SIES College of Arts, Science & Commerce

RISE WITH EDUCATION

NAAC REACCREDITED - 'A' GRADE

ISO 9001 : 2008

S.I.E.S College of Arts, Science and Commerce(Autonomous)
Sion(W), Mumbai – 400 022.

CERTIFICATE

This is to certify that **Miss.Kimaya Naik** Roll No. **TCS2324048** has successfully completed the necessary course of experiments in the subject of **Wireless Sensor Networks & Mobile Communication** during the academic year **2023 – 2024** complying with the requirements of University of Mumbai, for the course of **TYBSc Computer Science [Semester-VI].**

Prof. In-Charge
JESICA D'CRUZ

Examination date:
Examiner's Signature & Date:

Head of the Department
Prof. Manoj Singh

College Seal

Index Page

Wireless Sensor Networks & Mobile Communication

Practical No.1

DEPARTMENT OF COMPUTER SCIENCE

Name:	Kimaya Naik	**Roll Number**	TCS2324048
Paper Code:	SIUSCS61	**Class**	TYBSc(Computer Science)
Topic:	Sensor Node Hardware	**Batch**	I
Date:	08/1/24	**Practical No**	1

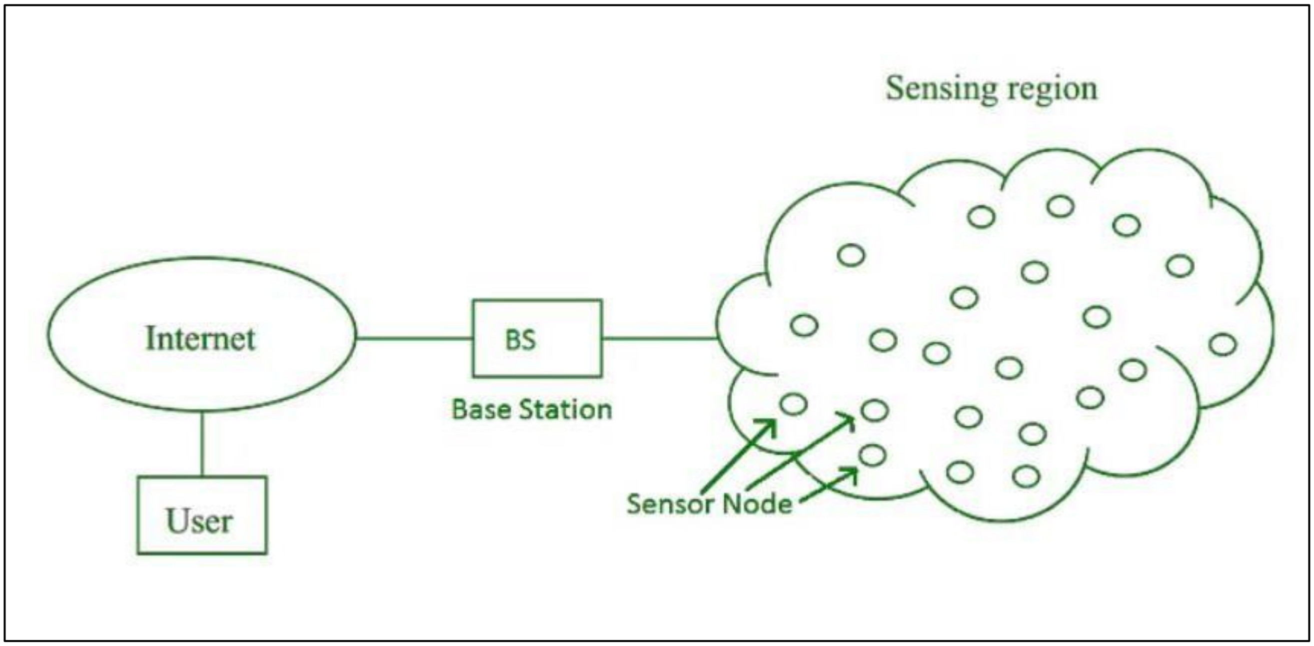
A) AIM: Understanding the Sensor Node Hardware

B) DESCRIPTION:

Wireless Sensor Network (WSN) is an infrastructure-less wireless network that is deployed in a large number of wireless sensors in an ad-hoc manner that is used to monitor the system, physical or environmental conditions.

Sensor nodes are used in WSN with the onboard processor that manages and monitors the environment in a particular area. They are connected to the Base Station which acts as a processing unit in the WSN System.
Base Station in a WSN System is connected through the Internet to share data.

C) NETWORK TOPOLOGY, CONFIGURATIONS AND OUTPUT:



Hardware of WSN:

1. **Sensors:**
   Sensors in WSN are used to capture the environmental variables and which is used for data acquisition. Sensor signals are converted into electrical signals.
   These components are responsible for data acquisition, i.e., they collect environmental data (variables) and convert it into electrical signals through a process known as transduction. In lay terms, sensors are tiny electric noses, ears, and fingers that 'feel' the environment and tell a computer what it 'senses' in a language it understands.

2. **Sensor Nodes:**
   A sensor node is a combination of different subunits and all they help to perform the functionality of the sensor node. Different units help to sense, record, monitor and analyze the data which is collected from physical conditions .Even though the name, a Sensor Node comprises not only the sensing component but also other important characteristics like processing of recorded data, communication with servers and storage units to store recorded data. With all these characteristics, components and enrichments, a Sensor Node takes responsibility for data collection, data correlation, and fusion of data from other sensors with its own data and network analysis.

3. **Radio Nodes**:

These components are equipped with a microcontroller for data processing, a transceiver for wireless communication, external memory for data storage, and a power source to remain operational. They receive the sensor's electrical signals and send this data to the WLAN access point.

4. **Wireless Access Point:**

A Wireless Access Point (WAP) is a networking device that allows connecting the devices with the wired network. A Wireless Access Point (WAP) is used to create the WLAN (Wireless Local Area Network), it is commonly used in large offices and buildings which have expanded businesses.

It is easier and simpler to understand and implant the device. It can be fixed, mobile or hybrid proliferated in the 21st century. The availability, confidentiality, and integrity of the communication and network are a responsibility and to be ensured about that.

A wireless AP connects the wired networks to the wireless client. It eases access to the network for mobile users which increases productivity and reduces the infrastructure cost.

5. **WLAN Access Point:**
   It receives the data which is sent by the Radio nodes wirelessly, generally through the internet.

6. **Evaluation Software:**
   The data received by the WLAN Access Point is processed by a software called as Evaluation Software for presenting the report to the users for further processing of the data which can be used for processing, analysis, storage, and mining of the data.

7. **Base Station:**
   A base station serves as a central connection point for a wireless device to communicate. It further connects the device to other networks or devices, usually through dedicated high bandwidth wire or fiber optic connections.
   The base station sends commands to the sensor nodes and the sensor node perform the task by collaborating with each other. After collecting the necessary data, the sensor nodes send the data back to the base station.
   A base station also acts as a gateway to other networks through the internet. After receiving the data from the sensor nodes, a base station performs simple data processing and sends the updated information to the user using internet.

   If each sensor node is connected to the base station, it is known as Single-hop network architecture. Although long distance transmission is possible, the energy consumption for communication will be significantly higher than data collection and computation.

8. **Graphical User Interface:**
A graphical user interface (GUI) is a digital interface in which a user interacts with graphical components such as icons, buttons, and menus.

9. **Actuators:**
Actuators allow a WSN node to influence its environment, providing a feedback channel through which its decisions can be enacted. Throughout the history of computing, there has been a trend for the ratio of processing elements to people to increase, resulting in the creation and popularization of new usage paradigms.

Wireless Sensor Networks & Mobile Communication

Practical No.2

DEPARTMENT OF COMPUTER SCIENCE

Name:	Kimaya Naik	**Roll Number**	TCS2324048
Paper Code:	SIUSCS61	**Class**	TYBSc(Computer Science)
Topic:	TinyOs and Tossim	**Batch**	I
Date:	08/1/24	**Practical No**	2

A) AIM:

1) Exploring and understanding TinyOS computational concepts: Events, Commands and Task

- nesC model

- nesC Components

2) Understanding TOSSIM for

- Mote-mote radio communication

- Mote-PC serial communication

B) DESCRIPTION:

TinyOS is an embedded, component-based operating system and platform for low-power wireless devices, such as those used in wireless sensor networks (WSNs), smartdust, ubiquitous computing, personal area networks, building automation, and smart meters. It is written in the programming language nesC, as a set of cooperating tasks and processes.

TOSSIM is a discrete event emulator for the execution of nesC model on TinyOS-Mica hardware [1]. In TOSSIM, an event is generated for each transmitted or received bit or packet.

C) NETWORK TOPOLOGY, CONFIGURATIONS AND OUTPUT:

A) Understanding TinyOS computational concepts: Events, Commands, Task, nesC components and model

A TinyOS program is a graph of components, each of which is an independent computational entity that exposes one or more interfaces. Components have three computational abstractions: commands, events, and tasks. Commands and events are mechanisms for inter-component communication, while tasks are used to express intra-component concurrency

Events:

An event in TinyOS is essentially a named placeholder for a function. It represents an occurrence or trigger that can be handled by the system. In TinyOS, events are a fundamental concept used for asynchronous programming. They provide a mechanism for handling and responding to external stimuli, such as sensor readings or incoming messages, in a non-blocking manner. Events are central to the event-driven programming paradigm in TinyOS, and they play a crucial role in managing the flow of execution in the system.

Commands:

A command is a named function or operation that a component provides to other components. It defines a specific action that can be requested by a client component. In TinyOS, commands are a mechanism for providing a standardized interface for components to interact with each other. Commands allow one component to request specific actions or services from another component in a well-defined manner. They are part of the component-based programming model used in TinyOS.

Tasks:

A task in TinyOS is a named function or operation that represents a unit of work to be performed. Tasks are used to structure the execution flow of the program and handle concurrent activities. A task in TinyOS is a named function or operation that represents a unit of work to be performed. Tasks are used to structure the execution flow of the program and handle concurrent activities.

NesC Components And Nesc Model:

In TinyOS, components are a fundamental building block of the programming model. Components are modular units of code that encapsulate functionality, providing a way to structure and organize the software. Components are written in the nesC (pronounced "nes-C") programming language, which is specifically designed for developing embedded systems and wireless sensor networks. Here are key points about nesC components in TinyOS:

1. **Definition:** A nesC component is a self-contained unit of code that encapsulates a specific functionality or service. Components in TinyOS are designed to be modular and composable, promoting code reuse and maintainability.
2. **Interface-Implementation Separation:** Components in TinyOS follow the interface-implementation separation principle. An interface declares a set of commands, events, and functions that a component provides or expects from other components. The implementation contains the actual code that realizes the behavior defined in the interface.
3. **Interfaces:** Interfaces define the communication contract between components. They specify a set of commands, events, and functions that components can use to interact with each other. Interfaces allow for the decoupling of components and enable the development of interchangeable and interoperable software modules.
4. **Component Wiring:** Components can be connected or "wired" together to form an application. Wiring is the process of specifying how components are connected, determining how they communicate and collaborate. This wiring is often done in the configuration file of a TinyOS application.
5. **Configuration:** Components can be parameterized and configured, allowing for flexibility in adapting their behavior to different requirements. Configuration settings can be adjusted at compile time, providing a way to customize the behavior of the software.

B) Understanding TOSSIM for

- **Mote-mote radio communication:**

Mote to mote communication is the radio communication in Tiny os. This introduces us the interfaces and components in Tiny os which supports the radio communication. And also we learn the basics how to use the message_ t that is a message buffer which is used to send the message buffer to the radio and receives the message buffer from the radio. Tiny os provides us with the interfaces and the components.

Interfaces are used to consider the existing communication services and the components are used to implement the interfaces. These components and interfaces use a message buffer called message_ t that is implemented as nesC structure. This message buffer message_ t was used as TOS_ Msg in the first version of Tiny os and in the latest version of it has been replaced as message_ t. In the first version of the tiny os the message buffers were accessed directly but in the latest version they cannot be accessed directly instead this function can be read and written in the form of mutator and accessor functions.

- **Mote-PC serial communication**

Mote-to-PC radio communication is a crucial aspect of wireless sensor networks (WSNs), facilitating the transfer of data between sensor nodes (motes) deployed in the field and a central personal computer (PC). This communication enables the collection, monitoring, and analysis of data generated by the sensor nodes. Here's a brief overview:

1. **Hardware Setup:**
    - Motes are equipped with radio transceivers for wireless communication, often following standards like IEEE 802.15.4 or Zigbee.
    - The PC may have a compatible radio interface or a dedicated gateway device to communicate with the sensor nodes.
2. **Communication Protocols:**
    - Motes and the PC use compatible communication protocols, defining how data is formatted, transmitted, and received.
    - Protocols may include addressing, synchronization, and error-handling mechanisms.
3. **Data Transmission:**
    - Motes periodically sense the environment, collect data, and transmit it wirelessly to the PC or through an intermediary gateway.
    - Data may include sensor readings, status information, or event notifications.
4. **Gateway Device:**
    - A gateway device may be employed to interface between the wireless sensor network and the PC. It receives data from sensor nodes, processes or aggregates it, and forwards it to the PC.

