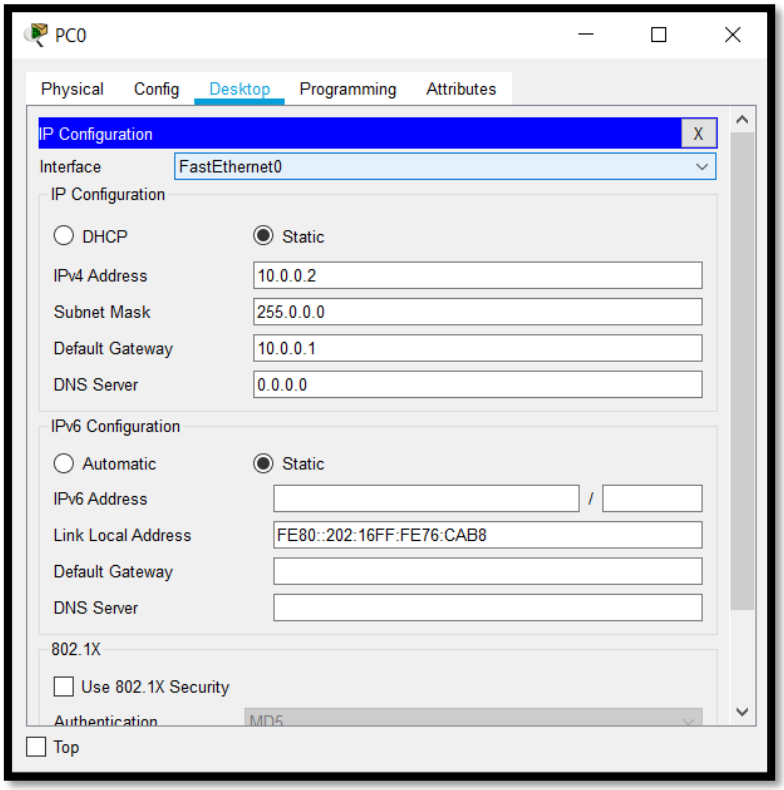
Step 3] Configure IP address for router 2

For Serial 2/0.

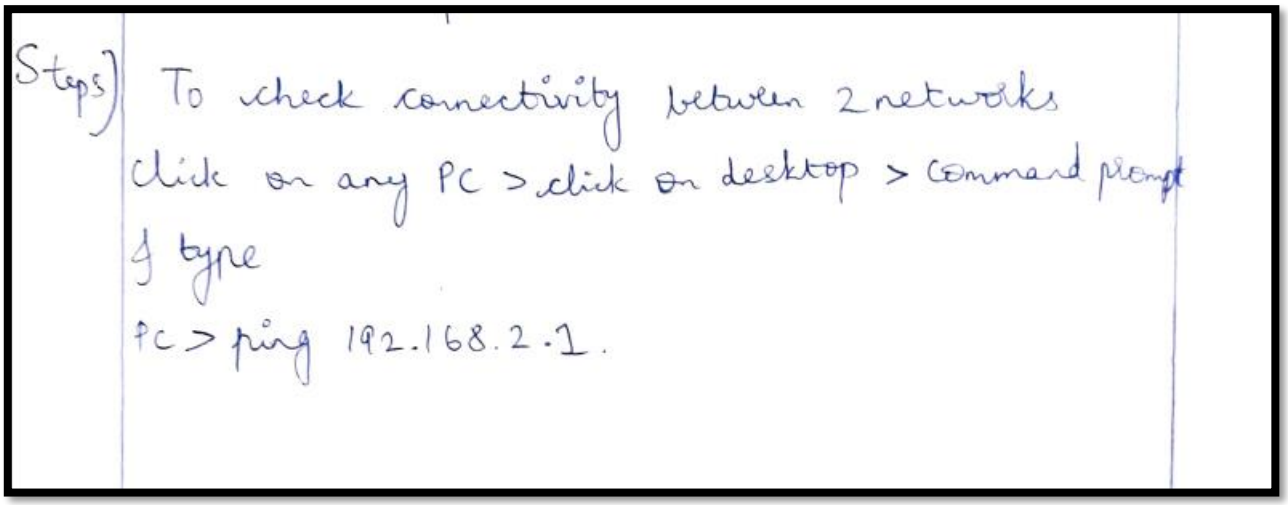

Router1
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
INTERFACE
FastEthernet0/0
FastEthernet1/0
Serial2/0
Serial3/0
FastEthernet4/0
FastEthernet5/0
Serial2/0
Port Status
On
Duplex
Full Duplex
Clock Rate
1200
IP Configuration
IPv4 Address
192.168.2.2
Subnet Mask
255.255.255.0
Tx Ring Limit
10
Equivalent IOS Commands
Router>enable
Router#
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface Serial2/0
Router(config-if)#
Top

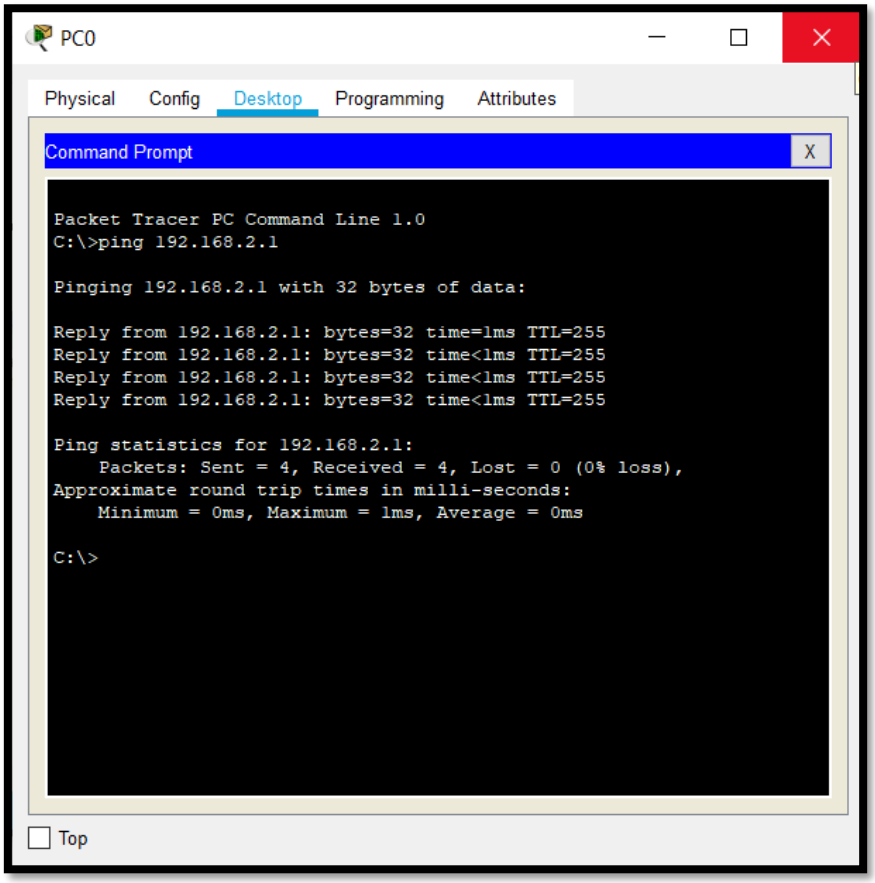
Step 4.


Step4 To setup Dynamic NAT
Router(config)# access-list 1 permit 10.0.0.2 0.0.0.0
Router(config)# access-list 1 permit 10.0.0.3 0.0.0.0
Router(config)# ip nat pool nslab 192.168.2.3 192.168.2.4
netmask 255.255.255.0
Router(config)# ip nat inside source list 1 pool nslab
16
20181CSE0621
Router(config)# int fa0/0
Router(config-if)# ip nat inside
Router(config-if)# exit
Router(config)# int Serial 2/0
Router(config-if) ip nat outside
Router# sh ip nat translation


PC0
Physical
Config
Desktop
Programming
Attributes
IP Configuration
X
Interface
FastEthernet0
IP Configuration
DHCP
Static
IPv4 Address
10.0.0.2
Subnet Mask
255.0.0.0
Default Gateway
10.0.0.1
DNS Server
0.0.0.0
IPv6 Configuration
Automatic
Static
IPv6 Address
Link Local Address
FE80::202:16FF:FE76:CAB8
Default Gateway
DNS Server
802.1X
Use 802.1X Security
Authentication
MD5
Top

Step 5.


Steps) To check connectivity between 2 networks
Click on any PC > click on desktop > Command prompt
& type
PC > ping 192.168.2.1.


PC0
Physical
Config
Desktop
Programming
Attributes
Command Prompt
X
Packet Tracer PC Command Line 1.0
C:\>ping 192.168.2.1
Pinging 192.168.2.1 with 32 bytes of data:
Reply from 192.168.2.1: bytes=32 time=1ms TTL=255
Reply from 192.168.2.1: bytes=32 time<1ms TTL=255
Reply from 192.168.2.1: bytes=32 time<1ms TTL=255
Reply from 192.168.2.1: bytes=32 time<1ms TTL=255
Ping statistics for 192.168.2.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
C:\>
Top

Module – 3

Socket Programming

Experiment – 1

20181CSE0621

MODULE-3

SOCKET PROGRAMMING

EXPERIMENT-1

– QUESTION: To find website address using Socket programming.

– CODE:

```
import java.net.*;
import java.util.*;
public class IPfinder{
    public static void main(String[] args)
    {
        String host;
        Scanner inp = new Scanner(System.in);
        System.out.print("\n Enter Hosts name:");
        host = input.next();
        try{
            InetAddress add = InetAddress.getByName(host);
            System.out.println(" IP : "+add.toString());
        }
        catch(UnknownHostException uhEx)
        {
            System.out.println("Couldn't find :"+host);
        }
    }
}
```

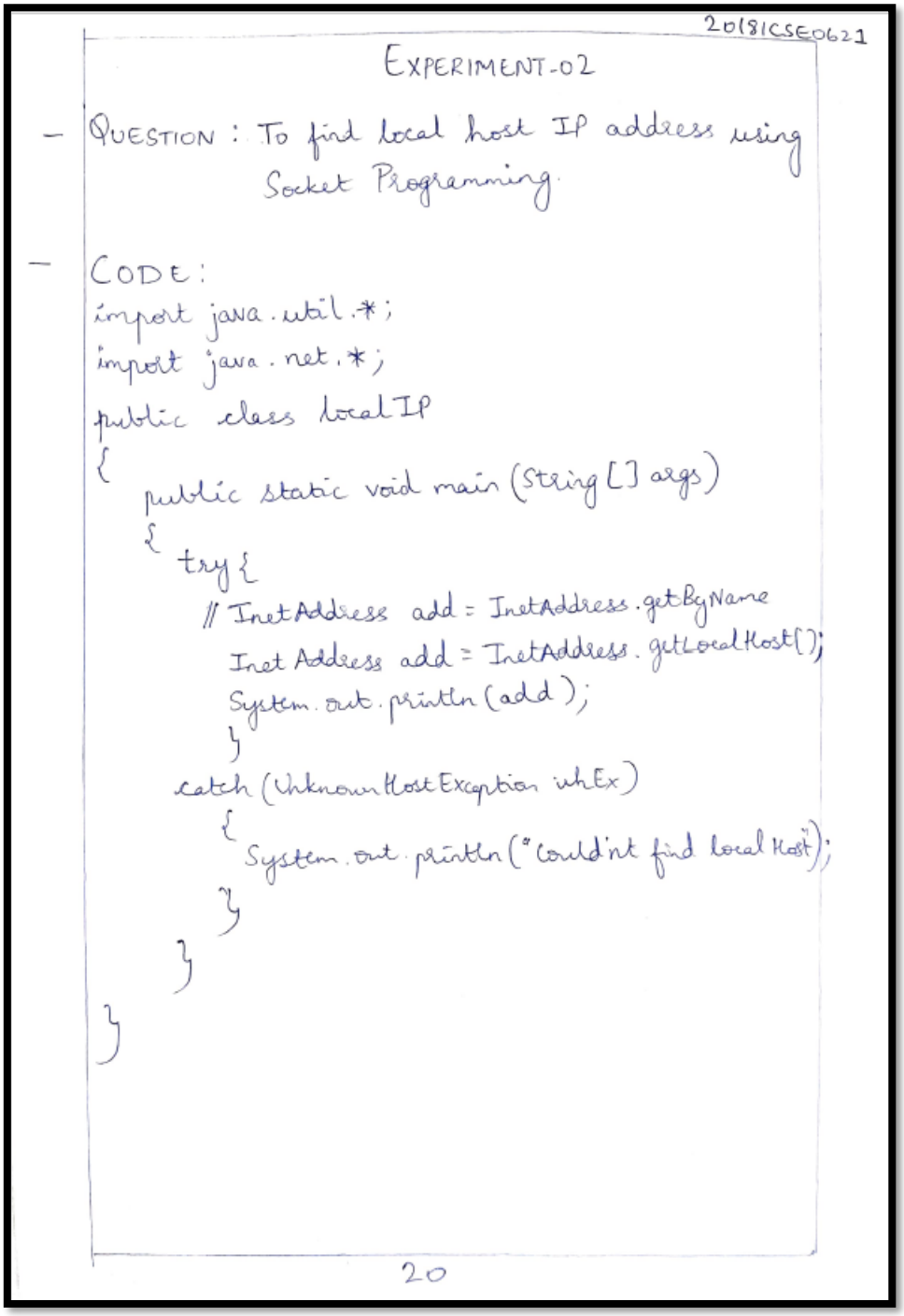
Output :

IPFinder.java

```
import java.util.*;
import java.net.*;
public class IPFinder {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        String host;
        Scanner input = new Scanner(System.in);
        System.out.println("Enter Host Name : ");
        host = input.next();
        try {
            InetAddress address = InetAddress.getByName(host);
            System.out.println("IP address: " + address.toString());
        }
        catch (UnknownHostException uhEx)
        {
            System.out.println("Could not find " + host);
        }

    }

}
```

Experiment – 2

To find the local host IP address using socket programming



20181CSE0621

EXPERIMENT-02

– QUESTION : To find local host IP address using Socket Programming.

– CODE :

```
import java.util.*;
import java.net.*;
public class localIP
{
   public static void main (String [] args)
   {
      try{
        // InetAddress add = InetAddress.getByName
          InetAddress add = InetAddress.getLocalHost();
          System.out.println(add);
          }
      catch (UnknownHostException unEx)
          {
            System.out.println("Could'nt find local Host");
          }
      }
}
```

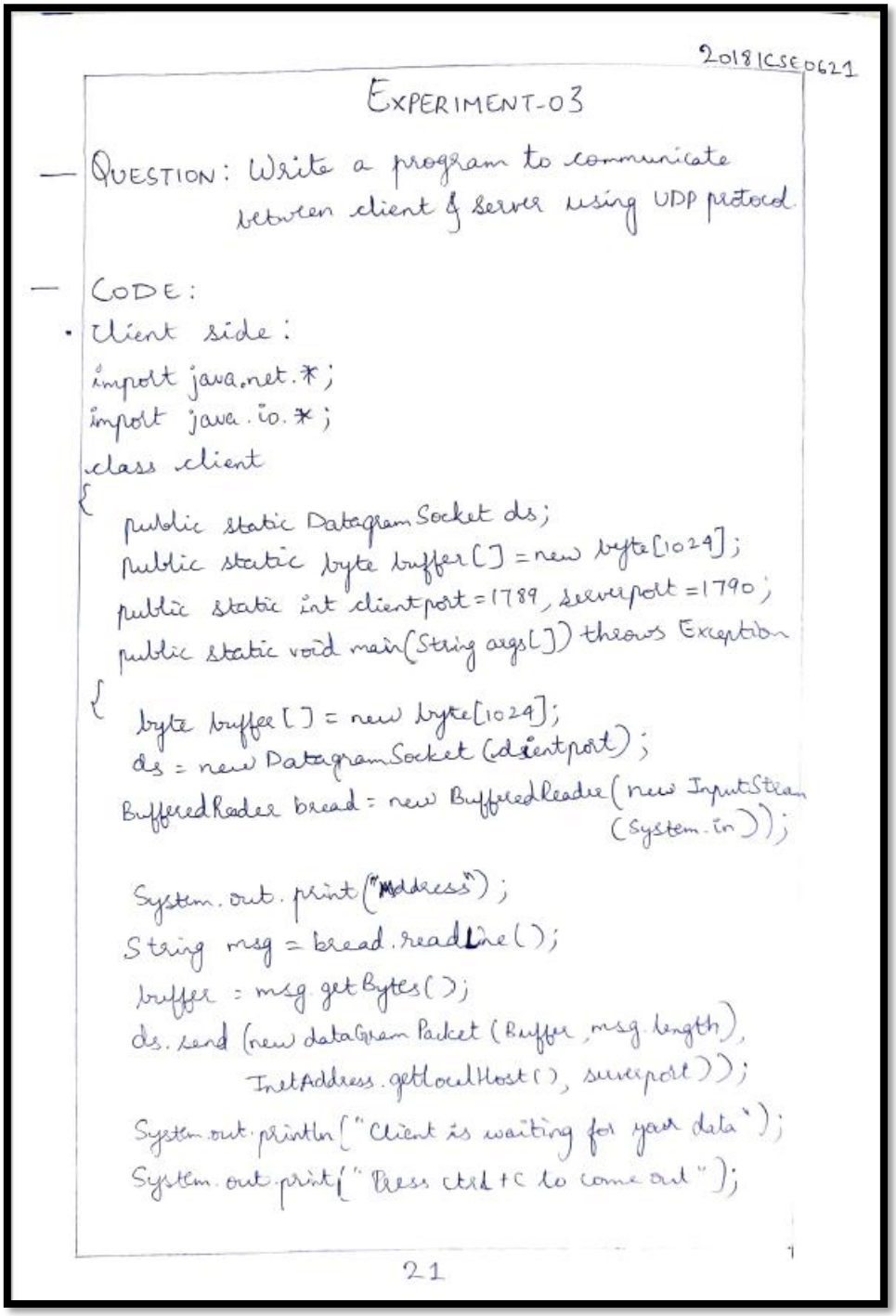
20

Output :


Mod-3 Socket Programming - Local_IPaddress/src/Local_ip_address.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
IPFinder
Local_IPaddress
JRE System Library [JavaSE-15]
src
(default package)
Local_ip_address.java
IPFinder.java
Local_ip_address.java
import java.util.*;
import java.net.*;
public class Local_ip_address
{
public static void main(String[] args)
{
try
{
InetAddress address =InetAddress.getLocalHost();
System.out.println("Your IP address is ==> " + address);
}
catch (UnknownHostException uhEx)
{
System.out.println("Could not find local address!");
}
}
}
Problems Javadoc Declaration Console
<terminated> Local_ip_address [Java Application] C:\Users\ram10\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0.2.v20210201-0955\jre\bin\javaw.exe (26-Mar-2021, 2:28:52 pm – 2:28:52 pm)
Your IP address is ==> LAPTOP-MVFDFOVD/192.168.0.104
Writable
Smart Insert
17 : 2 : 360

Experiment – 3

Write a program to communicate between client and server using UDP.



20181CSE0621

EXPERIMENT-03

— QUESTION: Write a program to communicate between client & server using UDP protocol.

