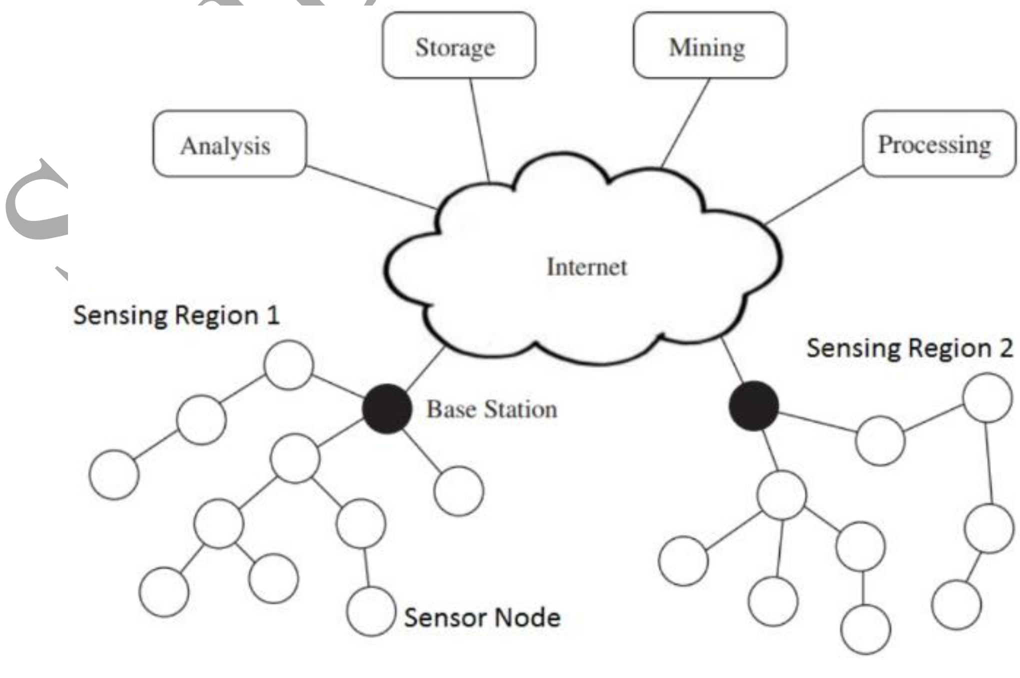
Aim: Understanding the Sensor Node Hardware.

Features of Sensor Node Hardware

Sensor nodes are fundamental building blocks in Wireless Sensor Networks (WSNs). These nodes contain various components that enable sensing, processing, and communication. Below are some key features of sensor node hardware:

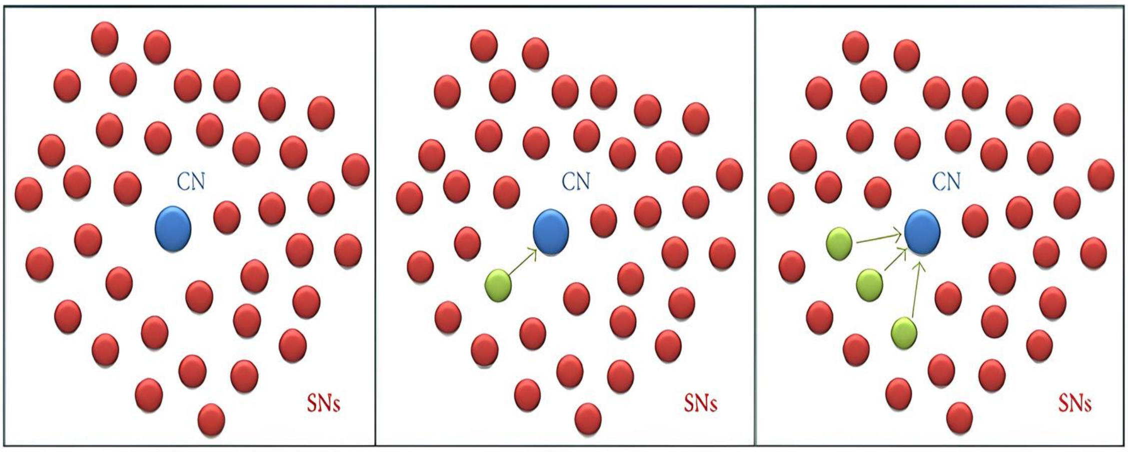
- **Low Power Consumption:** Sensor nodes are designed to operate on minimal power, often relying on batteries or energy-harvesting techniques to prolong their lifetime.
- **Wireless Communication:** Most sensor nodes communicate using wireless protocols such as Zigbee, Bluetooth, LoRa, and Wi-Fi to transmit data over networks.
- **Multi-Sensing Capabilities:** These nodes can integrate multiple sensors to measure environmental parameters such as temperature, humidity, pressure, motion, and light intensity.
- **Small Size & Portability:** Due to their compact design, sensor nodes can be deployed in various environments, including remote and hard-to-reach locations.
- **Scalability:** WSNs can support hundreds or thousands of sensor nodes, allowing large-scale deployment for diverse applications.
- **Self-Configuration & Adaptability:** Sensor nodes are capable of self-organizing into networks and dynamically adjusting their parameters based on environmental conditions.



Processing in Sensor Nodes

Processing in sensor nodes involves several critical components that handle data acquisition, computation, and transmission. These components include:

- **Microcontroller Unit (MCU):** The core processing unit that manages sensor data collection, processing, and transmission. Examples include ARM Cortex-M series, ATmega328 (used in Arduino), and ESP32.
- **Analog-to-Digital Converter (ADC):** Converts raw analog signals from sensors into digital data that can be processed and analyzed.
- **Memory Storage:** Stores collected data temporarily or permanently before transmission. Memory types include RAM, Flash, and EEPROM.
- **Real-Time Processing:** Enables sensor nodes to process data immediately and take necessary actions based on pre-programmed logic.
- **Communication Interfaces:** Sensor nodes use UART, SPI, I2C, and other protocols to interface with peripherals and transmit data wirelessly.

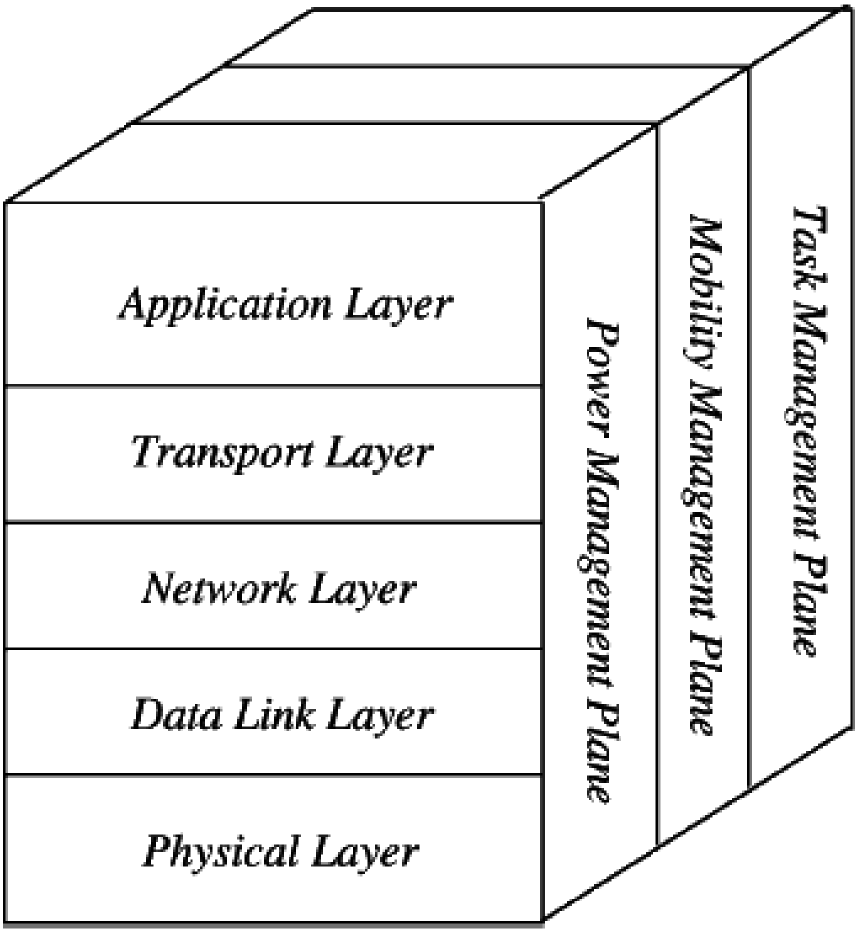


No packet transmission idle listening | Single packet transmission successful packet | Multiple packet transmission collision

Importance of Sensor Node Hardware

Sensor nodes play a crucial role in various real-world applications. Some of their key contributions include:

- **Environmental Monitoring:** Sensor nodes help track air quality, weather patterns, and pollution levels to ensure environmental sustainability.
- **Healthcare Applications:** Wearable sensor nodes enable remote patient monitoring, detecting vital signs such as heart rate, blood pressure, and body temperature.
- **Industrial Automation:** In smart factories, sensor nodes optimize manufacturing processes by monitoring equipment health and reducing downtime.
- **Security & Surveillance:** Wireless sensor networks support surveillance systems by detecting unauthorized activities and triggering alarms.
- **Smart Cities:** Sensor nodes contribute to intelligent traffic management, waste management, and efficient energy usage in smart cities.



Wireless Sensor Network Architecture

What is the Coding Behind Sensor Nodes?

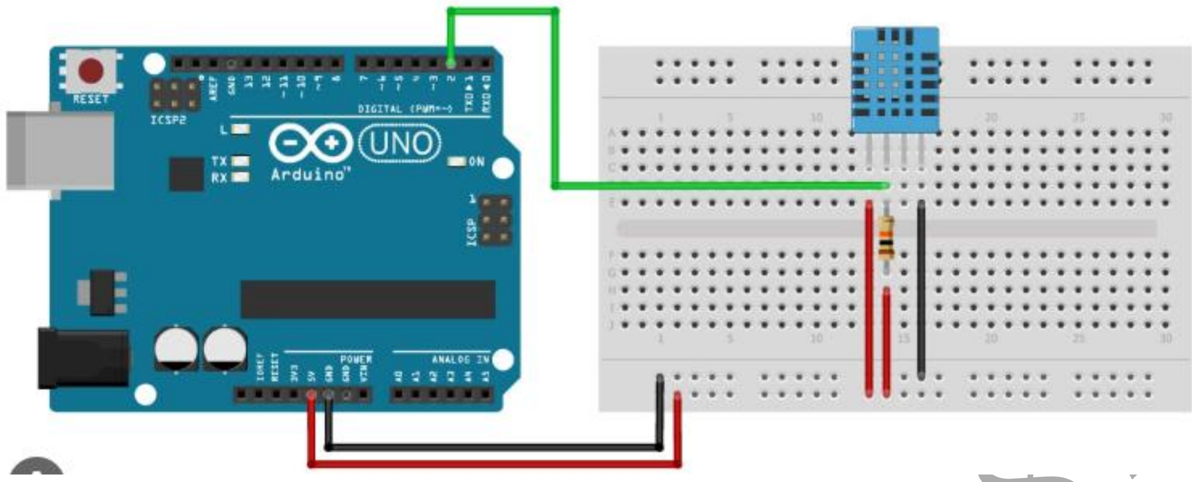
Programming sensor nodes requires embedded system development using languages such as **C, C++, Python, and Embedded C**. Some popular platforms and operating systems used for coding sensor nodes include:

- **Arduino IDE:** Used to program microcontroller-based sensor nodes such as Arduino Uno and ESP8266.
- **Contiki OS:** An open-source operating system for IoT applications supporting low-power sensor nodes.
- **TinyOS:** A lightweight OS specifically designed for low-power wireless sensor networks.
- **Node-RED:** A visual programming tool for designing IoT applications and managing sensor data.

==

Example Code (Arduino for Temperature Sensor - DHT11)

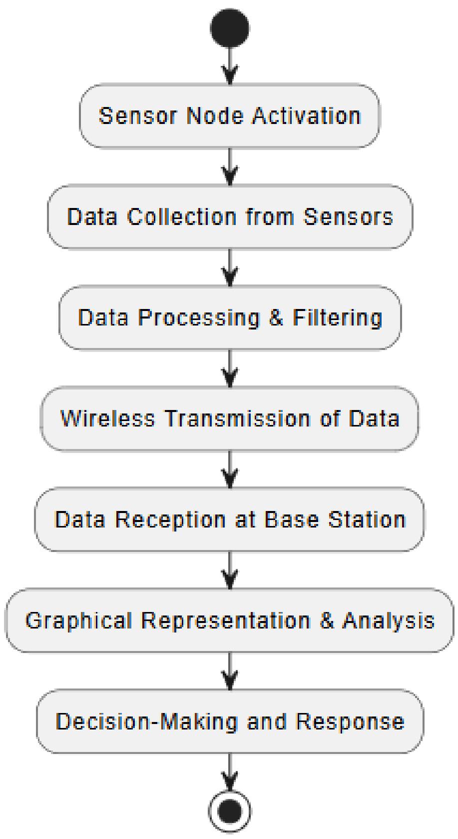
```
#include <DHT.h>
#define DHTPIN 2     // Pin where the sensor is connected
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);


void setup() {
  Serial.begin(9600);
  dht.begin();
}
void loop() {
  float temp = dht.readTemperature();
  Serial.print("Temperature: ");
  Serial.print(temp);
  Serial.println(" °C");
  delay(2000);
}
```



==

Flow Chart of Sensor Node Operation

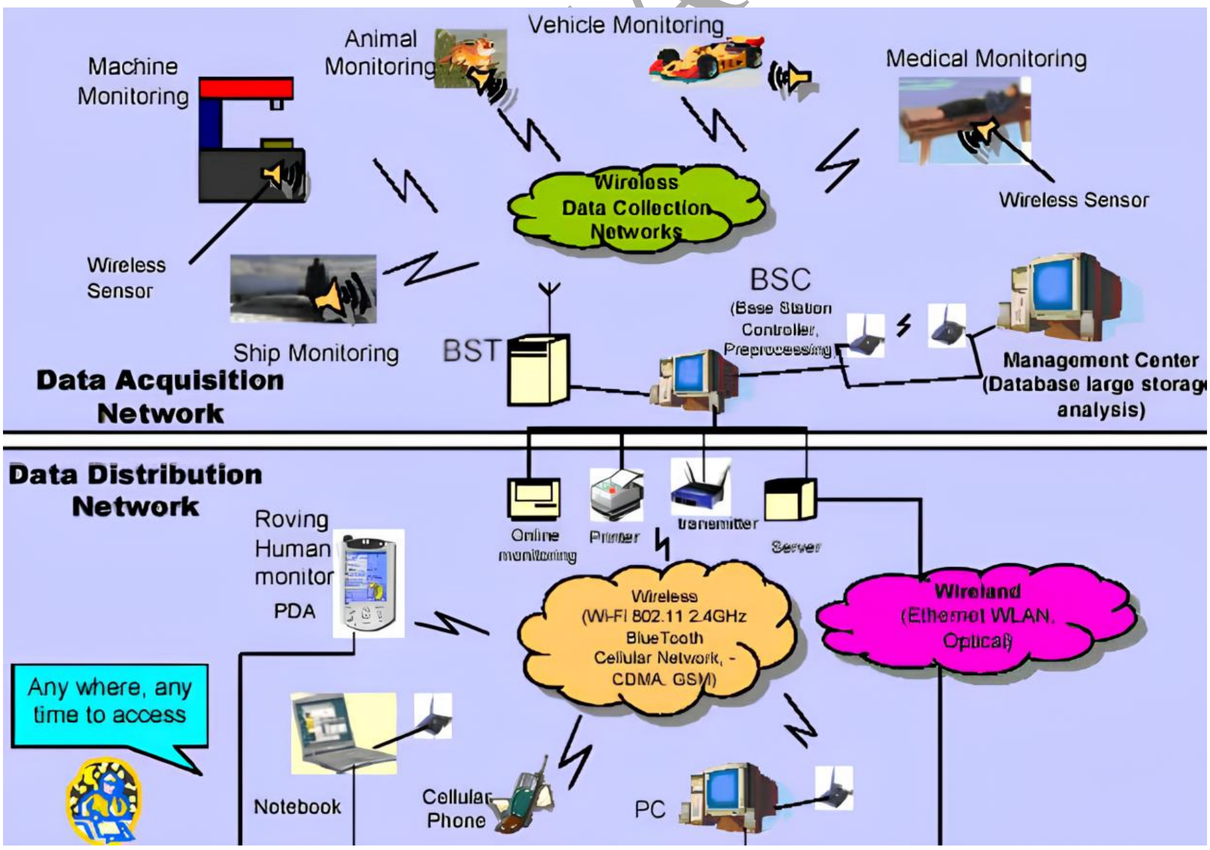
Below is the step-by-step operational flow of a sensor node in a WSN:



Basic Requirements for Sensor Nodes

For an efficient sensor node design, the following hardware and software components are necessary:

- **Microcontroller or Processor:** ARM Cortex-M, ATmega328, ESP32, etc.
- **Communication Module:** Zigbee, LoRa, Wi-Fi, or Bluetooth for data transmission.
- **Power Supply:** Battery-powered or energy-harvesting sources such as solar panels.
- **Sensors:** Various sensors depending on the application (Temperature, Humidity, Motion, Light, etc.).
- **Memory Unit:** EEPROM, Flash Storage, or SD cards for data logging and processing.
- **Graphical Interface:** Software tools such as MATLAB, Grafana, or Node-RED for data visualization and interpretation.



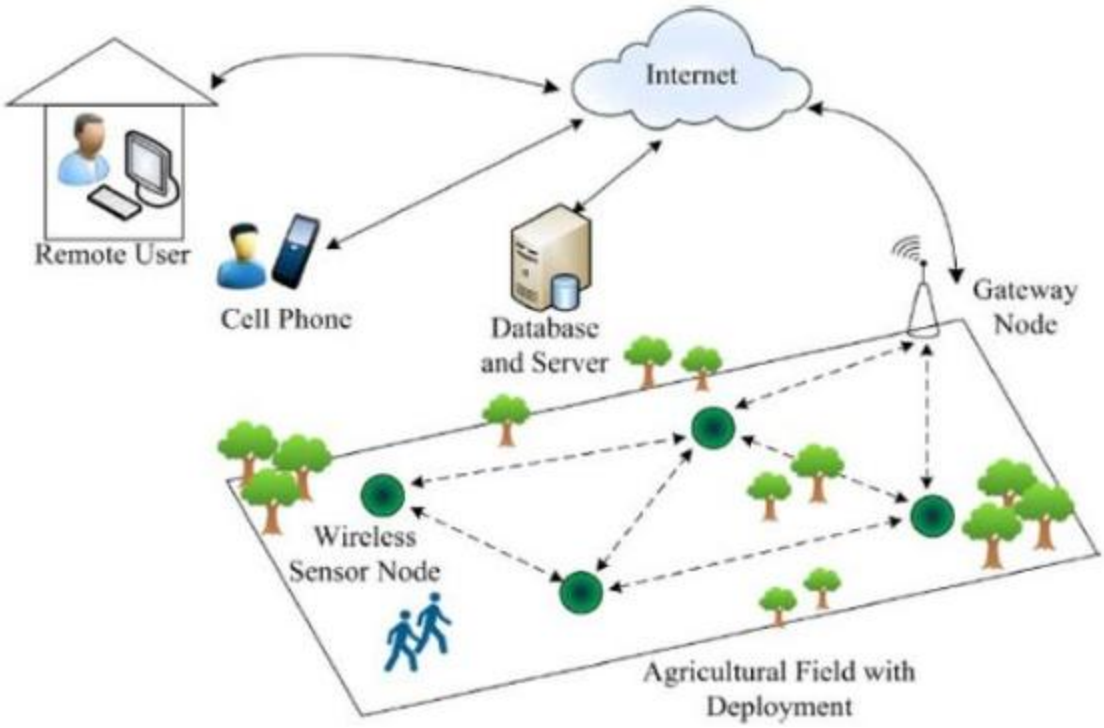
Maximum Memory Requirement for Sensor Nodes

The memory requirements for sensor nodes vary based on the complexity of applications. The typical requirements are:

- **Microcontroller Memory:** Generally, sensor nodes operate with **32 KB to 512 KB** of Flash storage.
- **RAM Requirement:** Depending on the complexity of data processing, RAM can range from **2 KB to 256 KB**.
- **External Storage (Optional):** SD cards up to **32 GB** can be used for extensive data logging in advanced applications.

Conclusion

Sensor nodes are the fundamental units of Wireless Sensor Networks, integrating low-power microcontrollers, sensors, and wireless communication modules. They play a vital role in real-time monitoring, data analysis, and automation in various fields, including healthcare, industrial automation, and environmental tracking. Efficient hardware selection, optimized programming, and intelligent power management strategies are essential for maximizing the effectiveness and longevity of sensor networks.