5. **PC Software:**
    - The PC runs software responsible for receiving, parsing, and processing data from the sensor nodes.
    - Communication drivers, middleware, and visualization components are commonly part of the software stack.
6. **Security Measures:**
    - Security measures, such as encryption and authentication, are often implemented to protect the communication between sensor nodes and the PC, ensuring data integrity and preventing unauthorized access.
7. **Data Analysis and Visualization:**
    - Once received by the PC, the data can be analyzed, visualized, and used for decision-making.
    - Visualization tools, databases, and analysis software help make sense of the collected data.
8. **Application Examples:**
    - Mote-to-PC communication is used in various applications, including environmental monitoring, industrial automation, healthcare, and smart agriculture.
    - It enables real-time tracking, control, and management of distributed sensor nodes.
9. **Energy Considerations:**
    - Energy-efficient communication strategies, such as duty cycling and data aggregation, are often employed in mote-to-PC communication to maximize the lifespan of sensor nodes.

In summary, mote-to-PC radio communication is a foundational element in WSNs, enabling the seamless exchange of data between sensor nodes and central processing units for analysis, monitoring, and decision-making in diverse applications.

SIES	Wireless Sensor Networks & Mobile Communication Practical No. 3A

DEPARTMENT OF COMPUTER SCIENCE

Name:	Kimaya Naik	**Roll Number**	TCS2324048
Paper Code:	SIUSCS61	**Class**	TYBSc(Computer Science)
Topic:	Smoke detection and fire prevention system	**Batch**	I
Date:	08/1/24	Practical No	3A

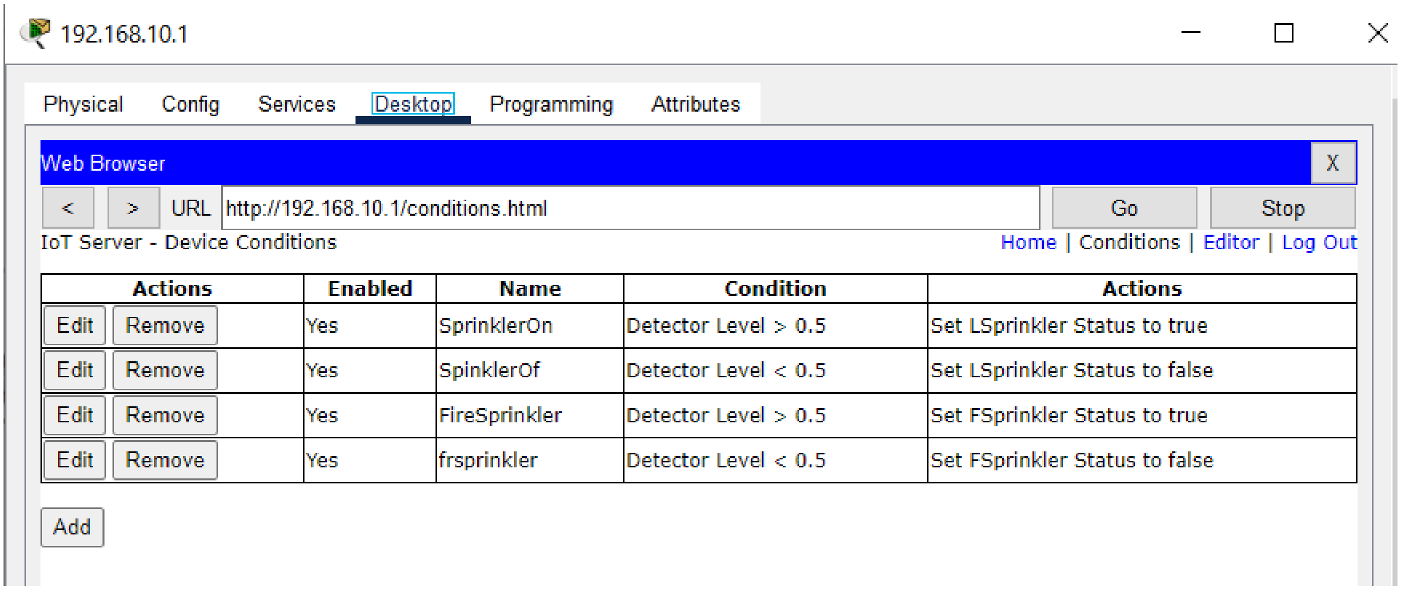
A) AIM: Design Smoke detection and fire prevention system using cisco packet tracer

B) DESCRIPTION:

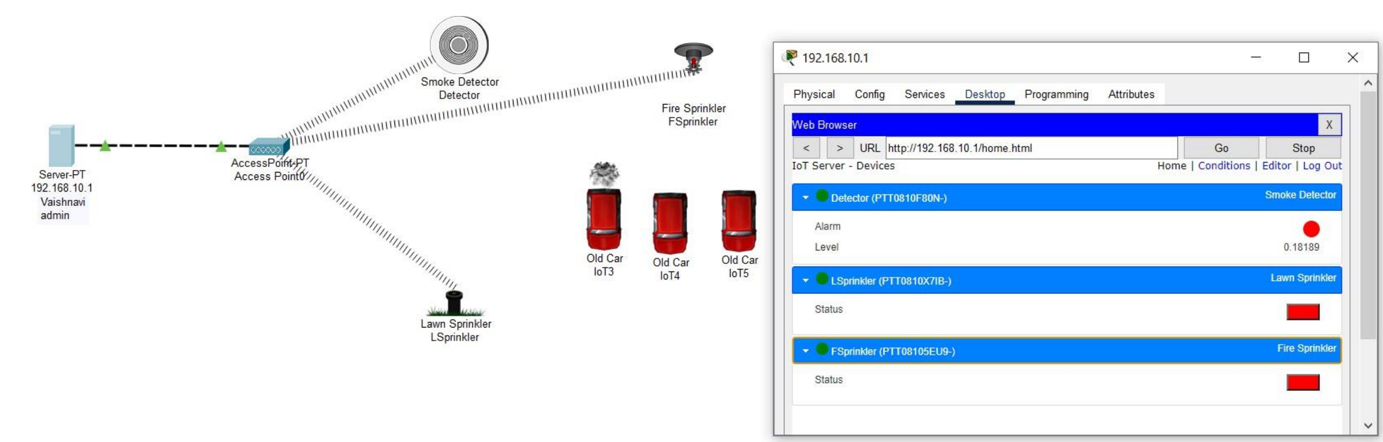
Sensor is a device that is used to gather information about a physical process or a physical phenomenon and translate it into electrical signals that can be processed, measured and analysed. The term physical process Sensor can be any real-world information like temperature, pressure, light, sound, motion, position, flow, humidity, radiation etc. **Here, Smoke Detector is a sensor.**

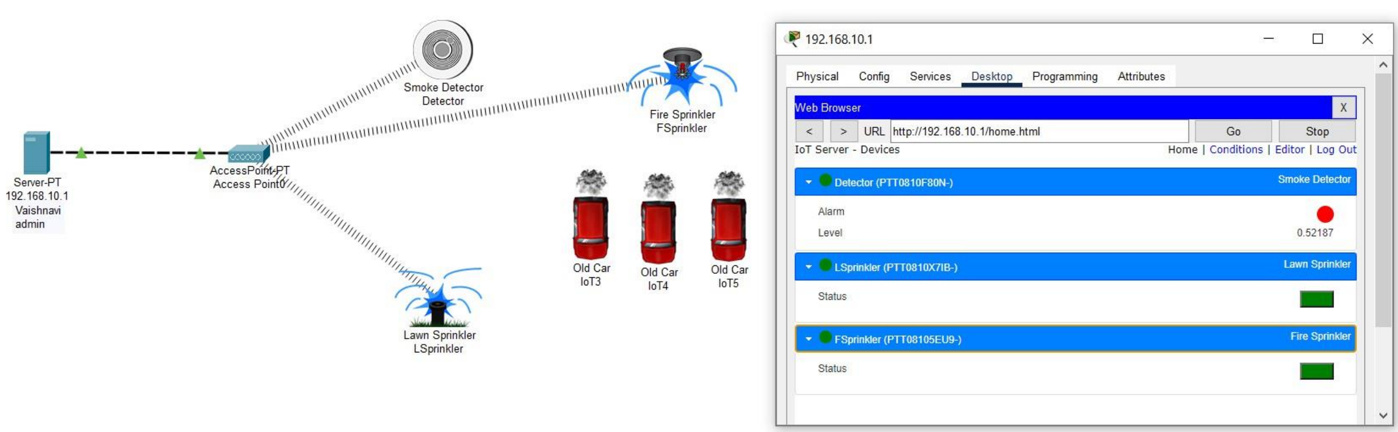
Actuator is a device that converts the electrical signals into the physical events or characteristics. It takes the input from the system and gives output to the environment. **Here, FireSprinkler and LawnSprinkler is the Actuator.**

C) CODE AND OUTPUT:

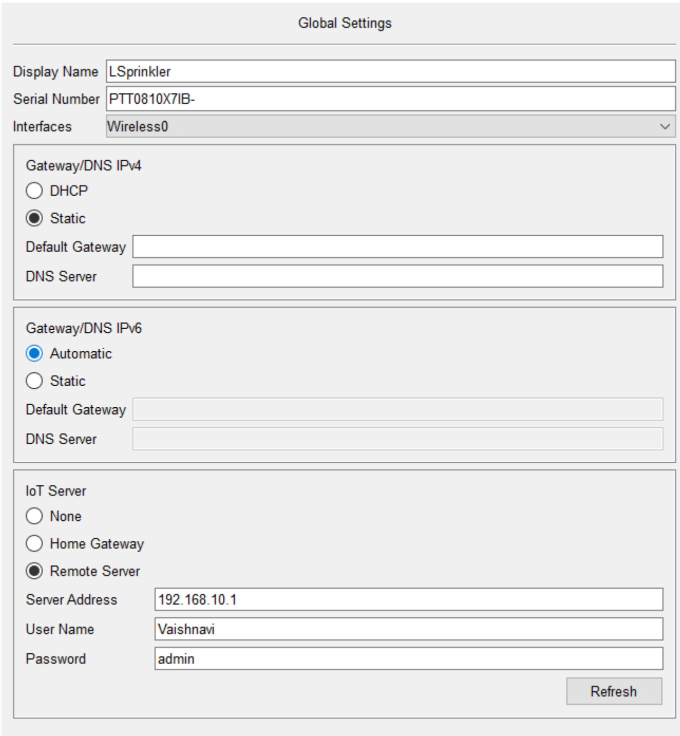


Actions	Enabled	Name	Condition	Actions
Edit Remove	Yes	SprinklerOn	Detector Level > 0.5	Set LSprinkler Status to true
Edit Remove	Yes	SpinklerOf	Detector Level < 0.5	Set LSprinkler Status to false
Edit Remove	Yes	FireSprinkler	Detector Level > 0.5	Set FSprinkler Status to true
Edit Remove	Yes	frsprinkler	Detector Level < 0.5	Set FSprinkler Status to false