— CODE:

• Client side:

```
import java.net.*;
import java.io.*;
class client
{
  public static DatagramSocket ds;
  public static byte buffer[] = new byte[1024];
  public static int clientport=1789, serverport=1790;
  public static void main(String args[]) throws Exception
{
  byte buffer[] = new byte[1024];
  ds = new DatagramSocket (clientport);
BufferedReader bread = new BufferedReader(new InputStream
                                    (System.in));

  System.out.print("Address");
String msg = bread.readLine();
  buffer = msg.getBytes();
ds.send (new dataGram Packet (Buffer, msg.length),
        InetAddress.getLocalHost(), serverport));
System.out.println("Client is waiting for your data");
System.out.print("Press ctrl+C to come out");
```

21

```
while(true)
{
  DatagramPacket dp = new DatagramPacket(buffer, buffer.length);
  ds.receive(dp)
  String pdata = new String(dp.getData(),0,dp.getLength());
  if(pdata.equals("End"))
      break;
  System.out.println(pdata);
  String str = breader.readLine();
  buffer = str.getBytes();
  ds.send(new DatagramPacket(buffer, str.length(),
        InetAddress.getLocalHost, serverport));
}
}
}
```

—> Server Side:

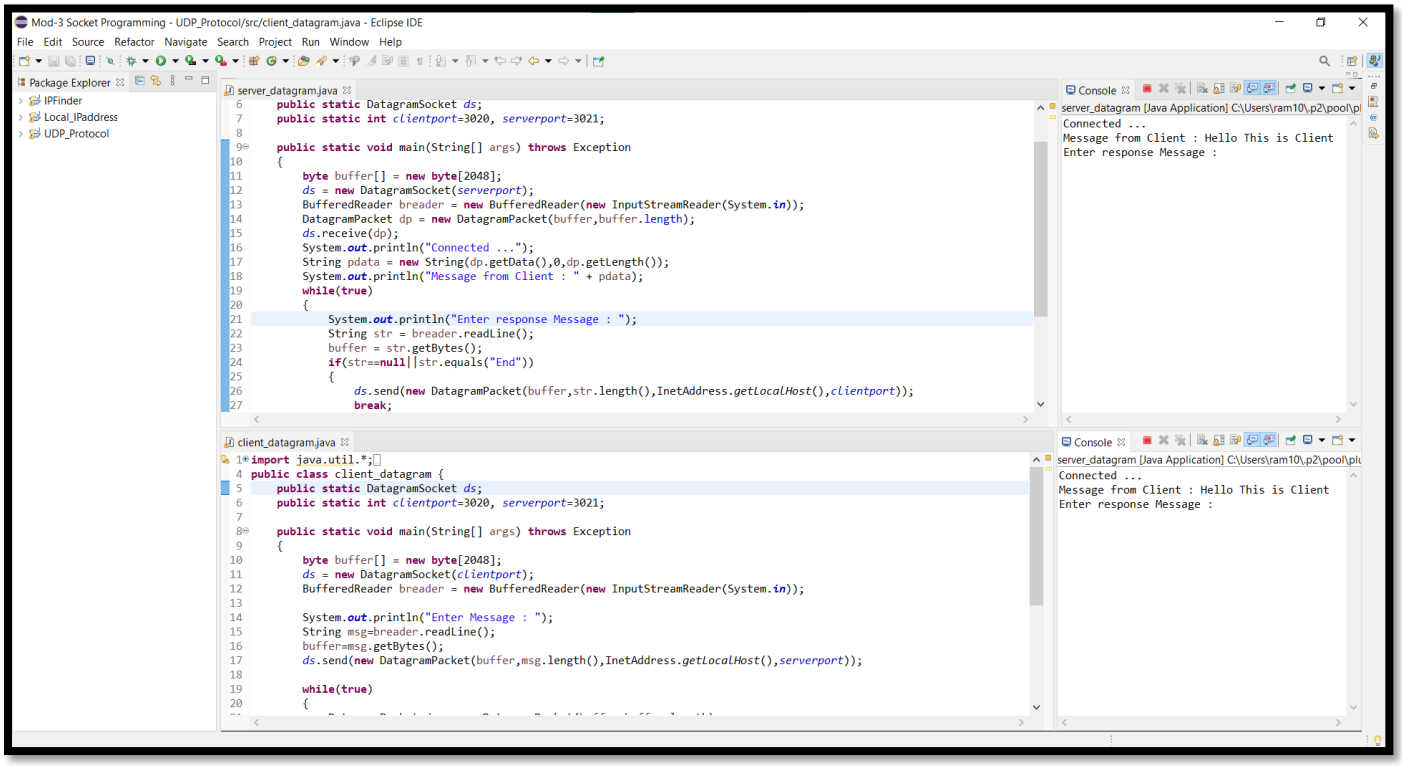
```
import java.net.*;
import java.io.*;
class Server
{
   public static DatagramSocket ds;
   public static int clientport = 1789, serverport = 1790;
   public static void main(String[] args)throws Exception
   {
      byte buffer = new byte[1024];
      ds = new DatagramSocket(serverport);
      BufferedReader breader = new BufferedReader(
      new InputStreamReader(System.in));
```

20181CSE0621

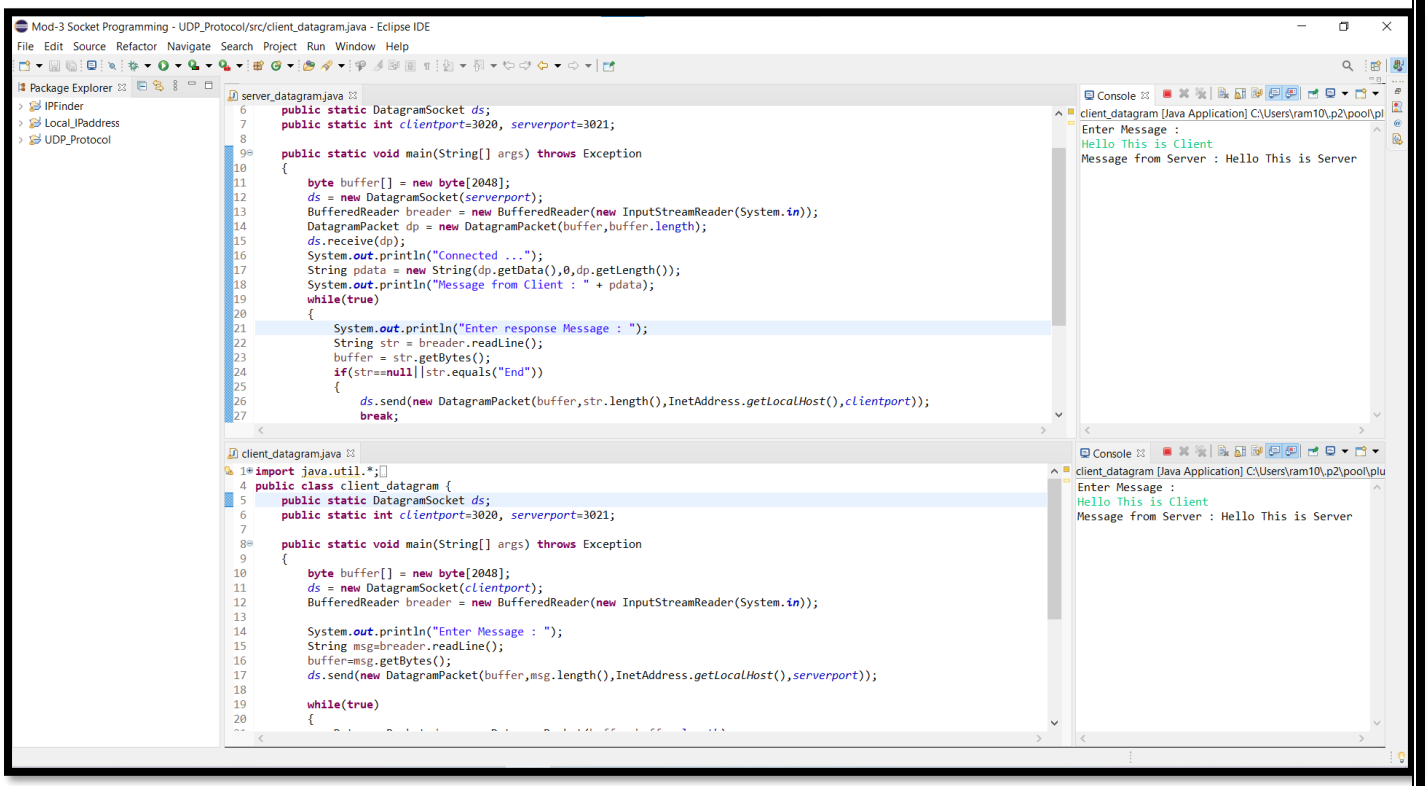
```
System.out.println("Waiting for connection");
DatagramPacket dp = new DatagramPacket(buffer, buffer.length);
ds.receive(dp);
String pdata = new String(dp.getData(), 0, dp.getLength());
System.out.println("Connected");
String st = "Hello";
buffer = st.getBytes();
ds.send(new DatagramPacket(buffer, str.length(), clientport));
  break;
while(true)
{
   ds.receive(dp);
   String recv = new String(dp.getData(), 0, dp.length());
   System.out.println(recv);
    buffer = str.getBytes();
   if( str == NULL || str.equals("End")
   { ds.send(new DatagramPacket(buffer, str.length,
     InetAddress.getLocalHost(), clientport));
      break;
   }
   ds.send(new DatagramPacket(buffer, str.length,
   InetAddress.getLocalHost(), clientport));
   }
}
}
```

Output :

Client Request



Server Response



Experiment – 4

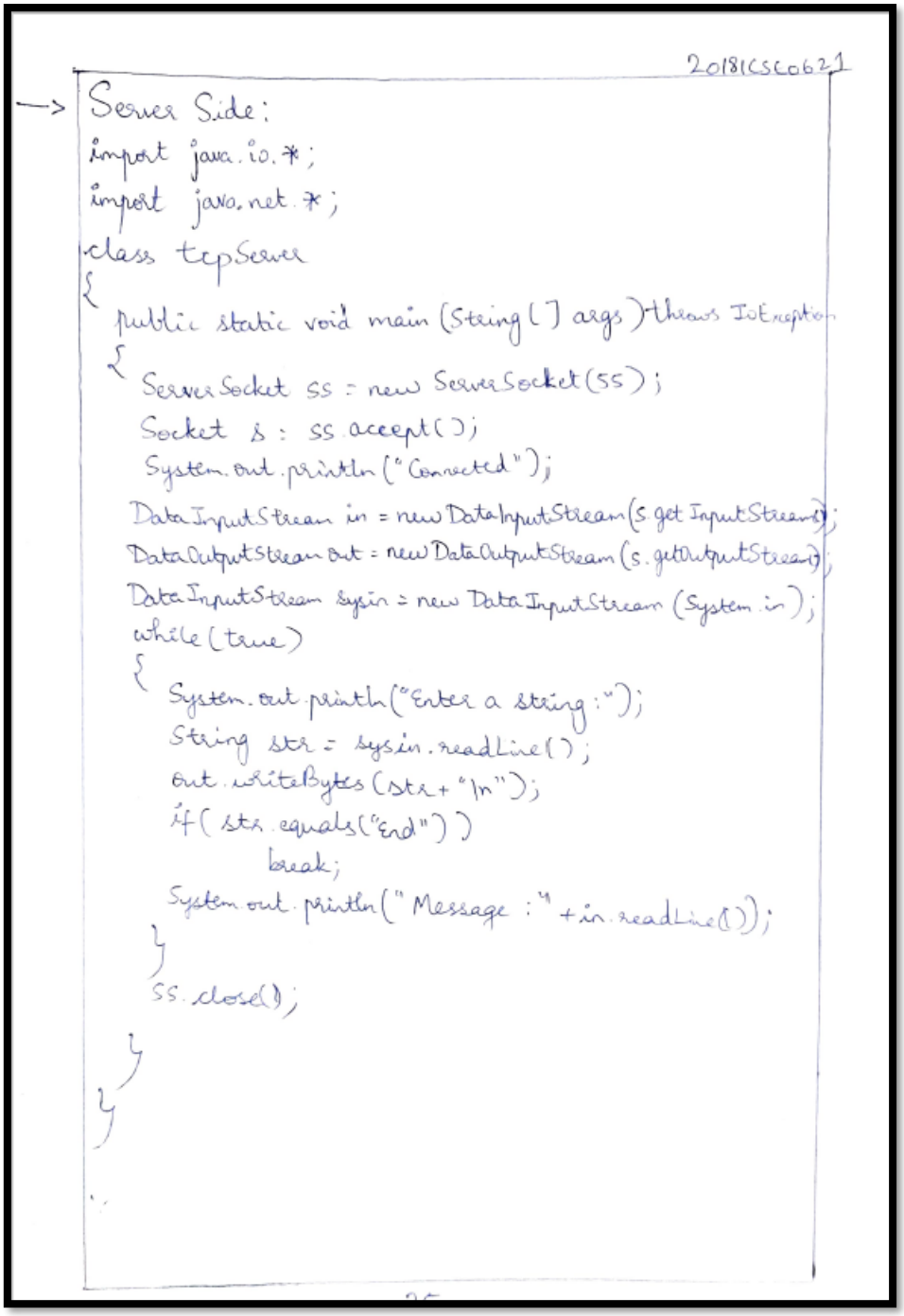
Write a program to communicate between client and server using TCP Protocol

20181CSE0621

EXPERIMENT-04

- QUESTION:- Write a program to communicate between client & server using TCP protocol.
- CODE:
- → Client side:

```
import java.net.*;
import java.io.*;
public class tcpclient
{
  public static void main(String[] args)throws IOException
  {
     Socket s = new Socket("localhost", 55);
  DataInputStream in = new DataInputStream(s.getInputStream());
  DataOutputStream out=new DataOutputStream(s.getOutputStream());
  DataInputStream sysin=new DataInputStream(System.in);
   while(true)
   {
        String str = in.readLine();
    System.out.println("Message from Server" + str);
    if(str.equals("End"))
        break;
    System.out.println("Enter reply:");
     String line = sysin.readLine();
     out.writeBytes(line + "\n");
   }
   s.close();
  }
}
```

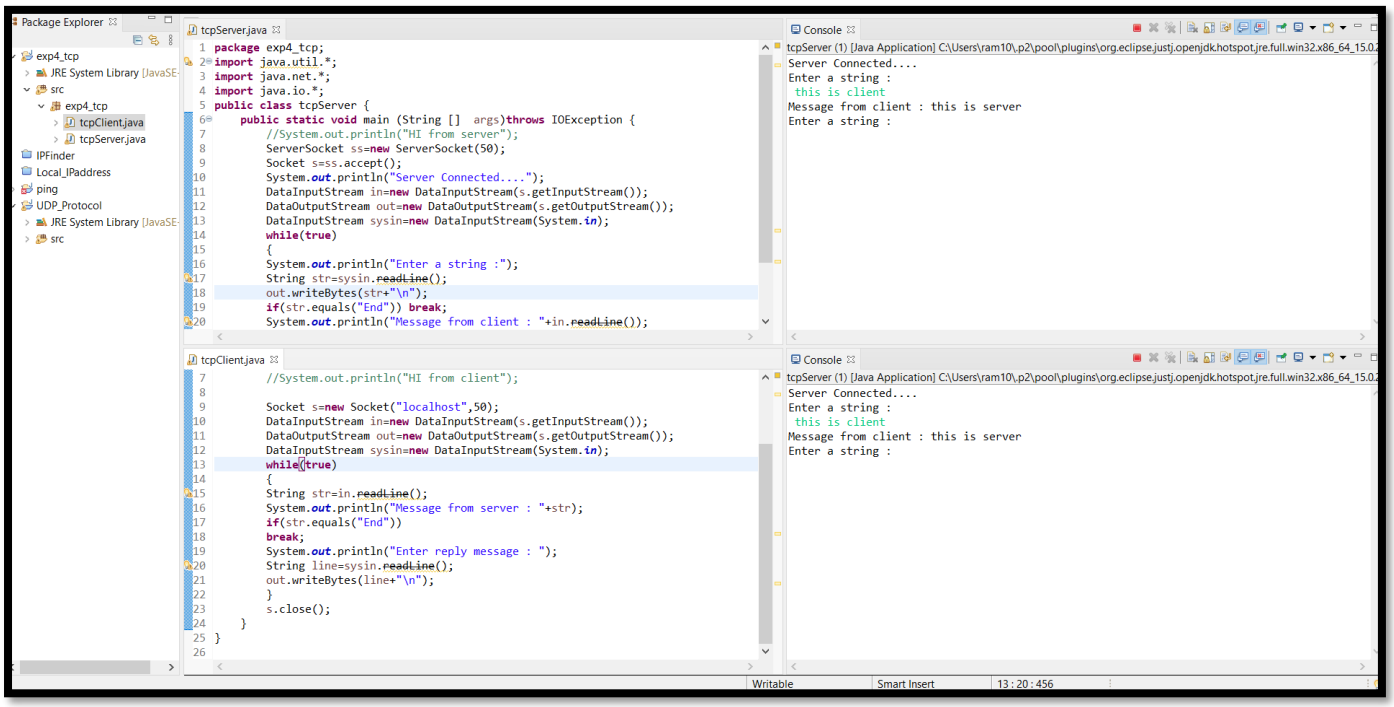


20181CSE0621

→ Server Side:

```
import java.io.*;
import java.net.*;
class tcpServer
{
  public static void main (String[] args) throws IOException
  {
    ServerSocket ss = new ServerSocket(55);
    Socket s = ss.accept();
    System.out.println("Connected");
   DataInputStream in = new DataInputStream(s.getInputStream());
   DataOutputStream out = new DataOutputStream(s.getOutputStream());
   DataInputStream sysin = new DataInputStream(System.in);
   while(true)
   {
     System.out.println("Enter a string:");
     String str = sysin.readLine();
     out.writeBytes(str+"\n");
     if(str.equals("End"))
          break;
     System.out.println("Message :"+in.readLine());
   }
   ss.close();
  }
}
```

Output :


Package Explorer
exp4_tcp
JRE System Library [JavaSE-
src
exp4_tcp
tcpClient.java
tcpServer.java
IPFinder
Local_IPaddress
ping
UDP_Protocol
JRE System Library [JavaSE-
src
tcpServer.java
package exp4_tcp;
import java.util.*;
import java.net.*;
import java.io.*;
public class tcpServer {
public static void main (String [] args)throws IOException {
//System.out.println("HI from server");
ServerSocket ss=new ServerSocket(50);
Socket s=ss.accept();
System.out.println("Server Connected....");
DataInputStream in=new DataInputStream(s.getInputStream());
DataOutputStream out=new DataOutputStream(s.getOutputStream());
DataInputStream sysin=new DataInputStream(System.in);
while(true)
{
System.out.println("Enter a string :");
String str=sysin.readLine();
out.writeBytes(str+"\n");
if(str.equals("End")) break;
System.out.println("Message from client : "+in.readLine());
Console
tcpServer (1) [Java Application] C:\Users\ram10\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0.2
Server Connected....
Enter a string :
this is client
Message from client : this is server
Enter a string :
tcpClient.java
//System.out.println("HI from client");
Socket s=new Socket("localhost",50);
DataInputStream in=new DataInputStream(s.getInputStream());
DataOutputStream out=new DataOutputStream(s.getOutputStream());
DataInputStream sysin=new DataInputStream(System.in);
while(true)
{
String str=in.readLine();
System.out.println("Message from server : "+str);
if(str.equals("End"))
break;
System.out.println("Enter reply message : ");
String line=sysin.readLine();
out.writeBytes(line+"\n");
}
s.close();
}
}
Console
tcpServer (1) [Java Application] C:\Users\ram10\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0.2
Server Connected....
Enter a string :
this is client
Message from client : this is server
Enter a string :
Writable
Smart Insert
13 : 20 : 456

Experiment – 5

Host Connectivity

20181CSE0621

EXPERIMENT-05

— QUESTION: Program to check connectivity of given Hostname.

— CODE:

```
import java.net.*;
import java.io.*;
public class ping{
  public static void main(String[] args) throws IoException {
  String Host = " ";
  Scanner inp = new Scanner(System.in);
  System.out.println("Enter host name: ");
  host = inp.next();
 try {
    InetAddress add = InetAddress.getByName(Host);
    System.out.println(" IP : " + add.toString());
    System.out.println("Sending ping request to" + host);
    if (add.isReachable(5000))
          System.out.println(host + "is reachable");
    else
          System.out.println( host + " is not reachable");
     }
 catch(UnknownHostException uhEx)
    {
       System.out.println(" Couldn't find" + host);
    }
  }
}
```

Output :

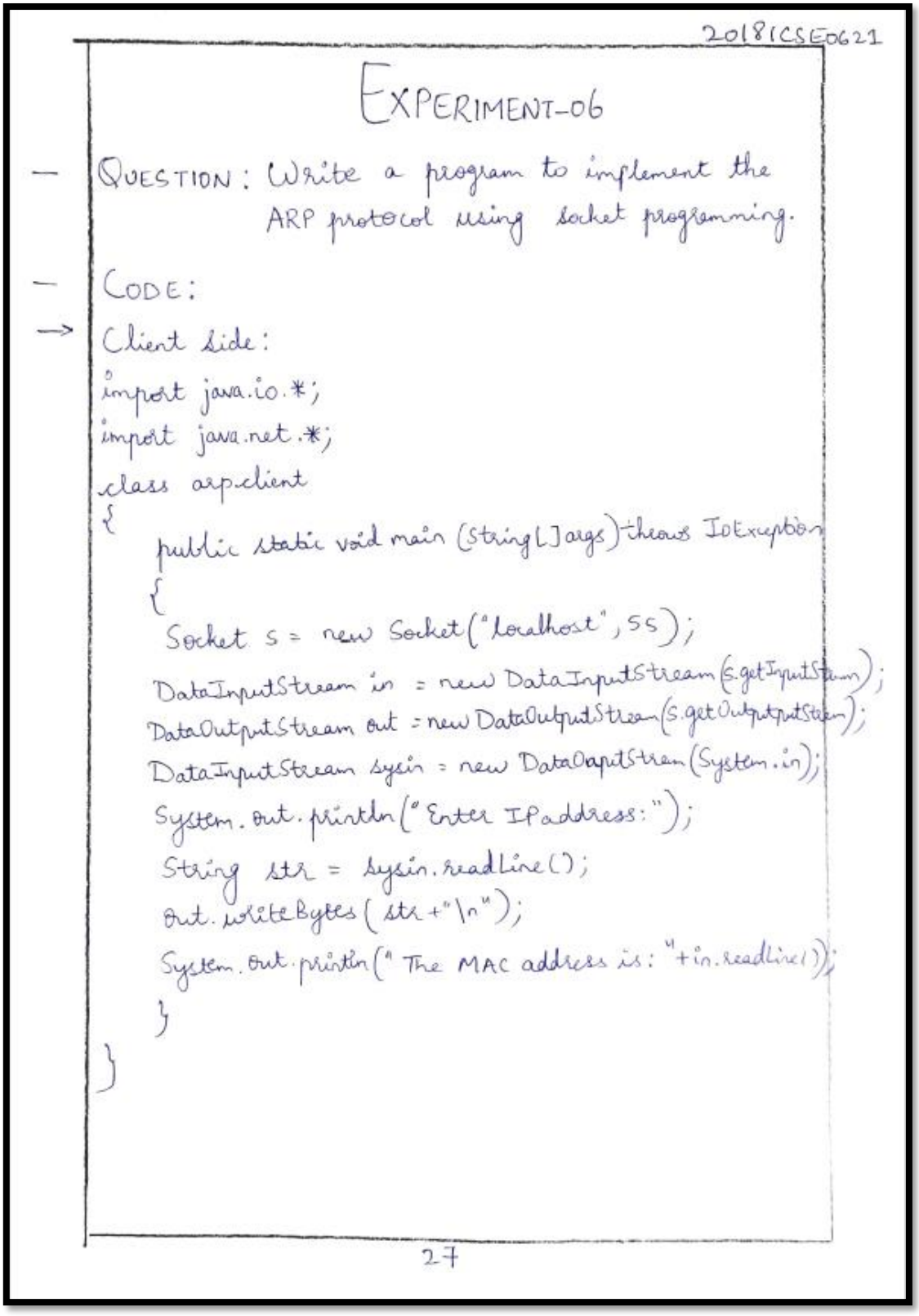

Console
<terminated> ping [Java Application] C:\Users\ram10\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0.2.v20210201-0955\jre\bin\javaw.exe (09-Apr-2021, 2:11:59 pm – 2:12:07 pm)
Enter host name: google.com
IP address: google.com/172.217.31.206
Sending Ping Request to google.com
google.com is reachable.