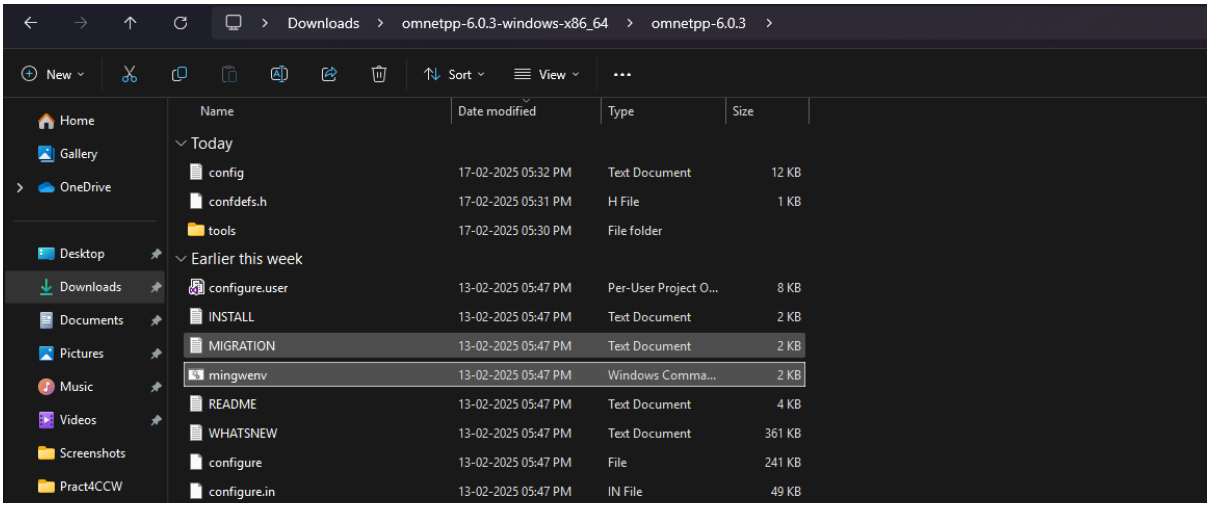


A typical WSN deployment for agriculture application

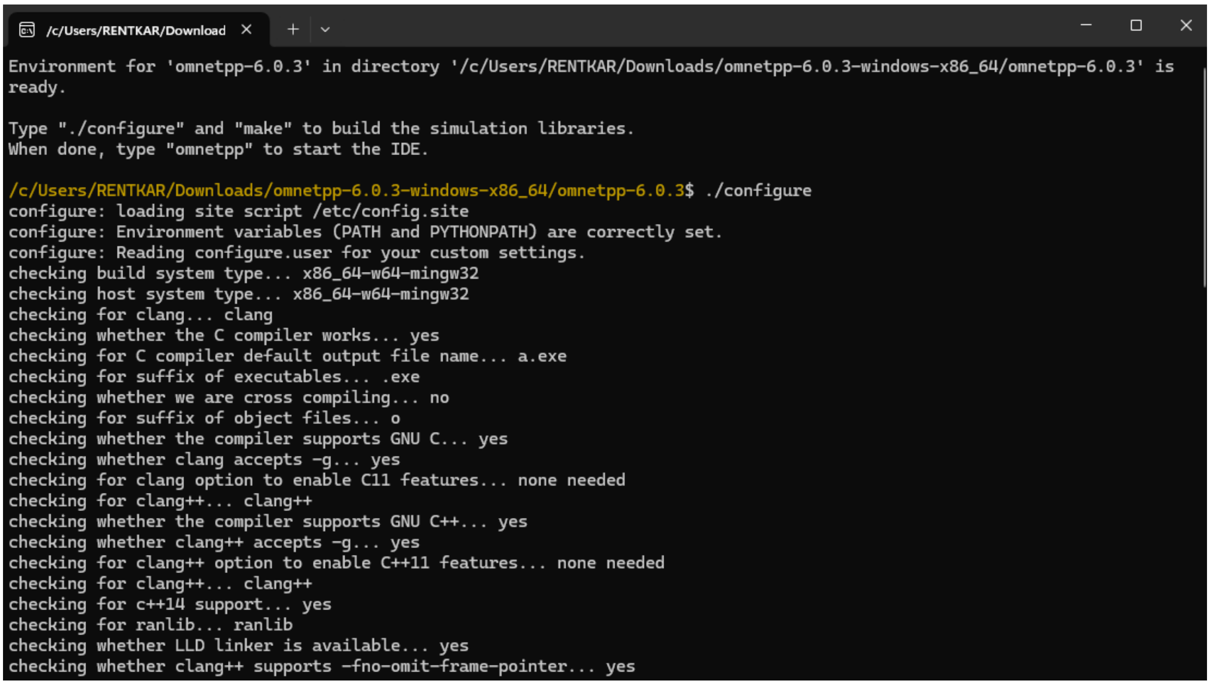
Practical 4

Aim: Create and simulate a simple adhoc network

Step1] In the extracted folder of omnet++ open the “mingwnv” then it will be open in command prompt

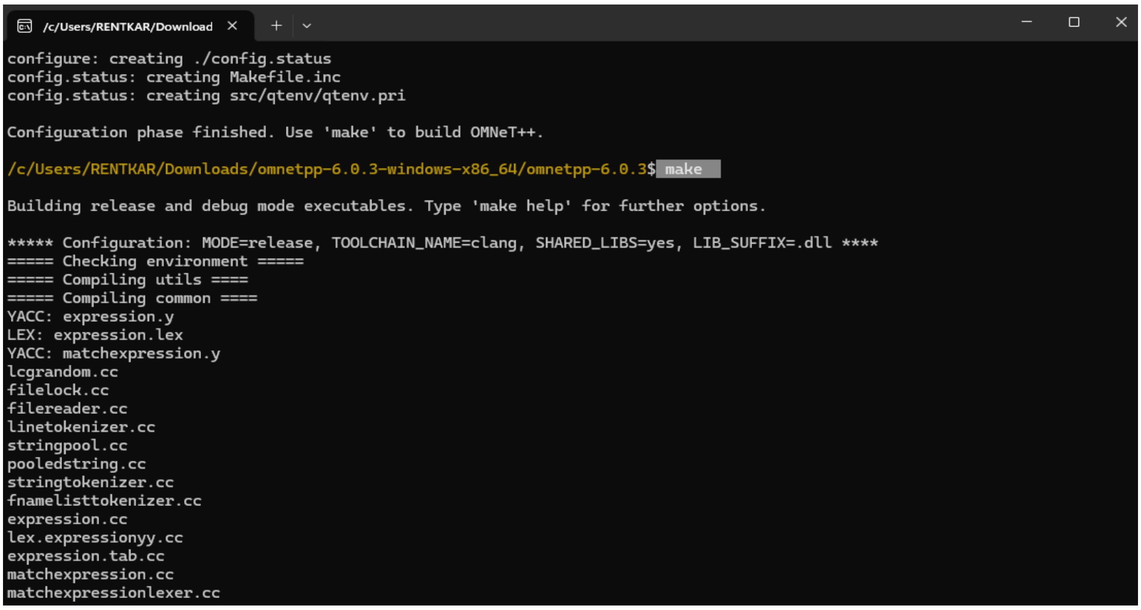


Step 2] Once it gets open then type “./configure”

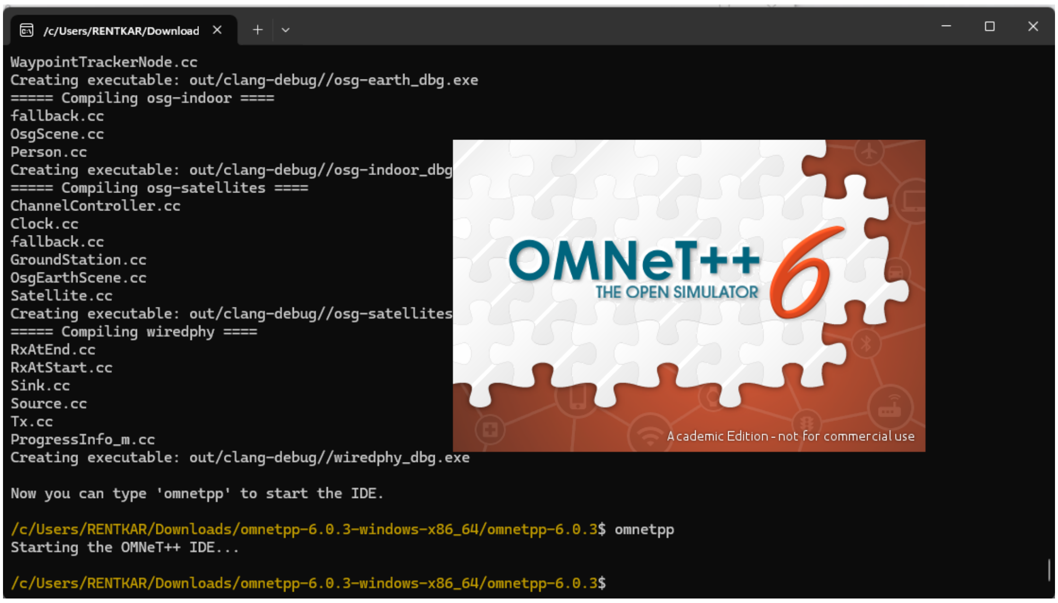


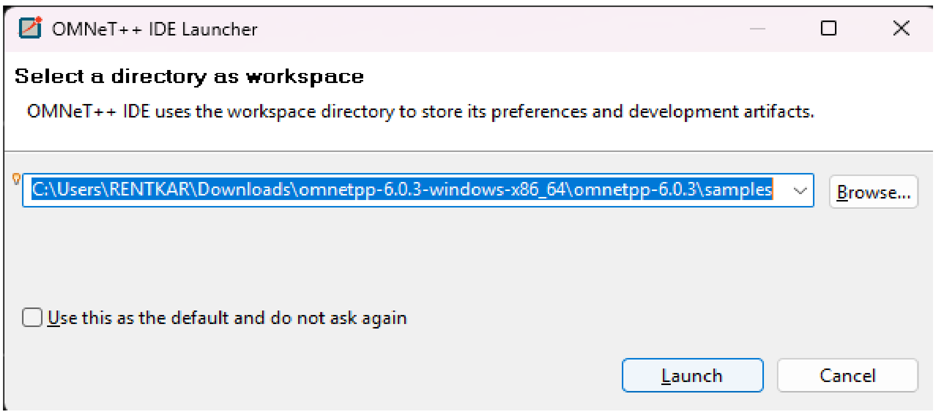
```
/c/Users/RENTKAR/Download
Environment for 'omnetpp-6.0.3' in directory '/c/Users/RENTKAR/Downloads/omnetpp-6.0.3-windows-x86_64/omnetpp-6.0.3' is
ready.

Type "./configure" and "make" to build the simulation libraries.
When done, type "omnetpp" to start the IDE.

/c/Users/RENTKAR/Downloads/omnetpp-6.0.3-windows-x86_64/omnetpp-6.0.3$ ./configure
configure: loading site script /etc/config.site
configure: Environment variables (PATH and PYTHONPATH) are correctly set.
configure: Reading configure.user for your custom settings.
checking build system type... x86_64-w64-mingw32
checking host system type... x86_64-w64-mingw32
checking for clang... clang
checking whether the C compiler works... yes
checking for C compiler default output file name... a.exe
checking for suffix of executables... .exe
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether clang accepts -g... yes
checking for clang option to enable C11 features... none needed
checking for clang++... clang++
checking whether the compiler supports GNU C++... yes
checking whether clang++ accepts -g... yes
checking for clang++ option to enable C++11 features... none needed
checking for clang++... clang++
checking for c++14 support... yes
checking for ranlib... ranlib
checking whether LLD linker is available... yes
checking whether clang++ supports -fno-omit-frame-pointer... yes
```

Step3] Then type “make” to build the simulation libraries

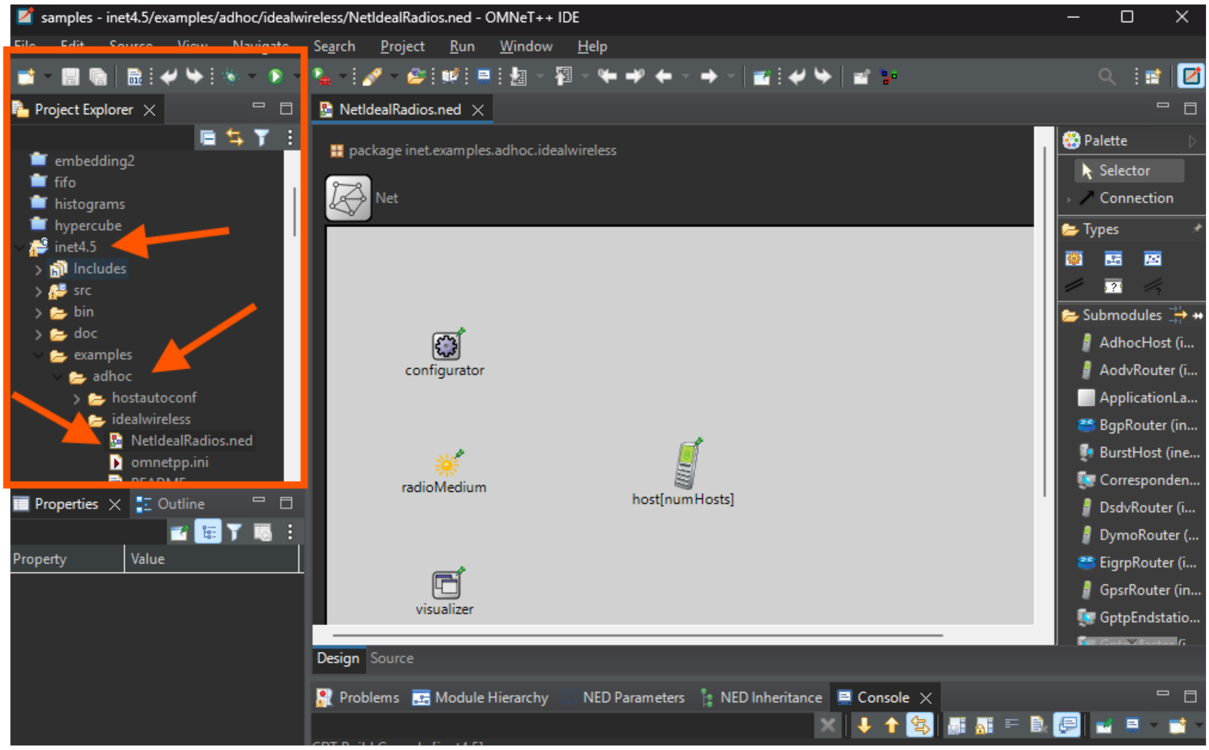


Step4] Now simulation libraries are build and omner++ is build now we have to type “omnetpp” this will launch the omnet++ ide

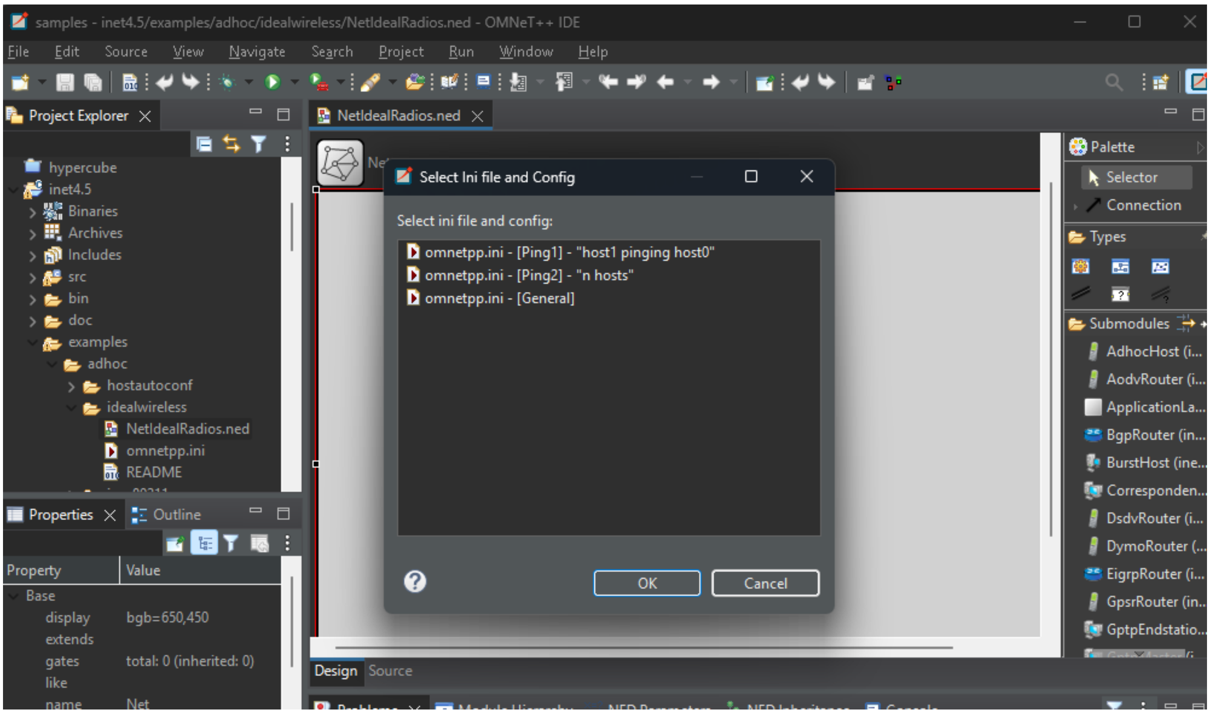




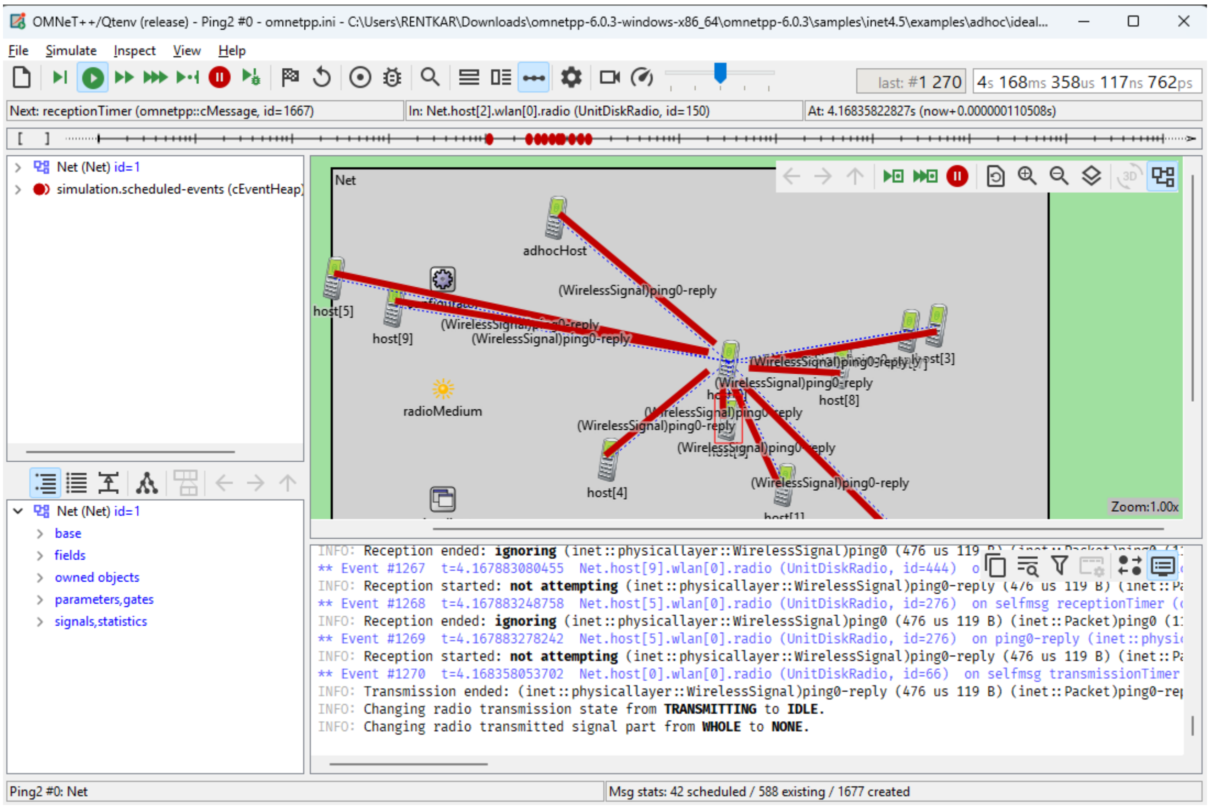
Step5] In the Ide Go to Project explore section then search inet framework then click on that and then click on examples> adhoc>NetidealRadios.ned



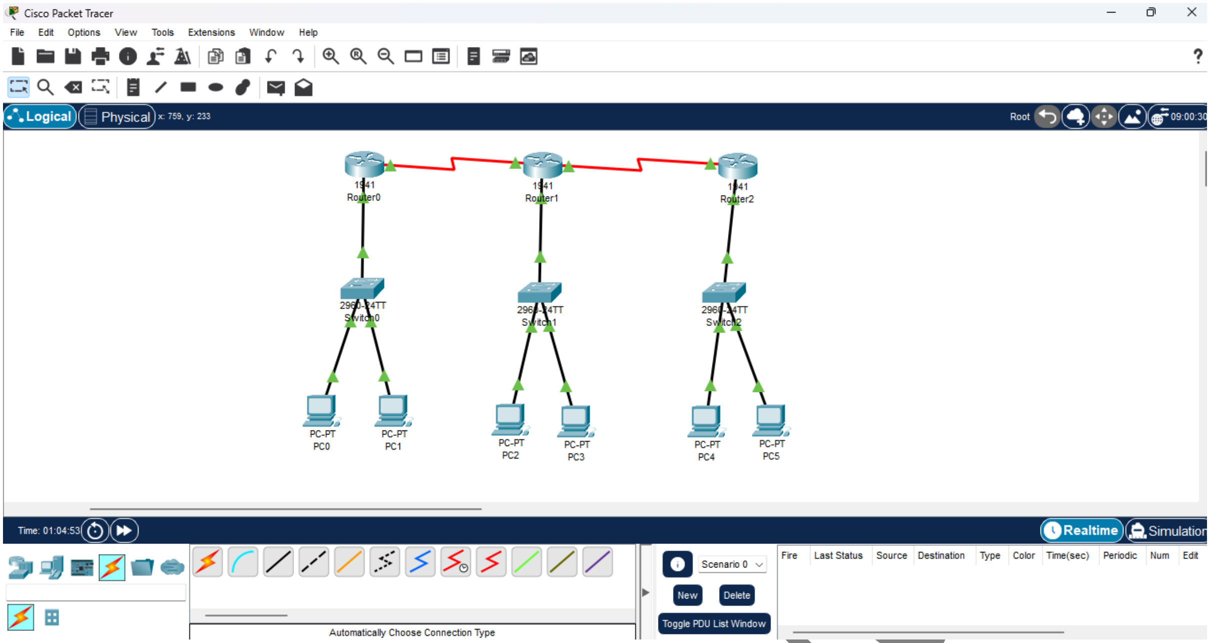
Step6] now everything is ready since this the prebuild simulation just need to run this so click on run and than this will build