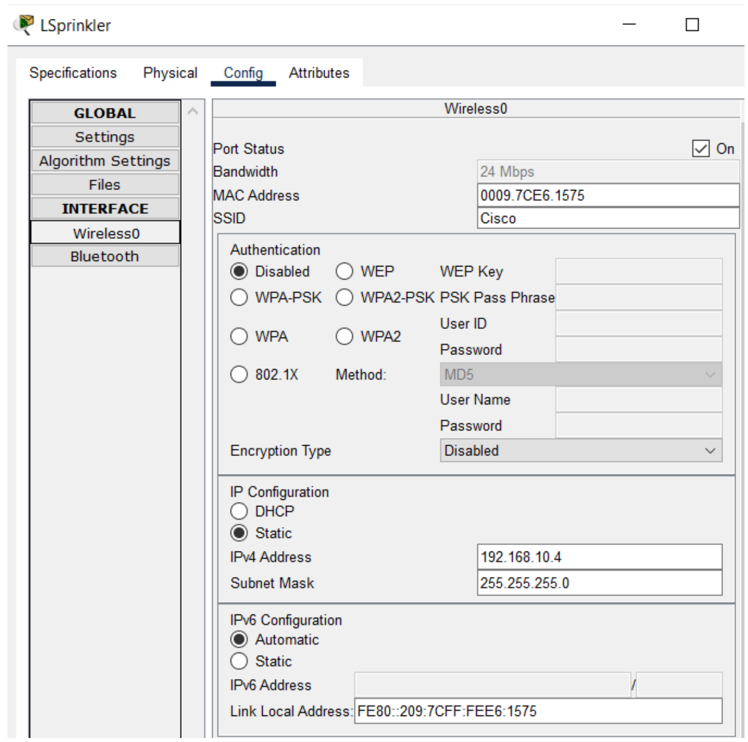




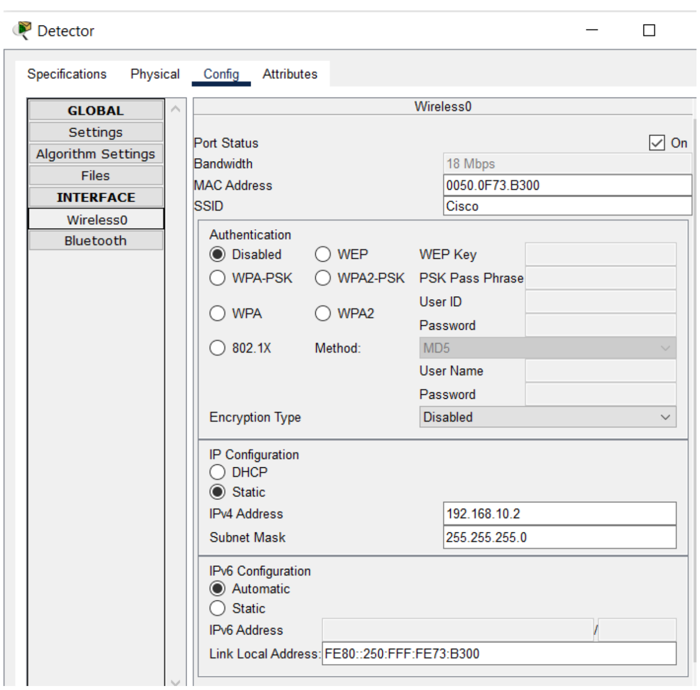
Global settings same for all



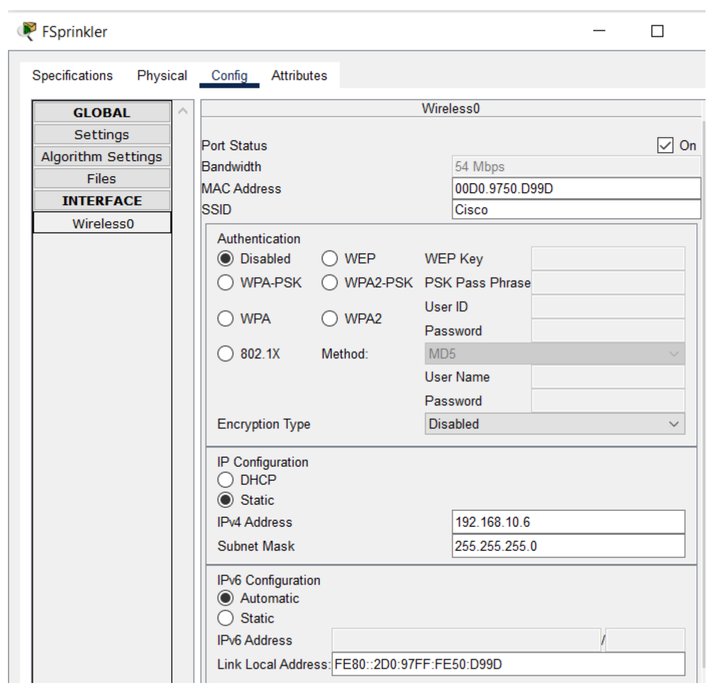
LSprinkler



Detector :



FSprinkler



Wireless Sensor Networks & Mobile Communication

Practical No. 3B

DEPARTMENT OF COMPUTER SCIENCE

Name:	Kimaya Naik	**Roll Number**	TCS2324048
Paper Code:	SIUSCS61	**Class**	TYBSc(Computer Science)
Topic:	Smart Garden System	**Batch**	I
Date:	15/1/24	Practical No	3A

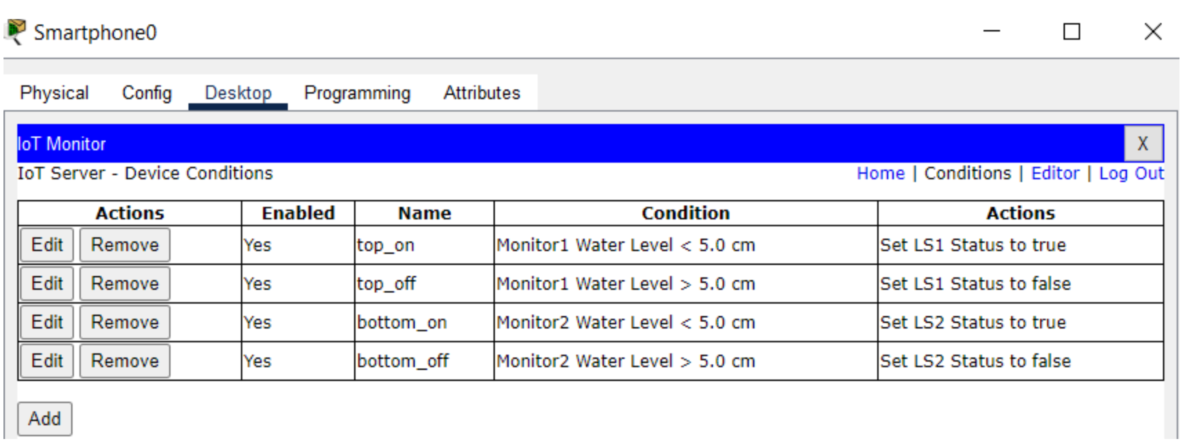
A) AIM: Design Smart Garden system system using cisco packet tracer

B) DESCRIPTION:

Sensor is a device that is used to gather information about a physical process or a physical phenomenon and translate it into electrical signals that can be processed, measured and analysed. The term physical process Sensor can be any real-world information like temperature, pressure, light, sound, motion, position, flow, humidity, radiation etc. **Here, Water Level Monitor is a sensor.**

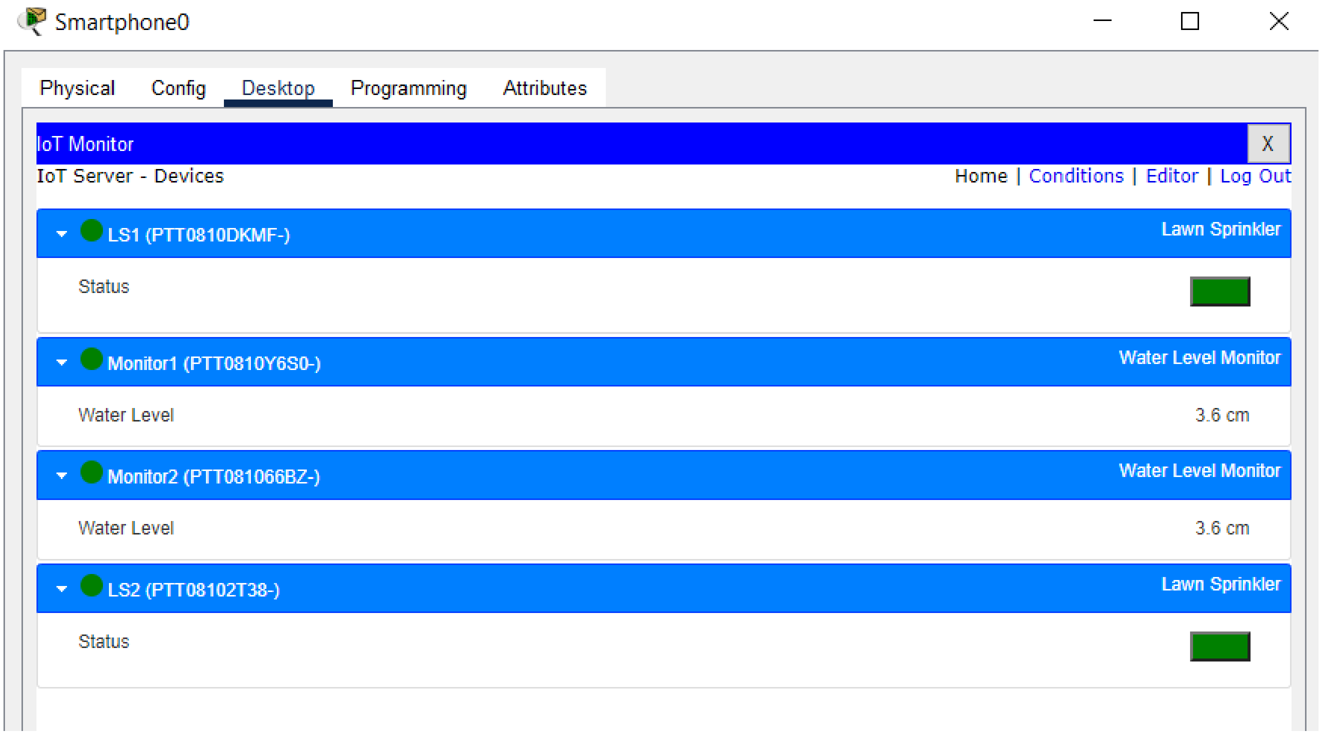
Actuator is a device that converts the electrical signals into the physical events or characteristics. It takes the input from the system and gives output to the environment. **Here, LawnSprinkler is the Actuator.**

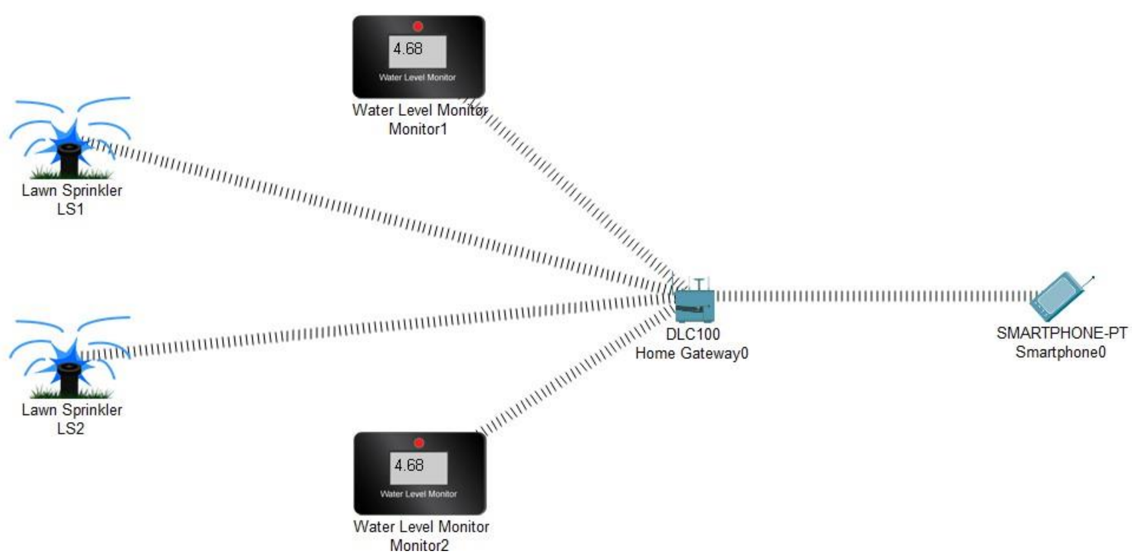
C) CODE AND OUTPUT:



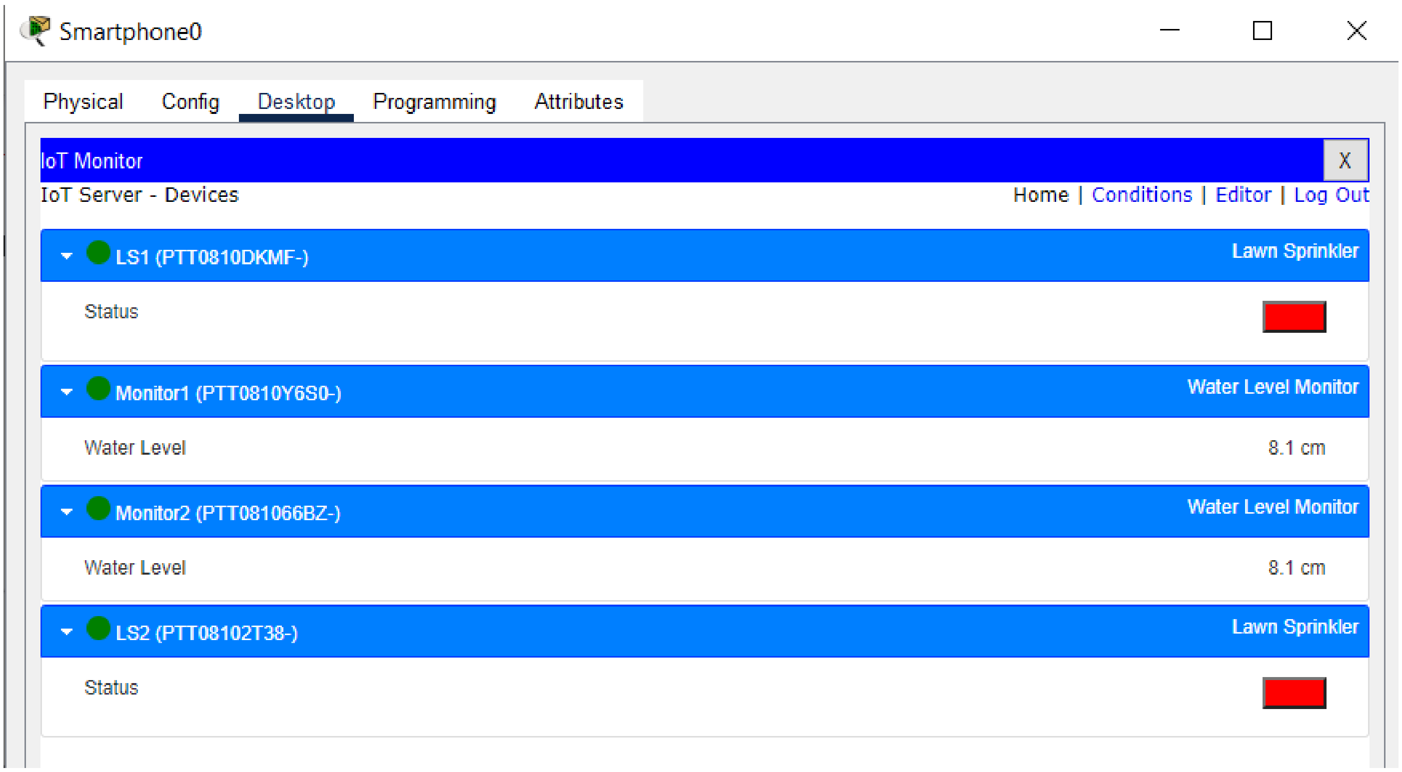
Actions	Enabled	Name	Condition	Actions
Edit Remove	Yes	top_on	Monitor1 Water Level < 5.0 cm	Set LS1 Status to true
Edit Remove	Yes	top_off	Monitor1 Water Level > 5.0 cm	Set LS1 Status to false
Edit Remove	Yes	bottom_on	Monitor2 Water Level < 5.0 cm	Set LS2 Status to true
Edit Remove	Yes	bottom_off	Monitor2 Water Level > 5.0 cm	Set LS2 Status to false