Console
<terminated> ping [Java Application] C:\Users\ram10\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0.2.v20210201-0955\jre\bin\javaw.exe (09-Apr-2021, 2:26:17 pm – 2:26:22 pm)
Enter host name: 123
IP address: /0.0.0.123
Sending Ping Request to 123
123 NOT reachable.

Experiment – 6

ARP Protocol

Client Side :



20181CSE0621

EXPERIMENT-06

— QUESTION: Write a program to implement the ARP protocol using socket programming.

— CODE:

→ Client side:

```
import java.io.*;
import java.net.*;
class arp.client
{
    public static void main (String[] args) throws IOException
    {
     Socket s = new Socket("localhost", 55);
    DataInputStream in = new DataInputStream(s.getInputStream);
    DataOutputStream out = new DataOutputStream(s.getOutputStream);
    DataInputStream sysin = new DataInputStream(System.in);
    System.out.println("Enter IPaddress:");
    String str = sysin.readLine();
    out.writeBytes(str+"\n");
    System.out.println("The MAC address is: "+in.readline());
    }
}
```

Server side:-

20181CSE0621

```
Server side:
import java.net.*;
import java.io.*;
public class arpserver
{
 public static void main (String[] args)
 { ServerSocket ss = new ServerSocket (55);
  Socket ss = s.accept();
  DataInputStream in = new DataInputStream(s.getInputStream);
  DataOutputStream out= new DataOutputStream(s.getOutputStream());
  String iparr[] = {"10.0.1.45","172.16.5.21","172.16.5.22"};
  String macarr[] = {"00-0c-6e-5c-3c-63","02-11-B6-F3-EF-21",
                     "03-12-B3-F3-EF-18"};
  String str = in.readLine();
  System.out.println("Ip received "+str);
  int flag=0;
  for(int i=0;i<3;i++)
  { if(str.equals(iparr[i]==true)
    {
      flag=1;
      String str1 = macarr[i];
      out.writeBytes(str+"\n");
      break;
    }
  } if(flag==0)
    {
      System.out.println("IP not in network");
    }
    s.close();
 }
}
```

OUTPUT

clientARP.java

```
1  package exp6_ARP;
2  import java.io.*;
4  class clientARP
5  {
6      public static void main(String args[])throws IOException
7      {
8          Socket s=new Socket("localhost",53);
9          DataInputStream in=new DataInputStream(s.getInputStream());
10         DataOutputStream out=new DataOutputStream(s.getOutputStream());
11         DataInputStream sysin=new DataInputStream(System.in);
12         System.out.println("Enter an IP Address:");
13         String str=sysin.readLine();
14         out.writeBytes(str+"\n");
15         System.out.println("The corresponding MAC address \n"+in.readLine());
16     }
17 }
```

serverARP.java

```
1  package exp6_ARP;
2  import java.net.*;
4  public class serverARP {
5
6      public static void main(String args[])throws IOException
7      {
8          ServerSocket ss=new ServerSocket(53);
9          Socket s=ss.accept();
10         DataInputStream in=new DataInputStream(s.getInputStream());
11         DataOutputStream out=new DataOutputStream(s.getOutputStream());
12         String iparr[]={"10.0.1.45","172.16.5.21","172.16..5.22"};
13         String macarr[]={"00-0c-6e-5c-3c-63","02-11-B6-F3-EF-21","03-12-B3-F3-EF-18"};
14         String str=in.readLine();
15         System.out.println("Ip Address received from server"+str);
16         int flag=0;
17         for(int i=0;i<3;i++)
18         {
19             if(str.equals(iparr[i])==true)
20             {
21                 flag=1;
22                 String str1=macarr[i];
23                 out.writeBytes(str1+"\n");
24                 break;
25             }
```

Console

```
<terminated> clientARP [Java Application] C:\Users\ram10\.p2\pool\plugins\org.eclipse.justj.openjdk.hots
Enter an IP Address:
10.0.1.45
The corresponding MAC address
00-0c-6e-5c-3c-63
```

Console

```
<terminated> clientARP [Java Application] C:\Users\ram10\.p2\pool\plugins\org.eclipse.justj.openjdk.hots
Enter an IP Address:
172.16.5.21
The corresponding MAC address
02-11-B6-F3-EF-21
```

Experiment – 7

FTP Protocol

Client Side: -

20181CSE0621

EXPERIMENT-07

QUESTION: Write a program to implement the FTP protocol using Socket programming.

CODE:-

Client side:

```
import java.io.*;
import java.net.*;
public class ftpclient {
   public static void main (String [] args)
{
Socket s = new Socket (InetAddress.getLocalHost( ),5555);
DataInputStream si = new DataInputStream(s.getInputStream());
DataInputStream inp = new DataInputStream(System.in);
DataOutputStream so = new DataOutputStream(s.getOutputStream());
System.out.println("Enter path:\n");
String str = inp.readline();
FileOutputStream fos = new FileOutputStream("output.txt");
int str1;
while ((str1 = si.read())!=-1)
        fos.write((char)str1);
System.out.println("File received\n");
si.close();
so.close();
inp.close();
S.close();
}
}
```

29

Server side: -

20181CSE0621

Server Side:

```
import java.net.*;
import java.io.*;
Class ftpserver
{
  public static void main (String[]args)
  {
   ServerSocket ss = new ServerSocket(5555);
   Socket  S = ss.accept();
   DataOutputStream dos = new DataOutputStream(s.getOutputStream());
   DataInputStream din = new DataInputStream(S.getInputStream());
   String s1;
   S1 = din.readLine();
   FileInputStream fin = new FileInputStream(s1);
   int str1;
  while ((str1 = fin.read()) != -1)
          dos.writeBytes("(char) str1);
  System.out.println("File Sent");
  dos.close();
  din.close();
  S.close();
}
}
```

OUTPUT

clientFTP.java

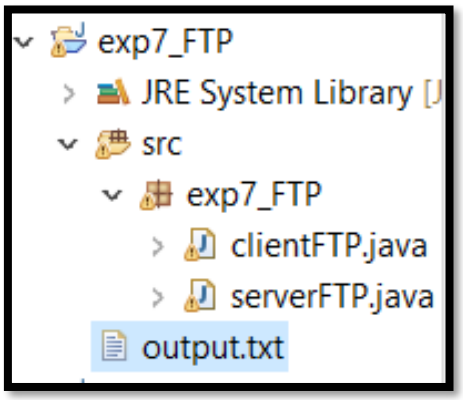
```
1  package exp7_FTP;
2  import java.io.*;
4  public class clientFTP
5  {
6      public static void main(String a[])throws IOException
7      {
8              Socket s=new Socket(InetAddress.getLocalHost(),5553);
9              DataInputStream s1=new DataInputStream(s.getInputStream());
10             DataInputStream inp=new DataInputStream(System.in);
11             DataOutputStream so=new DataOutputStream(s.getOutputStream());
12             System.out.println("\n enter the filename(path)");
13             String str=inp.readLine();
14             so.writeBytes(str+"\n");
15             FileOutputStream fos=new FileOutputStream("output.txt");
16             int str1;
17             while((str1=s1.read())!=-1)
18                 fos.write((char)str1);
19             System.out.println("\n file received successfully");
20             s1.close();
21             so.close();
22             inp.close();
23             s.close();
24         }
25 }
```

serverFTP.java

```
1  package exp7_FTP;
2  import java.io.*;
4  public class serverFTP
5  {
6      public static void main(String a[])throws IOException
7      {
8          ServerSocket ss=new ServerSocket(5553);
9          Socket s=ss.accept();
10         DataOutputStream dos=new DataOutputStream(s.getOutputStream());
11         DataInputStream din=new DataInputStream(s.getInputStream());
12         String s1;
13         s1=din.readLine();
14         FileInputStream fin=new FileInputStream(s1);
15         int str1;
16         while((str1=fin.read())!=-1)
17             dos.writeBytes(""+(char)str1);
18         System.out.println("\n file successfully sent");
19         dos.close();
20         din.close();
21         s.close();
22     }
23 }
```

output.txt

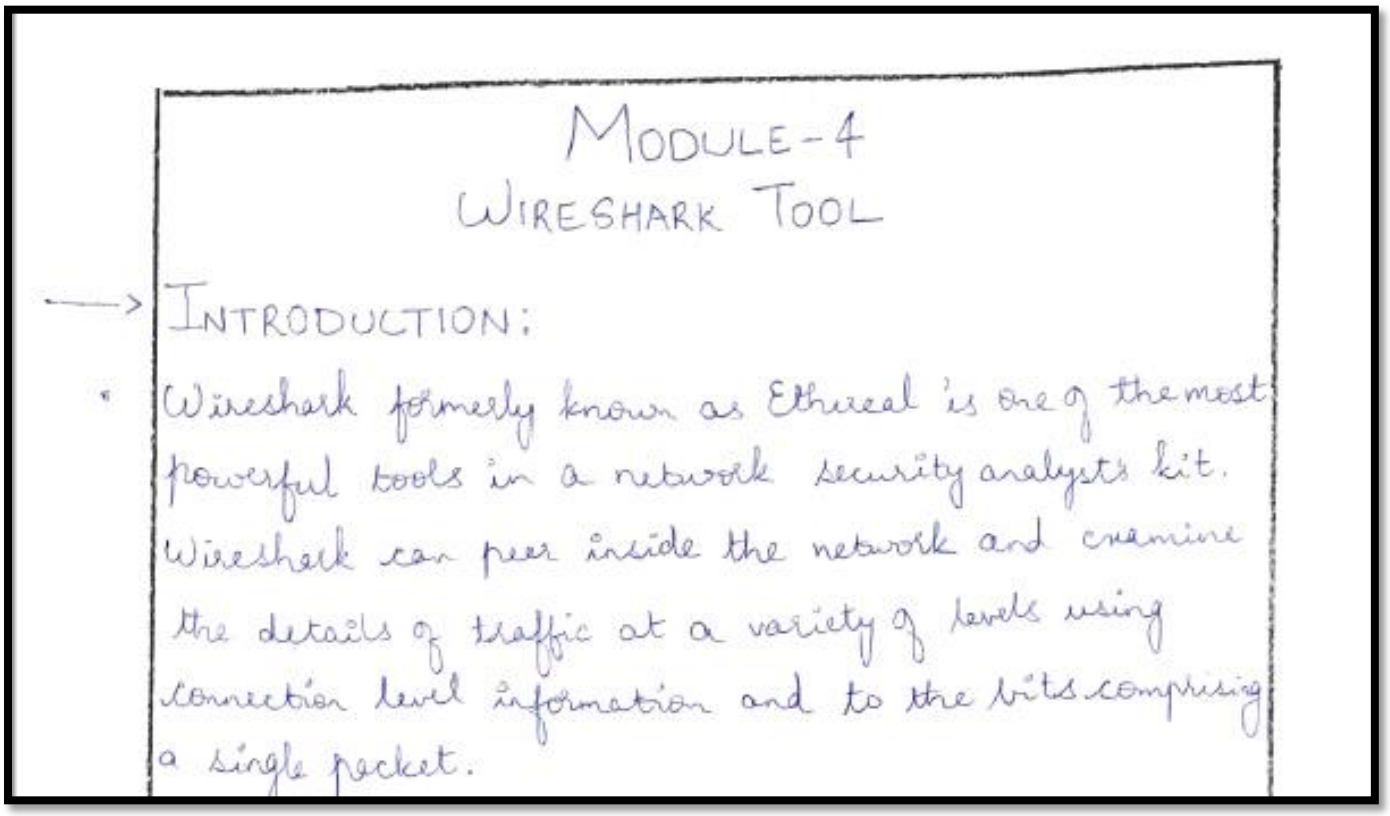
```
1 20181CSE0621
2
```



Module – 4

Wireshark Tool

Introduction



Features of Wireshark: -

- Features of Wireshark:
  1. Available in both UNIX and Windows.
  2. Ability to capture live packets from various interfaces
  3. Filters packet with many criteria.
  4. Can save & merge captured prokets.

- The flexibility and depth of inspection allows the valuable tool to analyze security events and troubleshoot network security device issues.

- The installation of this is easily available as we can find the source code at www.wireshark.org and we have the links that are compatible with Linux and x32 bit and x64 bit systems.

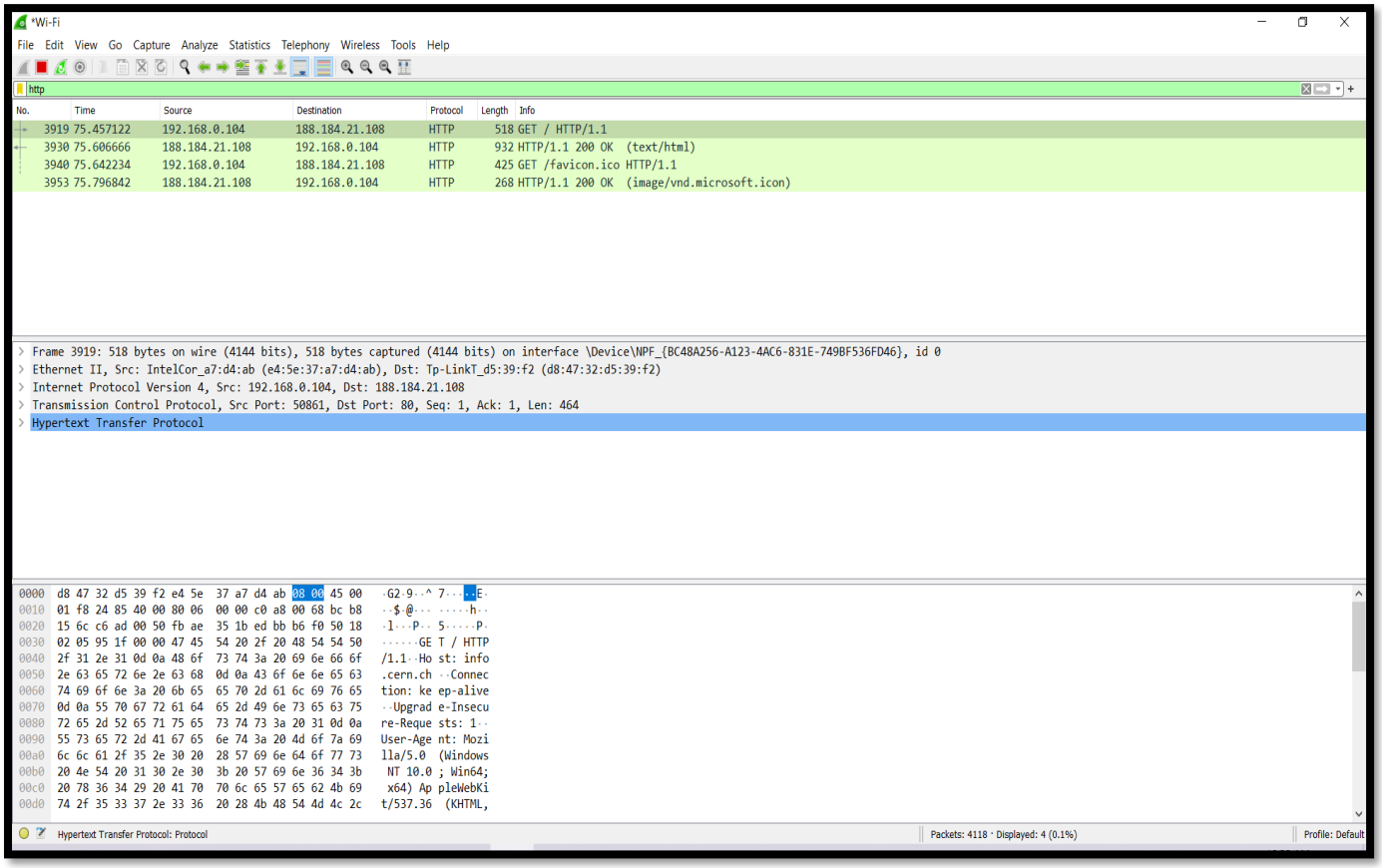
- Features of Wireshark:
  1. Available in both UNIX and Windows.
  2. Ability to capture live packets from various interfaces
  3. Filters packet with many criteria.
  4. Can save & merge captured prokets.

- The flexibility and depth of inspection allows the valuable tool to analyze security events and troubleshoot network security device issues.

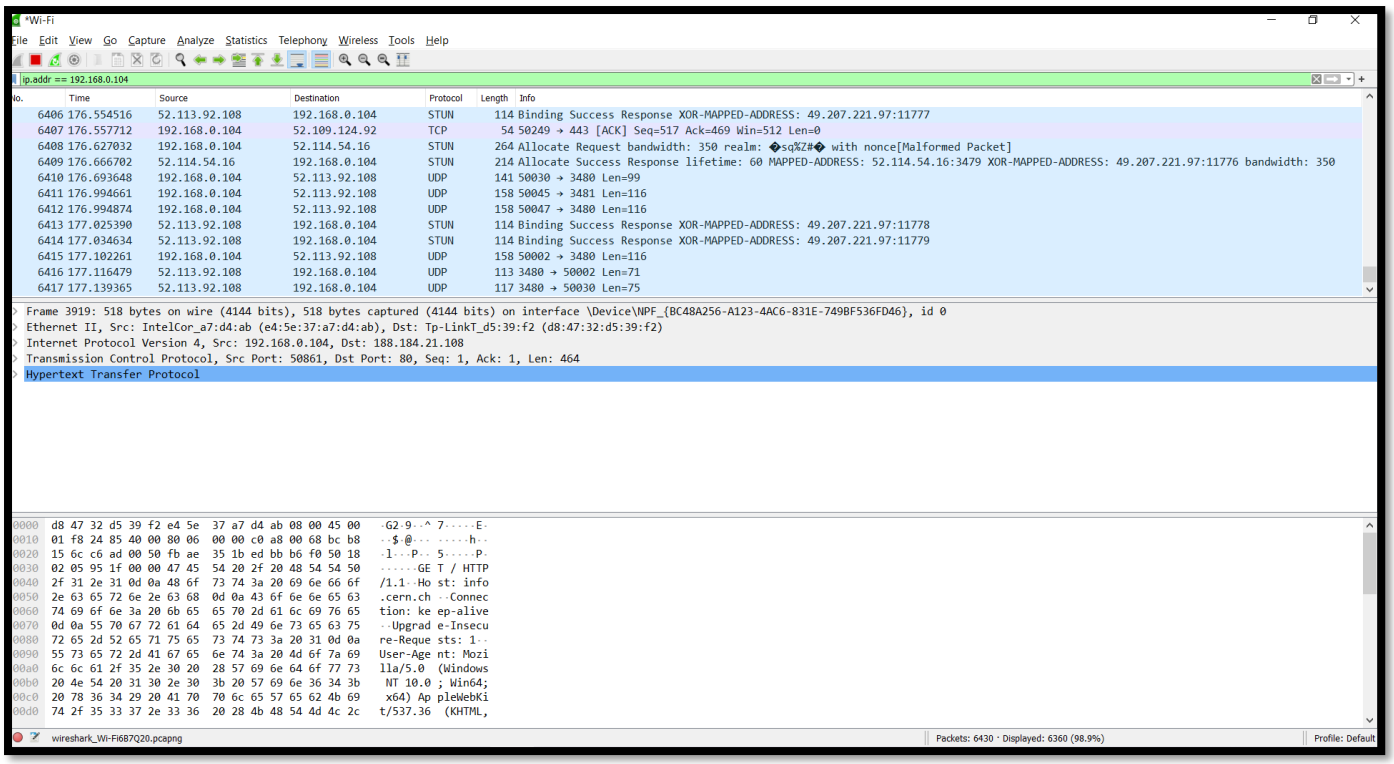
- The installation of this is easily available as we can find the source code at www.wireshark.org and we have the links that are compatible with Linux and x32 bit and x64 bit systems.