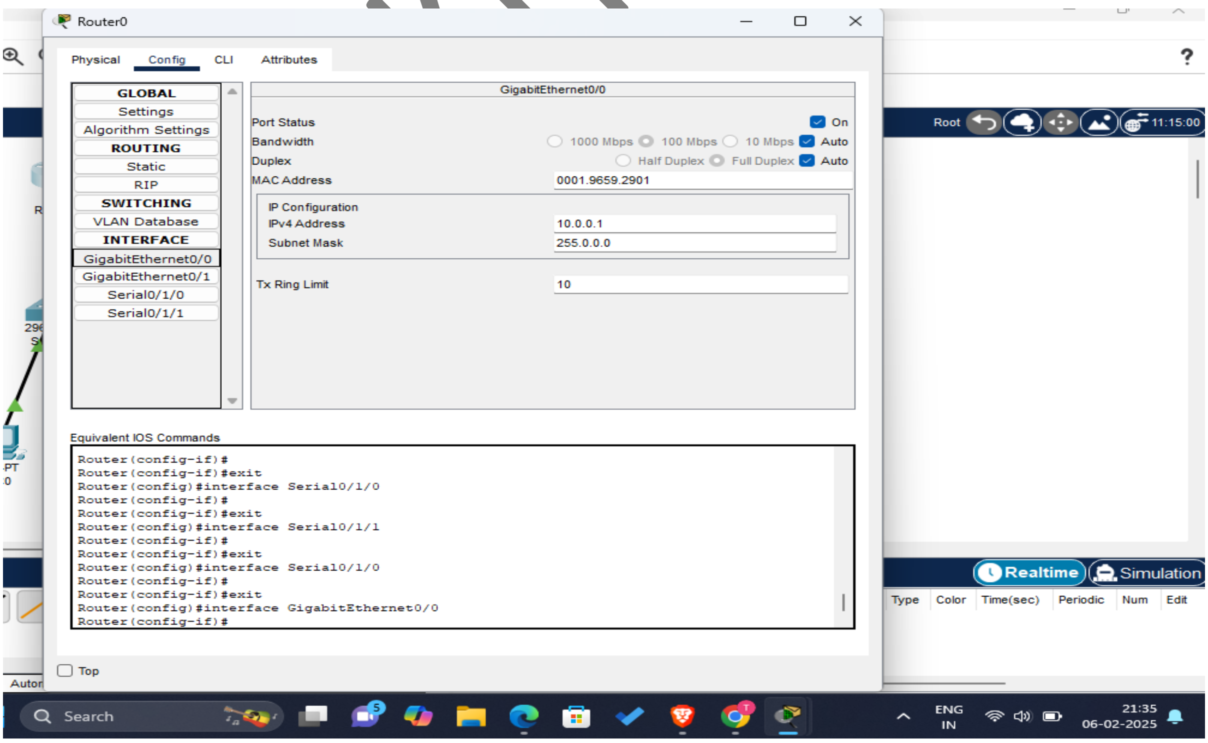


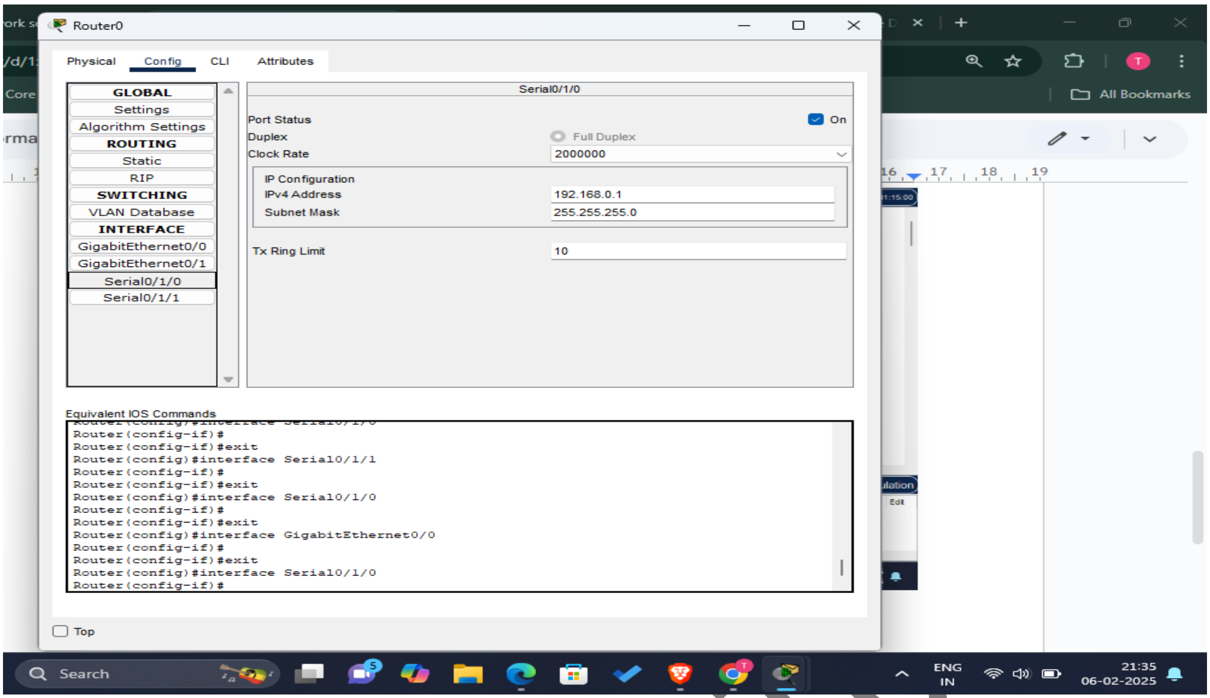
Step7] Now run the build and in the window will prompt up where it ask for number of host we have did with 10 and adhoc network is established

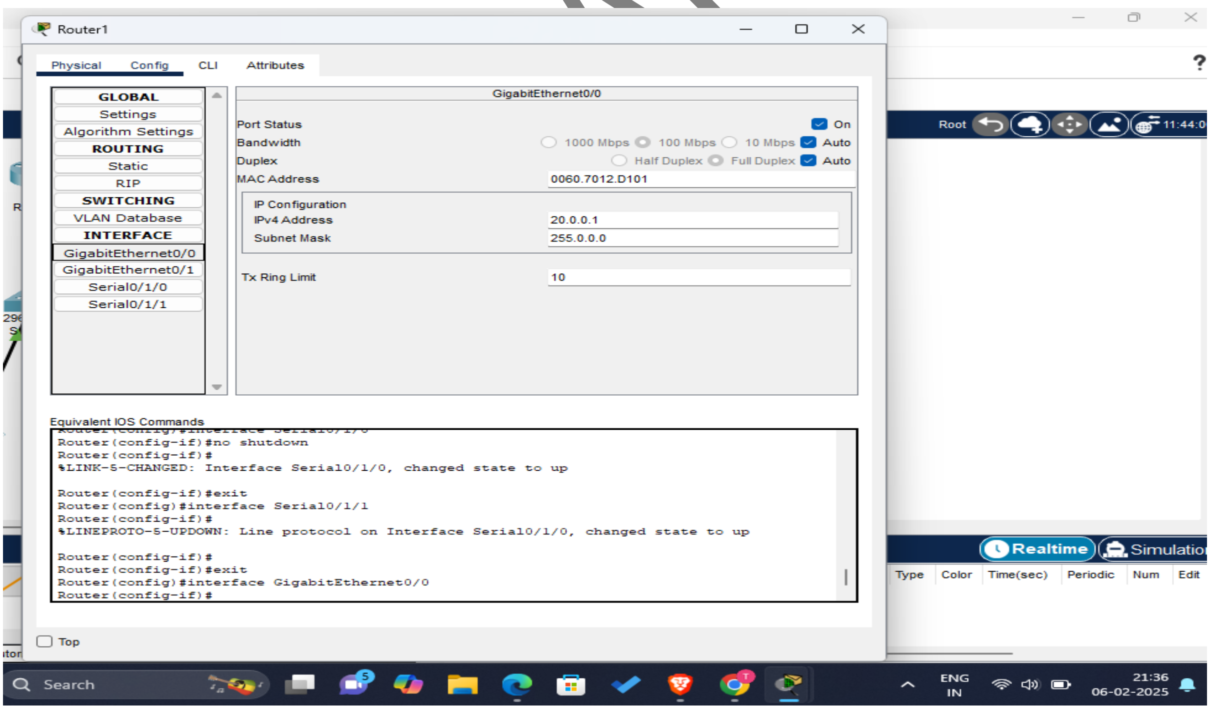


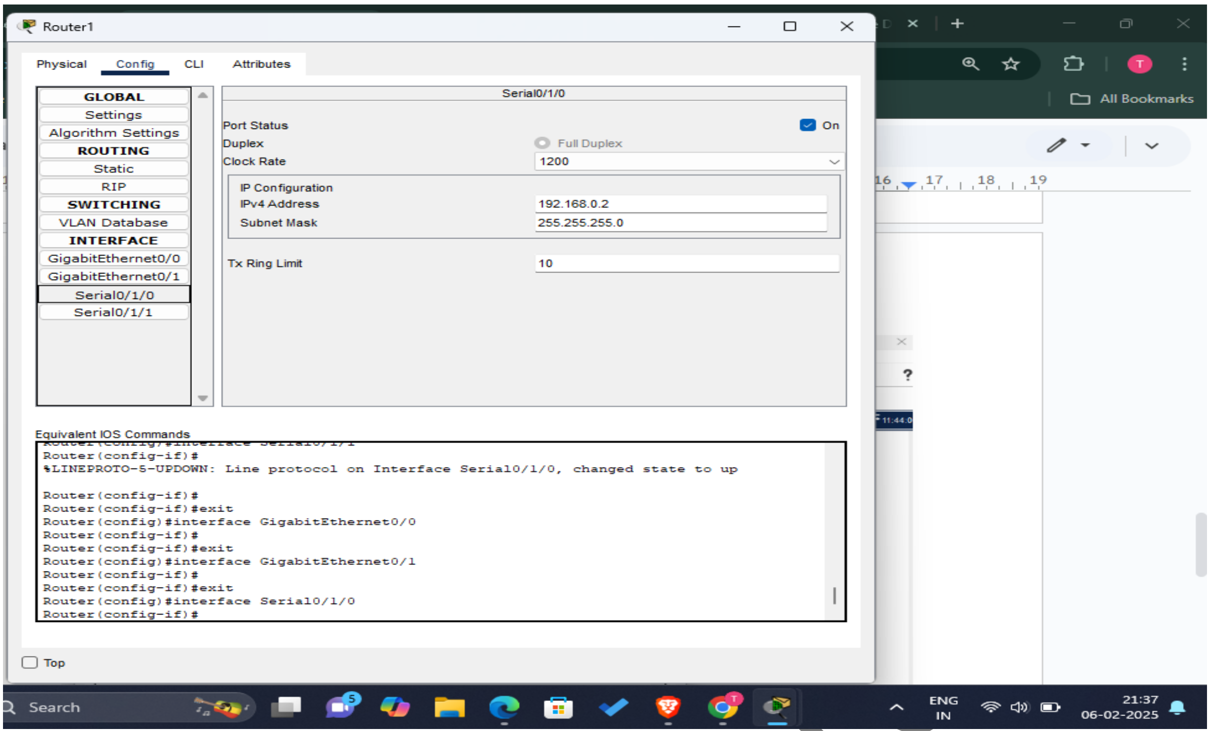
Aim: Understanding, Reading and Analyzing Routing Table of a network.

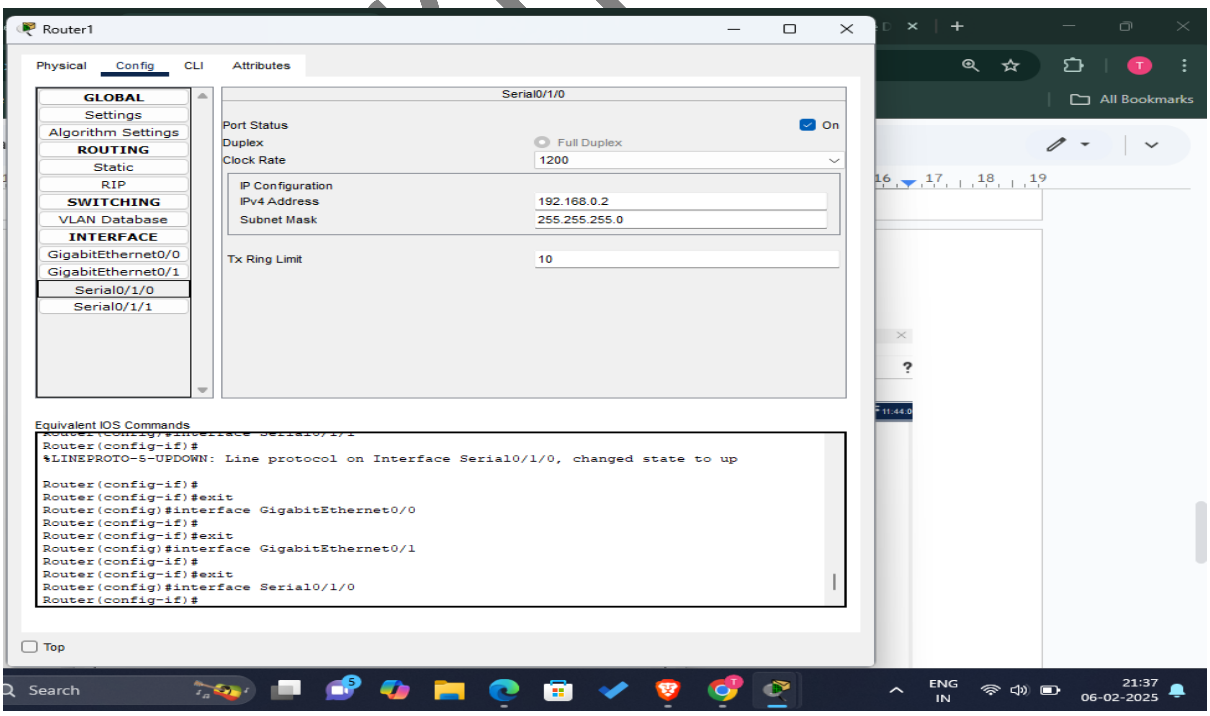


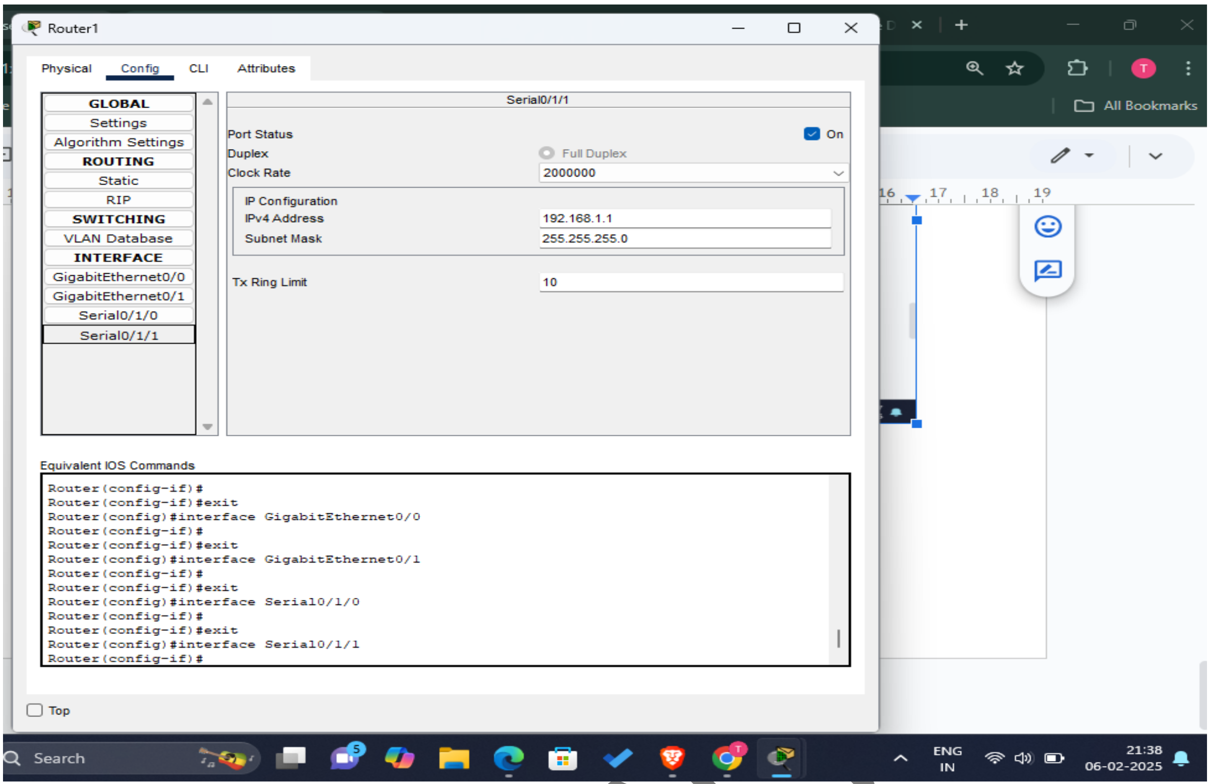


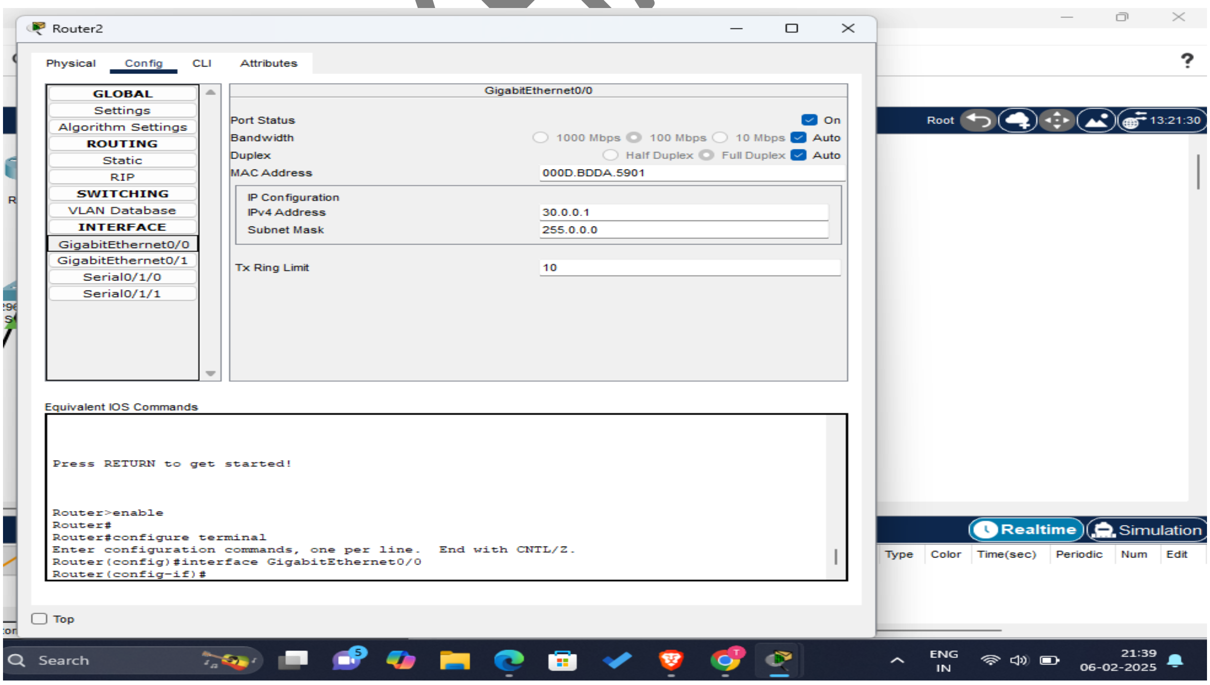

Router0
Physical Config CLI Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
GigabitEthernet0/0
GigabitEthernet0/1
Serial0/1/0
Serial0/1/1
Serial0/1/0
Port Status On
Duplex Full Duplex
Clock Rate 2000000
IP Configuration
IPv4 Address 192.168.0.1
Subnet Mask 255.255.255.0
Tx Ring Limit 10
Equivalent IOS Commands
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial0/1/1
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial0/1/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface GigabitEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial0/1/0
Router(config-if)#
Top
21:35
06-02-2025