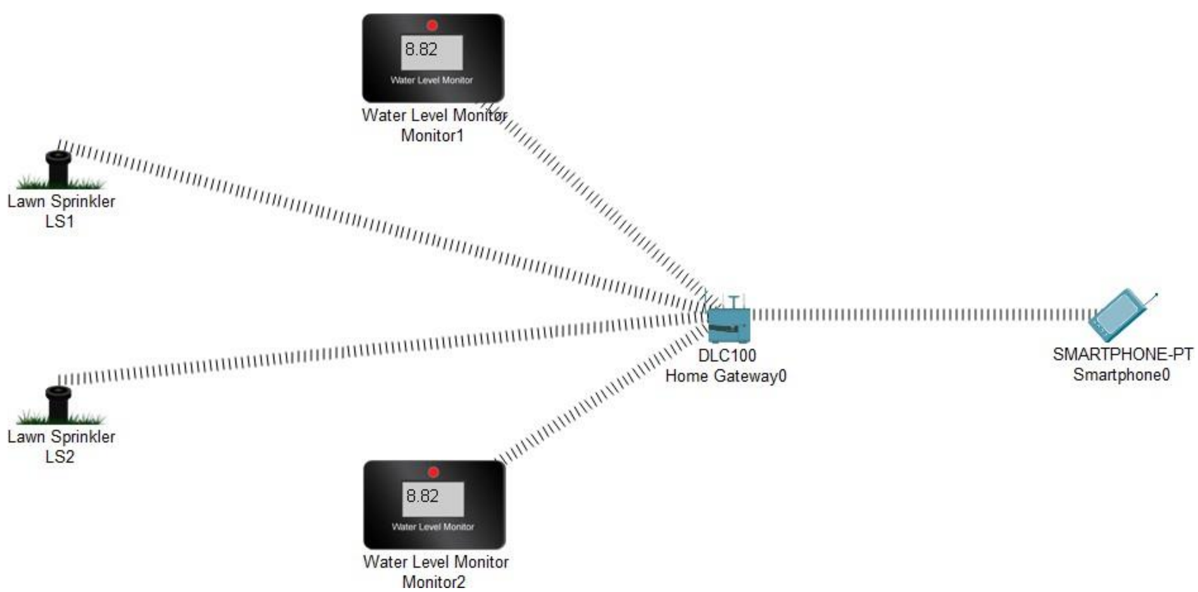
i) When water level goes below 5 cm : (LawnSprinkler is On)

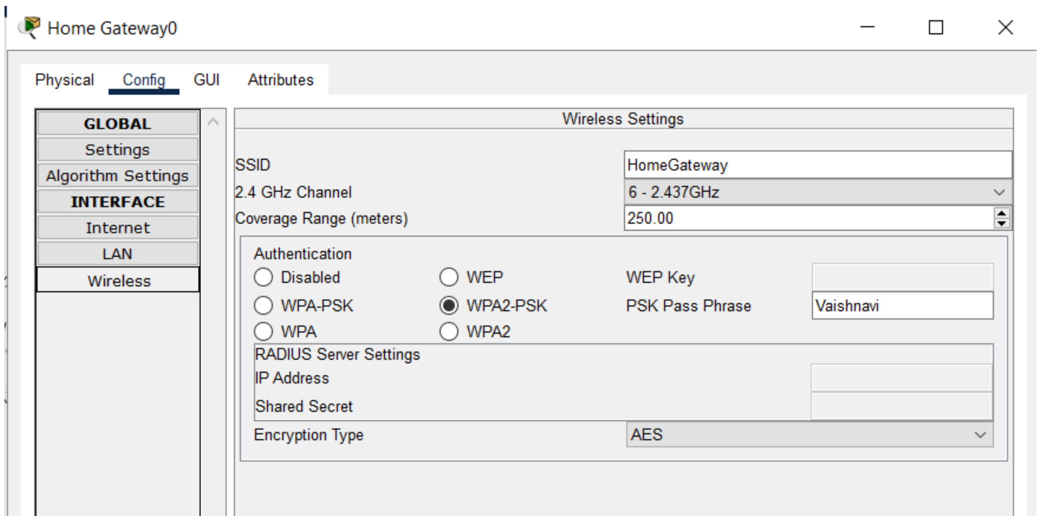


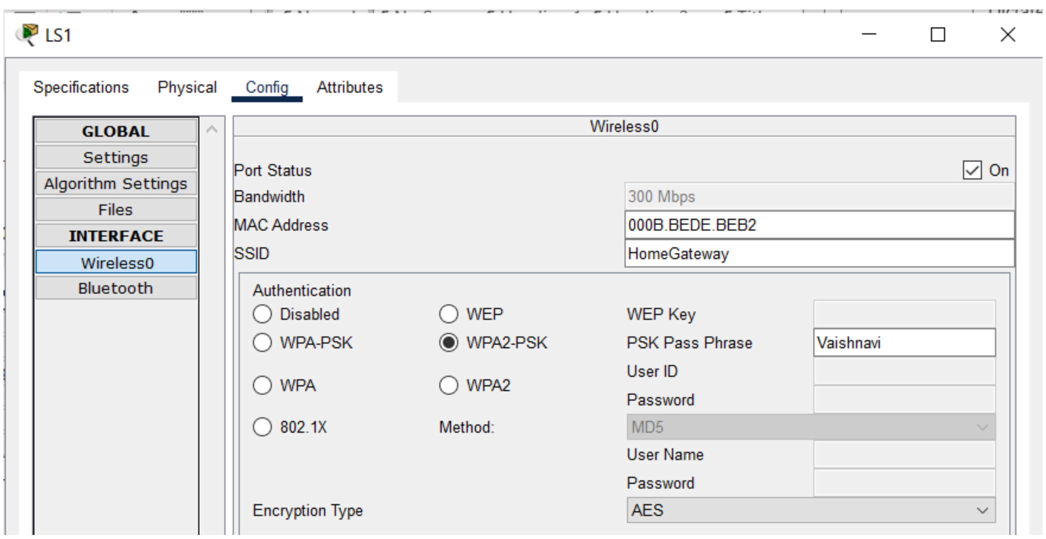


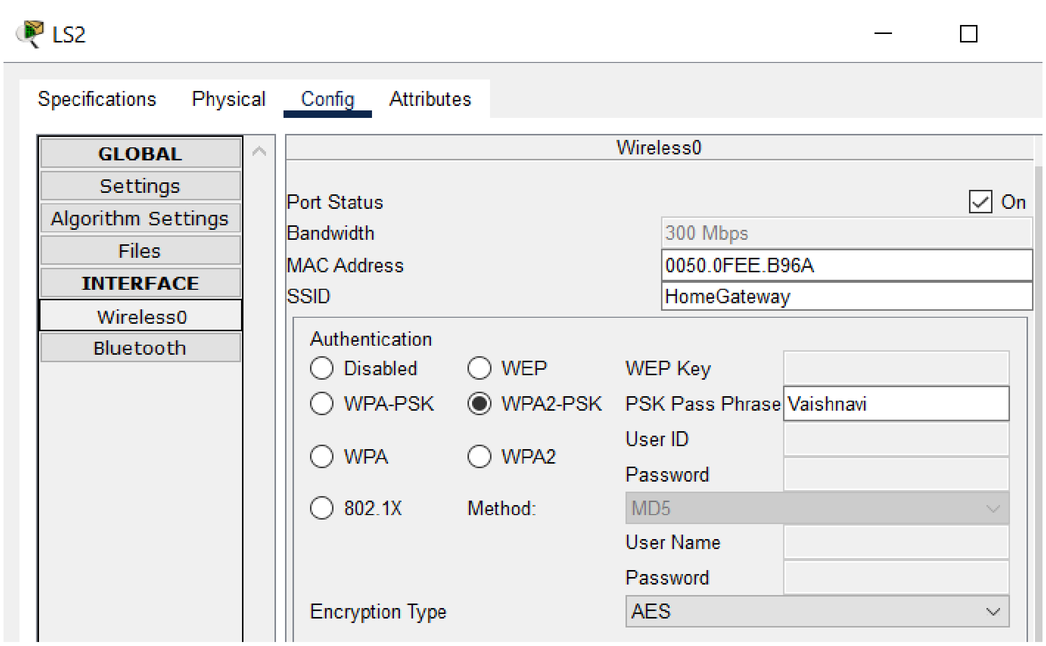
ii) when water level goes above 5cm : (LawnSprinkler is Off)

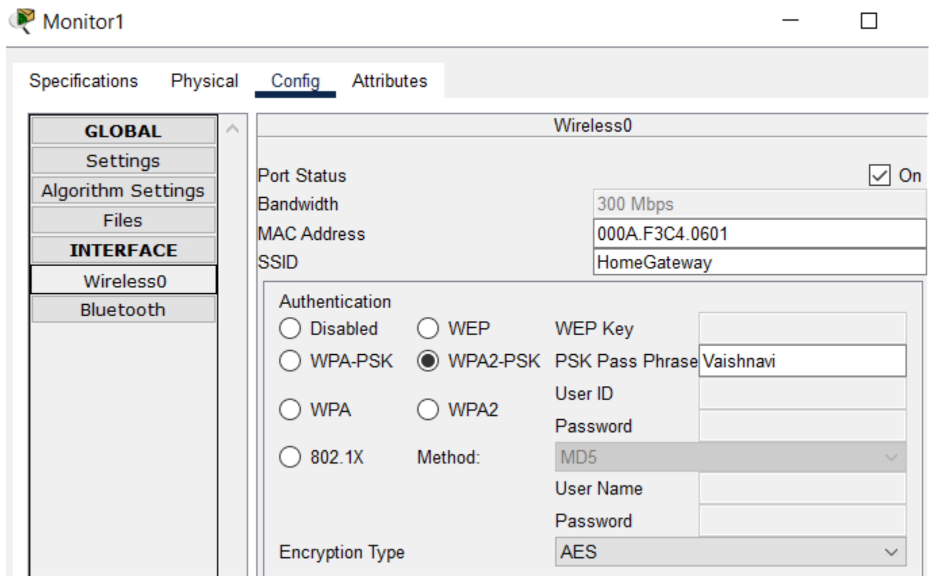


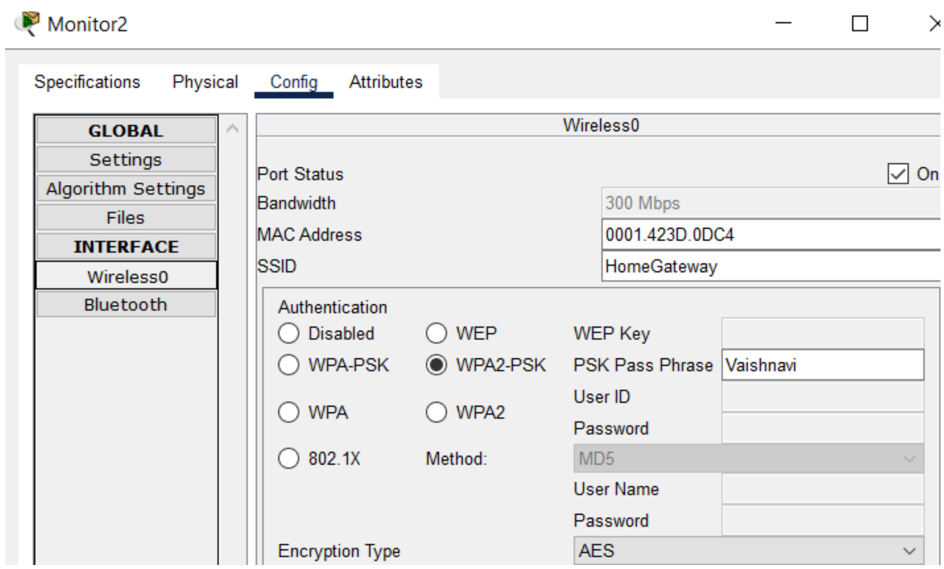


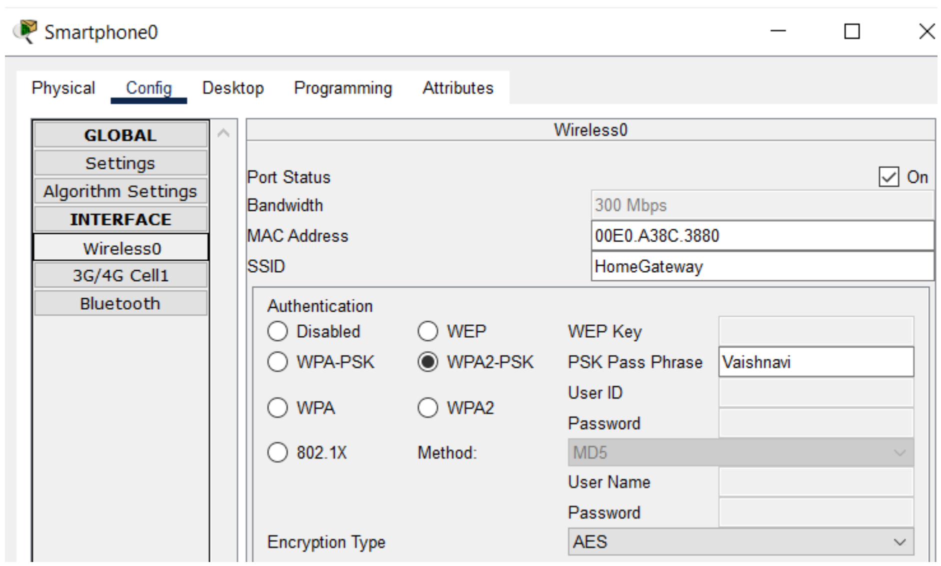

Home Gateway0
Physical
Config
GUI
Attributes
GLOBAL
Settings
Algorithm Settings
INTERFACE
Internet
LAN
Wireless
Wireless Settings
SSID
HomeGateway
2.4 GHz Channel
6 - 2.437GHz
Coverage Range (meters)
250.00
Authentication
Disabled
WEP
WEP Key
WPA-PSK
WPA2-PSK
PSK Pass Phrase
Vaishnavi
WPA
WPA2
RADIUS Server Settings
IP Address
Shared Secret
Encryption Type
AES