Commands in Wireshark

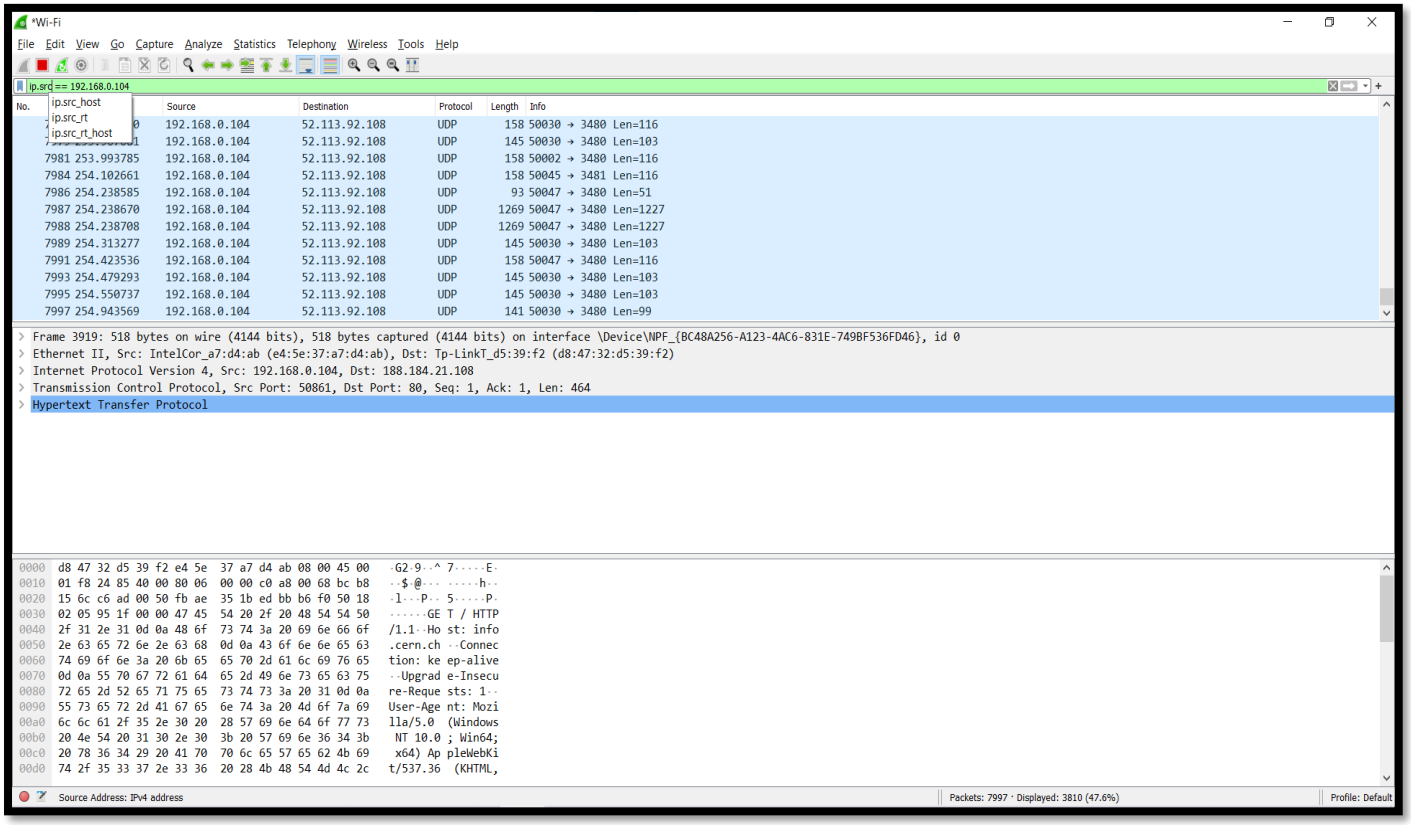
1. Http Filter



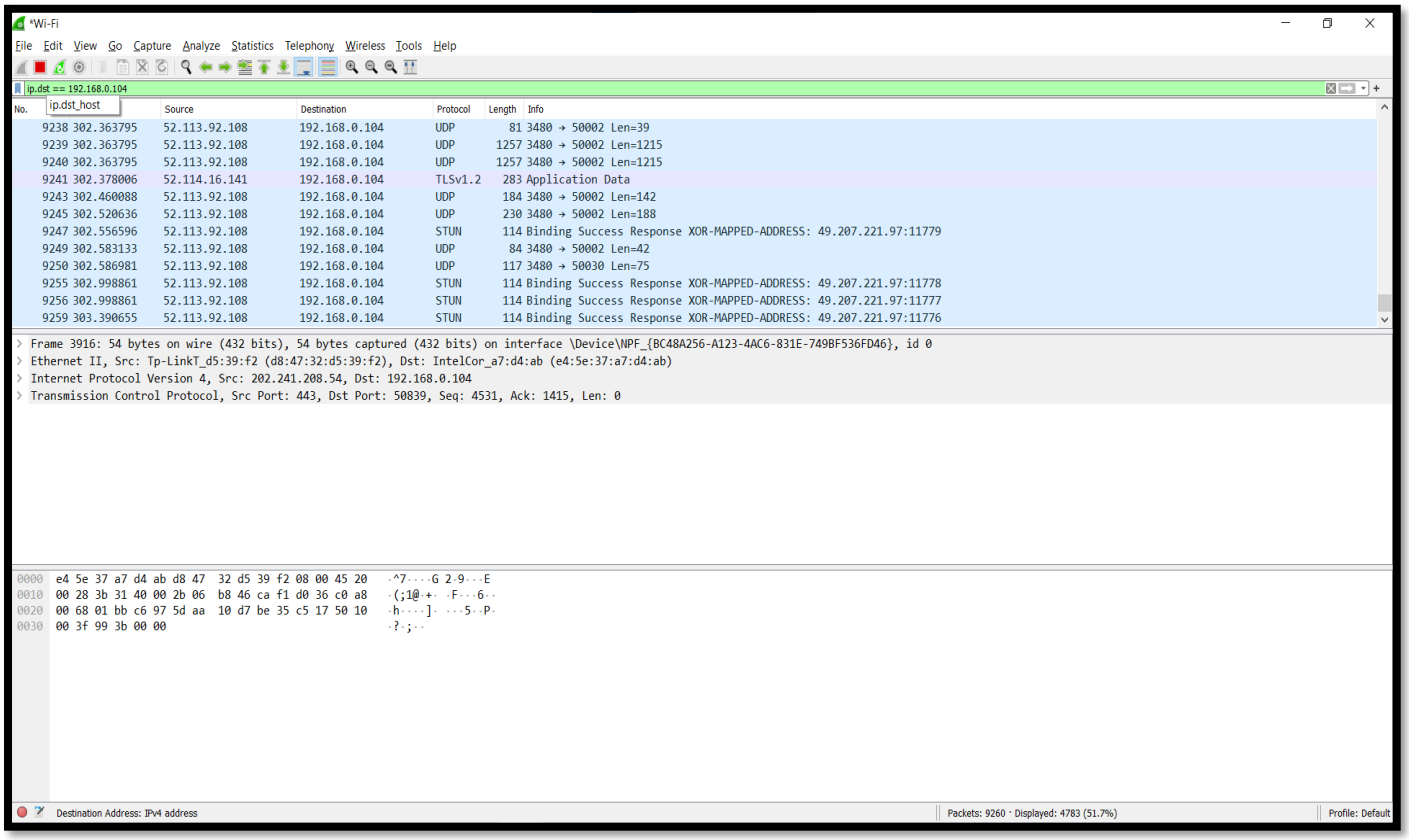
2. Ip.addr==192.168.0.__



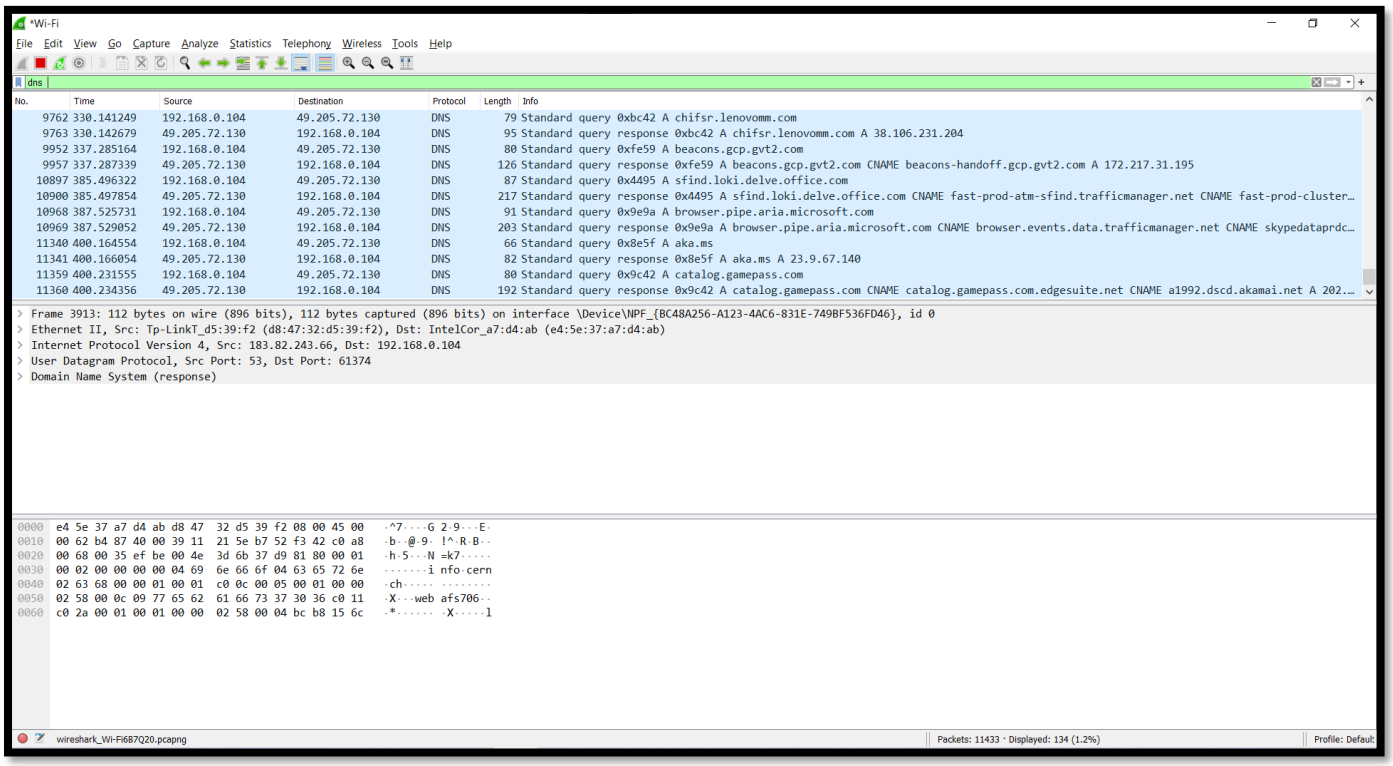
3. Ip.src == 192.168.0.104

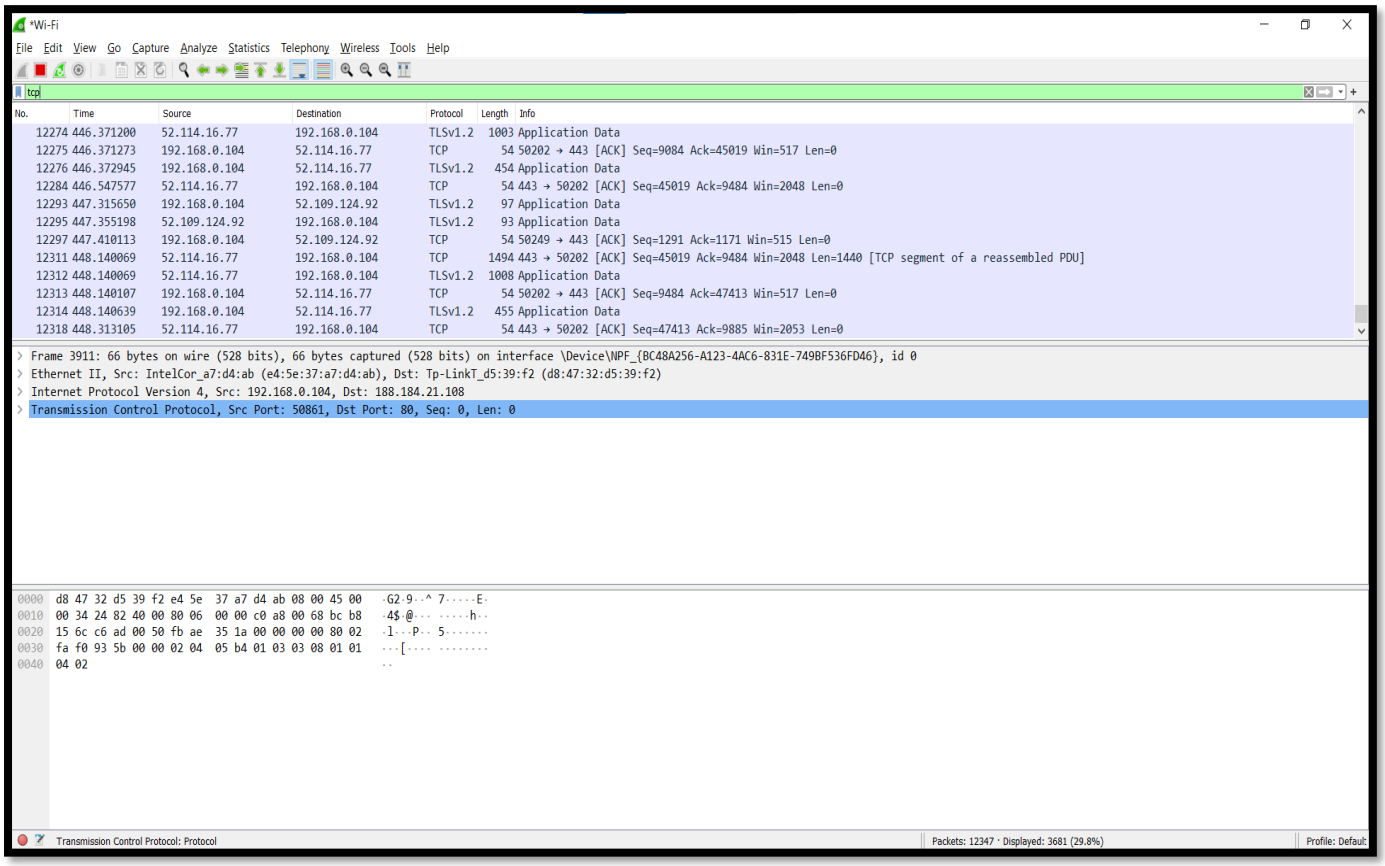


4. Ip.dst == 192.168.0.104

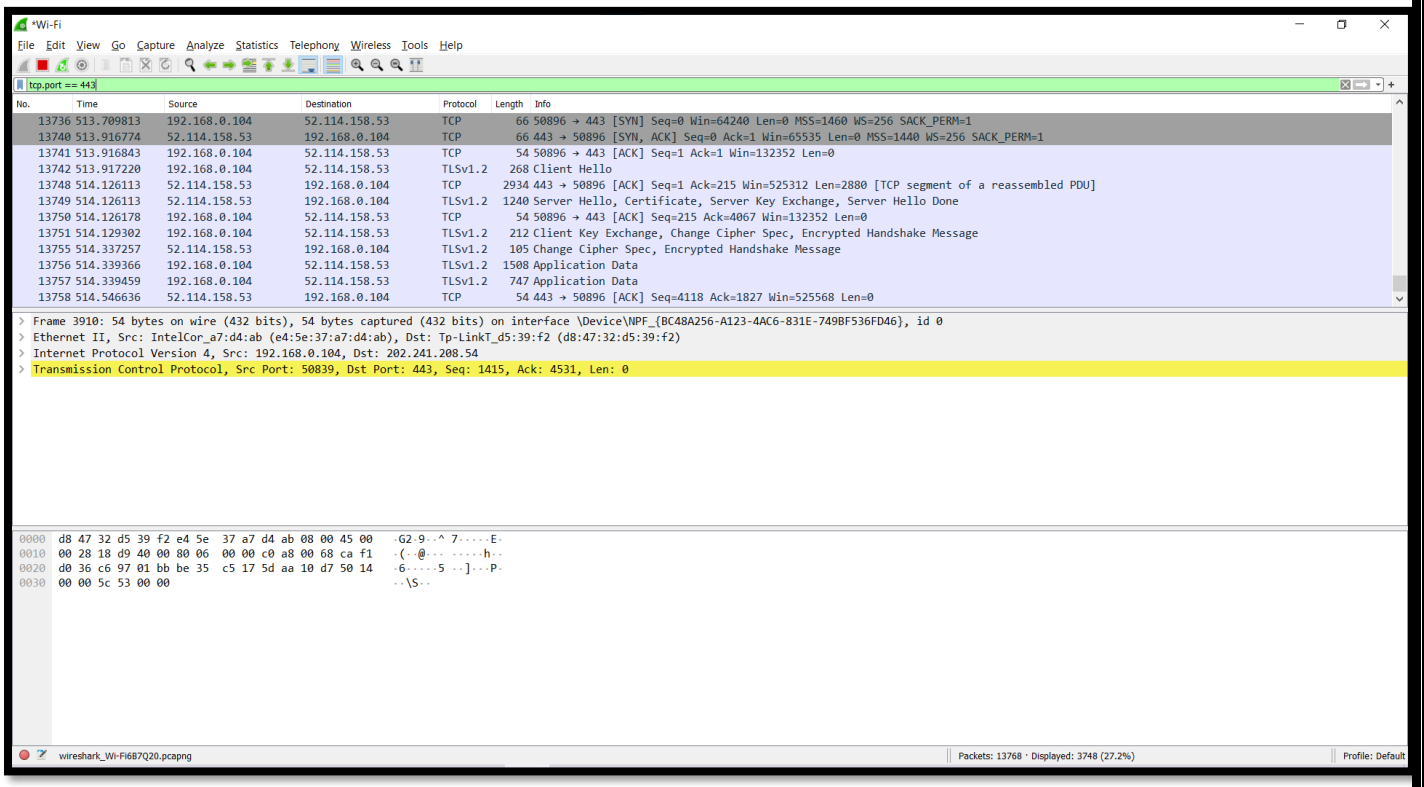


5. Dns or tcp

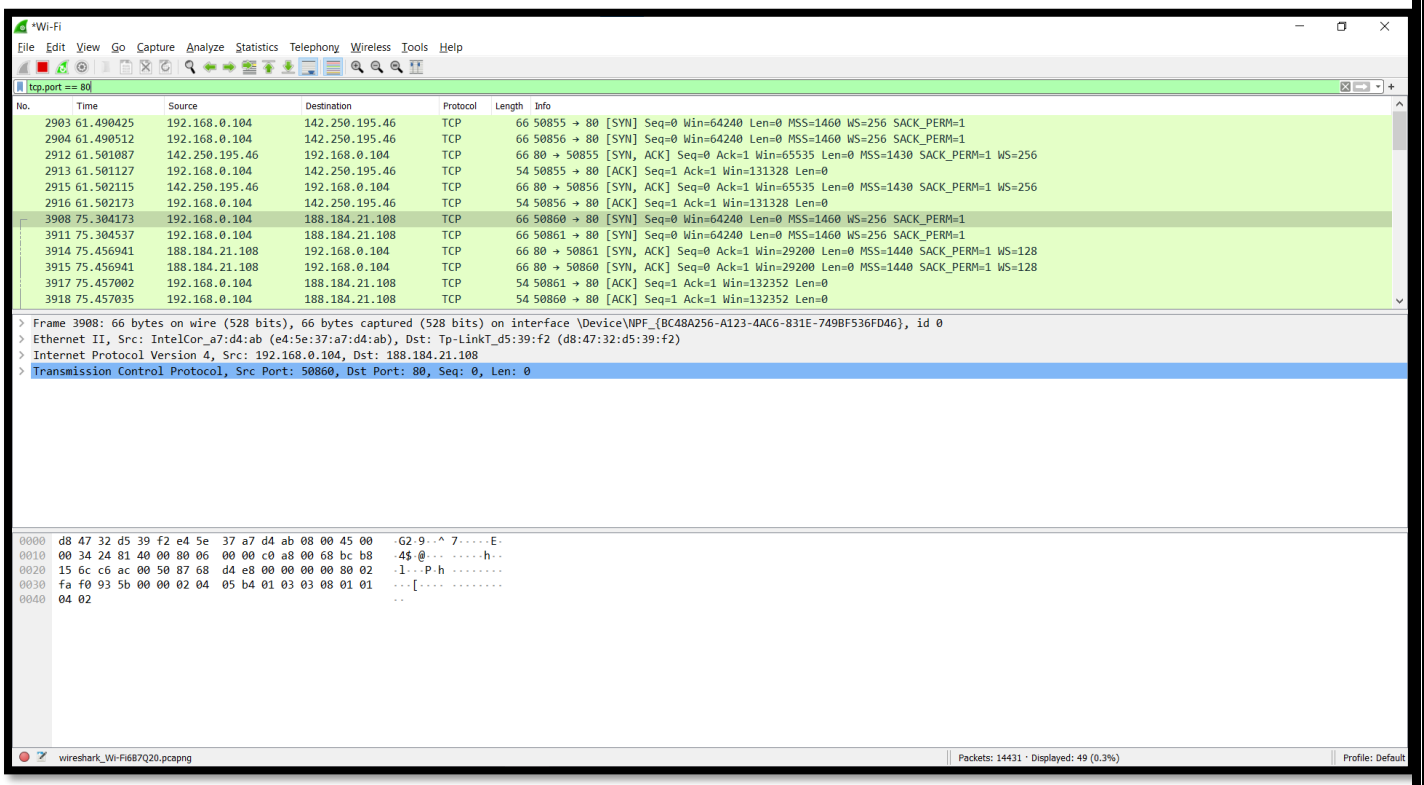

*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
dns
No. Time Source Destination Protocol Length Info
9762 330.141249 192.168.0.104 49.205.72.130 DNS 79 Standard query 0xbc42 A chifsr.lenovomm.com
9763 330.142679 49.205.72.130 192.168.0.104 DNS 95 Standard query response 0xbc42 A chifsr.lenovomm.com A 38.106.231.204
9952 337.285164 192.168.0.104 49.205.72.130 DNS 80 Standard query 0xfe59 A beacons.gcp.gvt2.com
9957 337.287339 49.205.72.130 192.168.0.104 DNS 126 Standard query response 0xfe59 A beacons.gcp.gvt2.com CNAME beacons-handoff.gcp.gvt2.com A 172.217.31.195
10897 385.496322 192.168.0.104 49.205.72.130 DNS 87 Standard query 0x4495 A sfind.loki.delve.office.com
10900 385.497854 49.205.72.130 192.168.0.104 DNS 217 Standard query response 0x4495 A sfind.loki.delve.office.com CNAME fast-prod-atm-sfind.trafficmanager.net CNAME fast-prod-cluster…
10968 387.525731 192.168.0.104 49.205.72.130 DNS 91 Standard query 0x9e9a A browser.pipe.aria.microsoft.com
10969 387.529052 49.205.72.130 192.168.0.104 DNS 203 Standard query response 0x9e9a A browser.pipe.aria.microsoft.com CNAME browser.events.data.trafficmanager.net CNAME skypedataprdc…
11340 400.164554 192.168.0.104 49.205.72.130 DNS 66 Standard query 0x8e5f A aka.ms
11341 400.166054 49.205.72.130 192.168.0.104 DNS 82 Standard query response 0x8e5f A aka.ms A 23.9.67.140
11359 400.231555 192.168.0.104 49.205.72.130 DNS 80 Standard query 0x9c42 A catalog.gamepass.com
11360 400.234356 49.205.72.130 192.168.0.104 DNS 192 Standard query response 0x9c42 A catalog.gamepass.com CNAME catalog.gamepass.com.edgesuite.net CNAME a1992.dscd.akamai.net A 202.…
Frame 3913: 112 bytes on wire (896 bits), 112 bytes captured (896 bits) on interface \Device\NPF_{BC48A256-A123-4AC6-831E-749BF536FD46}, id 0
Ethernet II, Src: Tp-LinkT_d5:39:f2 (d8:47:32:d5:39:f2), Dst: IntelCor_a7:d4:ab (e4:5e:37:a7:d4:ab)
Internet Protocol Version 4, Src: 183.82.243.66, Dst: 192.168.0.104
User Datagram Protocol, Src Port: 53, Dst Port: 61374
Domain Name System (response)
0000 e4 5e 37 a7 d4 ab d8 47 32 d5 39 f2 08 00 45 00
0010 00 62 b4 87 40 00 39 11 21 5e b7 52 f3 42 c0 a8
0020 00 68 00 35 ef be 00 4e 3d 6b 37 d9 81 80 00 01
0030 00 02 00 00 00 00 04 69 6e 66 6f 04 63 65 72 6e
0040 02 63 68 00 00 01 00 01 c0 0c 00 05 00 01 00 00
0050 02 58 00 0c 09 77 65 62 61 66 73 37 30 36 c0 11
0060 c0 2a 00 01 00 01 00 00 02 58 00 04 bc b8 15 6c
wireshark_Wi-Fi687Q20.pcapng
Packets: 11433 · Displayed: 134 (1.2%)
Profile: Default


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp
No. Time Source Destination Protocol Length Info
12274 446.371200 52.114.16.77 192.168.0.104 TLSv1.2 1003 Application Data
12275 446.371273 192.168.0.104 52.114.16.77 TCP 54 50202 → 443 [ACK] Seq=9084 Ack=45019 Win=517 Len=0
12276 446.372945 192.168.0.104 52.114.16.77 TLSv1.2 454 Application Data
12284 446.547577 52.114.16.77 192.168.0.104 TCP 54 443 → 50202 [ACK] Seq=45019 Ack=9484 Win=2048 Len=0
12293 447.315650 192.168.0.104 52.109.124.92 TLSv1.2 97 Application Data
12295 447.355198 52.109.124.92 192.168.0.104 TLSv1.2 93 Application Data
12297 447.410113 192.168.0.104 52.109.124.92 TCP 54 50249 → 443 [ACK] Seq=1291 Ack=1171 Win=515 Len=0
12311 448.140069 52.114.16.77 192.168.0.104 TCP 1494 443 → 50202 [ACK] Seq=45019 Ack=9484 Win=2048 Len=1440 [TCP segment of a reassembled PDU]
12312 448.140069 52.114.16.77 192.168.0.104 TLSv1.2 1008 Application Data
12313 448.140107 192.168.0.104 52.114.16.77 TCP 54 50202 → 443 [ACK] Seq=9484 Ack=47413 Win=517 Len=0
12314 448.140639 192.168.0.104 52.114.16.77 TLSv1.2 455 Application Data
12318 448.313105 52.114.16.77 192.168.0.104 TCP 54 443 → 50202 [ACK] Seq=47413 Ack=9885 Win=2053 Len=0
Frame 3911: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{BC48A256-A123-4AC6-831E-749BF536FD46}, id 0
Ethernet II, Src: IntelCor_a7:d4:ab (e4:5e:37:a7:d4:ab), Dst: Tp-LinkT_d5:39:f2 (d8:47:32:d5:39:f2)
Internet Protocol Version 4, Src: 192.168.0.104, Dst: 188.184.21.108
Transmission Control Protocol, Src Port: 50861, Dst Port: 80, Seq: 0, Len: 0
0000 d8 47 32 d5 39 f2 e4 5e 37 a7 d4 ab 08 00 45 00
0010 00 34 24 82 40 00 80 06 00 00 c0 a8 00 68 bc b8
0020 15 6c c6 ad 00 50 fb ae 35 1a 00 00 00 00 80 02
0030 fa f0 93 5b 00 00 02 04 05 b4 01 03 03 08 01 01
0040 04 02
Transmission Control Protocol: Protocol
Packets: 12347 · Displayed: 3681 (29.8%)
Profile: Default