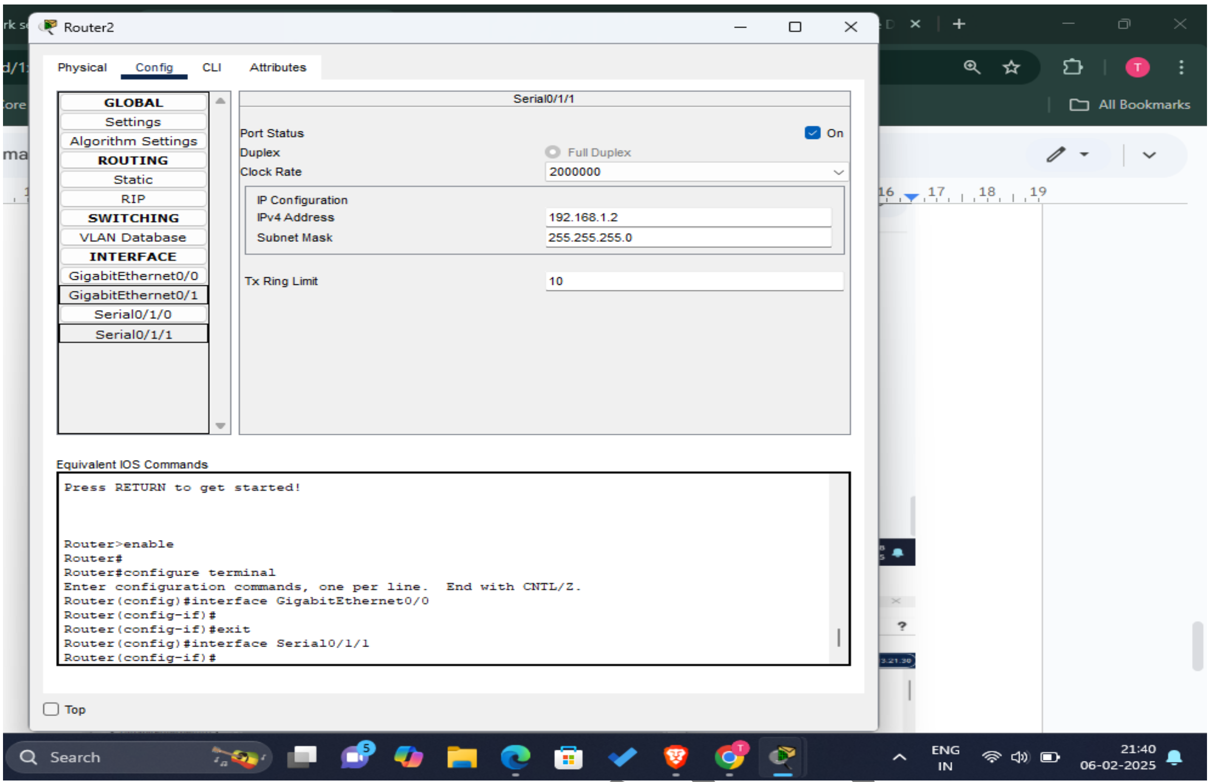

Router1
Physical Config CLI Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
GigabitEthernet0/0
GigabitEthernet0/1
Serial0/1/0
Serial0/1/1
GigabitEthernet0/0
Port Status On
Bandwidth 1000 Mbps 100 Mbps 10 Mbps Auto
Duplex Half Duplex Full Duplex Auto
MAC Address 0060.7012.D101
IP Configuration
IPv4 Address 20.0.0.1
Subnet Mask 255.0.0.0
Tx Ring Limit 10
Equivalent IOS Commands
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface Serial0/1/0, changed state to up
Router(config-if)#exit
Router(config)#interface Serial0/1/1
Router(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1/0, changed state to up
Router(config-if)#
Router(config-if)#exit
Router(config)#interface GigabitEthernet0/0
Router(config-if)#
Top
Realtime Simulation
21:36
06-02-2025

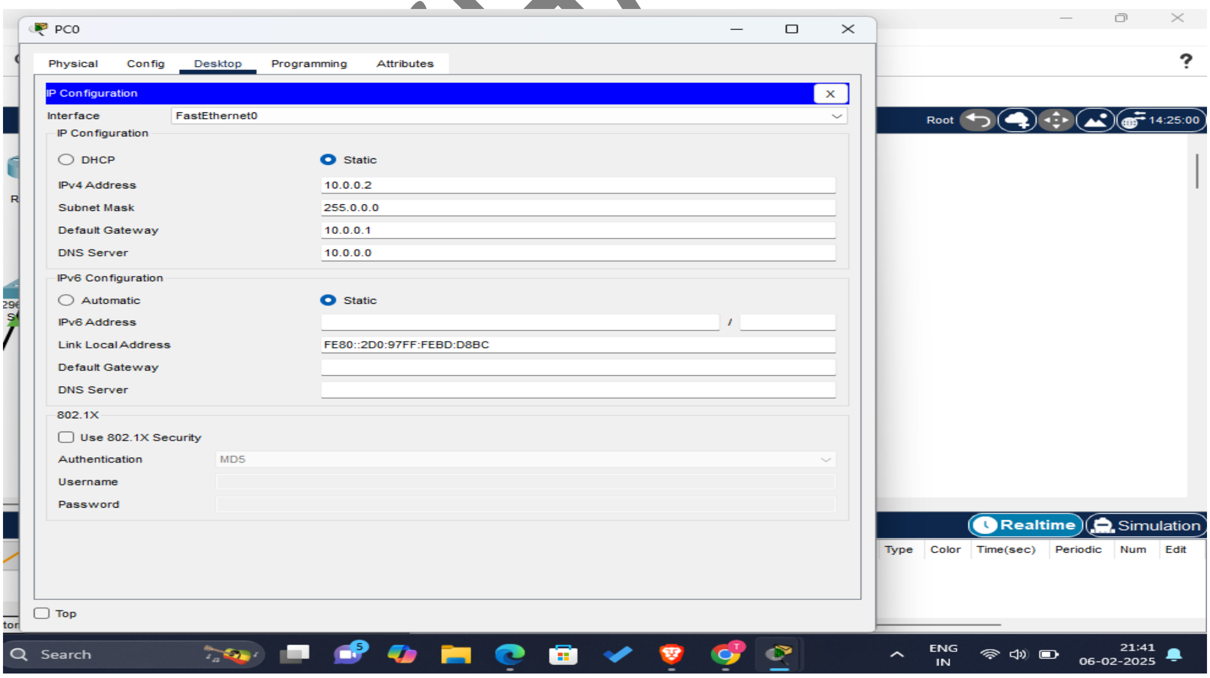

Router1
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
GigabitEthernet0/0
GigabitEthernet0/1
Serial0/1/0
Serial0/1/1
Serial0/1/0
Port Status
On
Duplex
Full Duplex
Clock Rate
1200
IP Configuration
IPv4 Address
192.168.0.2
Subnet Mask
255.255.255.0
Tx Ring Limit
10
Equivalent IOS Commands
Router(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1/0, changed state to up
Router(config-if)#
Router(config-if)#exit
Router(config)#interface GigabitEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface GigabitEthernet0/1
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial0/1/0
Router(config-if)#
Top
All Bookmarks
Search
ENG
IN
21:37
06-02-2025

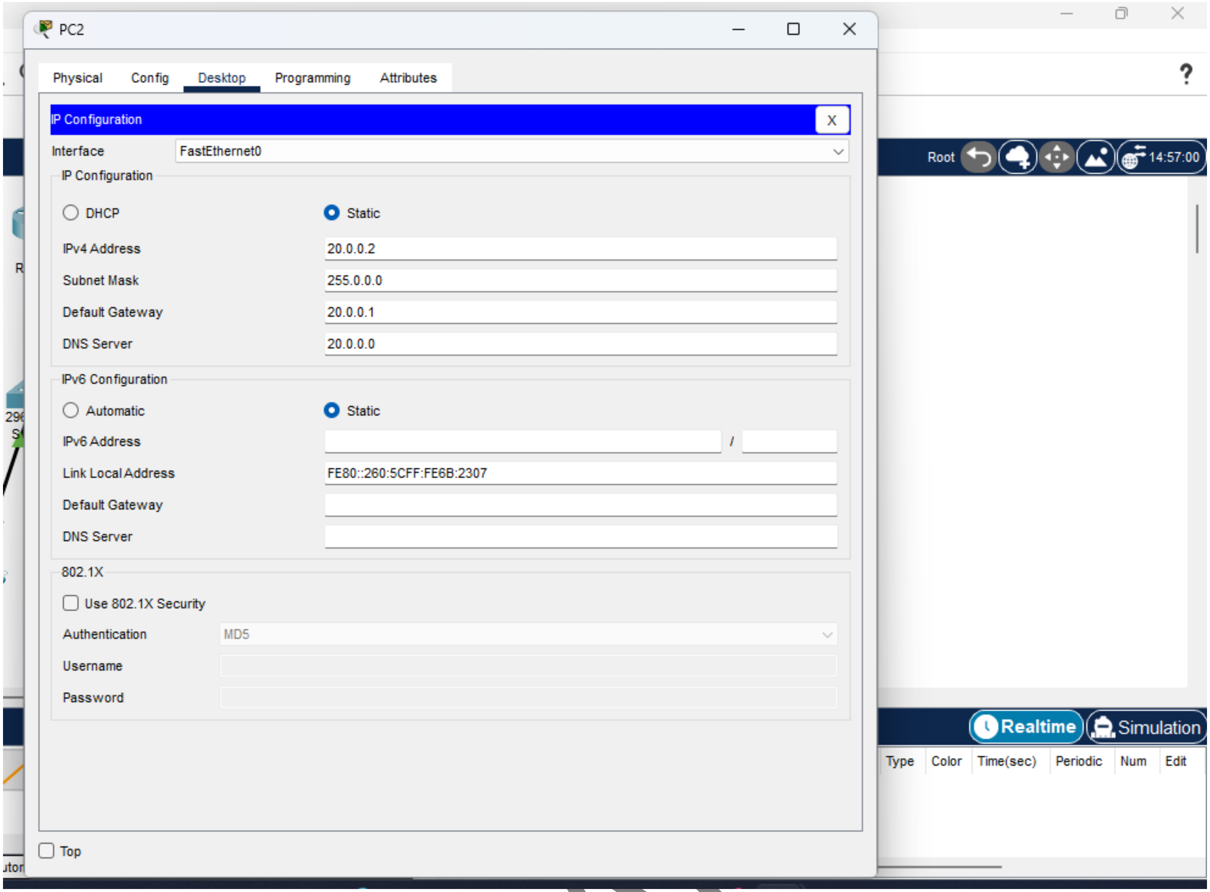

Router1
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
GigabitEthernet0/0
GigabitEthernet0/1
Serial0/1/0
Serial0/1/1
Serial0/1/0
Port Status
On
Duplex
Full Duplex
Clock Rate
1200
IP Configuration
IPv4 Address
192.168.0.2
Subnet Mask
255.255.255.0
Tx Ring Limit
10
Equivalent IOS Commands
Router(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1/0, changed state to up
Router(config-if)#
Router(config-if)#exit
Router(config)#interface GigabitEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface GigabitEthernet0/1
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial0/1/0
Router(config-if)#
Top
All Bookmarks
Search
ENG
IN
21:37
06-02-2025

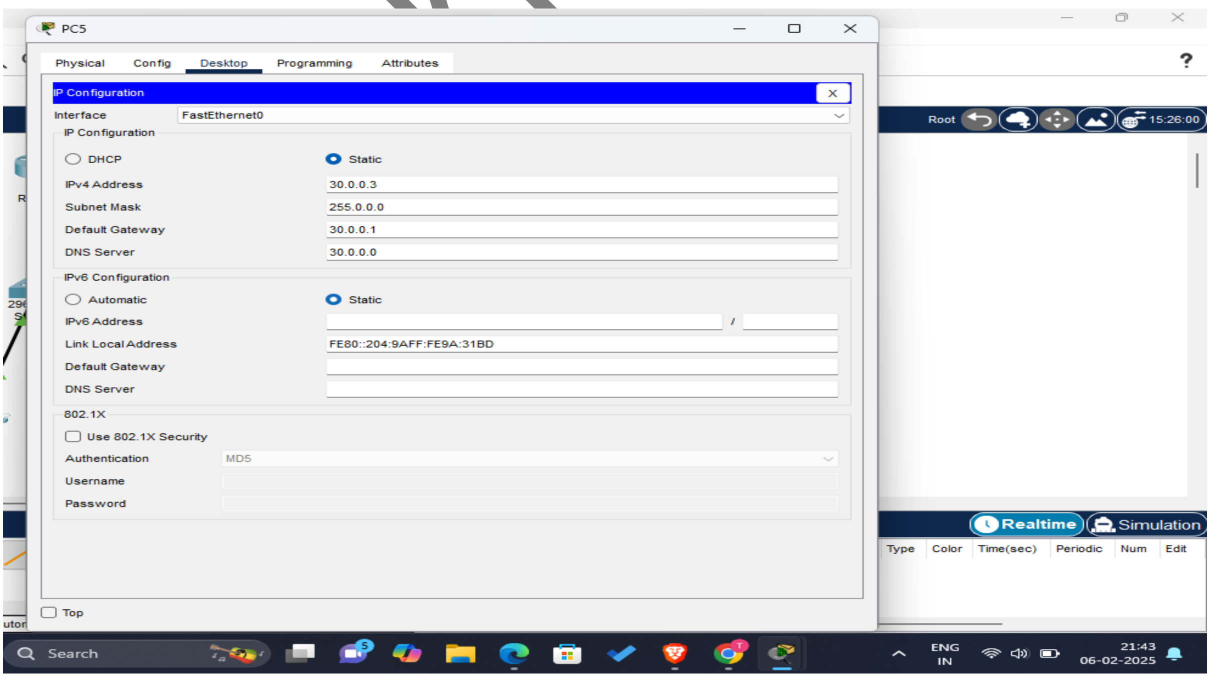

Router1
Physical Config CLI Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
GigabitEthernet0/0
GigabitEthernet0/1
Serial0/1/0
Serial0/1/1
Serial0/1/1
Port Status On
Duplex Full Duplex
Clock Rate 2000000
IP Configuration
IPv4 Address 192.168.1.1
Subnet Mask 255.255.255.0
Tx Ring Limit 10
Equivalent IOS Commands
Router(config-if)#
Router(config-if)#exit
Router(config)#interface GigabitEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface GigabitEthernet0/1
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial0/1/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial0/1/1
Router(config-if)#
Top
Search
ENG IN
21:38
06-02-2025

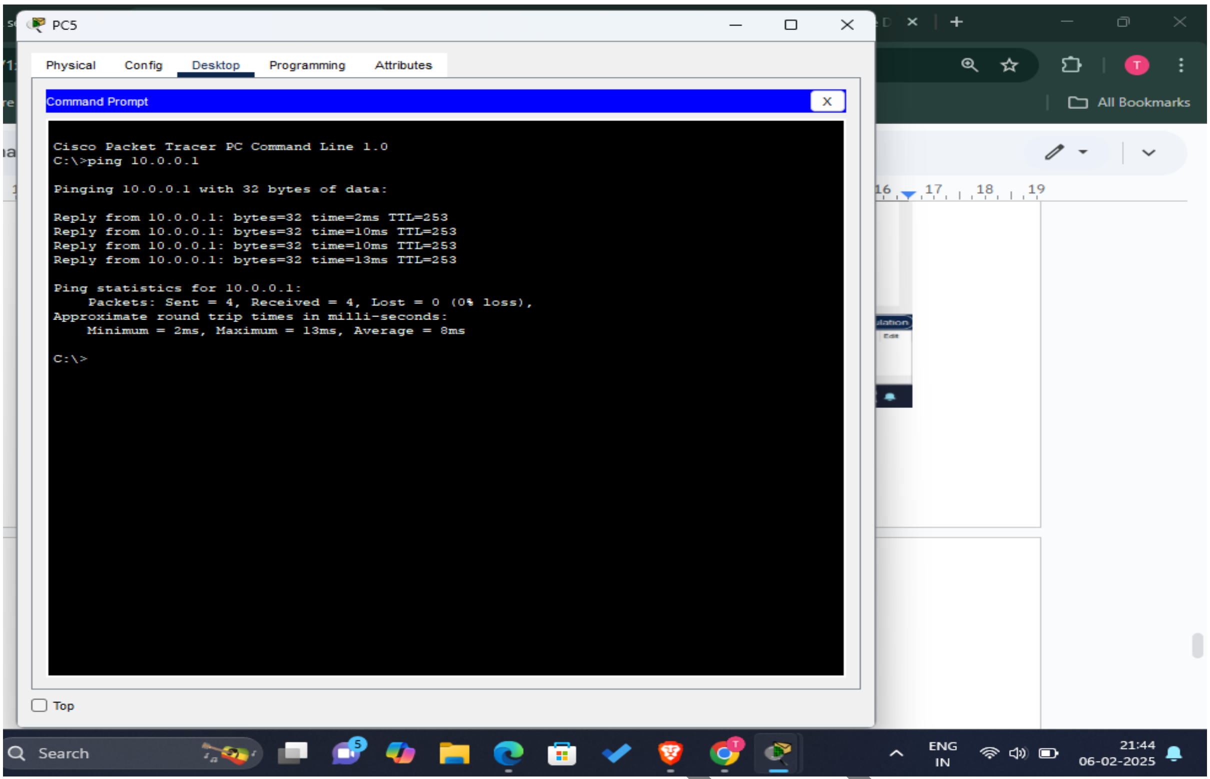

Router2
Physical Config CLI Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
GigabitEthernet0/0
GigabitEthernet0/1
Serial0/1/0
Serial0/1/1
GigabitEthernet0/0
Port Status On
Bandwidth 1000 Mbps 100 Mbps 10 Mbps Auto
Duplex Half Duplex Full Duplex Auto
MAC Address 000D.BDDA.5901
IP Configuration
IPv4 Address 30.0.0.1
Subnet Mask 255.0.0.0
Tx Ring Limit 10
Equivalent IOS Commands
Press RETURN to get started!
Router>enable
Router#
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface GigabitEthernet0/0
Router(config-if)#
Top
Realtime Simulation
Type Color Time(sec) Periodic Num Edit
Search
ENG IN
21:39
06-02-2025

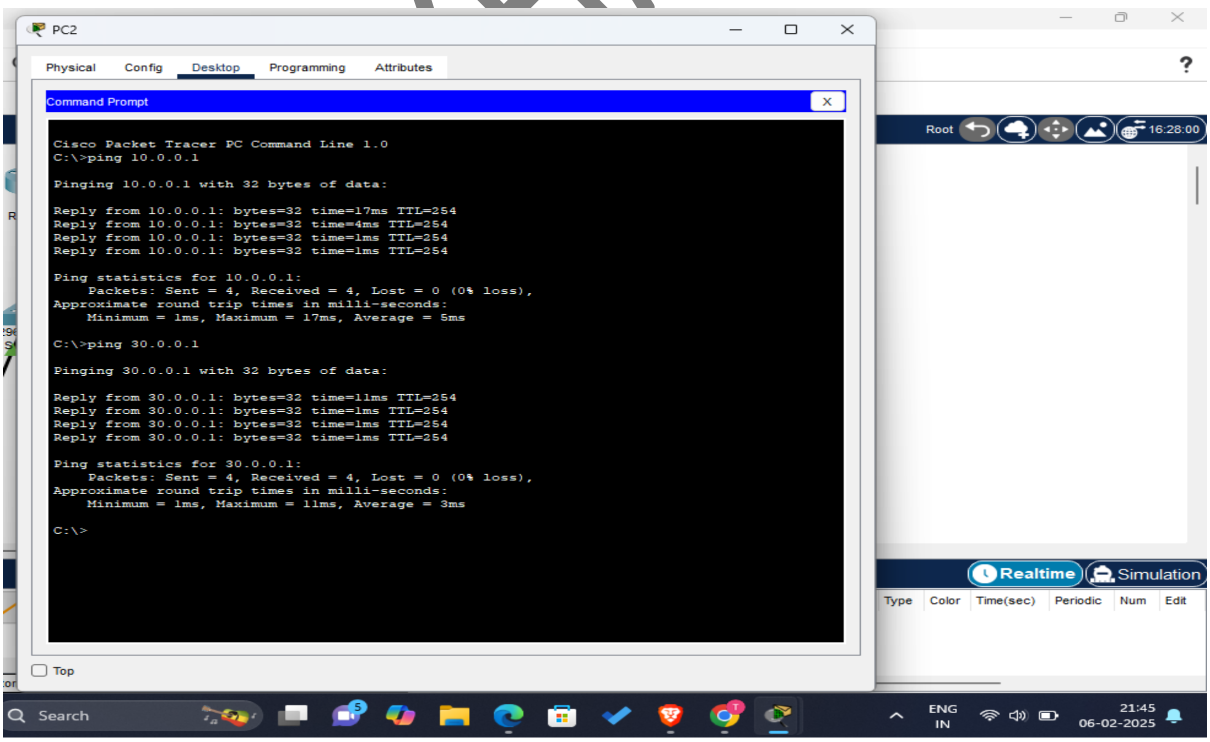

Router2
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
GigabitEthernet0/0
GigabitEthernet0/1
Serial0/1/0
Serial0/1/1
Serial0/1/1
Port Status
On
Duplex
Full Duplex
Clock Rate
2000000
IP Configuration
IPv4 Address
192.168.1.2
Subnet Mask
255.255.255.0
Tx Ring Limit
10
Equivalent IOS Commands
Press RETURN to get started!
Router>enable
Router#
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface GigabitEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial0/1/1
Router(config-if)#
Top
Search
ENG
IN
21:40
06-02-2025
All Bookmarks


PC0
Physical
Config
Desktop
Programming
Attributes
IP Configuration
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
10.0.0.0
IPv6 Configuration
Automatic
Static
IPv6 Address
Link Local Address
FE80::2D0:97FF:FEBD:D8BC
Default Gateway
DNS Server
802.1X
Use 802.1X Security
Authentication
MD5
Username
Password
Top
Root
14:25:00
Realtime
Simulation
Type
Color
Time(sec)
Periodic
Num
Edit
Search
ENG
IN
21:41
06-02-2025


PC2
Physical
Config
Desktop
Programming
Attributes
IP Configuration
Interface
FastEthernet0
IP Configuration
DHCP
Static
IPv4 Address
20.0.0.2
Subnet Mask
255.0.0.0
Default Gateway
20.0.0.1
DNS Server
20.0.0.0
IPv6 Configuration
Automatic
Static
IPv6 Address
Link Local Address
FE80::260:5CFF:FE6B:2307
Default Gateway
DNS Server
802.1X
Use 802.1X Security
Authentication
MD5
Username
Password
Top
Root
14:57:00
Realtime
Simulation
Type
Color
Time(sec)
Periodic
Num
Edit