LS1
Specifications
Physical
Config
Attributes
GLOBAL
Settings
Algorithm Settings
Files
INTERFACE
Wireless0
Bluetooth
Wireless0
Port Status
On
Bandwidth
300 Mbps
MAC Address
000B.BEDE.BEB2
SSID
HomeGateway
Authentication
Disabled
WEP
WEP Key
WPA-PSK
WPA2-PSK
PSK Pass Phrase
Vaishnavi
User ID
WPA
WPA2
Password
802.1X
Method:
MD5
User Name
Password
Encryption Type
AES


LS2
Specifications
Physical
Config
Attributes
GLOBAL
Settings
Algorithm Settings
Files
INTERFACE
Wireless0
Bluetooth
Wireless0
Port Status
On
Bandwidth
300 Mbps
MAC Address
0050.0FEE.B96A
SSID
HomeGateway
Authentication
Disabled
WEP
WEP Key
WPA-PSK
WPA2-PSK
PSK Pass Phrase
Vaishnavi
User ID
WPA
WPA2
Password
802.1X
Method:
MD5
User Name
Password
Encryption Type
AES


Monitor1
Specifications
Physical
Config
Attributes
GLOBAL
Settings
Algorithm Settings
Files
INTERFACE
Wireless0
Bluetooth
Wireless0
Port Status
On
Bandwidth
300 Mbps
MAC Address
000A.F3C4.0601
SSID
HomeGateway
Authentication
Disabled
WEP
WEP Key
WPA-PSK
WPA2-PSK
PSK Pass Phrase
Vaishnavi
User ID
WPA
WPA2
Password
802.1X
Method:
MD5
User Name
Password
Encryption Type
AES


Monitor2
Specifications
Physical
Config
Attributes
GLOBAL
Settings
Algorithm Settings
Files
INTERFACE
Wireless0
Bluetooth
Wireless0
Port Status
On
Bandwidth
300 Mbps
MAC Address
0001.423D.0DC4
SSID
HomeGateway
Authentication
Disabled
WEP
WEP Key
WPA-PSK
WPA2-PSK
PSK Pass Phrase
Vaishnavi
User ID
WPA
WPA2
Password
802.1X
Method:
MD5
User Name
Password
Encryption Type
AES


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
00E0.A38C.3880
SSID
HomeGateway
Authentication
Disabled
WEP
WEP Key
WPA-PSK
WPA2-PSK
PSK Pass Phrase
Vaishnavi
User ID
WPA
WPA2
Password
802.1X
Method:
MD5
User Name
Password
Encryption Type
AES

Wireless Sensor Networks & Mobile Communication

Practical No. 4

DEPARTMENT OF COMPUTER SCIENCE

Name:	Kimaya Naik	**Roll Number**	TCS2324048
Paper Code:	SIUSCS61	**Class**	TYBSc(Computer Science)
Topic:	Simple ADHOC Network	**Batch**	I
Date:	05/2/24	Practical No	4

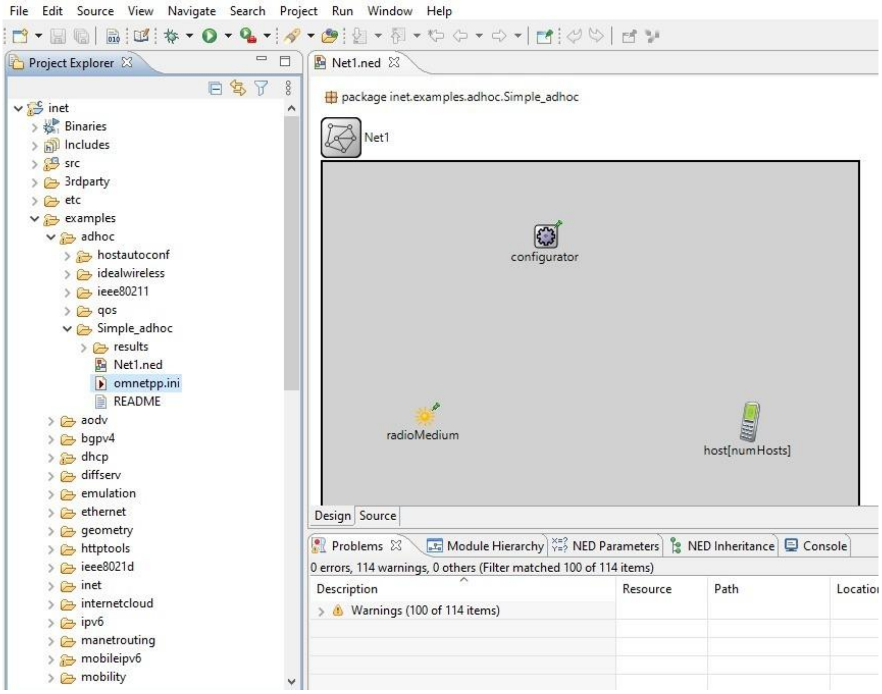
A) AIM: Create and simulate A SIMPLE ADHOC Network (OMNET)

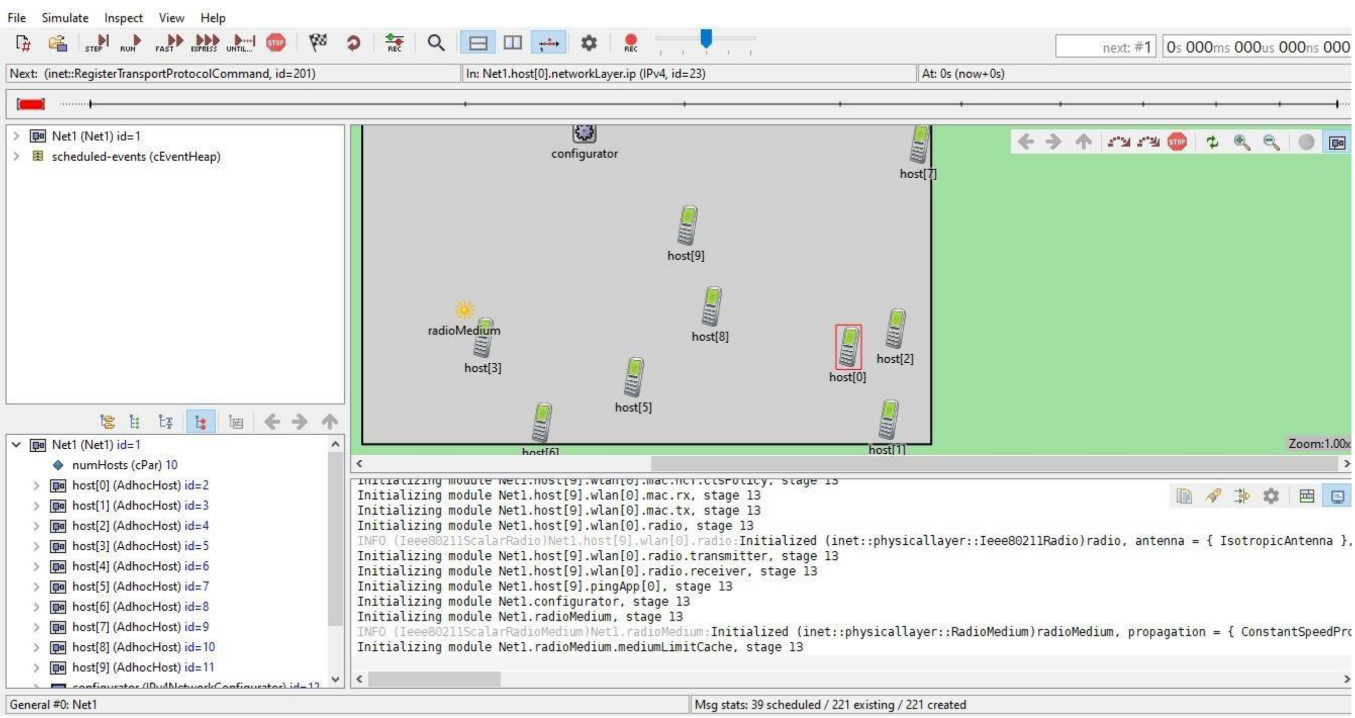
B) DESCRIPTION:

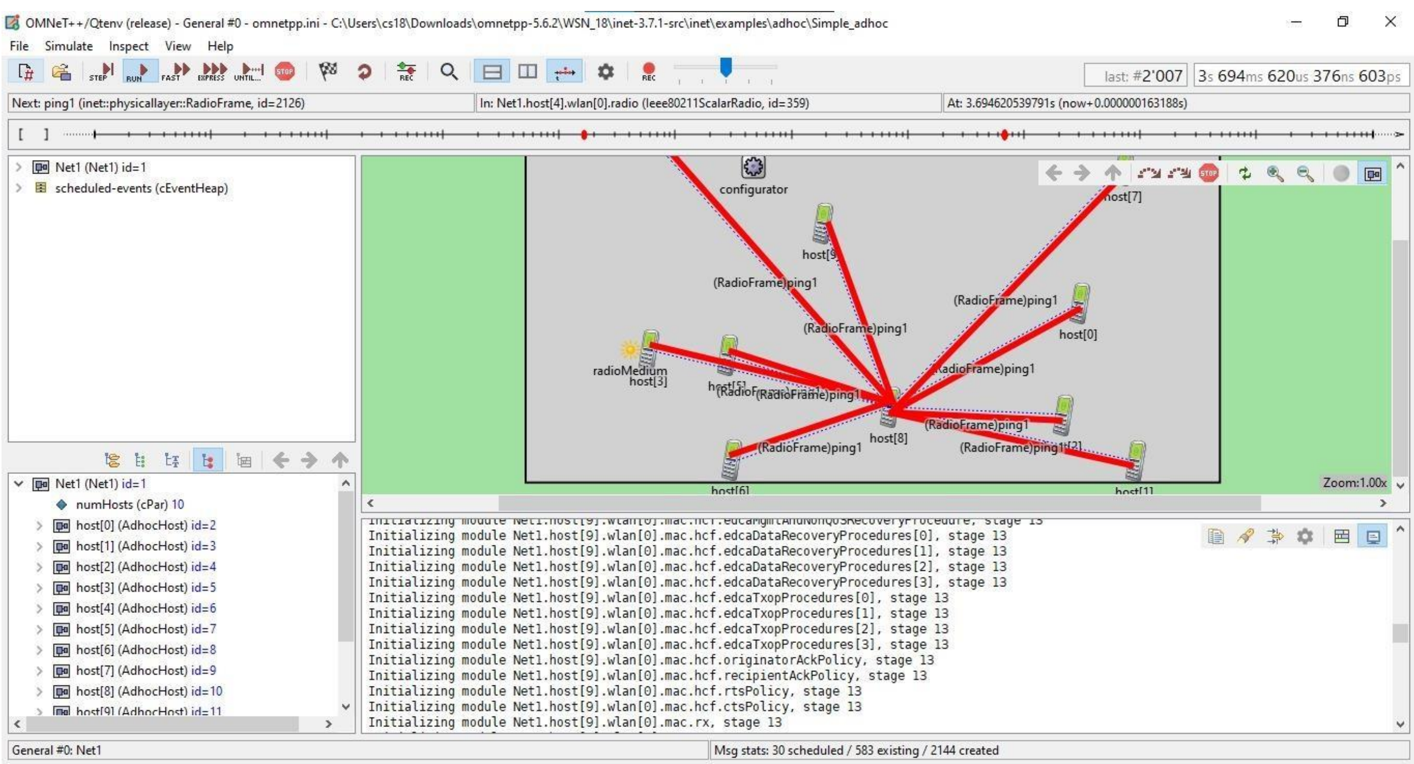
- An ad hoc network is a temporary network created between two devices without utilizing any other networking infrastructure. These networks exist for a single session, often for a specific purpose like transferring files or sharing an Internet connection. Ad hoc networks do not require a router or wireless access point.