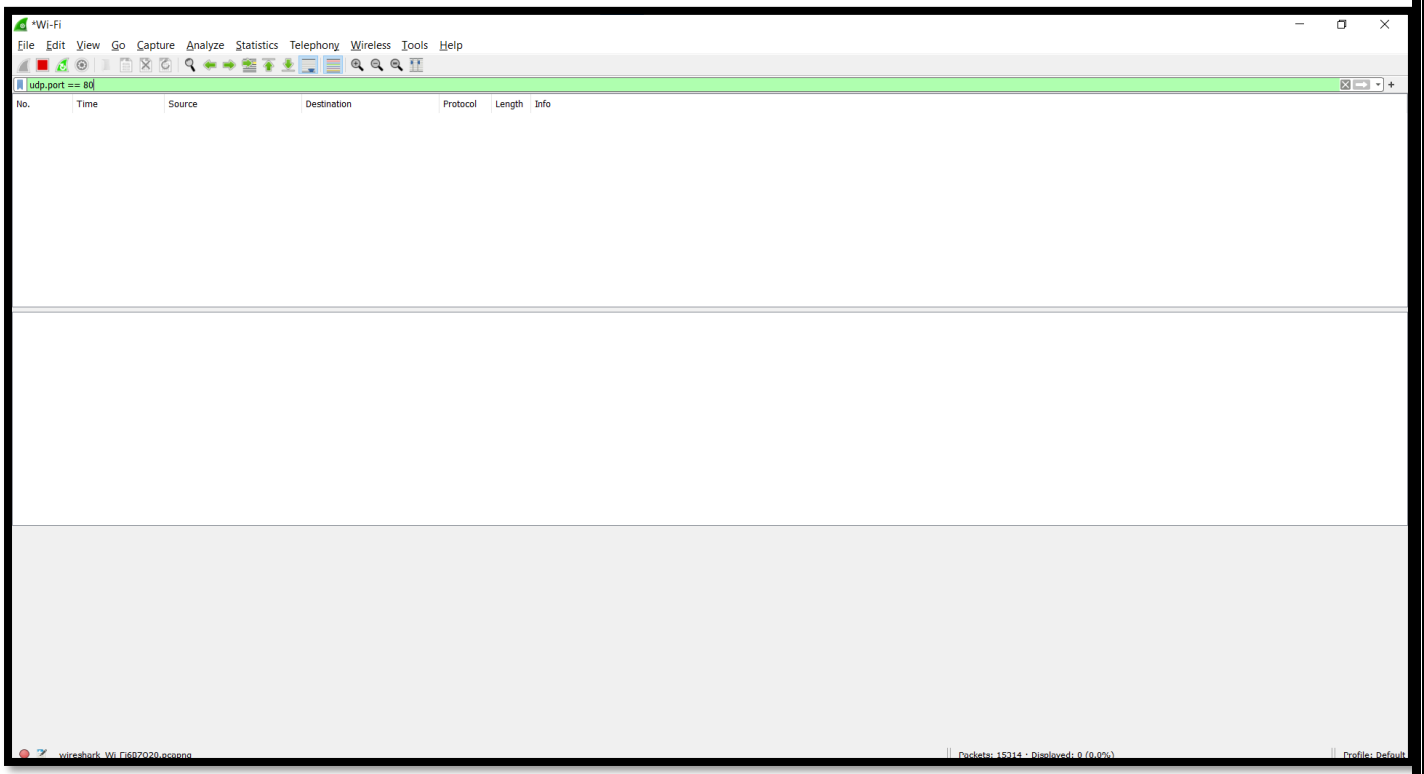
6. Tcp.port ==443



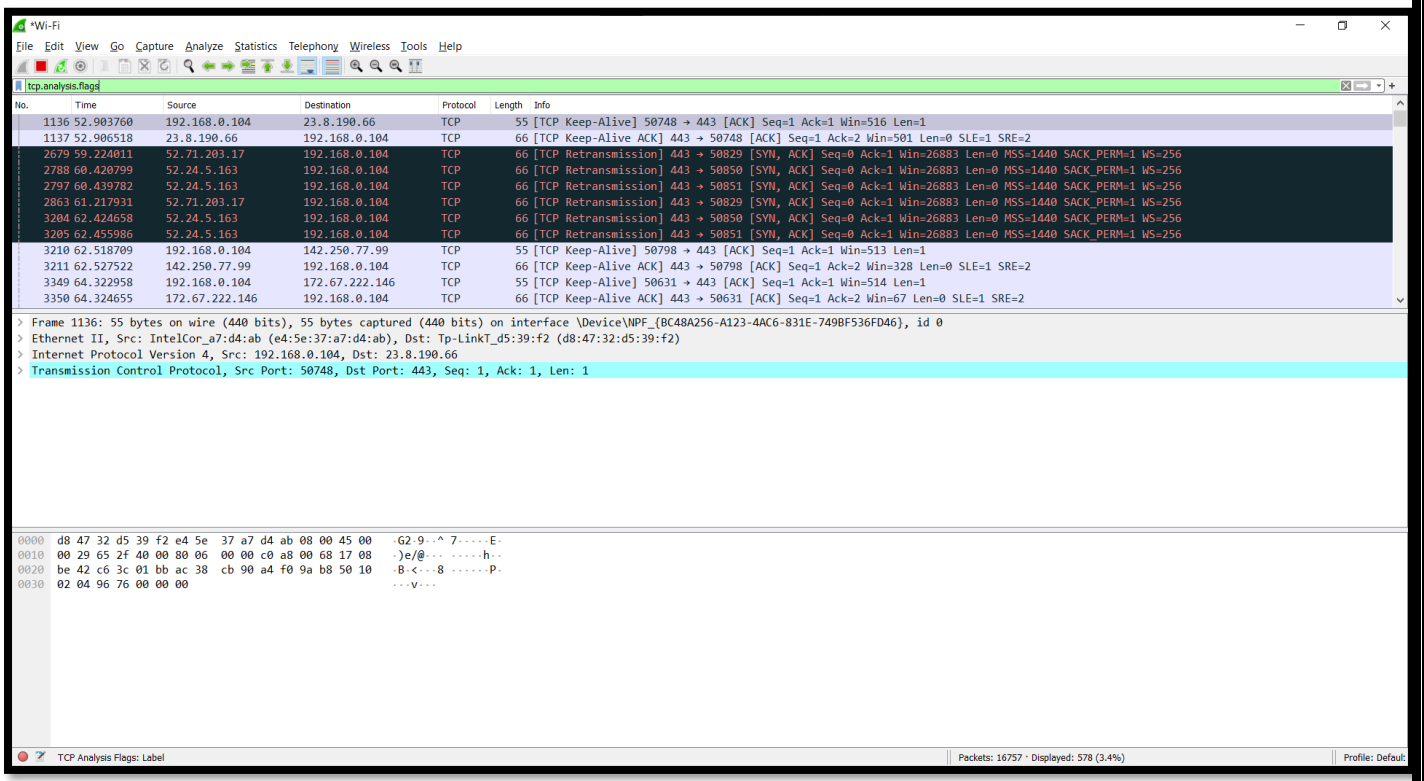
7. Tcp.port == 80



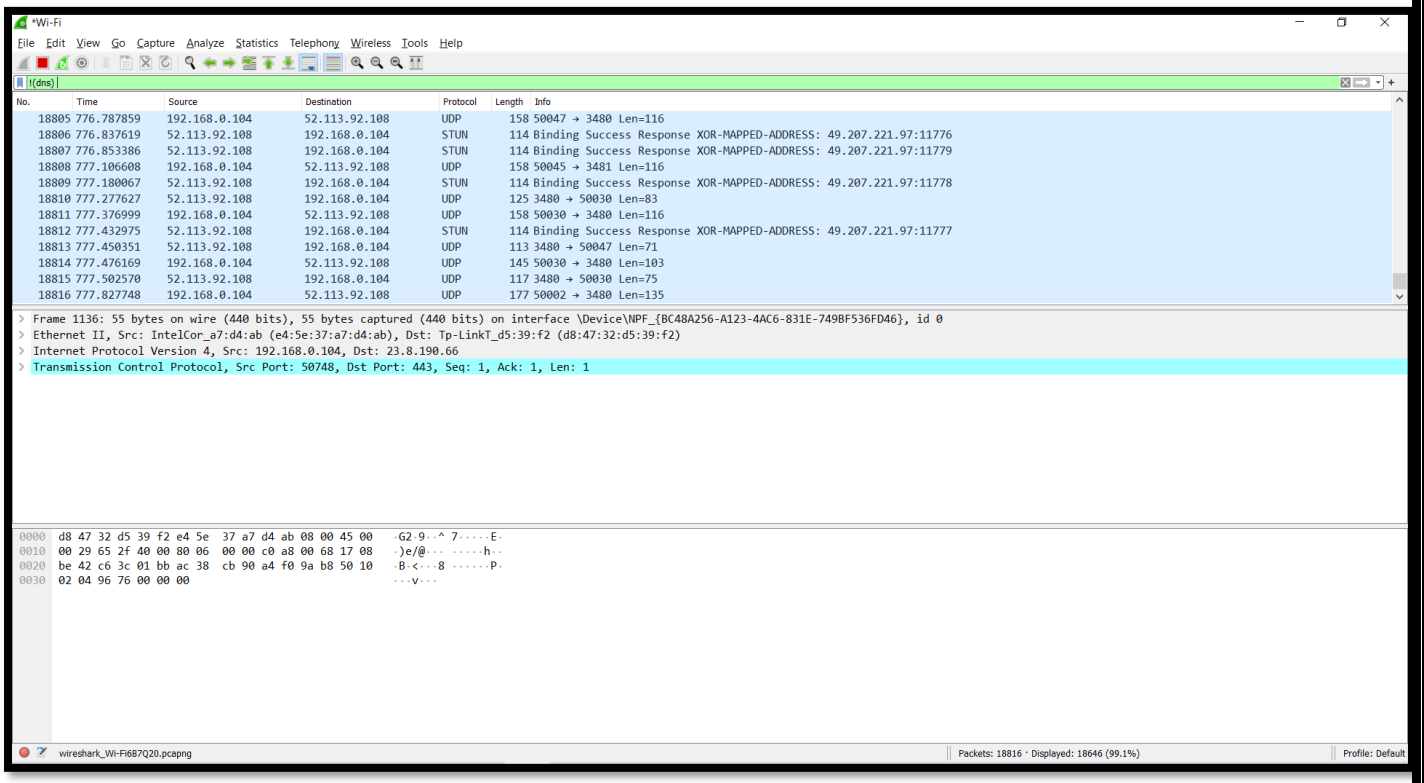
8. Udp.port == 80



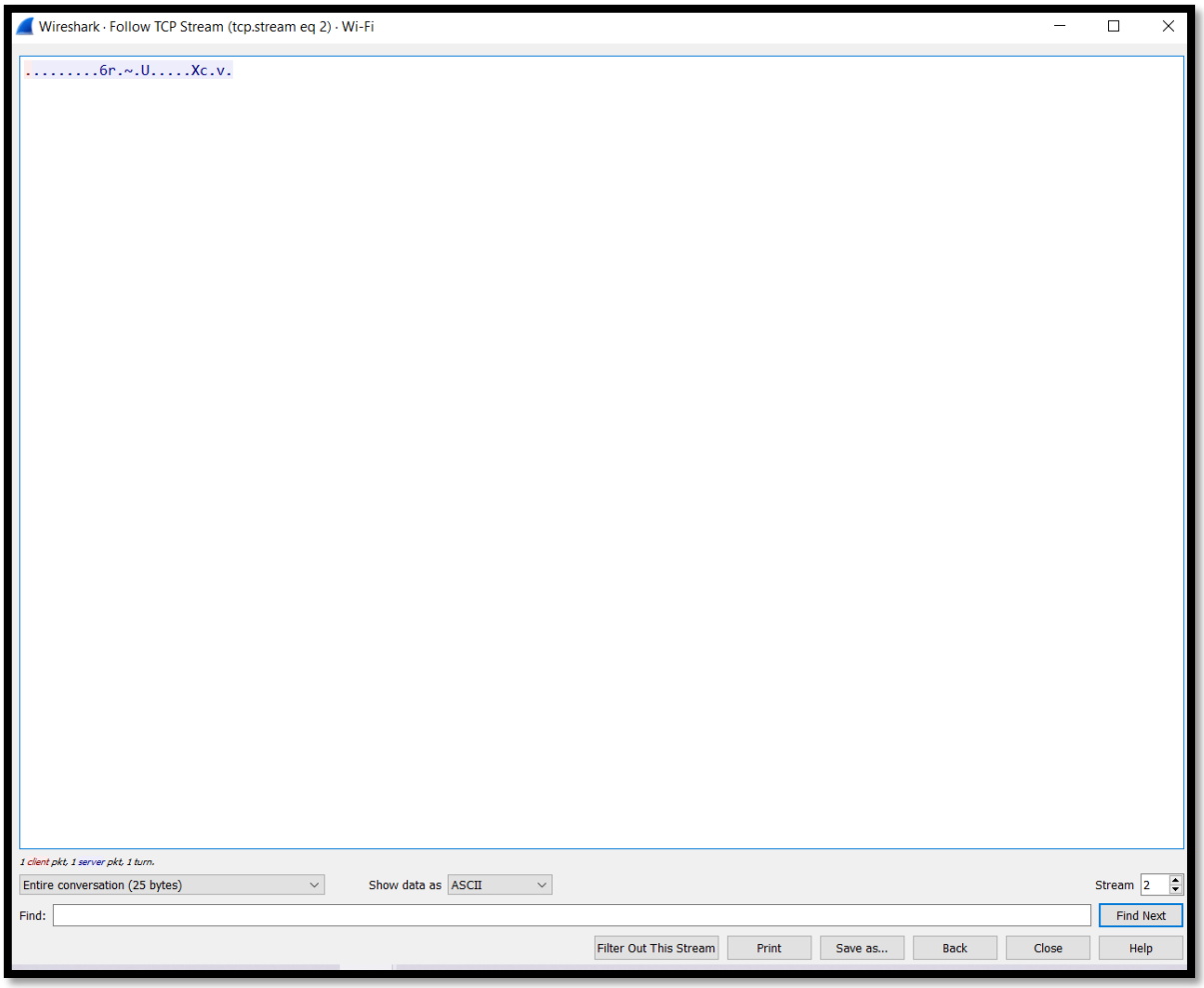
9. Tcp.analysis.flags



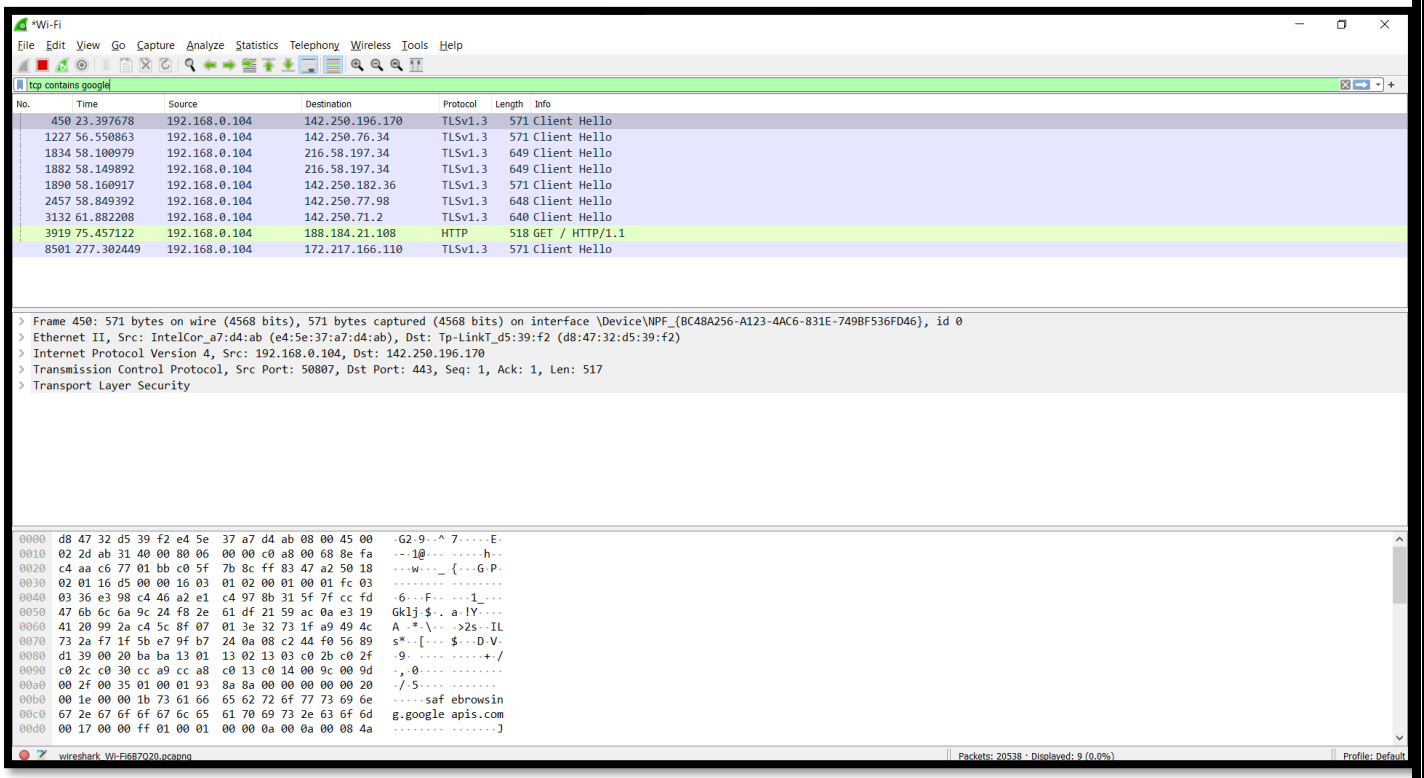
10. !(udp)



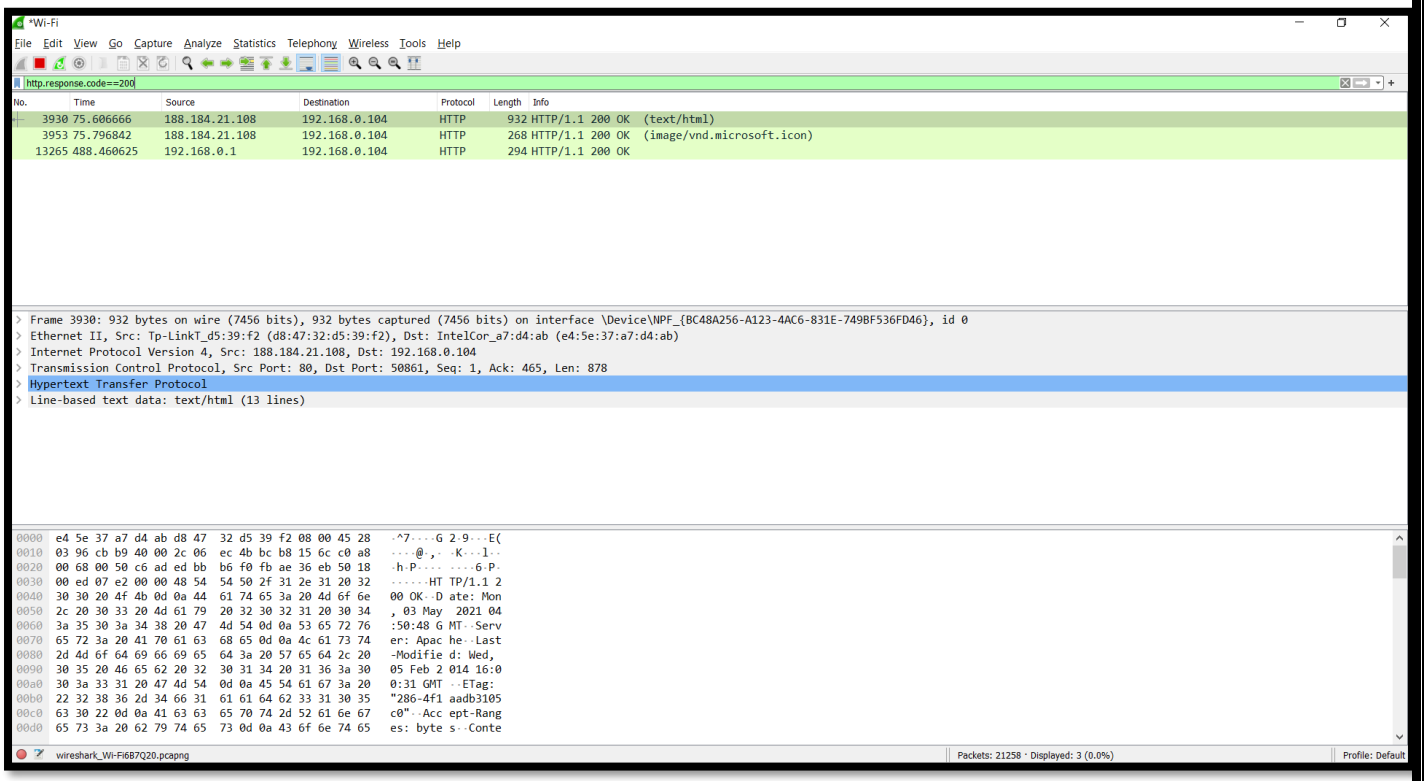
11. Follow tcp stream



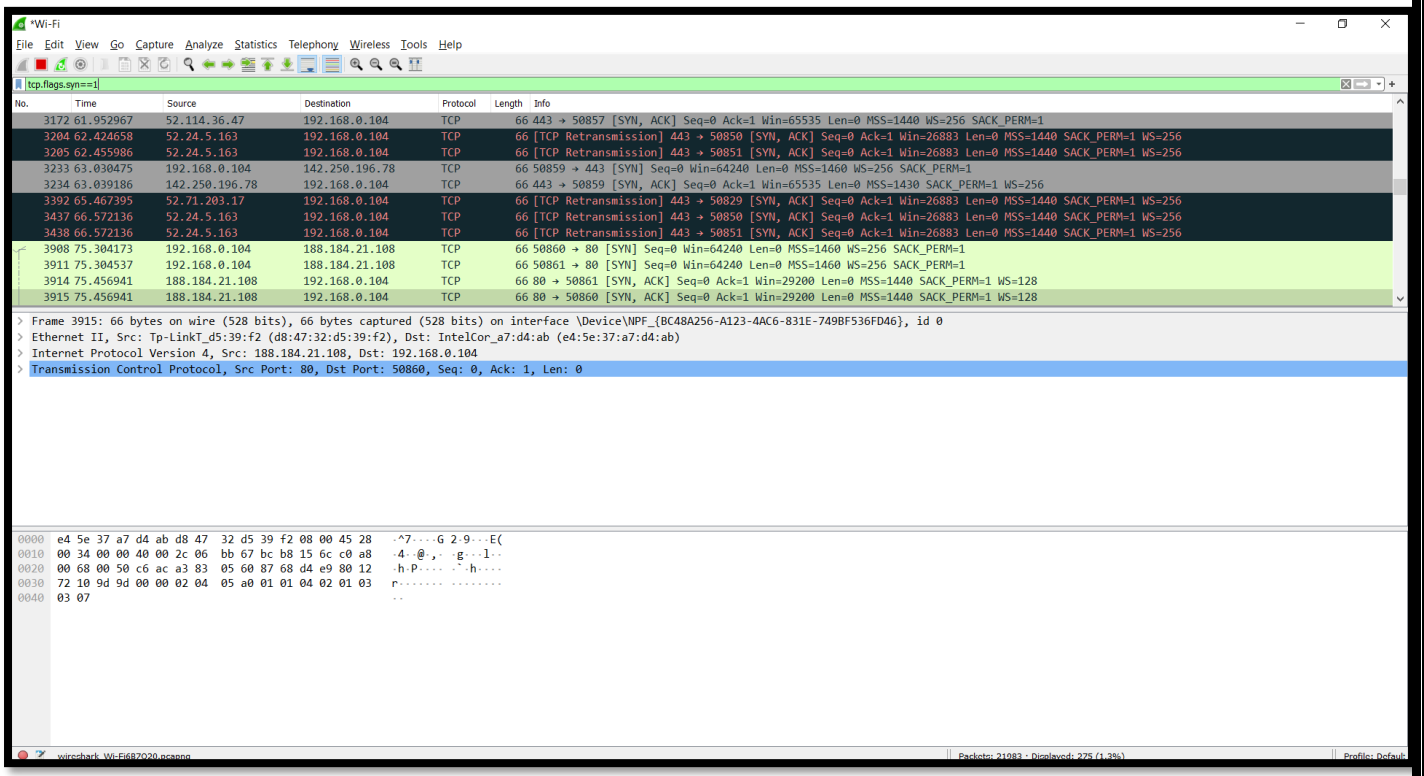
12. Tcp contains google



13. Tcp.response.code==200



14. Tcp.flags.syn==1



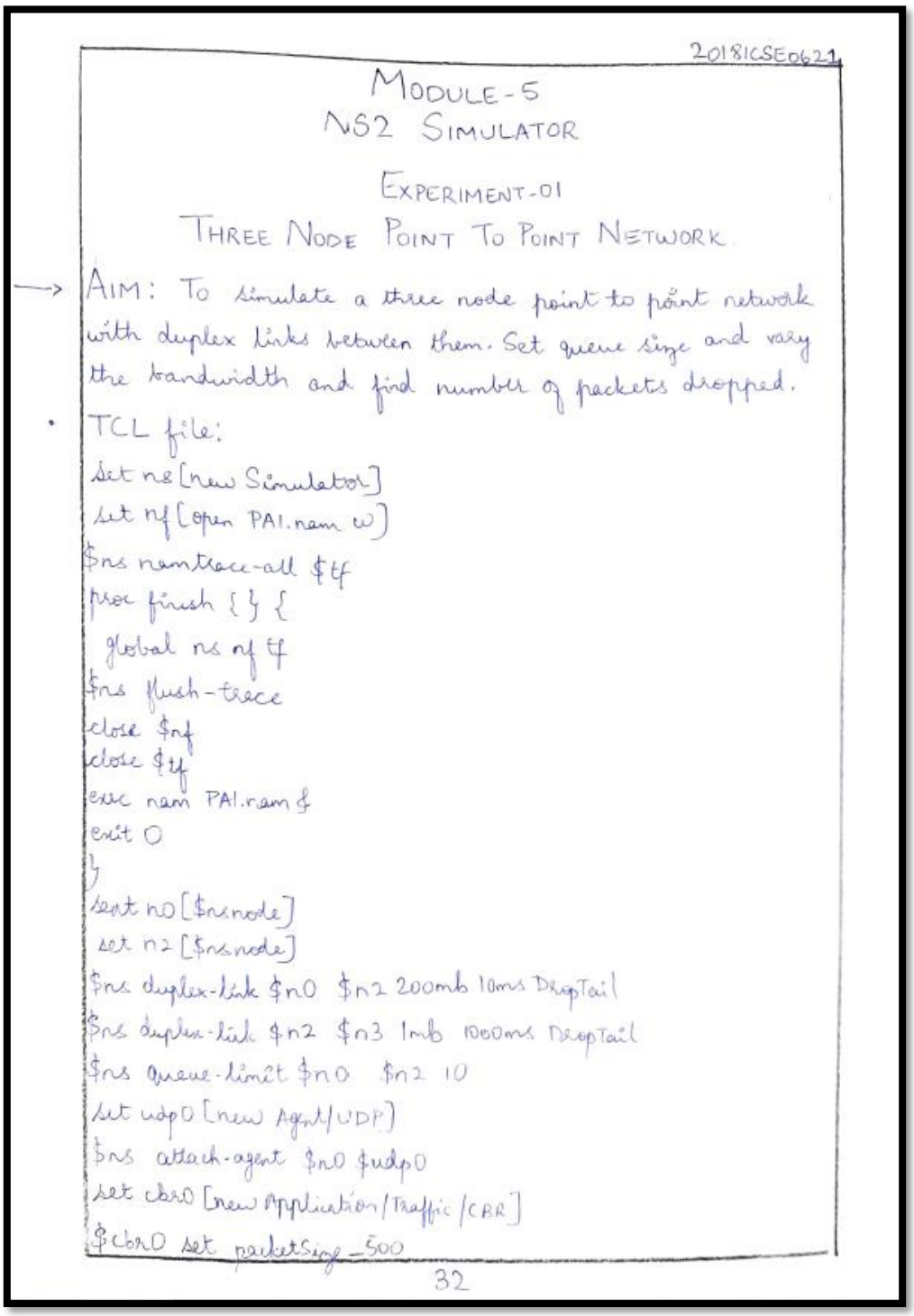
Module – 5

NS2 Simulator

Experiment – 01

Four Node Point to Point Network

TCL file :



20181CSE0621

MODULE-5
NS2 SIMULATOR

EXPERIMENT-01
THREE NODE POINT TO POINT NETWORK

→ AIM: To simulate a three node point to point network with duplex links between them. Set queue size and vary the bandwidth and find number of packets dropped.

• TCL file:

```
Set ns [new Simulator]
Set nf [open PA1.nam w]
$ns namtrace-all $tf
proc finish { } {
 global ns nf tf
$ns flush-trace
close $nf
close $tf
exec nam PA1.nam &
exit 0
}
Set n0 [$ns node]
 Set n2 [$ns node]
$ns duplex-link $n0 $n2 200mb 10ms DropTail
$ns duplex-link $n2 $n3 1mb 1000ms DropTail
$ns queue-limit $n0 $n2 10
Set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
Set cbr0 [new Application/Traffic/CBR]
$cbr0 Set packetSize_500
```

32

AWK file: -

20181CSE0621

```
$cbr0 set_interval_0.005
$ cbr0 attach-agent $udp0
set null0 [new Agnt/NULL)
$ns attach agnt $n3 $null0
$ns connect $udp0 $null0
$ ns at 0.1 "$cbr0 start"
$ns at 1.0 "finish"
$ns Run.
```

—> AWK file:

```
BEGIN {c=0;}
{
 if ($1=="d")
 { c++;
   printf("%s\t%s\n", $5,$11);
 }
 END{
 printf("The number of packets dropped =%d\n",c);
    }
```