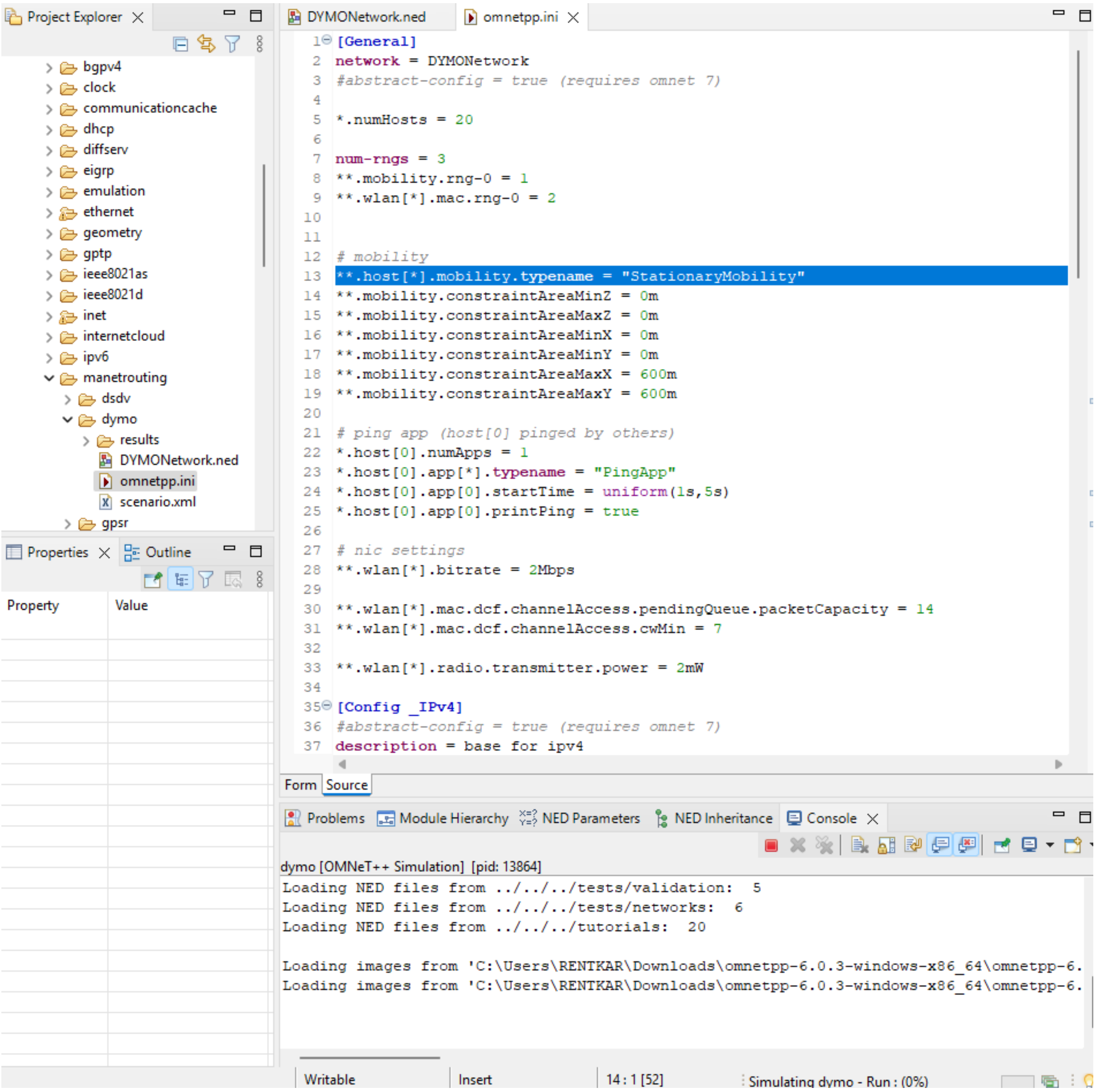

PC5
Physical
Config
Desktop
Programming
Attributes
IP Configuration
Interface
FastEthernet0
IP Configuration
DHCP
Static
IPv4 Address
30.0.0.3
Subnet Mask
255.0.0.0
Default Gateway
30.0.0.1
DNS Server
30.0.0.0
IPv6 Configuration
Automatic
Static
IPv6 Address
Link Local Address
FE80::204:9AFF:FE9A:31BD
Default Gateway
DNS Server
802.1X
Use 802.1X Security
Authentication
MD5
Username
Password
Top
Root
15:26:00
Realtime
Simulation
Type
Color
Time(sec)
Periodic
Num
Edit
Search
ENG
IN
21:43
06-02-2025


PC5
Physical
Config
Desktop
Programming
Attributes
Command Prompt
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=2ms TTL=253
Reply from 10.0.0.1: bytes=32 time=10ms TTL=253
Reply from 10.0.0.1: bytes=32 time=10ms TTL=253
Reply from 10.0.0.1: bytes=32 time=13ms TTL=253
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 2ms, Maximum = 13ms, Average = 8ms
C:\>
Top
All Bookmarks
Search
ENG
IN
21:44
06-02-2025


PC2
Physical
Config
Desktop
Programming
Attributes
Command Prompt
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.1
Pinging 10.0.0.1 with 32 bytes of data:
Reply from 10.0.0.1: bytes=32 time=17ms TTL=254
Reply from 10.0.0.1: bytes=32 time=4ms TTL=254
Reply from 10.0.0.1: bytes=32 time=1ms TTL=254
Reply from 10.0.0.1: bytes=32 time=1ms TTL=254
Ping statistics for 10.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 1ms, Maximum = 17ms, Average = 5ms
C:\>ping 30.0.0.1
Pinging 30.0.0.1 with 32 bytes of data:
Reply from 30.0.0.1: bytes=32 time=11ms TTL=254
Reply from 30.0.0.1: bytes=32 time=1ms TTL=254
Reply from 30.0.0.1: bytes=32 time=1ms TTL=254
Reply from 30.0.0.1: bytes=32 time=1ms TTL=254
Ping statistics for 30.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 1ms, Maximum = 11ms, Average = 3ms
C:\>
Top
Root
16:28:00
Realtime
Simulation
Type
Color
Time(sec)
Periodic
Num
Edit
Search
ENG
IN
21:45
06-02-2025

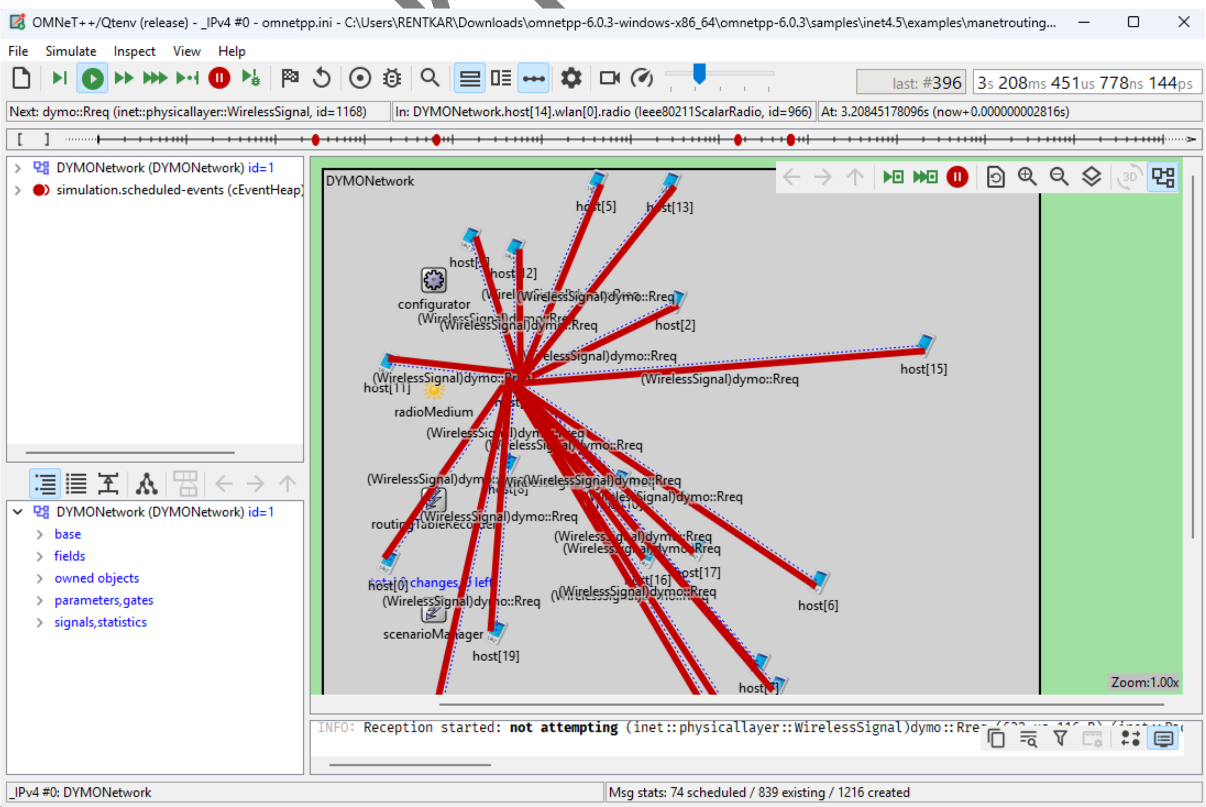
PRACTICAL 6

Aim: Create a basic MANET implementation simulation for Packet animation and Packet Trace.



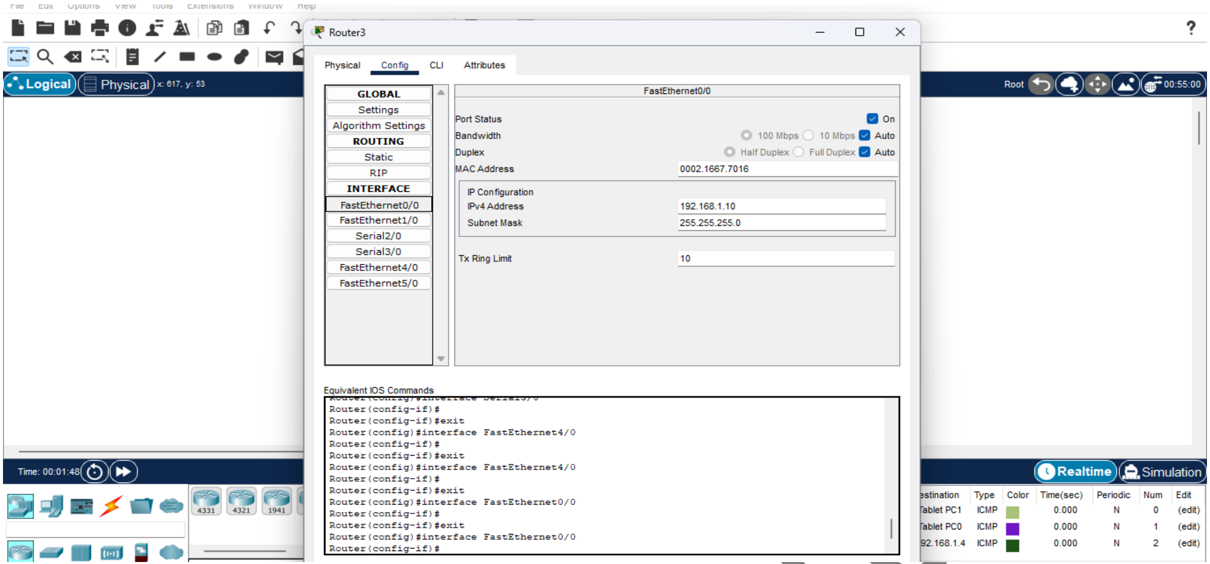
DYMONetwork.ned | omnetpp.ini

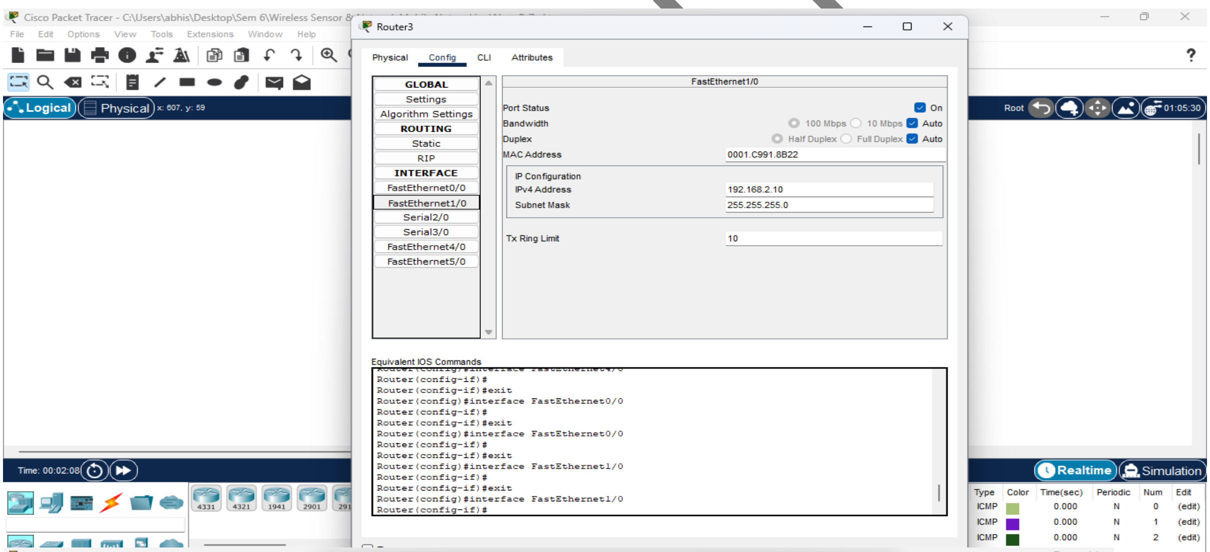
```
package inet.examples.manetrouting.dymo;

import inet.common.scenario.ScenarioManager;
import inet.networklayer.configurator.ipv4.Ipv4NetworkConfigurator;
import inet.networklayer.ipv4.RoutingTableRecorder;
import inet.node.dymo.DymoRouter;
import inet.physicallayer.wireless.ieee80211.packetlevel.Ieee80211ScalarRadioMedium;


//
// TODO
//
network DYMONetwork
{
    parameters:
        int numHosts;
        @display("bgb=650,650");
    submodules:
        radioMedium: Ieee80211ScalarRadioMedium {
            parameters:
                @display("p=100,200;is=s");
        }
        configurator: Ipv4NetworkConfigurator {
            parameters:
                config = xml("<config><interface hosts='*' address='145.236.x.x' netmask='255.255.0.0'/></config>");
                @display("p=100,100;is=s");
        }
        routingTableRecorder: RoutingTableRecorder {
            parameters:
                @display("p=100,300;is=s");
        }
        scenarioManager: ScenarioManager {
            parameters:
                script = default(xml("<scenario/>"));
                @display("p=100,400;is=s");
        }
```

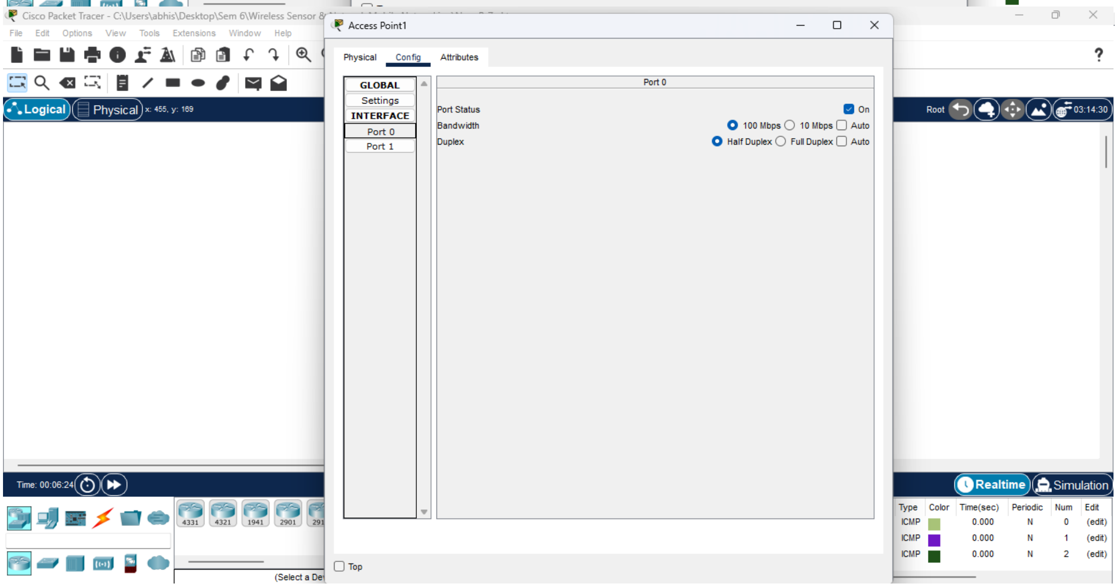


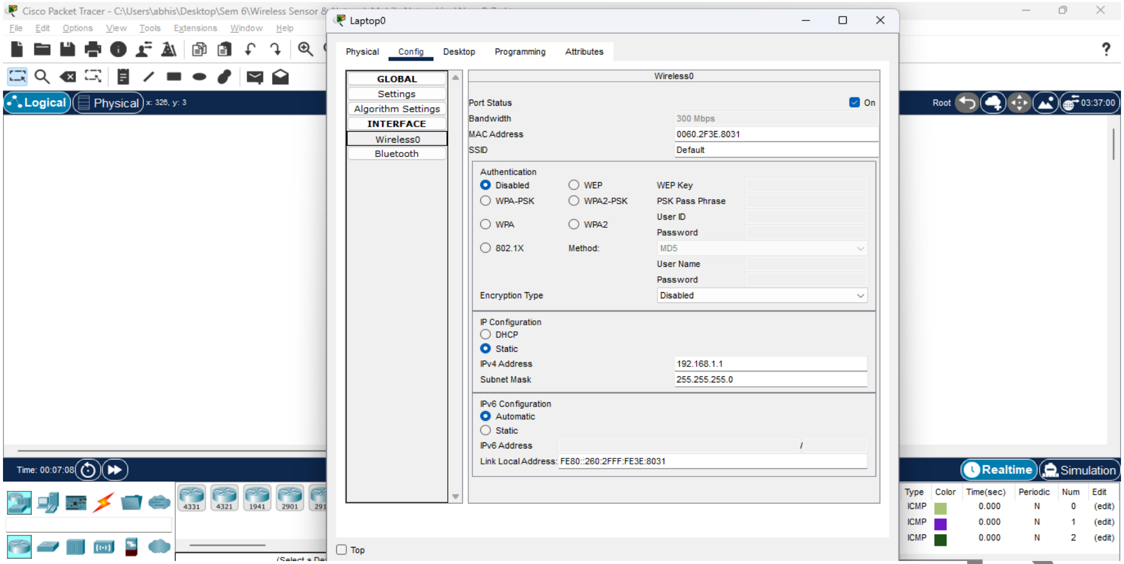
PRACTICAL 7

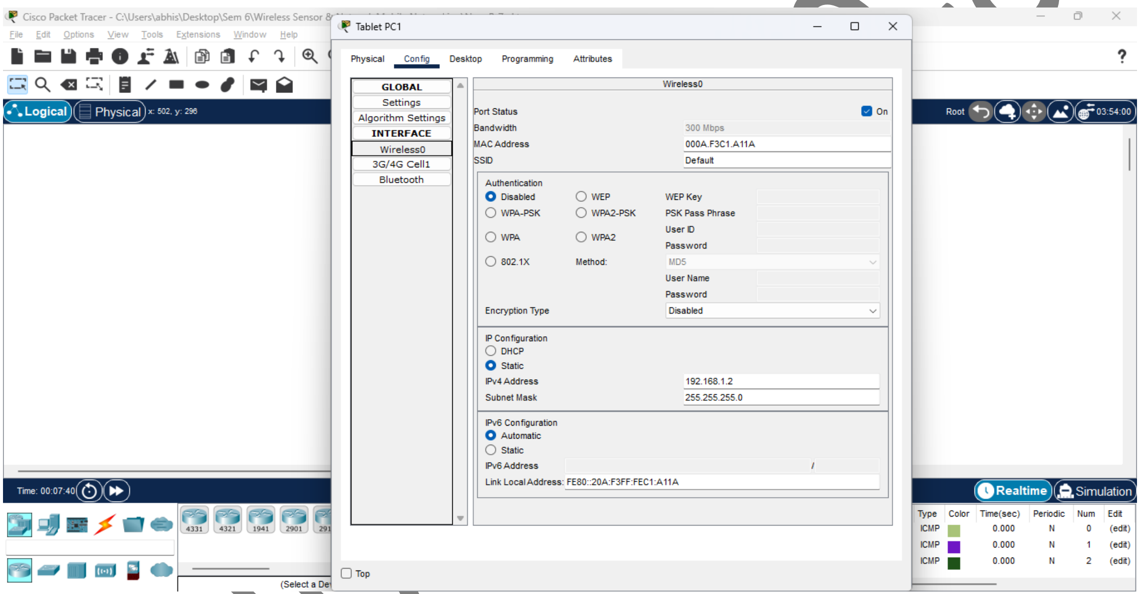
Aim: Create a basic MANET implementation simulation for Packet animation and Packet Trace.