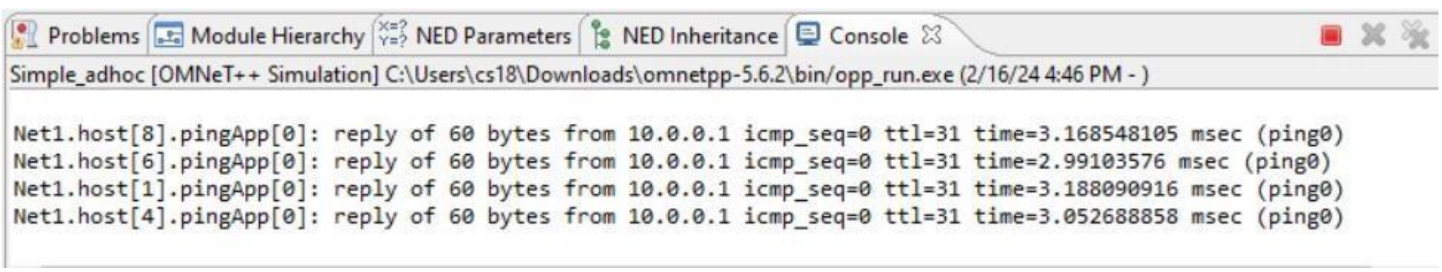
Code And Output :

```
package inet.examples.adhoc.Simple_adhoc;

// numOfHosts: 10
// parametric: true
// static:     false

import inet.networklayer.configurator.ipv4.IPv4NetworkConfigurator;
import inet.node.inet.AdhocHost;
import inet.physicallayer.ieee80211.packetlevel.Ieee80211ScalarRadioMedium;

network Net1
{
    parameters:
        int numHosts;
    submodules:
        host[numHosts]: AdhocHost {
            parameters:
                @display("r=,,#707070");
        }

        configurator: IPv4NetworkConfigurator {
            @display("p=219,73");
        }
        radioMedium: Ieee80211ScalarRadioMedium {
            parameters:
                @display("p=100,250");
        }
}
```


File Edit Source View Navigate Search Project Run Window Help
Project Explorer
Net1.ned
inet
Binaries
Includes
src
3rdparty
etc
examples
adhoc
hostautoconf
idealwireless
ieee80211
qos
Simple_adhoc
results
Net1.ned
omnetpp.ini
README
aodv
bgpv4
dhcp
diffserv
emulation
ethernet
geometry
httptools
ieee8021d
inet
internetcloud
ipv6
manetrouting
mobileipv6
mobility
package inet.examples.adhoc.Simple_adhoc
Net1
configurator
radioMedium
host[numHosts]
Design Source
Problems
Module Hierarchy
NED Parameters
NED Inheritance
Console
0 errors, 114 warnings, 0 others (Filter matched 100 of 114 items)
Description
Resource
Path
Location
Warnings (100 of 114 items)


File Simulate Inspect View Help
next: #1
Next: (inet::RegisterTransportProtocolCommand, id=201)
In: Net1.host[0].networkLayer.ip (IPv4, id=23)
At: 0s (now+0s)
Net1 (Net1) id=1
scheduled-events (cEventHeap)
configurator
host[7]
host[9]
radioMedium
host[8]
host[2]
host[3]
host[0]
host[5]
host[6]
host[1]
Zoom:1.00x
numHosts (cPar) 10
host[0] (AdhocHost) id=2
host[1] (AdhocHost) id=3
host[2] (AdhocHost) id=4
host[3] (AdhocHost) id=5
host[4] (AdhocHost) id=6
host[5] (AdhocHost) id=7
host[6] (AdhocHost) id=8
host[7] (AdhocHost) id=9
host[8] (AdhocHost) id=10
host[9] (AdhocHost) id=11
Initializing module Net1.host[9].wlan[0].mac.rx, stage 13
Initializing module Net1.host[9].wlan[0].mac.tx, stage 13
Initializing module Net1.host[9].wlan[0].radio, stage 13
INFO (Ieee80211ScalarRadio)Net1.host[9].wlan[0].radio:Initialized (inet::physicallayer::Ieee80211Radio)radio, antenna = { IsotropicAntenna },
Initializing module Net1.host[9].wlan[0].radio.transmitter, stage 13
Initializing module Net1.host[9].wlan[0].radio.receiver, stage 13
Initializing module Net1.host[9].pingApp[0], stage 13
Initializing module Net1.configurator, stage 13
Initializing module Net1.radioMedium, stage 13
INFO (Ieee80211ScalarRadioMedium)Net1.radioMedium:Initialized (inet::physicallayer::RadioMedium)radioMedium, propagation = { ConstantSpeedPro
Initializing module Net1.radioMedium.mediumLimitCache, stage 13
General #0: Net1
Msg stats: 39 scheduled / 221 existing / 221 created


OMNeT++/Qtenv (release) - General #0 - omnetpp.ini - C:\Users\cs18\Downloads\omnetpp-5.6.2\WSN_18\inet-3.7.1-src\inet\examples\adhoc\Simple_adhoc
File Simulate Inspect View Help
last: #2'007
3s 694ms 620us 376ns 603ps
Next: ping1 (inet::physicallayer::RadioFrame, id=2126)
In: Net1.host[4].wlan[0].radio (Ieee80211ScalarRadio, id=359)
At: 3.694620539791s (now+0.000000163188s)
Net1 (Net1) id=1
scheduled-events (cEventHeap)
configurator
radioMedium
host[0]
host[3]
host[7]
host[8]
(RadioFrame)ping1
Zoom:1.00x
numHosts (cPar) 10
host[0] (AdhocHost) id=2
host[1] (AdhocHost) id=3
host[2] (AdhocHost) id=4
host[3] (AdhocHost) id=5
host[4] (AdhocHost) id=6
host[5] (AdhocHost) id=7
host[6] (AdhocHost) id=8
host[7] (AdhocHost) id=9
host[8] (AdhocHost) id=10
Initializing module Net1.host[9].wlan[0].mac.hcf.edcaDataRecoveryProcedures[0], stage 13
Initializing module Net1.host[9].wlan[0].mac.hcf.edcaDataRecoveryProcedures[1], stage 13
Initializing module Net1.host[9].wlan[0].mac.hcf.edcaDataRecoveryProcedures[2], stage 13
Initializing module Net1.host[9].wlan[0].mac.hcf.edcaDataRecoveryProcedures[3], stage 13
Initializing module Net1.host[9].wlan[0].mac.hcf.edcaTxopProcedures[0], stage 13
Initializing module Net1.host[9].wlan[0].mac.hcf.edcaTxopProcedures[1], stage 13
Initializing module Net1.host[9].wlan[0].mac.hcf.edcaTxopProcedures[2], stage 13
Initializing module Net1.host[9].wlan[0].mac.hcf.edcaTxopProcedures[3], stage 13
Initializing module Net1.host[9].wlan[0].mac.hcf.originatorAckPolicy, stage 13
Initializing module Net1.host[9].wlan[0].mac.hcf.recipientAckPolicy, stage 13
Initializing module Net1.host[9].wlan[0].mac.hcf.rtsPolicy, stage 13
Initializing module Net1.host[9].wlan[0].mac.hcf.ctsPolicy, stage 13
Initializing module Net1.host[9].wlan[0].mac.rx, stage 13
General #0: Net1
Msg stats: 30 scheduled / 583 existing / 2144 created


Problems
Module Hierarchy
NED Parameters
NED Inheritance
Console
Simple_adhoc [OMNeT++ Simulation] C:\Users\cs18\Downloads\omnetpp-5.6.2\bin\opp_run.exe (2/16/24 4:46 PM -)
Net1.host[8].pingApp[0]: reply of 60 bytes from 10.0.0.1 icmp_seq=0 ttl=31 time=3.168548105 msec (ping0)
Net1.host[6].pingApp[0]: reply of 60 bytes from 10.0.0.1 icmp_seq=0 ttl=31 time=2.99103576 msec (ping0)
Net1.host[1].pingApp[0]: reply of 60 bytes from 10.0.0.1 icmp_seq=0 ttl=31 time=3.188090916 msec (ping0)
Net1.host[4].pingApp[0]: reply of 60 bytes from 10.0.0.1 icmp_seq=0 ttl=31 time=3.052688858 msec (ping0)

Wireless Sensor Networks & Mobile Communication

Practical No. 5

DEPARTMENT OF COMPUTER SCIENCE

Name:	Kimaya Naik	**Roll Number**	TCS2324048
Paper Code:	SIUSCS61	**Class**	TYBSc(Computer Science)
Topic:	OSPF and RIP Protocol	**Batch**	I
Date:	29/1/24	Practical No	5

A) AIM: Understanding, Reading and Analyzing Routing Table of Network

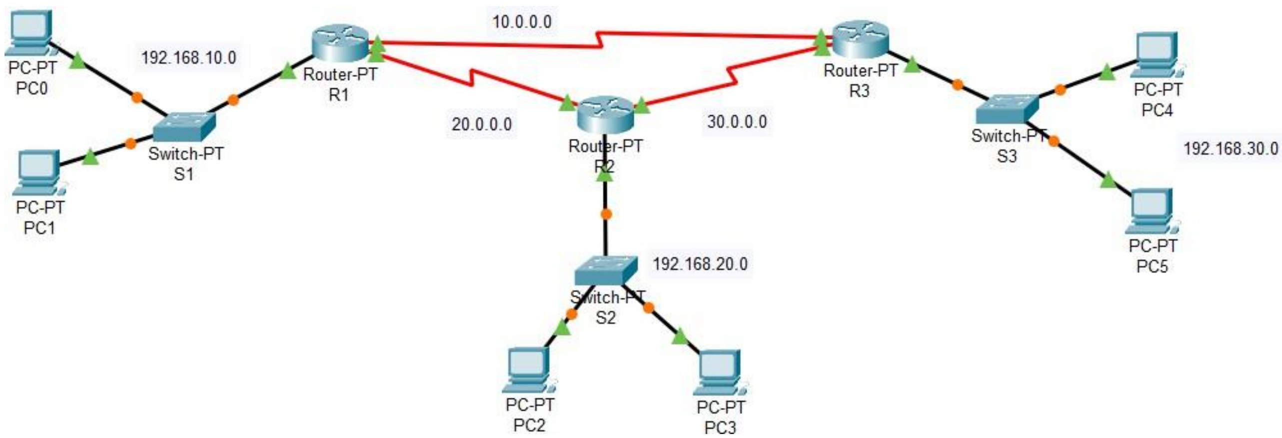
B) DESCRIPTION:

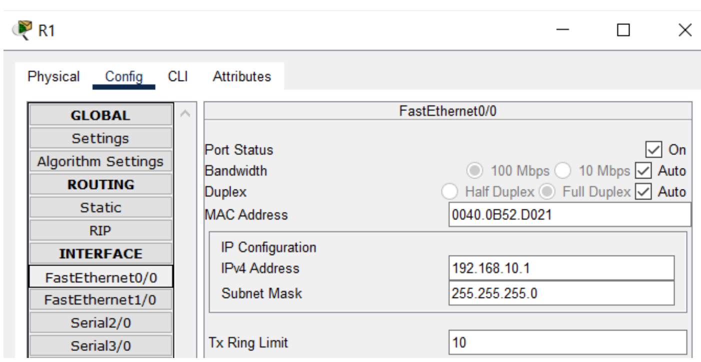
- RIP stands for Routing Information Protocol in which distance vector routing protocol is used for data/packet transmission. In the Routing Information Protocol (RIP), the maximum number of Hop is 15, because it prevents routing loops from source to destination. Compared to other routing protocols, RIP (Routing Information Protocol) is poor and limited in size i.e. small network. The main advantage of using RIP is it uses the UDP (User Datagram Protocol).

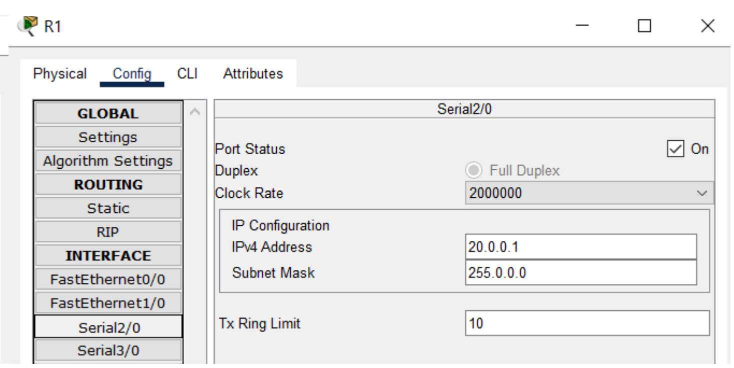
- OSPF stands for Open Shortest Path First which uses a link-state routing algorithm. Using the link state information which is available in routers, it constructs the topology in which topology determines the routing table for routing decisions. It is a router protocol which is used to find the best path for packets when they are passing through the set of connected networks simultaneously. The main disadvantage of OSPF is that it is difficult than other protocols.