33

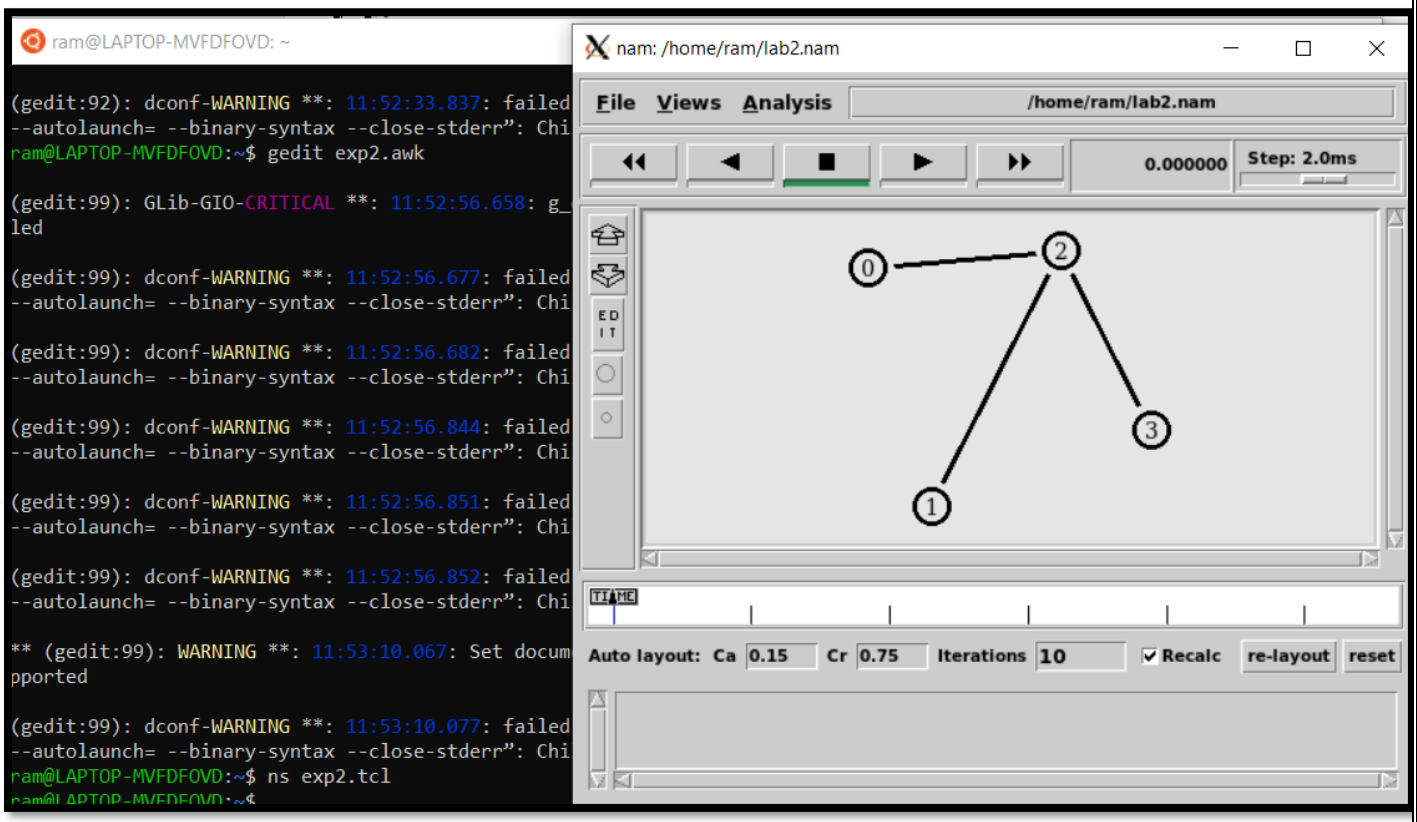
OUTPUT

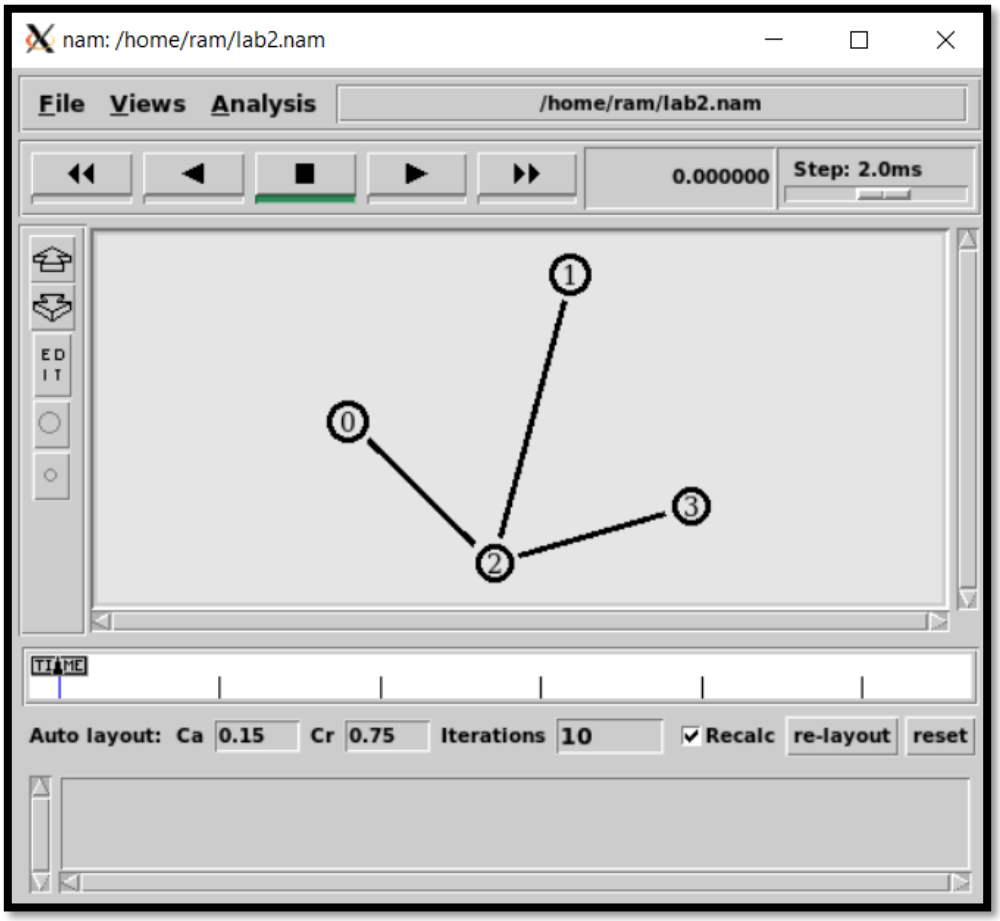
exp2.tcl

```
set ns [new Simulator]
set nf [open lab2.nam w]
$ns namtrace-all $nf
set tf [open lab2.tr w]
$ns trace-all $tf
proc finish { } {
global ns nf tf
$ns flush-trace
close $nf
close $tf
exec nam lab2.nam &
exit 0
}
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
$ns duplex-link $n0 $n2 10Mb 1ms DropTail
$ns duplex-link $n1 $n2 10Mb 1ms DropTail
$ns duplex-link $n2 $n3 10Mb 1ms DropTail
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1
set null0 [new Agent/Null]
$ns attach-agent $n3 $null0
set sink0 [new Agent/TCPSink]
$ns attach-agent $n3 $sink0
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1
$ns connect $tcp0 $sink0
$ns connect $udp1 $null0
$ns at 0.1 "$cbr1 start"
$ns at 0.2 "$ftp0 start"
$ns at 0.5 "finish"
$ns run
```

exp2.awk

```
BEGIN{
udp=0;
tcp=0;
}
{
if($1= = “r” && $5 = = “cbr”)
{
udp++;
}
else if($1 = = “r” && $5 = = “tcp”)
{
tcp++;
}
}
END{
printf(“Number of packets sent by TCP = %d\n”, tcp); printf(“Number of packets sent by
UDP=%d\n”,udp);
}
```


ram@LAPTOP-MVFDFOVD: ~
(gedit:92): dconf-WARNING **: 11:52:33.837: failed
--autolaunch= --binary-syntax --close-stderr": Chi
ram@LAPTOP-MVFDFOVD:~$ gedit exp2.awk
(gedit:99): GLib-GIO-CRITICAL **: 11:52:56.658: g_
led
(gedit:99): dconf-WARNING **: 11:52:56.677: failed
--autolaunch= --binary-syntax --close-stderr": Chi
(gedit:99): dconf-WARNING **: 11:52:56.682: failed
--autolaunch= --binary-syntax --close-stderr": Chi
(gedit:99): dconf-WARNING **: 11:52:56.844: failed
--autolaunch= --binary-syntax --close-stderr": Chi
(gedit:99): dconf-WARNING **: 11:52:56.851: failed
--autolaunch= --binary-syntax --close-stderr": Chi
(gedit:99): dconf-WARNING **: 11:52:56.852: failed
--autolaunch= --binary-syntax --close-stderr": Chi
** (gedit:99): WARNING **: 11:53:10.067: Set docum
pported
(gedit:99): dconf-WARNING **: 11:53:10.077: failed
--autolaunch= --binary-syntax --close-stderr": Chi
ram@LAPTOP-MVFDFOVD:~$ ns exp2.tcl
nam: /home/ram/lab2.nam
File Views Analysis
/home/ram/lab2.nam
0.000000
Step: 2.0ms
0
1
2
3
TIME
Auto layout: Ca 0.15 Cr 0.75 Iterations 10 Recalc re-layout reset


nam: /home/ram/lab2.nam
File Views Analysis
/home/ram/lab2.nam
0.000000
Step: 2.0ms
0
1
2
3
TIME
Auto layout: Ca 0.15 Cr 0.75 Iterations 10 Recalc re-layout reset

Experiment – 2

Transmission of Ping Message

TCL File: -

20181CSE0621

EXPERIMENT-02

TRANSMISSION OF PING MESSAGE

AIM: To simulate transmission of ping message over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion.

TCL file:

```
set ns [new Simulator]
set nf [open lab4.nam w]
$ns nam-trace -all $nf
set tf [open lab4.tr w]
$ns trace-all $tf
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$ns duplex-link $n0 $n4 100mb 1ms DropTail
$ns duplex-link $n1 $n4 50mb 1ms DropTail
$ns duplex-link $n2 $n4 200mb 1ms DropTail
$ns duplex-link $n3 $n4 200mb 1ms DropTail
$ns duplex-link $n4 $n5 1mb 1ms DropTail
set p1 [new Agent/ping]
$ns attach-agent $n0 $n1
$p1 packet_size_50000
$p1 set_interval_0.0001
set p2 [new Agent/Ping]
$ns attach-agent $n2 $p3
```

34

20181CSE0621

```
$p3 set packetSize_ 30000
$p3 set_interval_ 0.00001
set p4 [new Agent/Ping]
ns attach-agent $n3 $p4
set p5 [new Agent/Ping]
$ns queue-limit $n0 $n4 5
$ns queue-limit $n2 $n4 3
$ns queue-limit $n4 $n5 2
Agent/Ping instproc recv {from rtt} {
$self instvar node_
puts "node [$node_id] from $from with $rtt msec.
$ns connect $p1 $p5
$ns connect $p3 $p4
proc finish {} {
global ns tf nf
$ns flush-trace
close $nf
close $tf
exec nam lab4.nam &
exit 0
}
$ns at 0.1 "$p1 send"
$ns at 0.2 "$p1 send"
        :
$ns at 0.9 "$p1 send"
$ns at 1.0 "$p1 send"
$ns at 1.1 "$p1 send"
        :
$ns at 1.9 "$p1 send"
```

```
20181CSE0621
$ns at 2.0 "$p3 send"
$ns at 2.1 "$p3 send"
$ns at 2.9 "$p3 send"
$ns at 0.1 "p3 send"
$ns at 0.9 "p3 send"
$ns at 1.0 "p3 send"
$ns at 1.1 "p3 send"
$ns at 2.0 "p3 send"
$ns at 2.1 "p3 send"
$ns at 2.9 "p3 send"
$ns at 3.0 "finish"
$ns run
```

AWK File: -

```
Awk file:
BEGIN{
drop=0;
}{
  if($1=="d")
  {
    drop++;
  }
  END{
    printf("Total %s packets dropped: %d\n",$5,drop);
    }
```

OUTPUT

TCL

exp3.tcl

```
set ns [ new Simulator ]
set nf [ open lab4.nam w ]
$ns namtrace-all $nf
set tf [ open lab4.tr w ]
$ns trace-all $tf
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$ns duplex-link $n0 $n4 1005Mb 1ms DropTail
$ns duplex-link $n1 $n4 50Mb 1ms DropTail
$ns duplex-link $n2 $n4 2000Mb 1ms DropTail
$ns duplex-link $n3 $n4 200Mb 1ms DropTail
$ns duplex-link $n4 $n5 1Mb 1ms DropTail
set p1 [new Agent/Ping]
$ns attach-agent $n0 $p1
$p1 set packetSize_ 50000
$p1 set interval_ 0.0001
set p2 [new Agent/Ping]
$ns attach-agent $n1 $p2
set p3 [new Agent/Ping]
$ns attach-agent $n2 $p3
$p3 set packetSize_ 30000
$p3 set interval_ 0.00001
set p4 [new Agent/Ping]
$ns attach-agent $n3 $p4
set p5 [new Agent/Ping]
$ns attach-agent $n5 $p5
$ns queue-limit $n0 $n4 5
$ns queue-limit $n2 $n4 3
$ns queue-limit $n4 $n5 2
Agent/Ping instproc recv {from rtt} {
$self instvar node_
puts "node [$node_ id]received answer from $from with round trip time $rtt msec" }

$ns connect $p1 $p5
$ns connect $p3 $p4
```

exp3.tcl

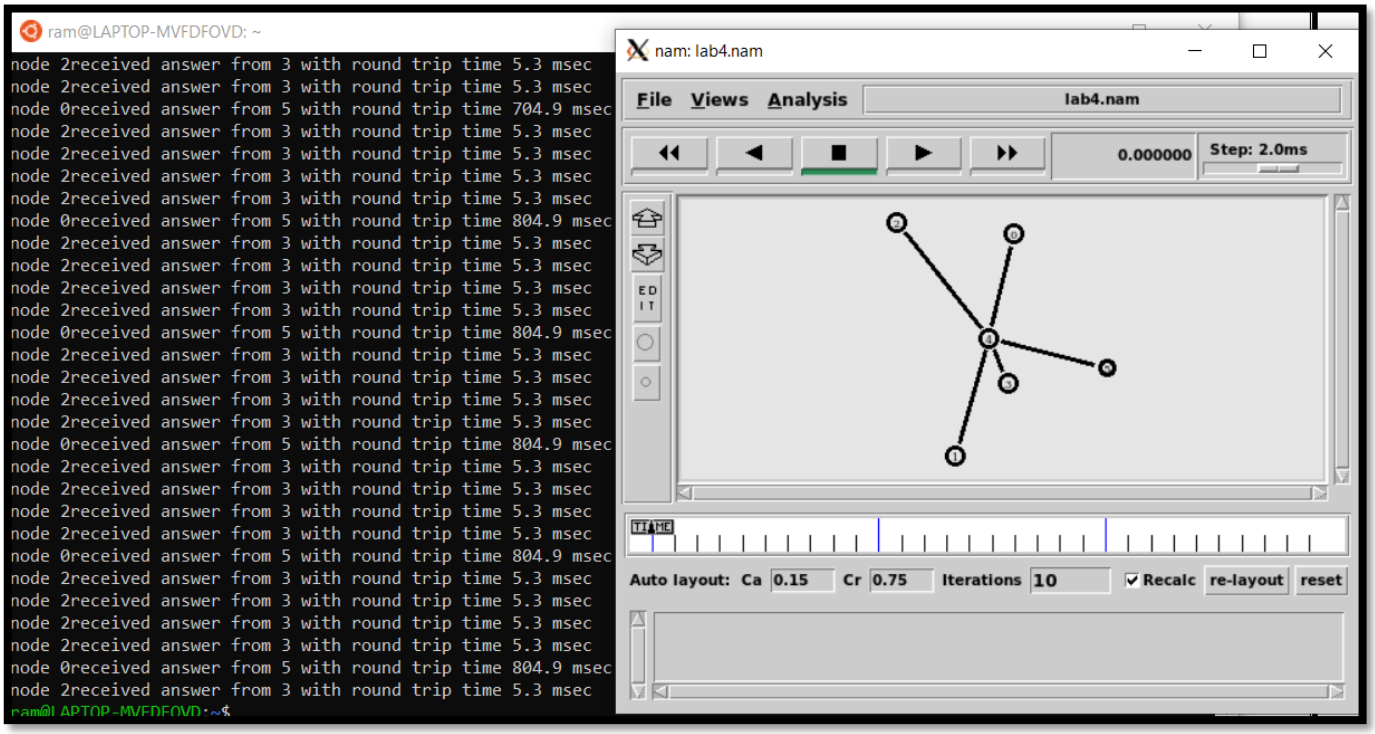
```
$ns at 2.2 "$p1 send"
$ns at 2.3 "$p1 send"
$ns at 2.4 "$p1 send"
$ns at 2.5 "$p1 send"
$ns at 2.6 "$p1 send"
$ns at 2.7 "$p1 send"
$ns at 2.8 "$p1 send"
$ns at 2.9 "$p1 send"
$ns at 0.1 "$p3 send"
$ns at 0.2 "$p3 send"
$ns at 0.3 "$p3 send"
$ns at 0.4 "$p3 send"
$ns at 0.5 "$p3 send"
$ns at 0.6 "$p3 send"
$ns at 0.7 "$p3 send"
$ns at 0.8 "$p3 send"
$ns at 0.9 "$p3 send"
$ns at 1.0 "$p3 send"
$ns at 1.1 "$p3 send"
$ns at 1.2 "$p3 send"
$ns at 1.3 "$p3 send"
$ns at 1.4 "$p3 send"
$ns at 1.5 "$p3 send"
$ns at 1.6 "$p3 send"
$ns at 1.7 "$p3 send"
$ns at 1.8 "$p3 send"
$ns at 1.9 "$p3 send"
$ns at 2.0 "$p3 send"
$ns at 2.1 "$p3 send"
$ns at 2.2 "$p3 send"
$ns at 2.3 "$p3 send"
$ns at 2.4 "$p3 send"
$ns at 2.5 "$p3 send"
$ns at 2.6 "$p3 send"
$ns at 2.7 "$p3 send"
$ns at 2.8 "$p3 send"
$ns at 2.9 "$p3 send"
$ns at 3.0 "finish"
$ns run
```

Tcl Tab Width: 8 Ln 1, Col 1 INS

AWK

exp3.awk

```
BEGIN{
drop=0;
}
{
if($1=="d")
{
drop++;
}
}
END{
printf("Total number of %s packets dropped due to congestion =%d\n",$5,drop);
}
```



Experiment – 3

Ethernet LAN using N-Nodes

TCL File: -

20181CSE0621

EXPERIMENT-03

ETHERNET LAN USING n nodes and set multiple traffic nodes and plot congestion window.

AIM: To simulate an Ethernet LAN using n nodes and set multiple traffic nodes and plot congestion window for different source/destination.

TCL file:-

```
Set ns [new Simulator]
set tf [open p7.tr w]
$ns trace-all $tf
set nf [open p7.nam w]
$ns namtrace-all $nf
set n0 [$ns node]
$n0 color "magenta"
$n0 label "src1"
set n1 [$ns node]
set n2 [$ns node]
$n2 color "magenta"
$n2 label "src2"
set n3 [$ns node]
$n3 color "blue"
$n3 label "dest2"
set n4 [$ns node]
set n5 [$ns node]
$n5 color "blue"
$n5 label "dest1"
$ns make-lan "$n0 $n1 $n2 $n3 $n4" 100mb 100ms LL Queue/DropTail MAC/802_3
```

37

20181CSE0621

```
$ns duplex-link $n4 $n5 1mb 1ms DropTail
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set ftp0 [new Appliation/TCP]
$ ftp0 attach-agent tcp0
$ ftp0 set packet_size 500
$ ftp set interval 0.0001
sent sink5 [new Agent/TCPSink]
$ns attach-agent $n5 $sink5
set tcp2 [new Agent/TCP]
$ns attach-agent $n2 $tcp2
set ftp2 [new Appliation/FTP]
$tcp2 attach$file2
$tcp0 trace cwnd_
cwnd_
proc finish{} {
 global ns nf tf
$ns flush-trace
close $tf
close $nf
exec nam p7.nam &
exit 0
}
```

20181CSE0621

```
$ns at 0.1 "$ftp0 start"
$ns at 5 "$ftp0 stop"
$ns at 7 "$ftp0 start"
$ns at 0.2 "$ftp2 start"
$ns at 8 "$ftp2 stop"
$ns at 14 "$ftp0 stop"
$ns at 10 "$ftp2 start"
$ns at 15 "$ftp2 stop"
$ns at 16 "finish"
$ns run.
```

AWK File: -

AWK file:

```
BEGIN {
{
  if ($6 == "cwnd_")
  printf("%f\t%f\t\n", $1, $7);
}
END{
}
```

OUTPUT

TCL

exp5.tcl

```
set ns [new Simulator]
set tf [open pgm7.tr w]
$ns trace-all $tf
set nf [open pgm7.nam w]
$ns namtrace-all $nf
set n0 [$ns node]
$n0 color "magenta"
$n0 label "src1"
set n1 [$ns node]
set n2 [$ns node]
$n2 color "magenta"
$n2 label "src2"
set n3 [$ns node]
$n3 color "blue"
$n3 label "dest2"
set n4 [$ns node]
set n5 [$ns node]
$n5 color "blue"
$n5 label "dest1"
$ns make-lan "$n0 $n1 $n2 $n3 $n4" 100Mb 100ms LL Queue/ DropTail Mac/802_3

$ns duplex-link $n4 $n5 1Mb 1ms DropTail
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ftp0 set packetSize_ 500
$ftp0 set interval_ 0.0001
set sink5 [new Agent/TCPSink]
$ns attach-agent $n5 $sink5
$ns connect $tcp0 $sink5
set tcp2 [new Agent/TCP]
$ns attach-agent $n2 $tcp2
set ftp2 [new Application/FTP]
$ftp2 attach-agent $tcp2
$ftp2 set packetSize_ 600
$ftp2 set interval_ 0.001
set sink3 [new Agent/TCPSink]
$ns attach-agent $n3 $sink3
```