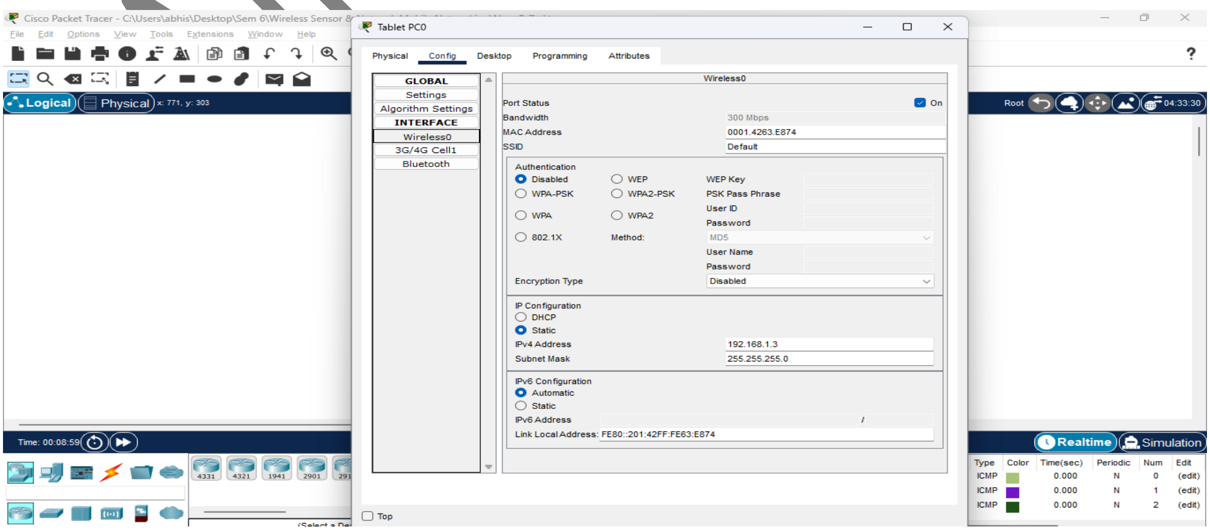


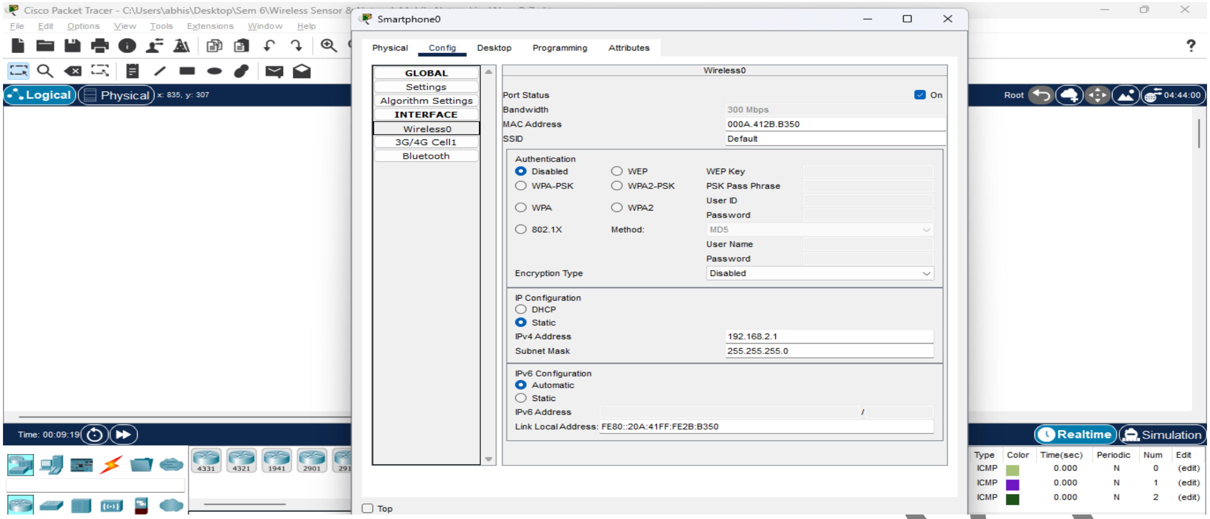


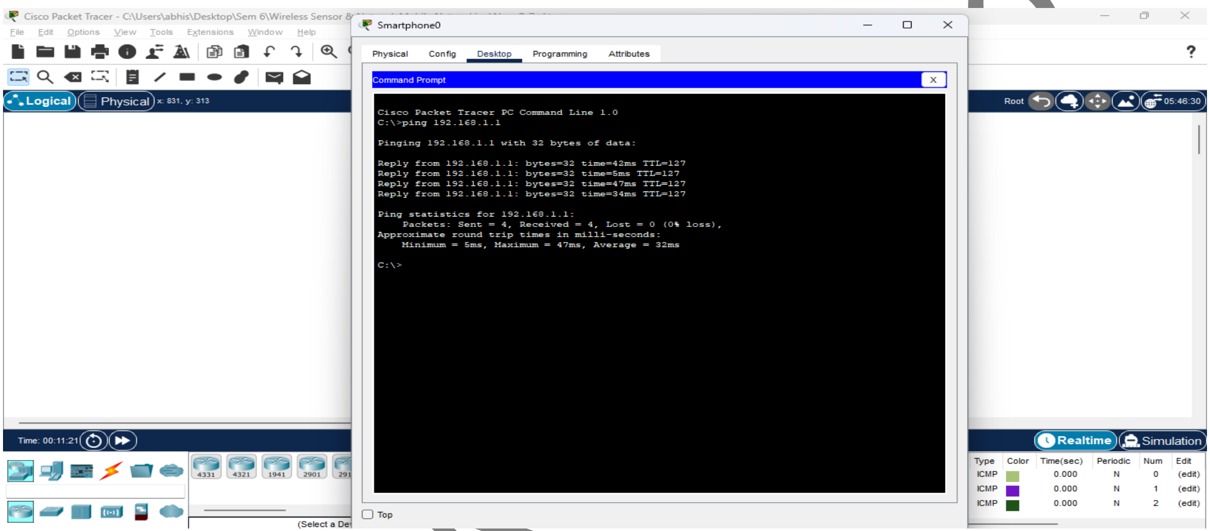


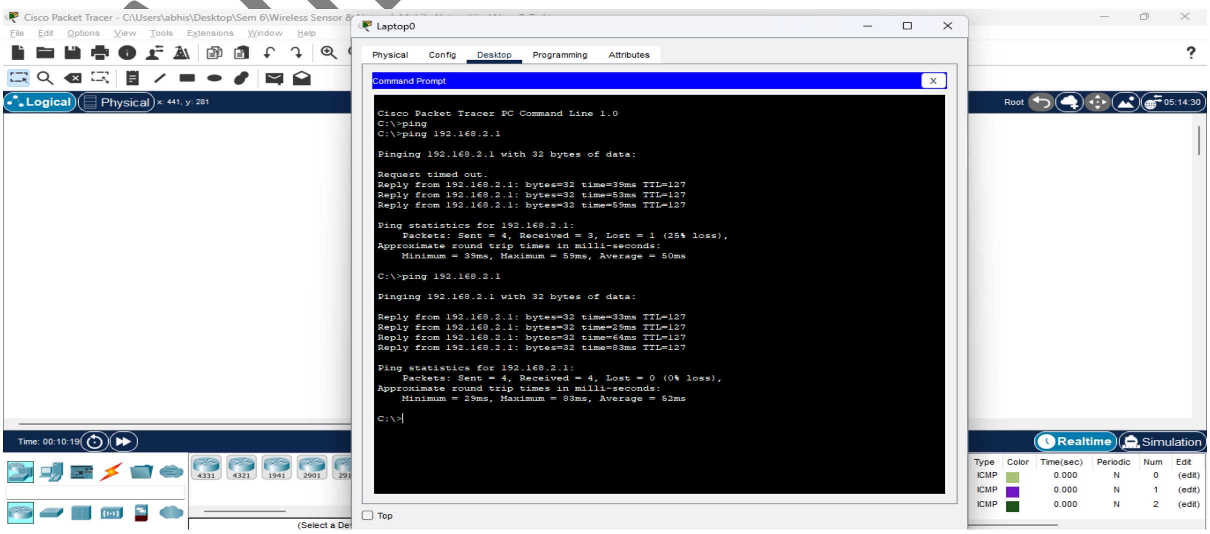






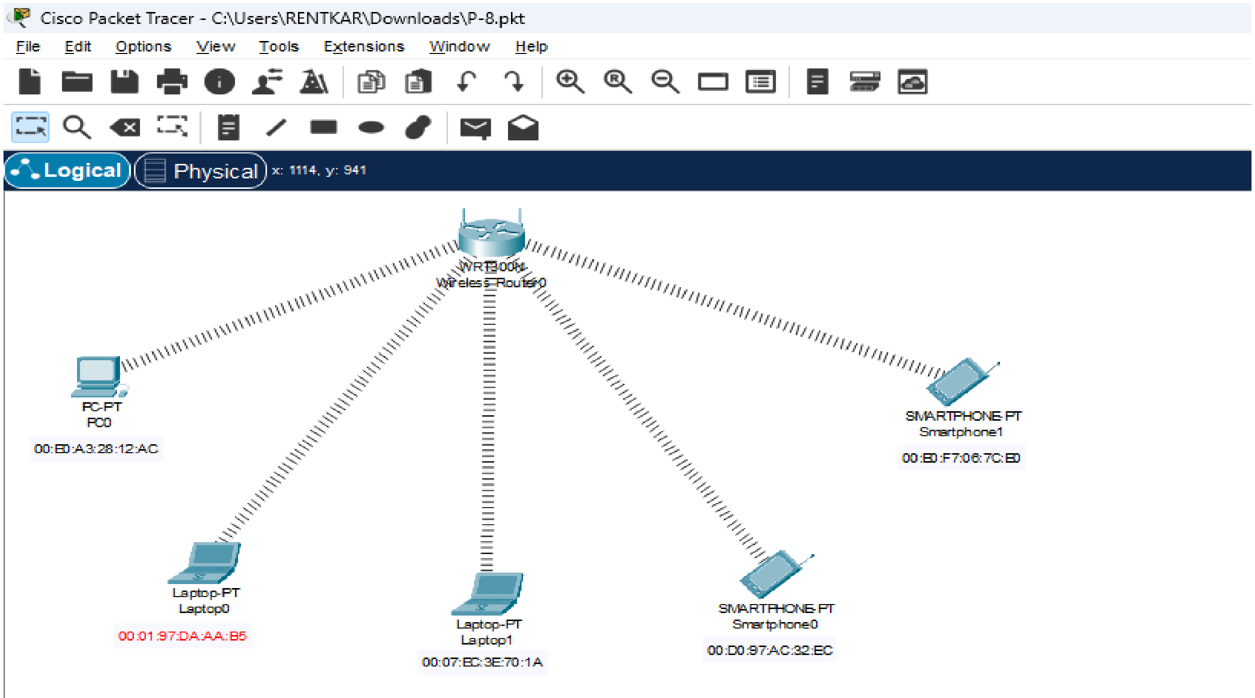

Smartphone0
Physical Config Desktop Programming Attributes
GLOBAL
Settings
Algorithm Settings
INTERFACE
Wireless0
3G/4G Cell1
Bluetooth
Wireless0
Port Status On
Bandwidth 300 Mbps
MAC Address 000A.412B.B350
SSID Default
Authentication
Disabled WEP WEP Key
WPA-PSK WPA2-PSK PSK Pass Phrase
User ID
WPA WPA2 Password
802.1X Method: MD5
User Name
Password
Encryption Type Disabled
IP Configuration
DHCP
Static
IPv4 Address 192.168.2.1
Subnet Mask 255.255.255.0
IPv6 Configuration
Automatic
Static
IPv6 Address
Link Local Address: FE80::20A:41FF:FE2B:B350
Top

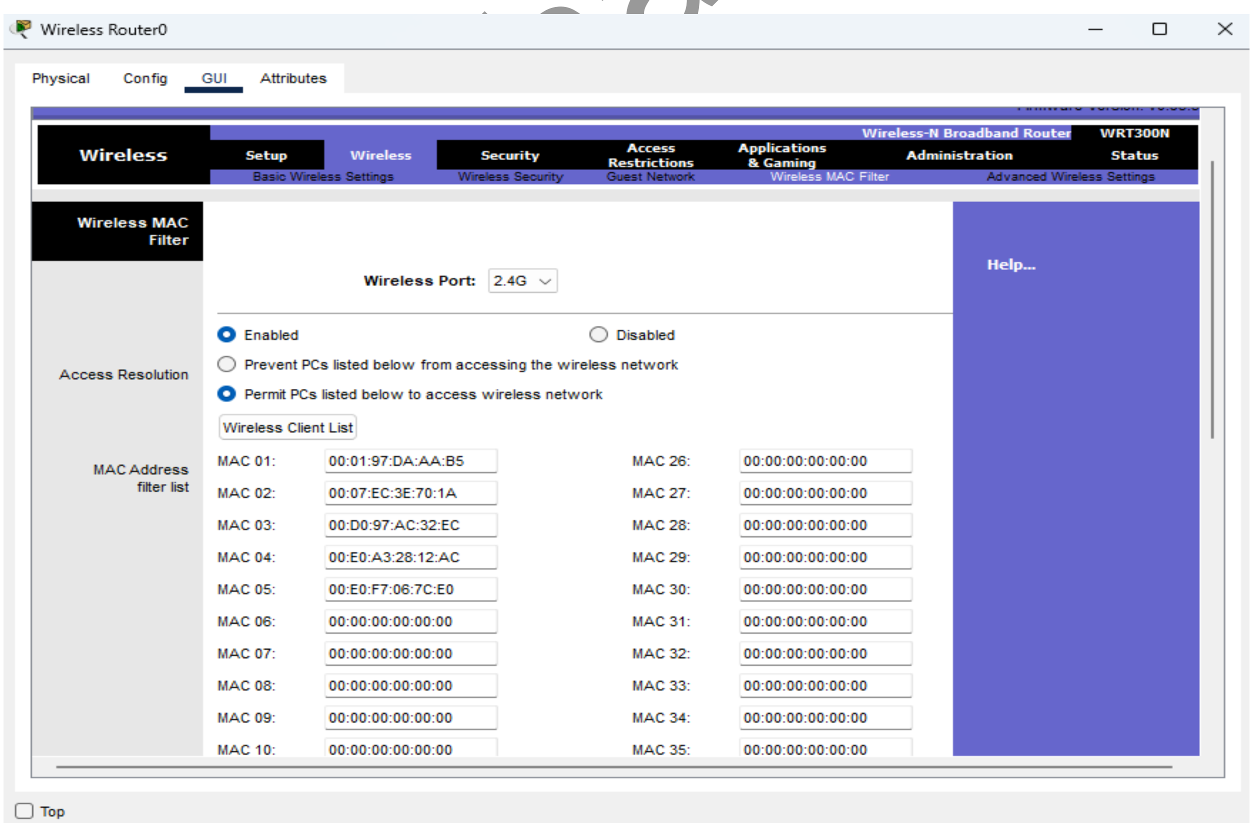

Smartphone0
Physical Config Desktop Programming Attributes
Command Prompt
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.1.1
Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time=42ms TTL=127
Reply from 192.168.1.1: bytes=32 time=5ms TTL=127
Reply from 192.168.1.1: bytes=32 time=47ms TTL=127
Reply from 192.168.1.1: bytes=32 time=34ms TTL=127
Ping statistics for 192.168.1.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 5ms, Maximum = 47ms, Average = 32ms
C:\>
Top


Laptop0
Physical Config Desktop Programming Attributes
Command Prompt
Cisco Packet Tracer PC Command Line 1.0
C:\>ping
C:\>ping 192.168.2.1
Pinging 192.168.2.1 with 32 bytes of data:
Request timed out.
Reply from 192.168.2.1: bytes=32 time=39ms TTL=127
Reply from 192.168.2.1: bytes=32 time=53ms TTL=127
Reply from 192.168.2.1: bytes=32 time=59ms TTL=127
Ping statistics for 192.168.2.1:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 39ms, Maximum = 59ms, Average = 50ms
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=33ms TTL=127
Reply from 192.168.2.1: bytes=32 time=29ms TTL=127
Reply from 192.168.2.1: bytes=32 time=64ms TTL=127
Reply from 192.168.2.1: bytes=32 time=83ms TTL=127
Ping statistics for 192.168.2.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 29ms, Maximum = 83ms, Average = 52ms
C:\>
Top

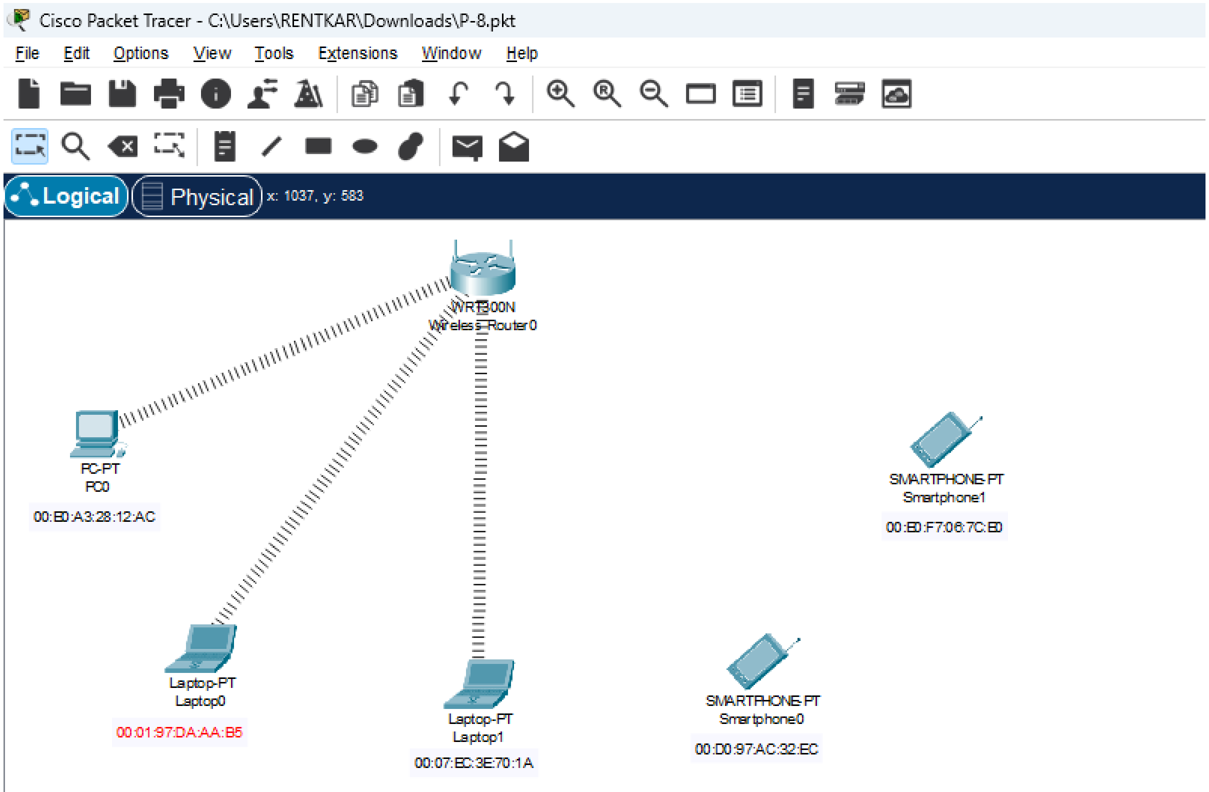
SHETH L.U.J. SIR M.V. COLLEGE
PRACTICAL 8

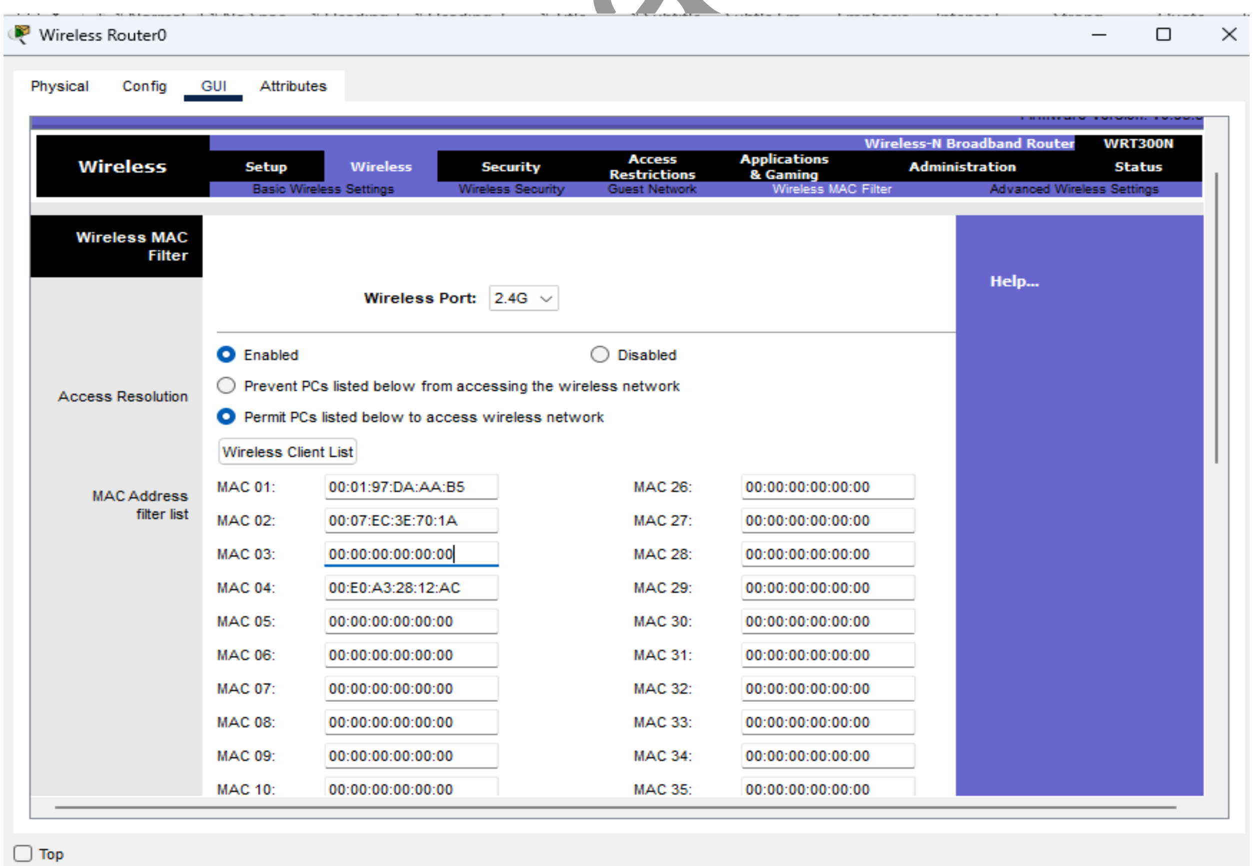
Aim: Create MAC protocol simulation implementation for wireless sensor Network