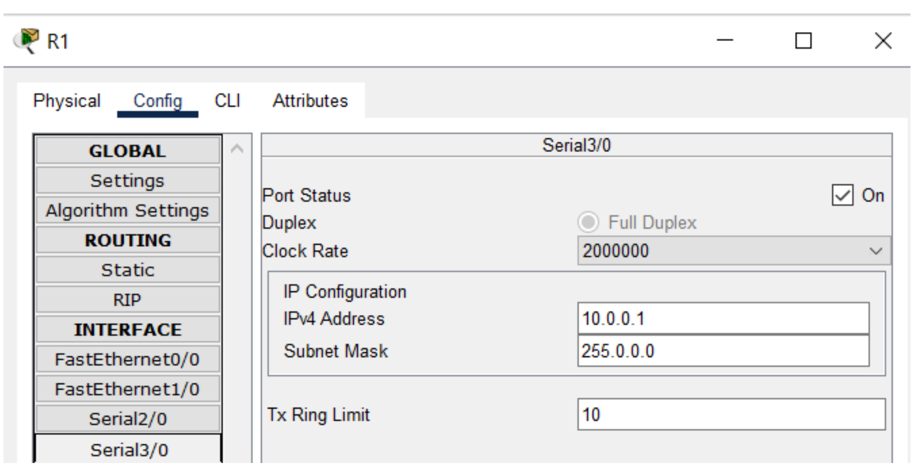
Code And Output :

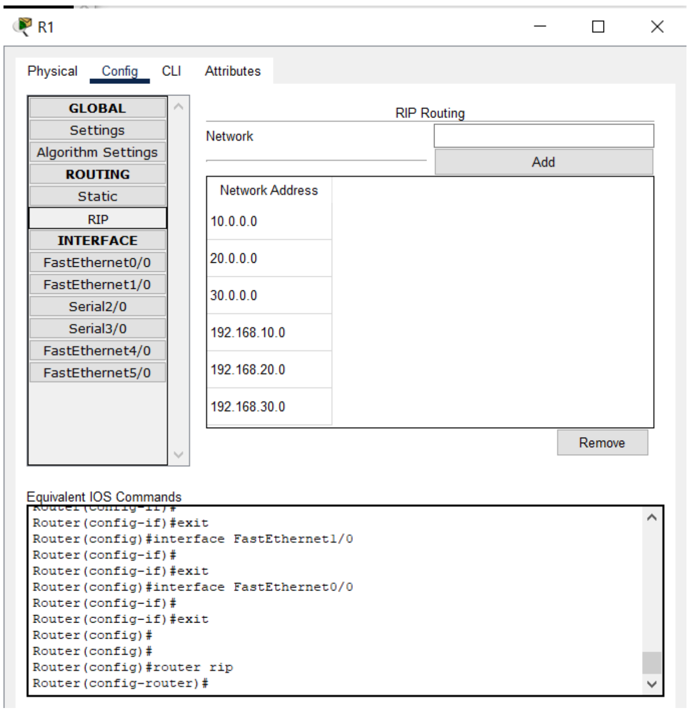
A) RIP Protocol :

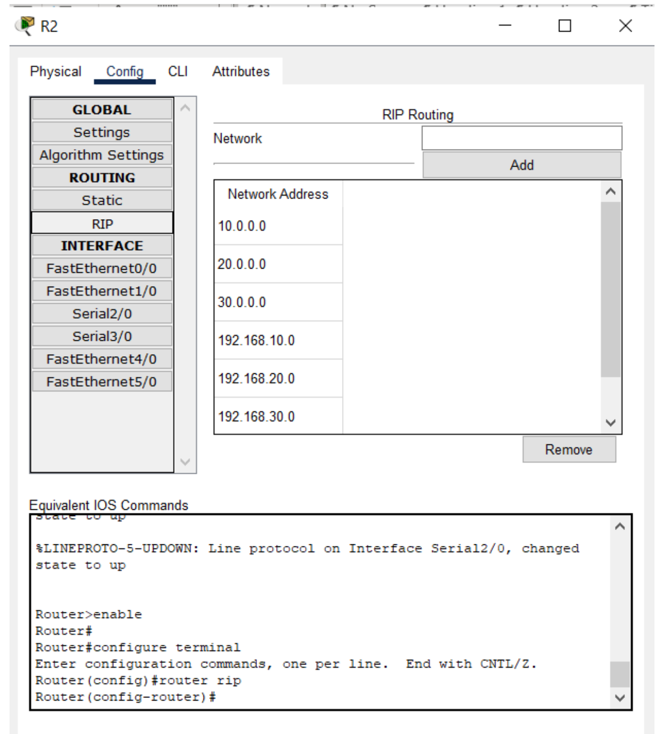


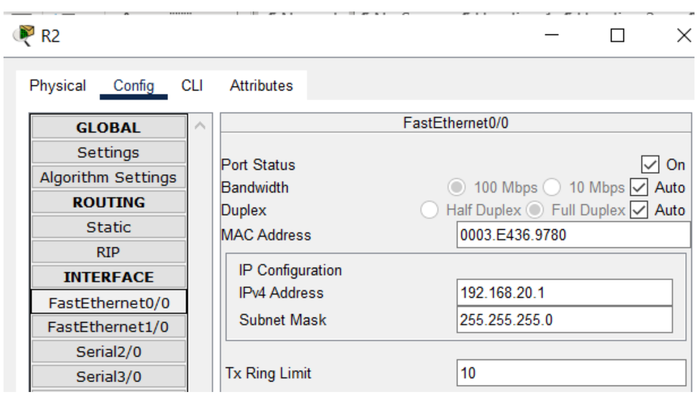


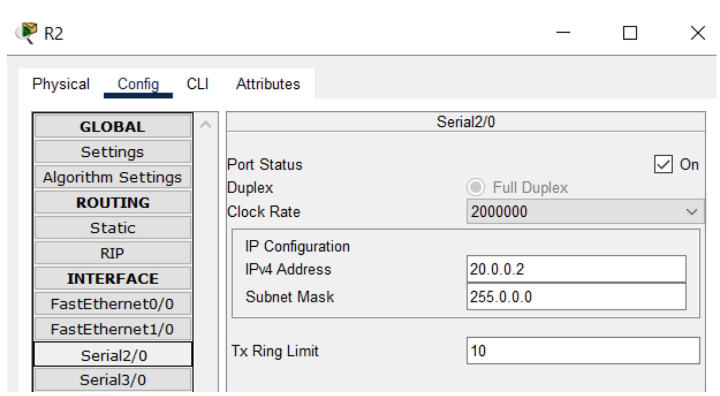


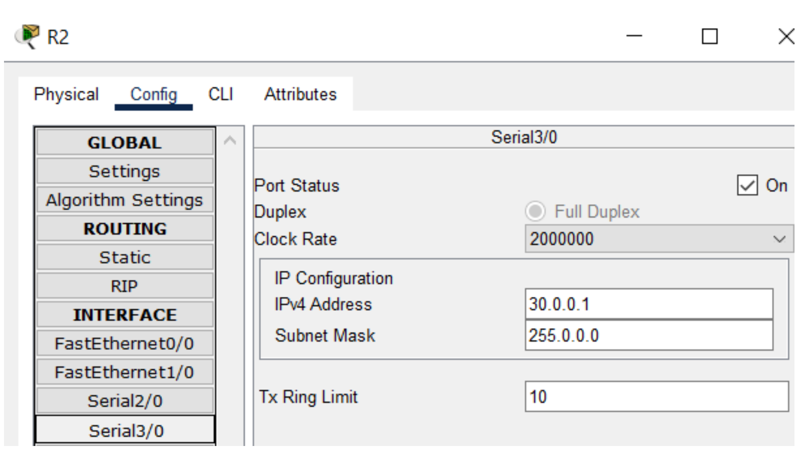


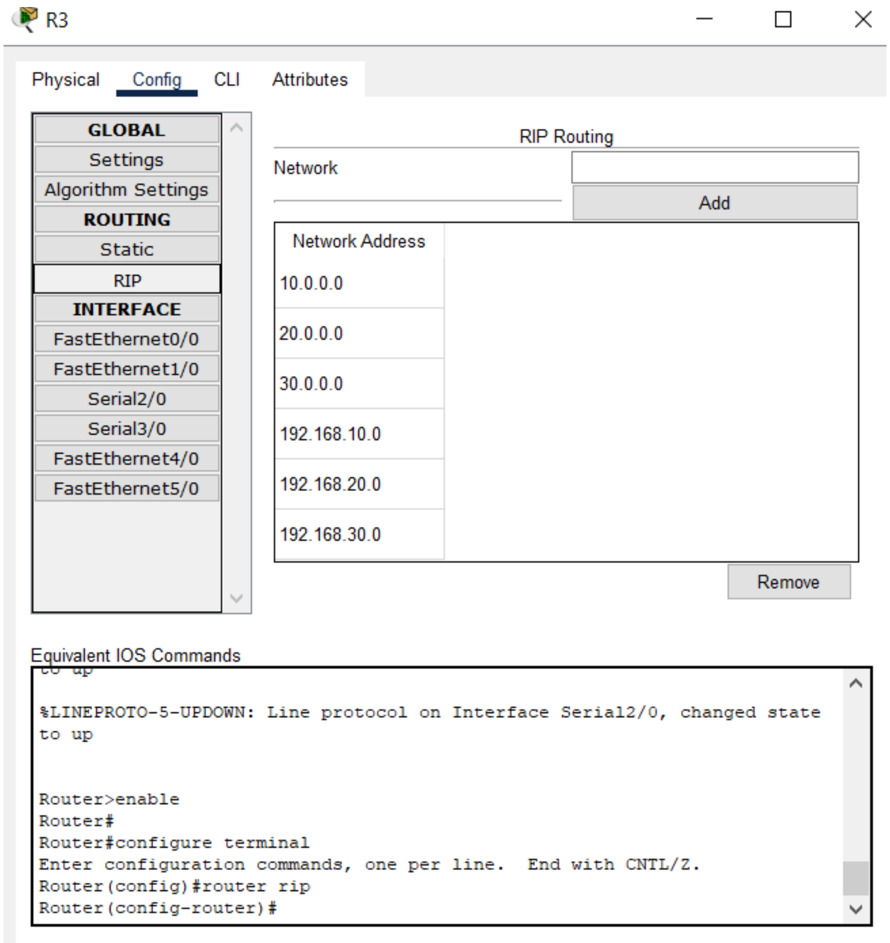

R1
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
INTERFACE
FastEthernet0/0
FastEthernet1/0
Serial2/0
Serial3/0
FastEthernet4/0
FastEthernet5/0
RIP Routing
Network
Add
Network Address
10.0.0.0
20.0.0.0
30.0.0.0
192.168.10.0
192.168.20.0
192.168.30.0
Remove
Equivalent IOS Commands
Router(config-if)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface FastEthernet0/0
Router(config-if)#
Router(config-if)#exit
Router(config)#
Router(config)#
Router(config)#router rip
Router(config-router)#

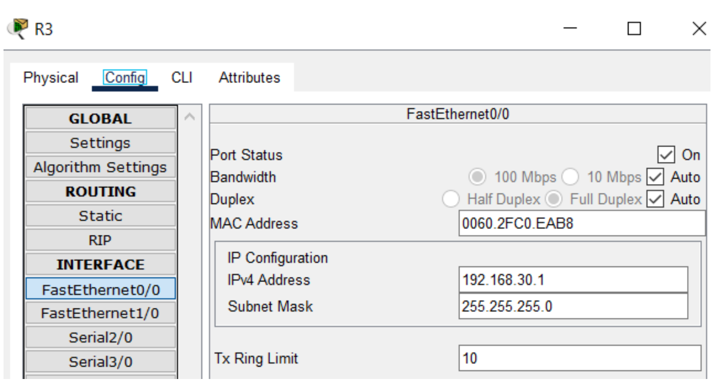

R2
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
INTERFACE
FastEthernet0/0
FastEthernet1/0
Serial2/0
Serial3/0
FastEthernet4/0
FastEthernet5/0
RIP Routing
Network
Add
Network Address
10.0.0.0
20.0.0.0
30.0.0.0
192.168.10.0
192.168.20.0
192.168.30.0
Remove
Equivalent IOS Commands
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
Router>enable
Router#
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#

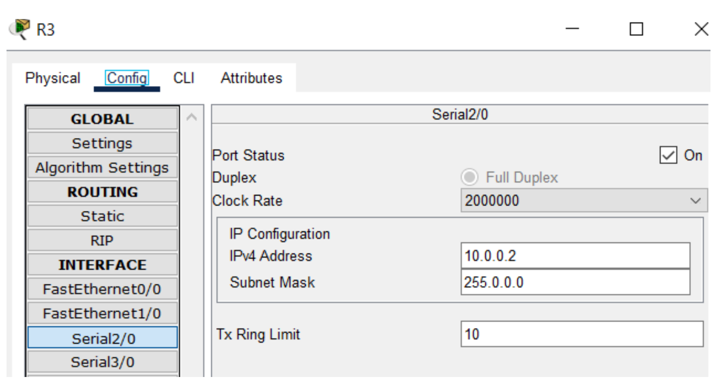

R2
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
INTERFACE
FastEthernet0/0
FastEthernet1/0
Serial2/0
Serial3/0
FastEthernet0/0
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
0003.E436.9780
IP Configuration
IPv4 Address
192.168.20.1
Subnet Mask
255.255.255.0
Tx Ring Limit
10