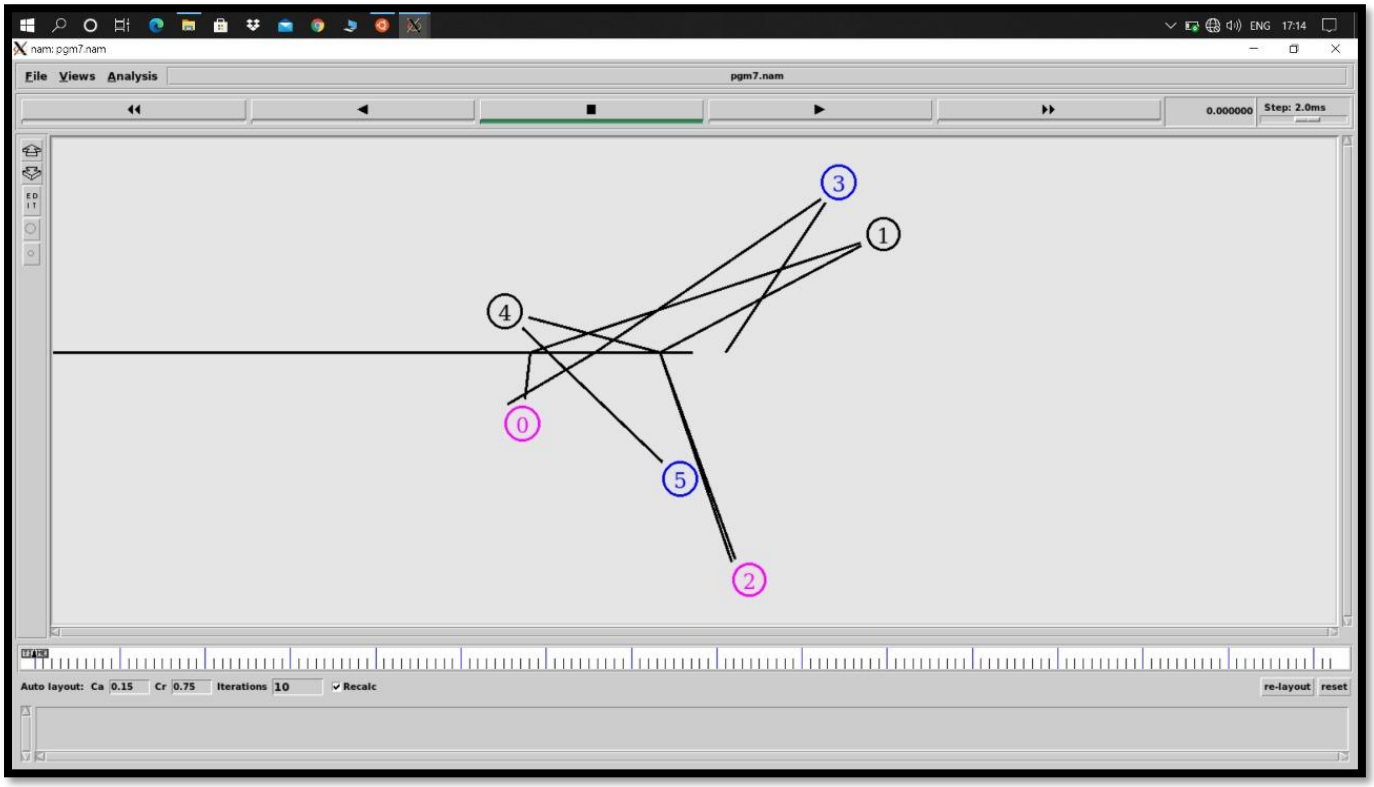
```
set sink5 [new Agent/TCPSink]
$ns attach-agent $n5 $sink5
$ns connect $tcp0 $sink5
set tcp2 [new Agent/TCP]
$ns attach-agent $n2 $tcp2
set ftp2 [new Application/FTP]
$ftp2 attach-agent $tcp2
$ftp2 set packetSize_ 600
$ftp2 set interval_ 0.001
set sink3 [new Agent/TCPSink]
$ns attach-agent $n3 $sink3
$ns connect $tcp2 $sink3
set file1 [open file1.tr w]
$tcp0 attach $file1
set file2 [open file2.tr w]
$tcp2 attach $file2
$tcp0 trace cwnd_
proc finish { } {
global ns nf tf
$ns flush-trace
close $tf
close $nf
exec nam pgm7.nam &
exit 0
}
$ns at 0.1 "$ftp0 start"
$ns at 5 "$ftp0 stop"
$ns at 7 "$ftp0 start"
$ns at 0.2 "$ftp2 start"
$ns at 8 "$ftp2 stop"
$ns at 14 "$ftp0 stop"
$ns at 10 "$ftp2 start"
$ns at 15 "$ftp2 stop"
$ns at 16 "finish"
$ns run
```

AWK

exp3.awk

```
BEGIN{
drop=0;
}
{
if($1=="d")
{
drop++;
}
}
END{
printf("Total number of %s packets dropped due to congestion =%d\n",$5,drop);
}
```

Simulation:



Experiment – 4

Simple ESS With Wireless LAN

TCL File: -

20181CSE0621

EXPERIMENT-04

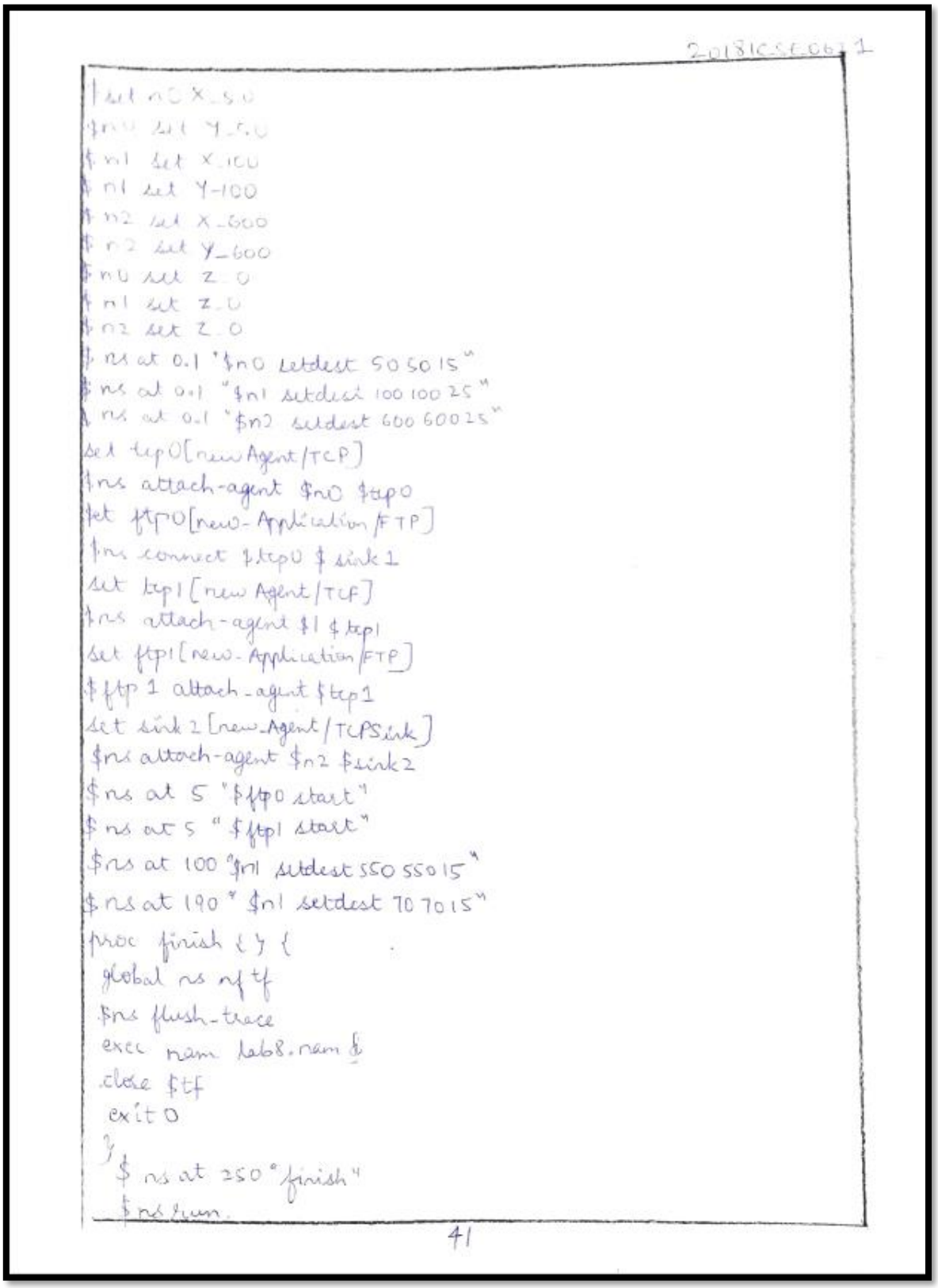
SIMPLE ESS WITH WIRELESS LAN

→ AIM : To simulate simple ESS with transmitting nodes in wireless LAN.

→ TCL file:

```
set ns [new Simulator]
set tf [open lab8.tr w]
$ns trace-all $tf
set topo [new topography]
$ topo load_flatgrid 1000 1000
set nf [open lab8.nam w]
$ns nametrace-all-wireless $nf 1000 1000
$ns node-config -adhocRouting DSDV \
-llType LL
-macType Mac/802_11 \
-ifqType queue/DropTail
-ifqlen 50 \
-phyType Phy/WirelessPhy \
-channelType /Wireless channel \
-antType antenna /Omni Antenna
-TopoInstance $topo \
-agentTrace ON \
-routerTrace ON
create-god 3
set no [$ns node]
set n1 [$ns node]
set n2 [$ns node]
$n0 label "tcp0"
$n1 label "sink-1/tcp1"
```

40



```
20181CSE0621
set n0 X_50
$n0 set Y_50
$n1 set X_100
$n1 set Y_100
$n2 set X_600
$n2 set Y_600
$n0 set Z_0
$n1 set Z_0
$n2 set Z_0
$ns at 0.1 "$n0 setdest 50 50 15"
$ns at 0.1 "$n1 setdest 100 100 25"
$ns at 0.1 "$n2 setdest 600 600 25"
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set ftp0 [new-Application/FTP]
$ns connect $tcp0 $sink1
set tcp1 [new Agent/TCP]
$ns attach-agent $1 $tcp1
set ftp1 [new-Application/FTP]
$ftp1 attach-agent $tcp1
set sink2 [new Agent/TCPSink]
$ns attach-agent $n2 $sink2
$ns at 5 "$ftp0 start"
$ns at 5 "$ftp1 start"
$ns at 100 "$n1 setdest 550 550 15"
$ns at 190 "$n1 setdest 70 70 15"
proc finish { } {
 global ns nf tf
 $ns flush-trace
 exec nam lab8.nam &
 close $tf
 exit 0
}
 $ns at 250 "finish"
 $ns run
```

AWK File: -

20181CSE0621

AWK FILE :

```
BEGIN {
  count1 = 0
  count2 = 0
  pack1 = 0
  pack2 = 0
  time1 = 0
  time2 = 0
      }
{ if ($1 = = "r" && $3 = = "1" && $4 == "AGT")
  { count1++
    pack1 = pack1 + $8
    time1 = $2
  }
  if ($1 = "r" && $3 == "2" && $4 = "AGT")
  { count2 ++
    pack2 = pack2 + $8
    time2 = $2
  }
}
END {
printf(" Throughput from n0 to n1 : %f mbps \n",
    ((count1 * count1 * 8) / (time1 * 1000000));
printf(" Throughput from n1 to n2 : %f mbps \n",
    ((count2 * pack2 * 8) / (time2 * 1000000));
```

42

OUTPUT

TCL

Open exp4.tcl ~/ Save

```
#20181CSE0621
set ns [new Simulator]
set tf [open lab8.tr w]
$ns trace-all $tf
set topo [new Topography]
$topo load_flatgrid 1000 1000
set nf [open lab8.nam w]
$ns namtrace-all-wireless $nf 1000 1000
$ns node-config -adhocRouting DSDV \
-llType LL \
-macType Mac/802_11 \
-ifqType Queue/DropTail \
-ifqLen 50 \
-phyType Phy/WirelessPhy \
-channelType Channel/WirelessChannel \
-prrootype Propagation/TwoRayGround \
-antType Antenna/OmniAntenna \
-topoInstance $topo \
-agentTrace ON \
-routerTrace ON
create-god 3
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
$n0 label "tcp0"
$n1 label "sink1/tcp1"
$n2 label "sink2"
$n0 set X_ 50
$n0 set Y_ 50
$n0 set Z_ 0
$n1 set X_ 100
$n1 set Y_ 100
$n1 set Z_ 0
$n2 set X_ 600
$n2 set Y_ 600
$n2 set Z_ 0
$ns at 0.1 "$n0 setdest 50 50 15"
$ns at 0.1 "$n1 setdest 100 100 25"
$ns at 0.1 "$n2 setdest 600 600 25"
set tcp0 [new Agent/TCP]
```

```
set sink1 [new Agent/TCPSink]
$ns attach-agent $n1 $sink1
$ns connect $tcp0 $sink1
set tcp1 [new Agent/TCP]
$ns attach-agent $n1 $tcp1
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
set sink2 [new Agent/TCPSink]
$ns attach-agent $n2 $sink2
$ns connect $tcp1 $sink2
$ns at 5 "$ftp0 start"
$ns at 5 "$ftp1 start"
$ns at 100 "$n1 setdest 550 550 15"
$ns at 190 "$n1 setdest 70 70 15"
proc finish { } {
global ns nf tf
$ns flush-trace
exec nam lab8.nam &
close $tf
exit 0
}
$ns at 250 "finish"
$ns run
```

Tcl Tab Width: 8 Ln 1, Col 14 INS

AWK

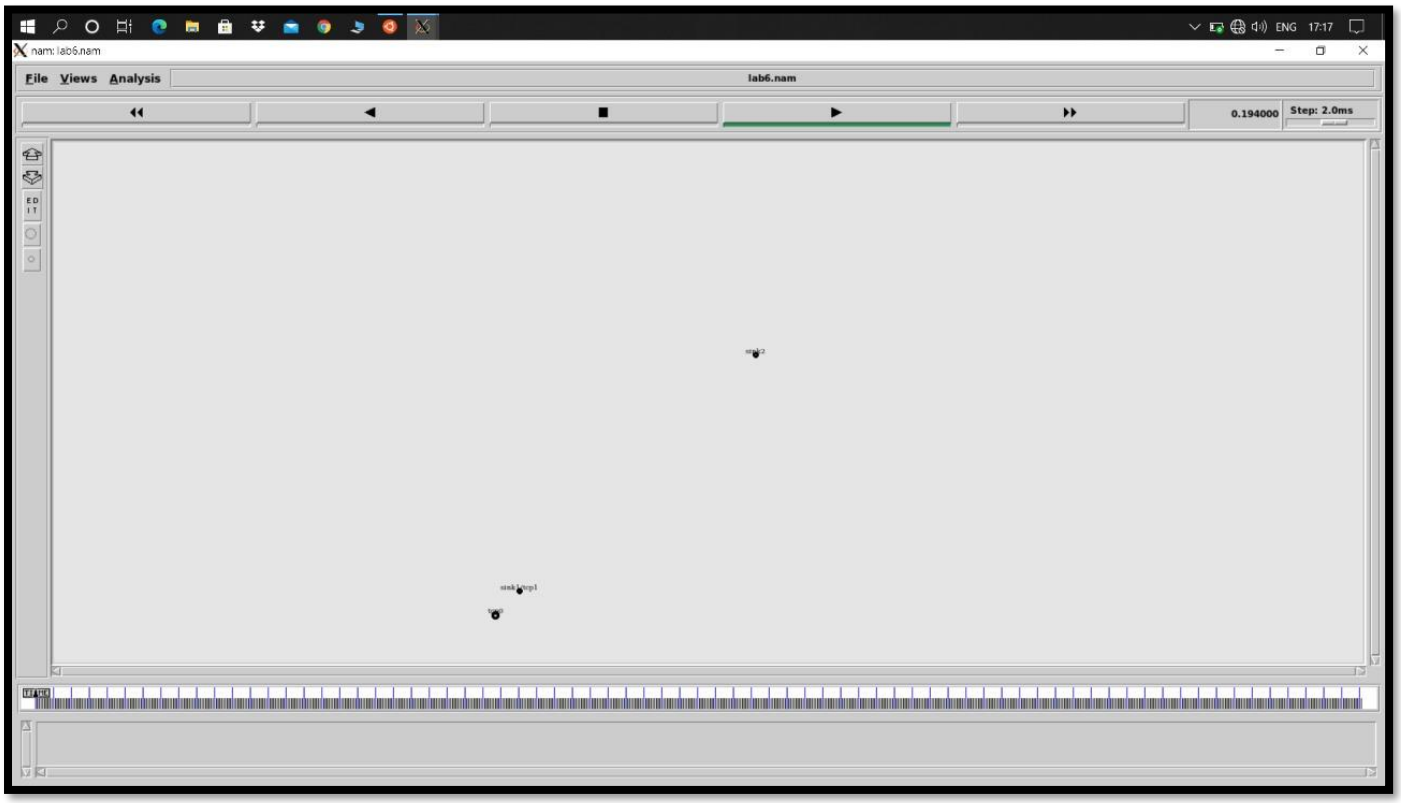
exp4.awk

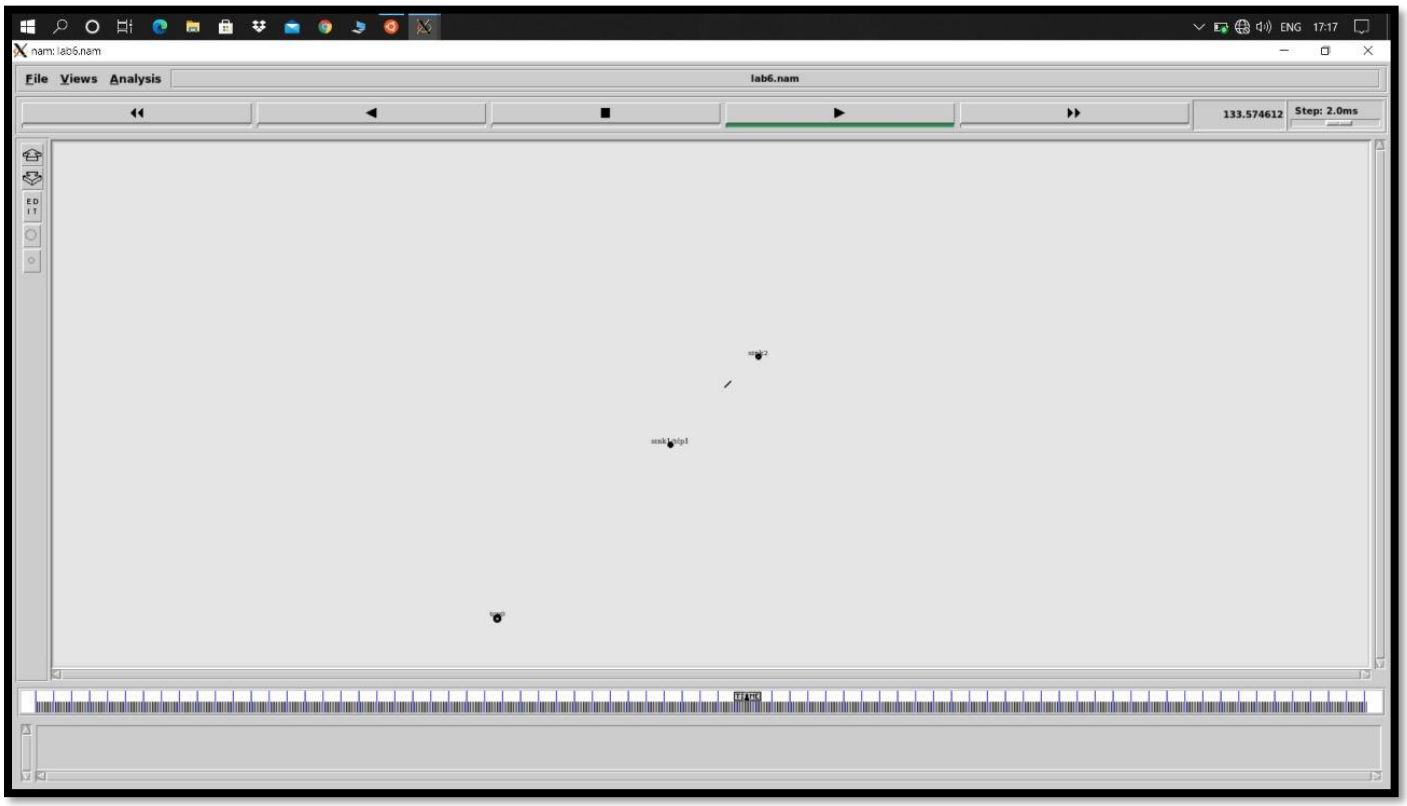
```
# 20181CSE0621
BEGIN{
count1=0
count2=0
pack1=0
pack2=0
time1=0
time2=0
}
{
if($1= ="r"&& $3= ="1" && $4= ="AGT")
{
count1++
pack1=pack1+$8
time1=$2
}
if($1= ="r" && $3= ="2" && $4= ="AGT")
{
count2++
pack2=pack2+$8
time2=$2
}
}
END{
printf("The Throughput from n0 to n1: %f Mbps \n", ((count1*pack1*8)/(time1*1000000)));
printf("The Throughput from n1 to n2: %f Mbps", ((count2*pack2*8)/(time2*1000000)));
}
```

Simulation

```
ram@LAPTOP-MVFDFOVD: ~
(gedit:84): dconf-WARNING **: 17:12:01.575: failed to commit changes to dconf: Error spawning command line “dbus-launch
--autolaunch= --binary-syntax --close-stderr”: Child process exited with code 1

(gedit:84): dconf-WARNING **: 17:12:01.583: failed to commit changes to dconf: Error spawning command line “dbus-launch
--autolaunch= --binary-syntax --close-stderr”: Child process exited with code 1

(gedit:84): dconf-WARNING **: 17:12:01.585: failed to commit changes to dconf: Error spawning command line “dbus-launch
--autolaunch= --binary-syntax --close-stderr”: Child process exited with code 1

** (gedit:84): WARNING **: 17:12:27.294: Set document metadata failed: Setting attribute metadata::gedit-position not su
pported

(gedit:84): dconf-WARNING **: 17:12:27.306: failed to commit changes to dconf: Error spawning command line “dbus-launch
--autolaunch= --binary-syntax --close-stderr”: Child process exited with code 1
ram@LAPTOP-MVFDFOVD:~$ ns exp7.tcl
```

Simulation:


nam: lab6.nam
File Views Analysis
lab6.nam
0.194000
Step: 2.0ms


nam: lab6.nam
File Views Analysis
lab6.nam
133.574612
Step: 2.0ms