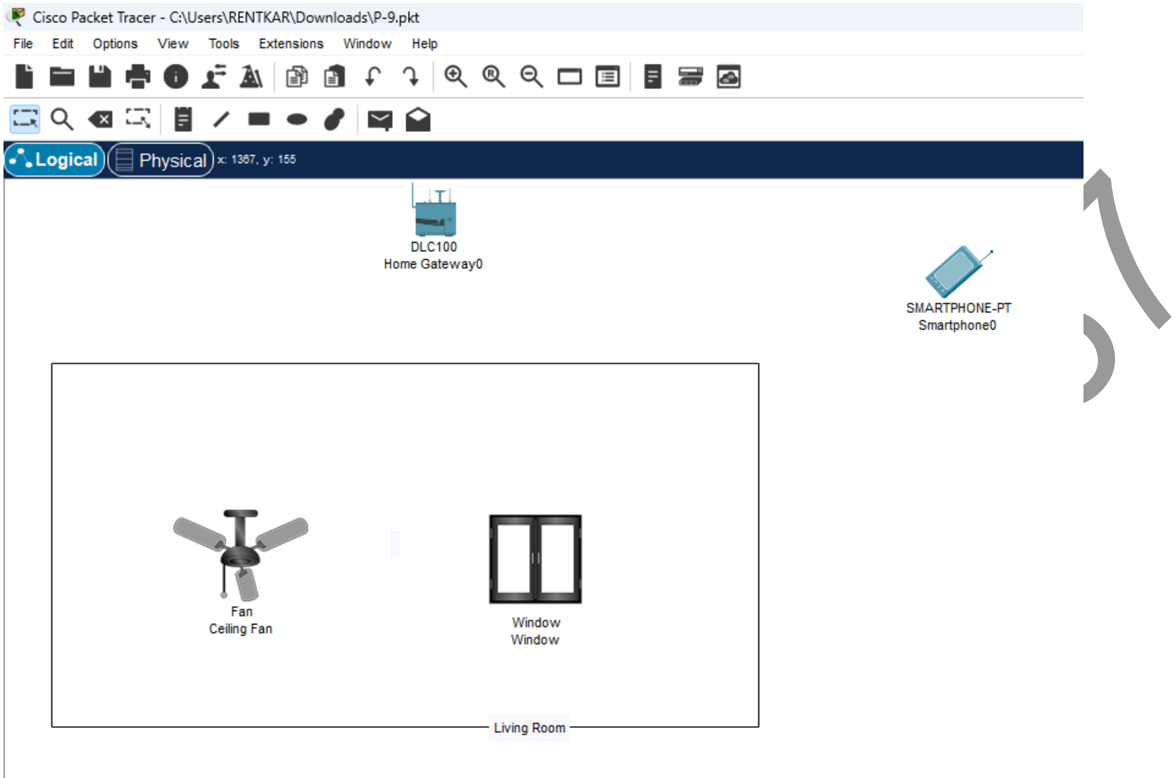


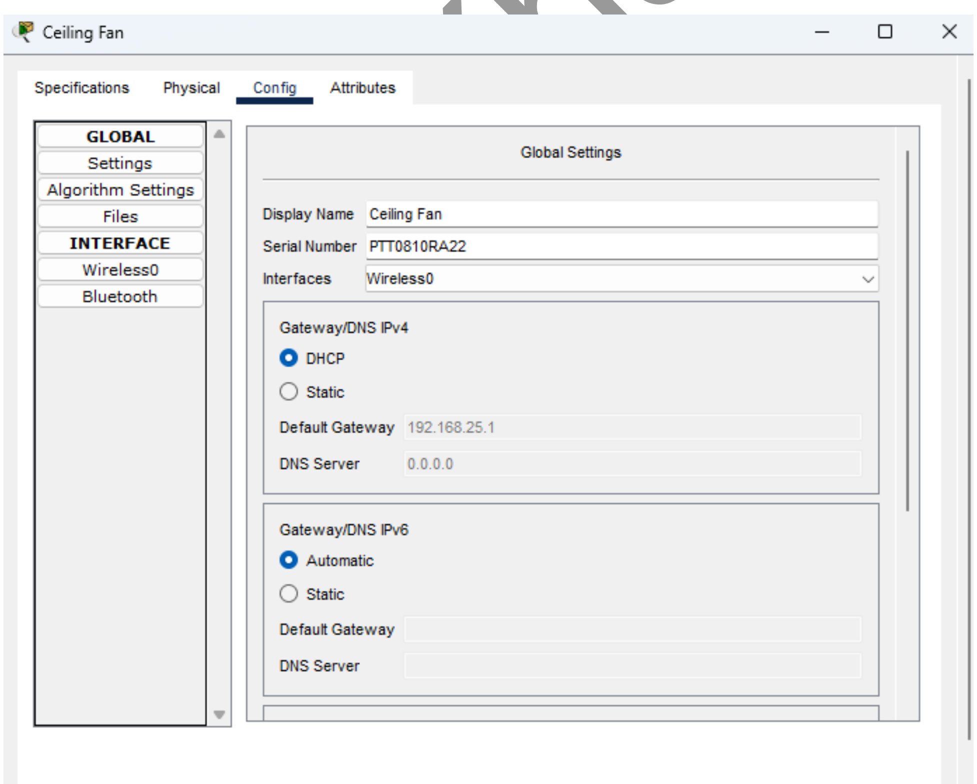
SHETH L.U.J. SIR M.V. COLLEGE
PRACTICAL 8



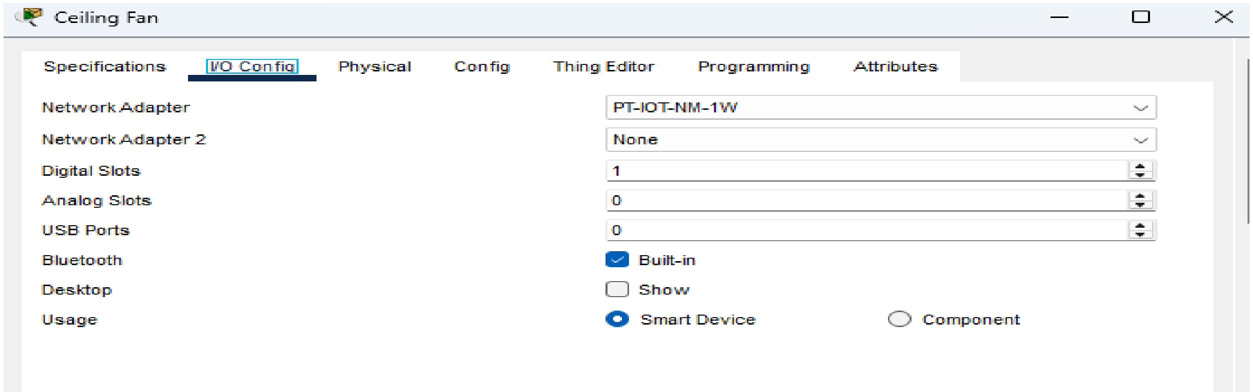


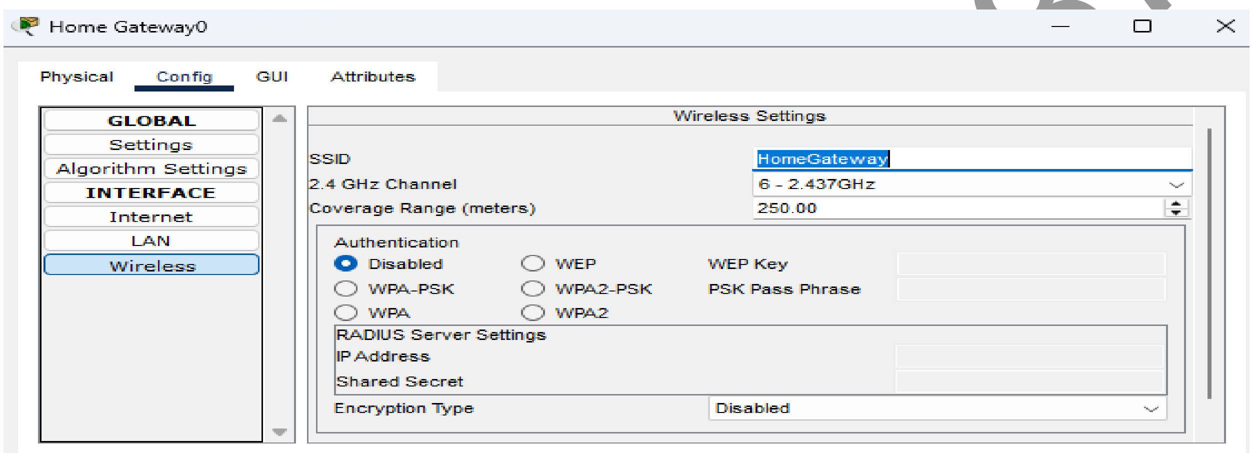
Aim: Simulate Mobile Adhoc Network with Directional Antenna

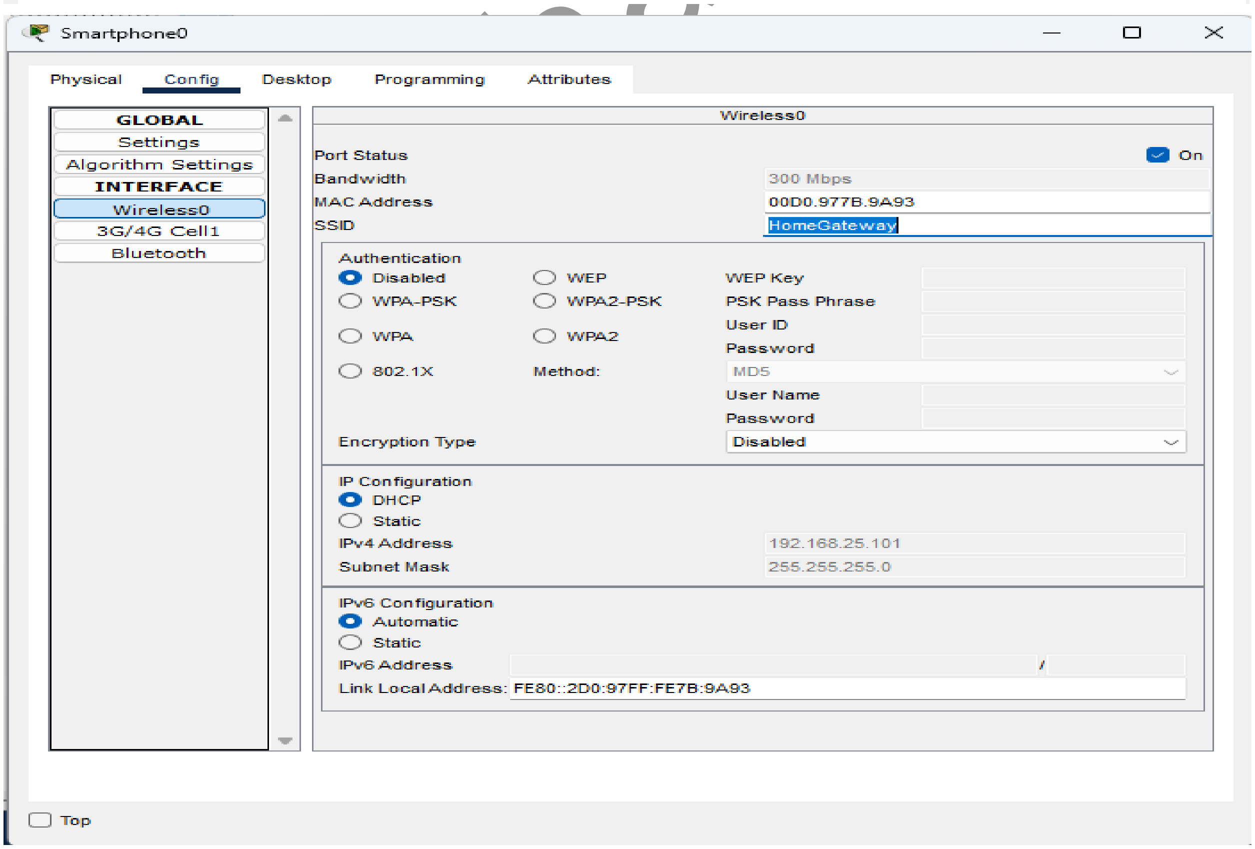


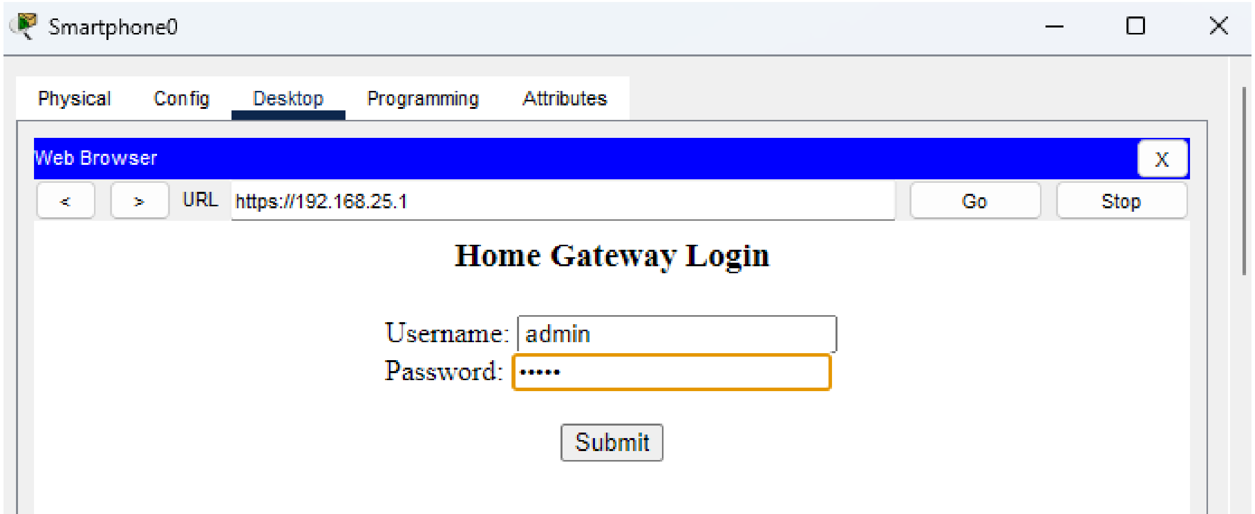


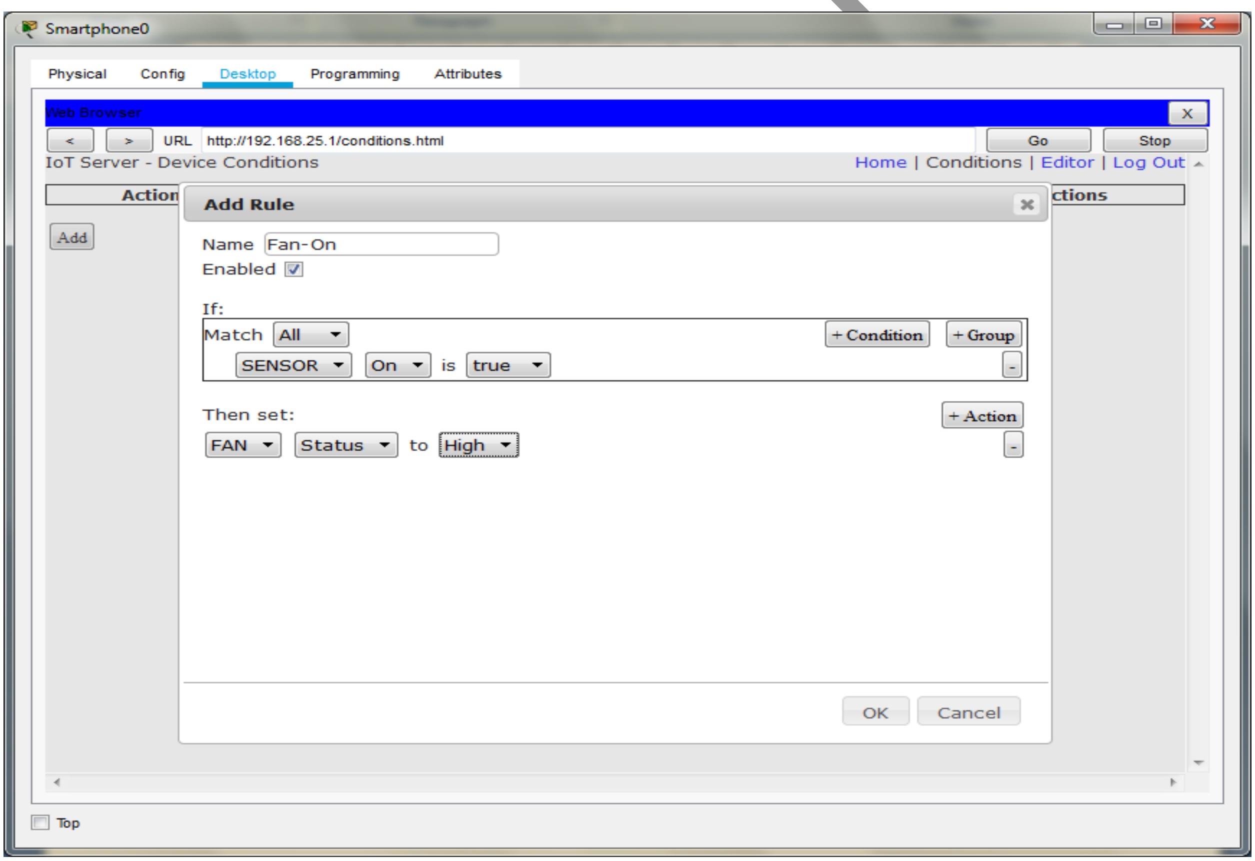
SHETH L.U.J. SIR M.V. COLLEGE
PRACTICAL 9

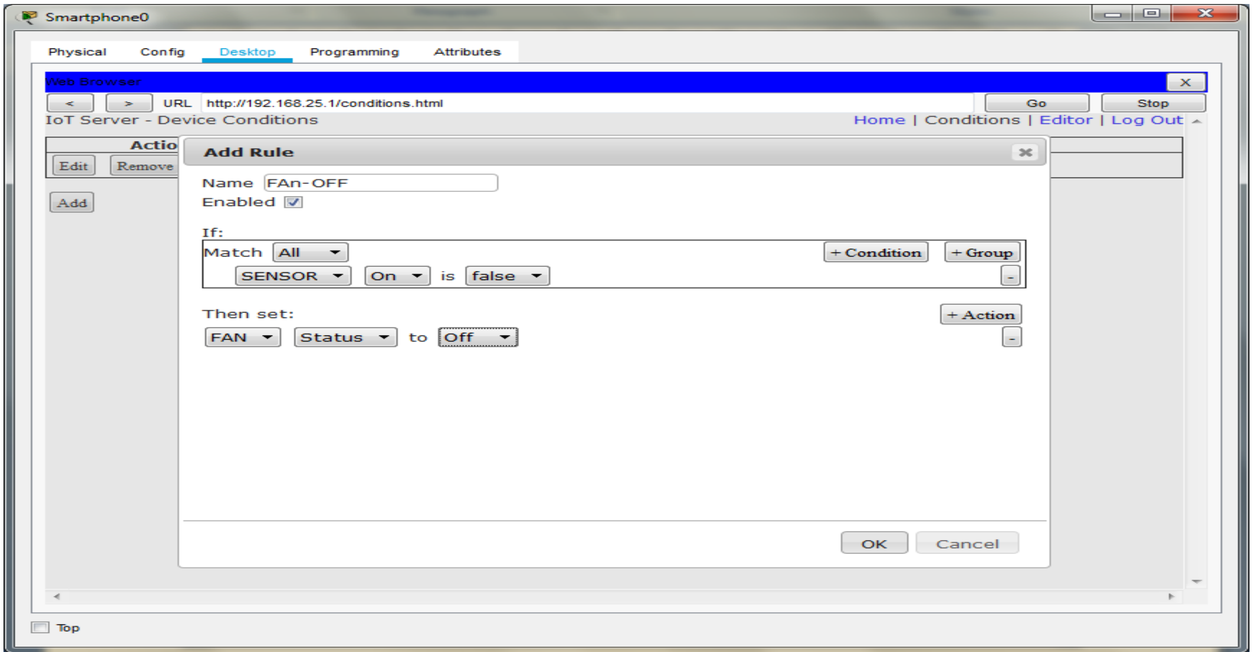


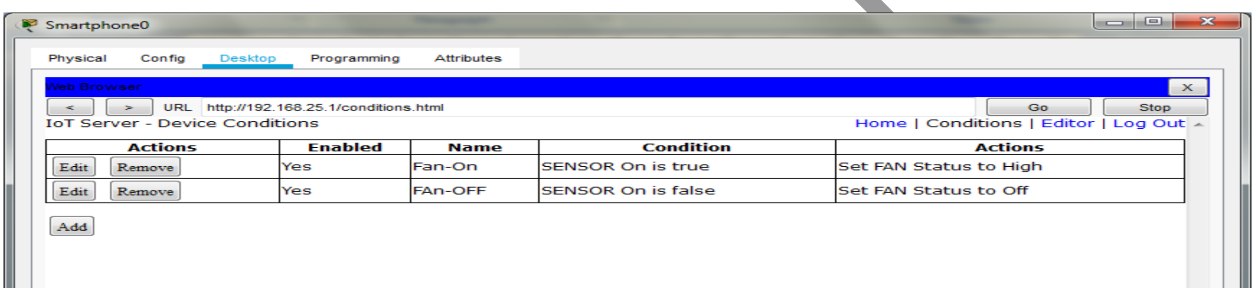


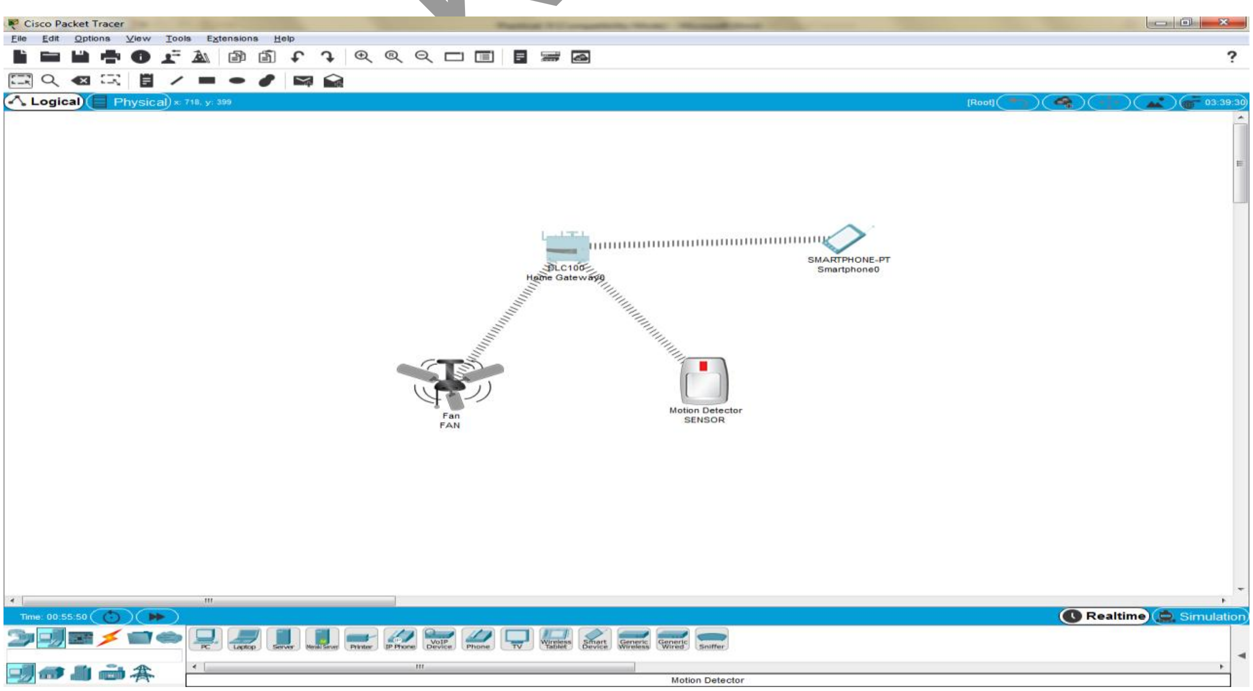



Smartphone0
Physical
Config
Desktop
Programming
Attributes
Web Browser
URL
https://192.168.25.1
Go
Stop
Home Gateway Login
Username:
admin
Password:
Submit


Smartphone0
Physical
Config
Desktop
Programming
Attributes
Web Browser
URL
http://192.168.25.1/conditions.html
Go
Stop
IoT Server - Device Conditions
Home | Conditions | Editor | Log Out
Add Rule
Name Fan-On
Enabled
If:
Match All
+ Condition
+ Group
SENSOR On is true
Then set:
+ Action
FAN Status to High
OK
Cancel
Top

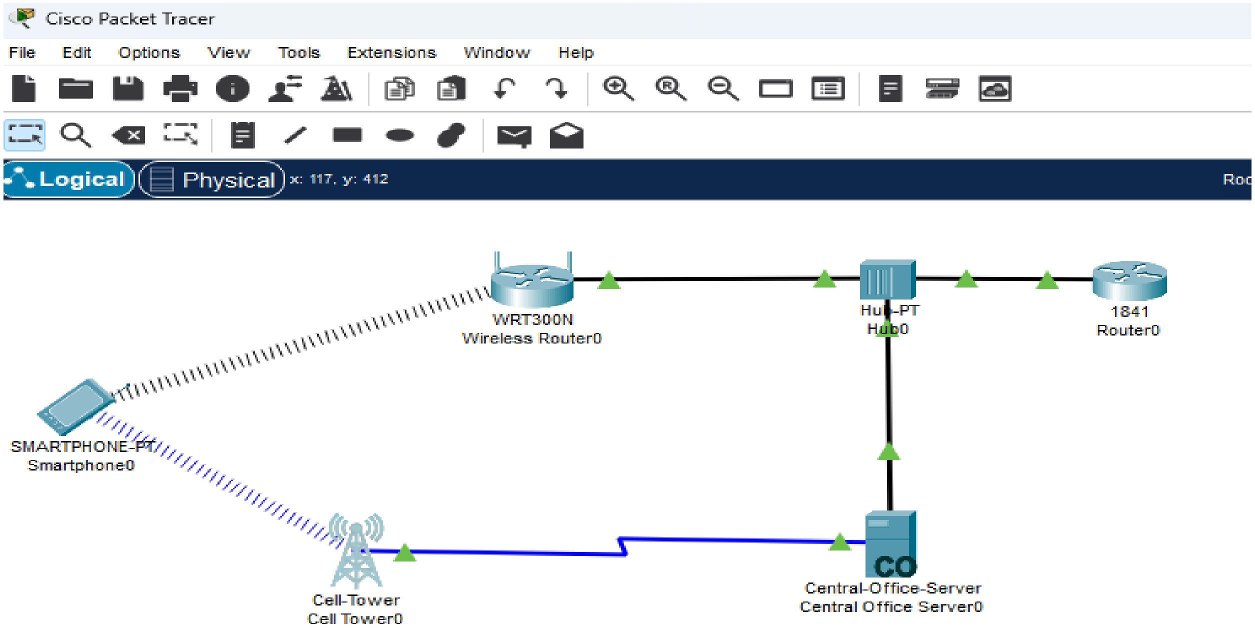

Smartphone0
Physical
Config
Desktop
Programming
Attributes
Web Browser
URL http://192.168.25.1/conditions.html
Go
Stop
IoT Server - Device Conditions
Home | Conditions | Editor | Log Out
Add Rule
Name FAn-OFF
Enabled
If:
Match All
+ Condition
+ Group
SENSOR On is false
Then set:
+ Action
FAN Status to Off
OK
Cancel
Top

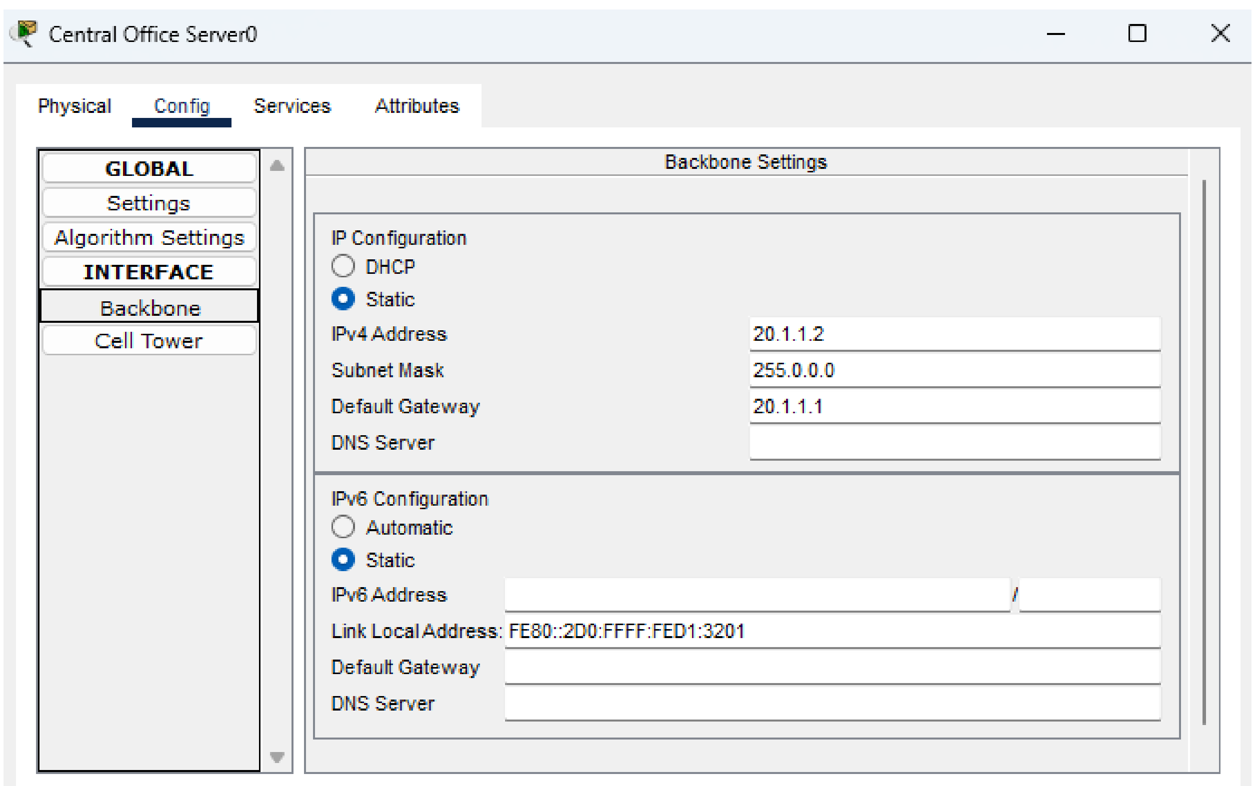

Smartphone0
Physical
Config
Desktop
Programming
Attributes
Web Browser
URL http://192.168.25.1/conditions.html
Go
Stop
IoT Server - Device Conditions
Home | Conditions | Editor | Log Out
Actions
Enabled
Name
Condition
Actions
Edit Remove
Yes
Fan-On
SENSOR On is true
Set FAN Status to High
Edit Remove
Yes
FAn-OFF
SENSOR On is false
Set FAN Status to Off
Add

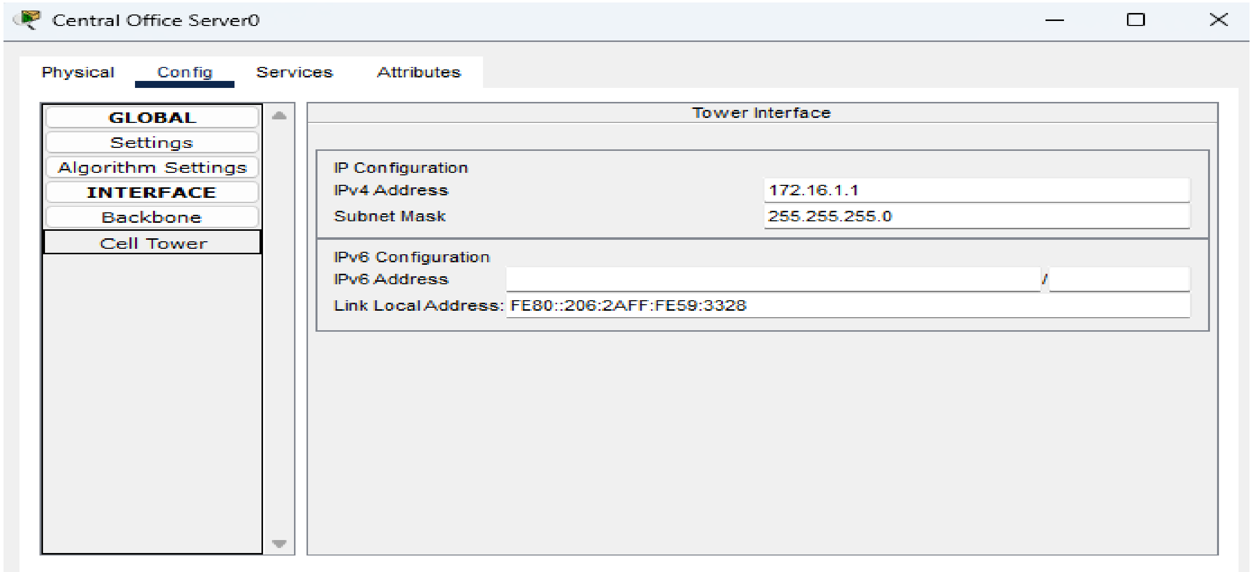

Cisco Packet Tracer
Logical
Physical
DLC100
Home Gateway0
SMARTPHONE-PT
Smartphone0
Fan
FAN
Motion Detector
SENSOR
Realtime
Simulation
Motion Detector

SHETH L.U.J. SIR M.V. COLLEGE
PRACTICAL 9

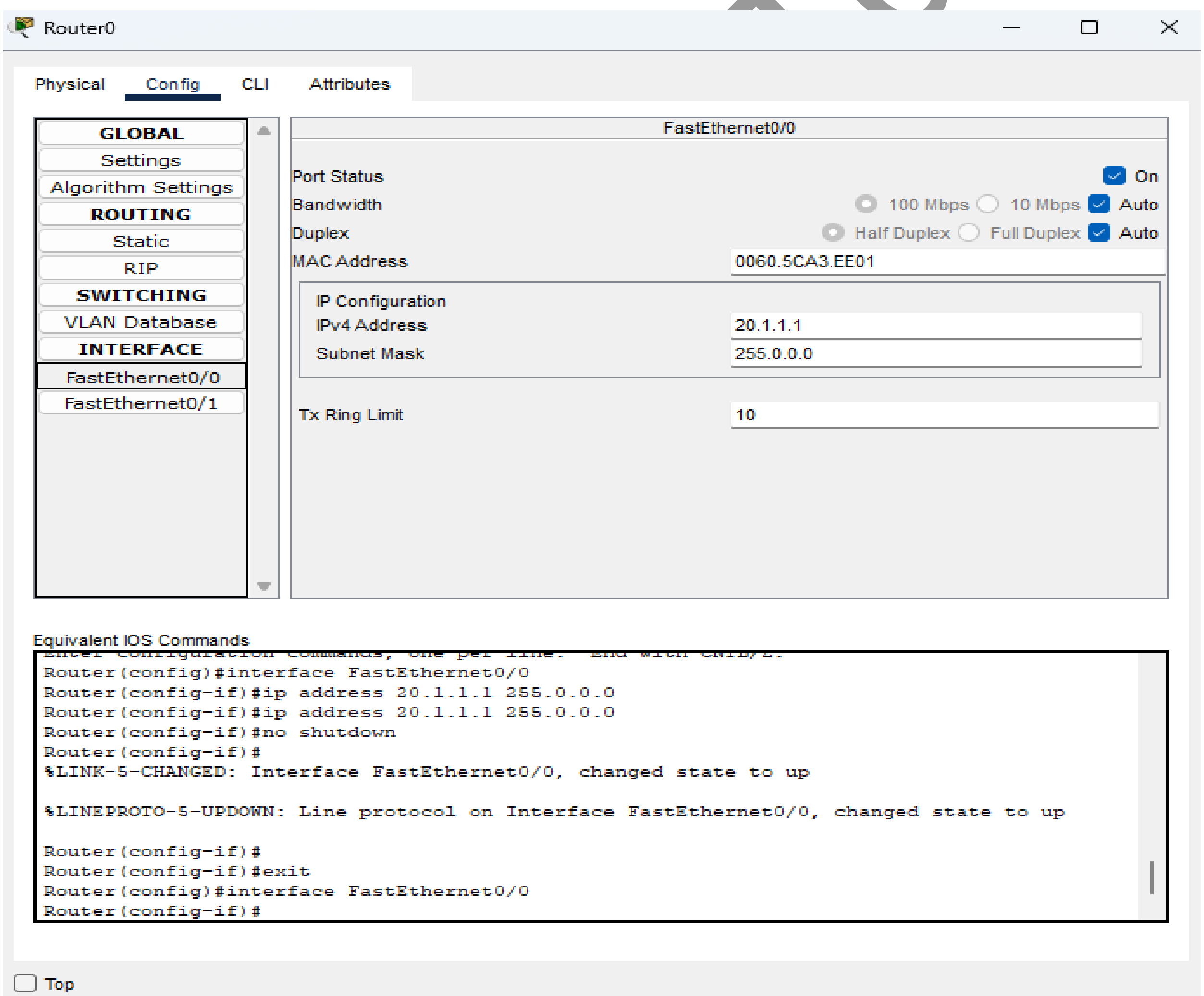
Aim: Create a mobile network using Cell Tower, Central Office Server, Web browser and Web Server. Simulate connection between them.



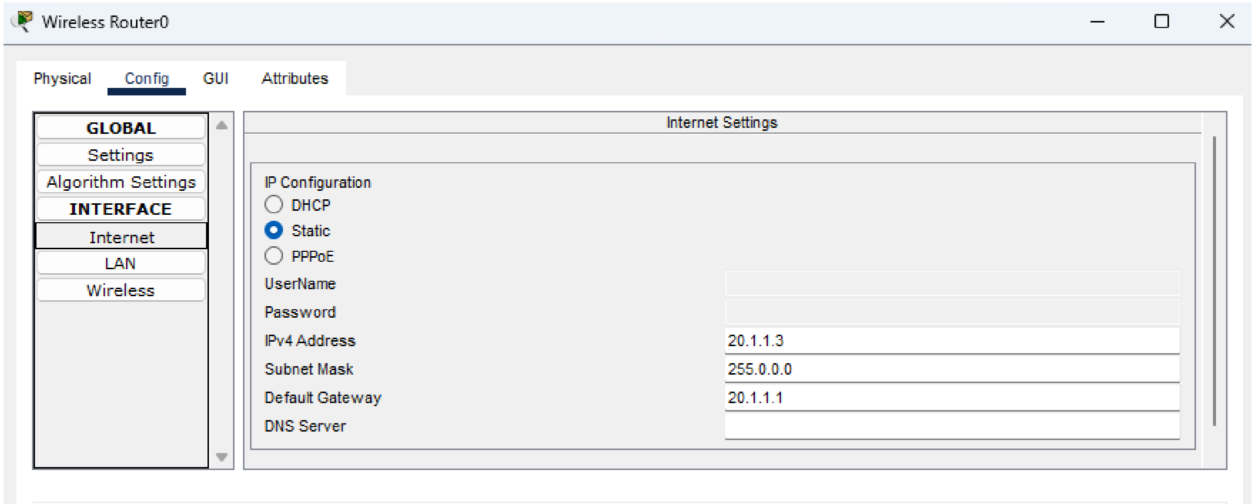


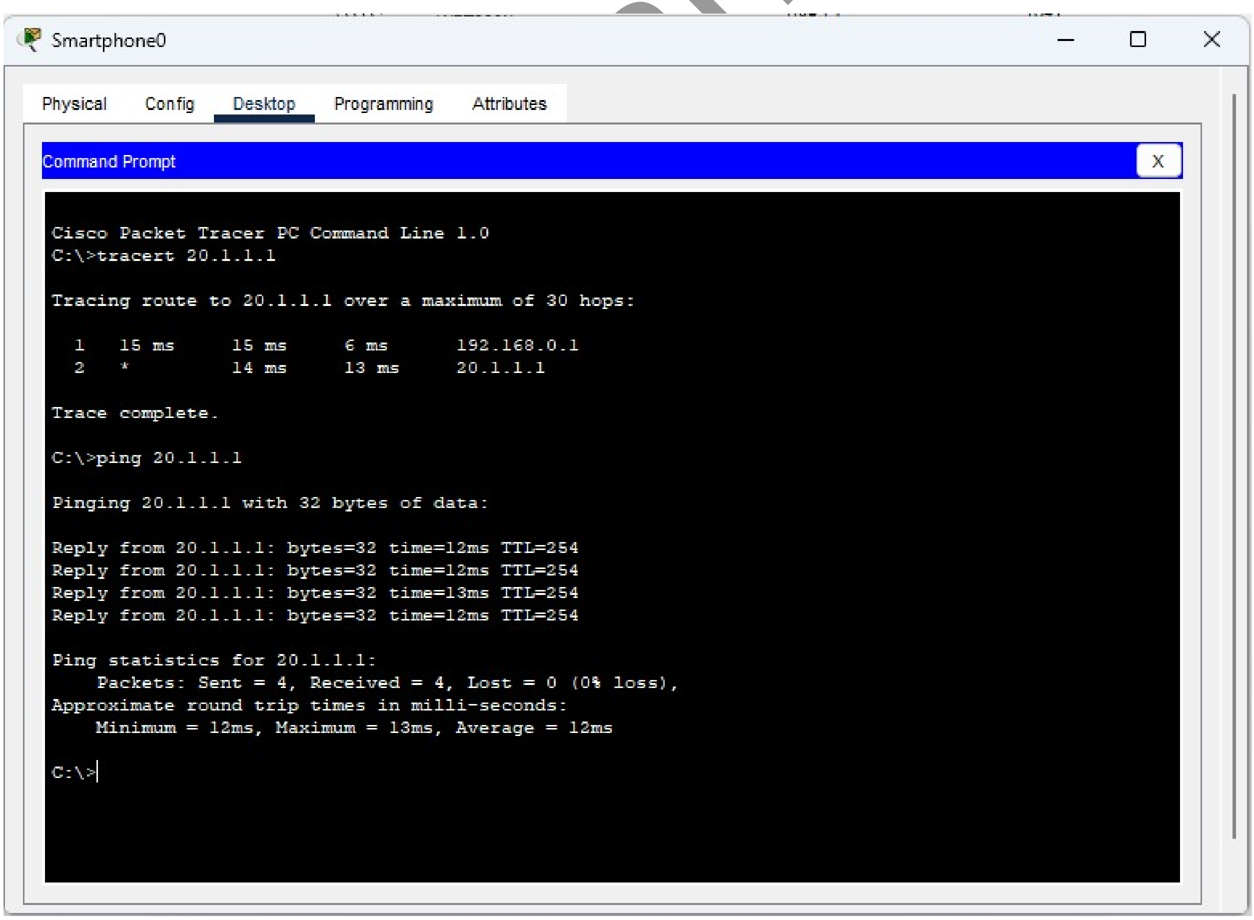


Central Office Server0
Physical
Config
Services
Attributes
GLOBAL
Settings
Algorithm Settings
INTERFACE
Backbone
Cell Tower
Tower Interface
IP Configuration
IPv4 Address
172.16.1.1
Subnet Mask
255.255.255.0
IPv6 Configuration
IPv6 Address
/
Link Local Address: FE80::206:2AFF:FE59:3328



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
0060.5CA3.EE01
IP Configuration
IPv4 Address
20.1.1.1
Subnet Mask
255.0.0.0
Tx Ring Limit
10
Equivalent IOS Commands
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 20.1.1.1 255.0.0.0
Router(config-if)#ip address 20.1.1.1 255.0.0.0
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)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#
Top


Wireless Router0
Physical
Config
GUI
Attributes
GLOBAL
Settings
Algorithm Settings
INTERFACE
Internet
LAN
Wireless
Internet Settings
IP Configuration
DHCP
Static
PPPoE
UserName
Password
IPv4 Address
20.1.1.3
Subnet Mask
255.0.0.0
Default Gateway
20.1.1.1
DNS Server


Smartphone0
Physical
Config
Desktop
Programming
Attributes
Command Prompt
X
Cisco Packet Tracer PC Command Line 1.0
C:\>tracert 20.1.1.1
Tracing route to 20.1.1.1 over a maximum of 30 hops:
1 15 ms 15 ms 6 ms 192.168.0.1
2 * 14 ms 13 ms 20.1.1.1
Trace complete.
C:\>ping 20.1.1.1
Pinging 20.1.1.1 with 32 bytes of data:
Reply from 20.1.1.1: bytes=32 time=12ms TTL=254
Reply from 20.1.1.1: bytes=32 time=12ms TTL=254
Reply from 20.1.1.1: bytes=32 time=13ms TTL=254
Reply from 20.1.1.1: bytes=32 time=12ms TTL=254
Ping statistics for 20.1.1.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 12ms, Maximum = 13ms, Average = 12ms
C:\>