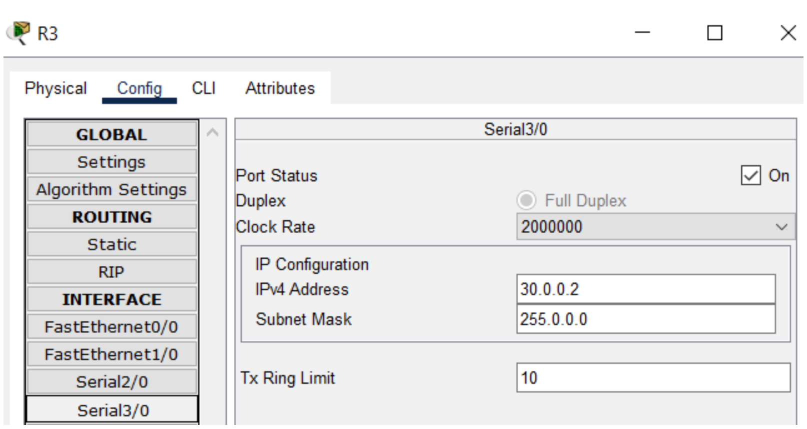

R2
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
INTERFACE
FastEthernet0/0
FastEthernet1/0
Serial2/0
Serial3/0
Serial2/0
Port Status
On
Duplex
Full Duplex
Clock Rate
2000000
IP Configuration
IPv4 Address
20.0.0.2
Subnet Mask
255.0.0.0
Tx Ring Limit
10


R2
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
INTERFACE
FastEthernet0/0
FastEthernet1/0
Serial2/0
Serial3/0
Serial3/0
Port Status
On
Duplex
Full Duplex
Clock Rate
2000000
IP Configuration
IPv4 Address
30.0.0.1
Subnet Mask
255.0.0.0
Tx Ring Limit
10


R3
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
INTERFACE
FastEthernet0/0
FastEthernet1/0
Serial2/0
Serial3/0
FastEthernet4/0
FastEthernet5/0
RIP Routing
Network
Add
Network Address
10.0.0.0
20.0.0.0
30.0.0.0
192.168.10.0
192.168.20.0
192.168.30.0
Remove
Equivalent IOS Commands
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
Router>enable
Router#
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#

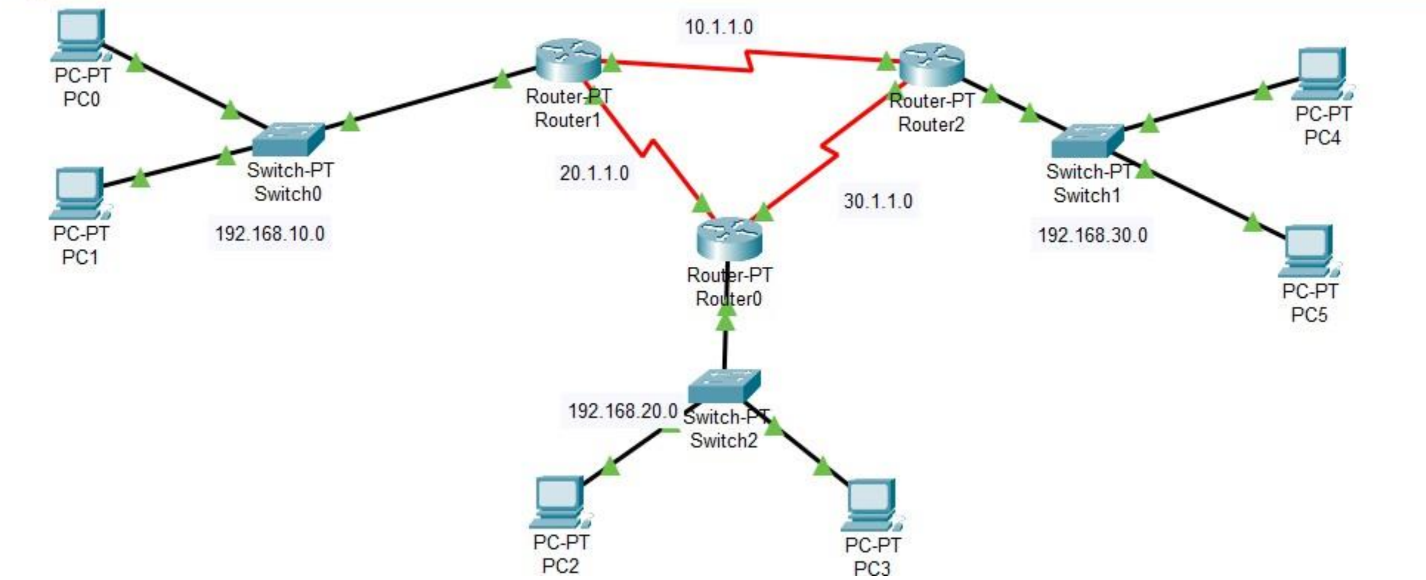

R3
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
INTERFACE
FastEthernet0/0
FastEthernet1/0
Serial2/0
Serial3/0
FastEthernet0/0
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
0060.2FC0.EAB8
IP Configuration
IPv4 Address
192.168.30.1
Subnet Mask
255.255.255.0
Tx Ring Limit
10


R3
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
INTERFACE
FastEthernet0/0
FastEthernet1/0
Serial2/0
Serial3/0
Serial2/0
Port Status
On
Duplex
Full Duplex
Clock Rate
2000000
IP Configuration
IPv4 Address
10.0.0.2
Subnet Mask
255.0.0.0
Tx Ring Limit
10


R3
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
INTERFACE
FastEthernet0/0
FastEthernet1/0
Serial2/0
Serial3/0
Serial3/0
Port Status
On
Duplex
Full Duplex
Clock Rate
2000000
IP Configuration
IPv4 Address
30.0.0.2
Subnet Mask
255.0.0.0
Tx Ring Limit
10

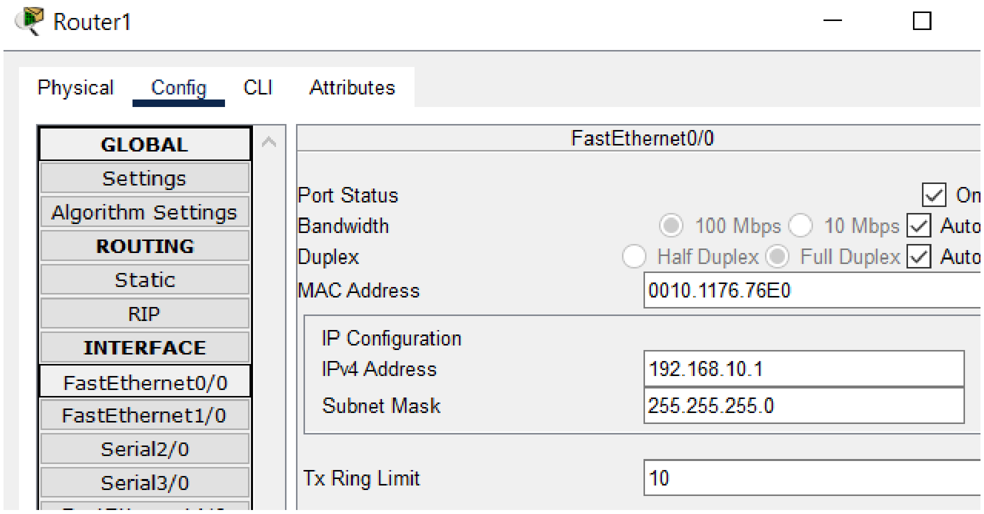
Code And Output :

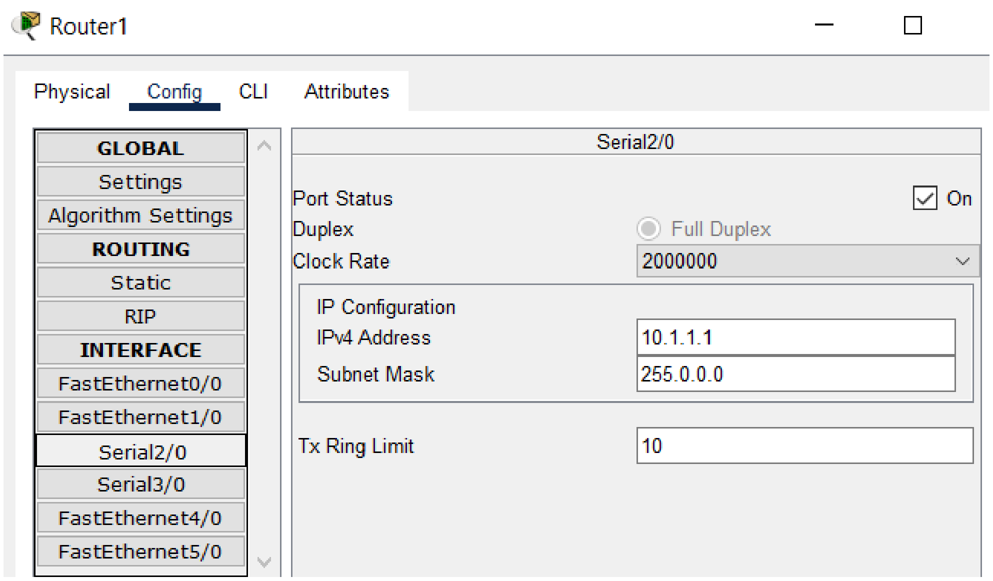
B) OSPF Protocol :

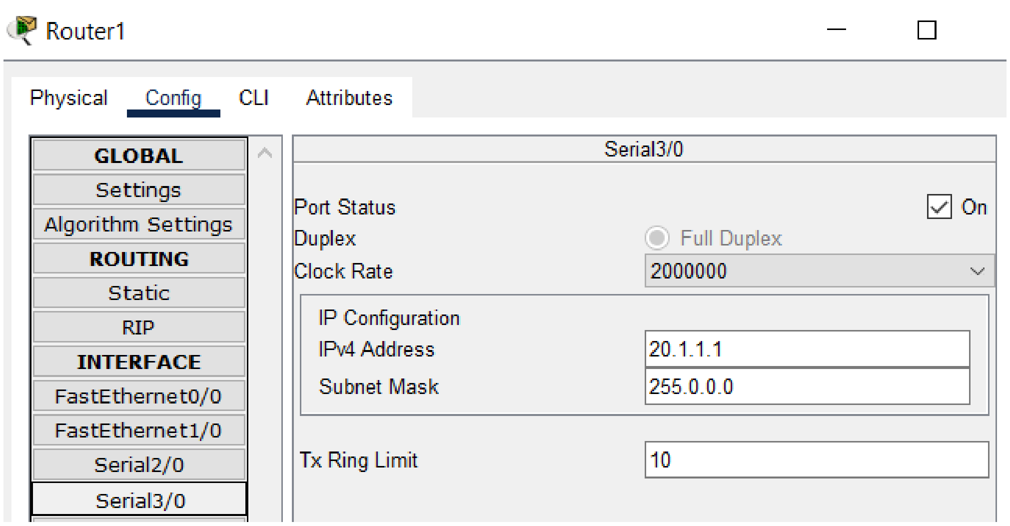


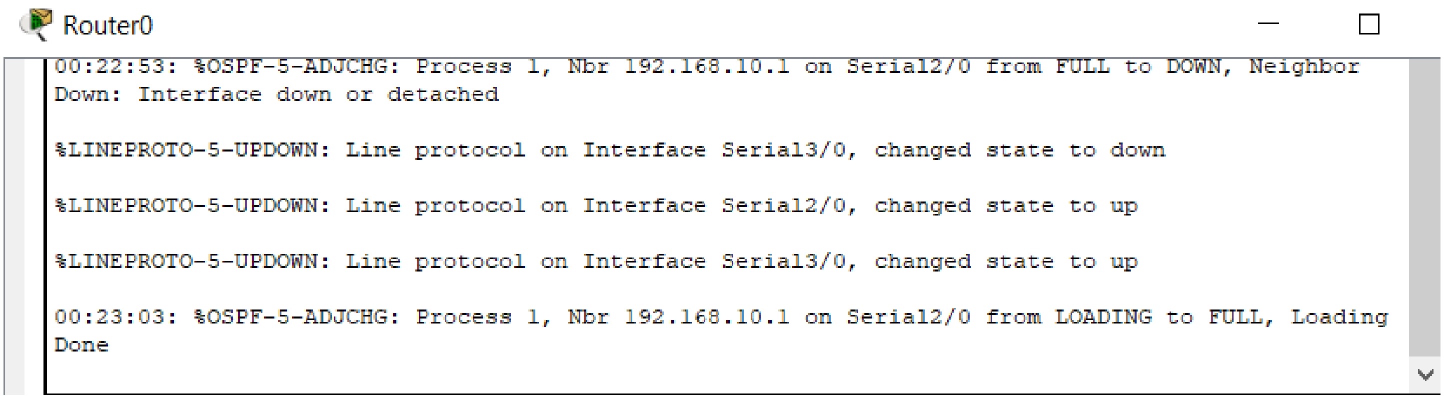
Router1

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to down

00:22:53: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.30.1 on Serial3/0 from FULL to DOWN, Neighbor
Down: Interface down or detached

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to down

00:22:53: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.30.1 on Serial2/0 from FULL to DOWN, Neighbor
Down: Interface down or detached

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

00:23:03: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.30.1 on Serial3/0 from LOADING to FULL, Loading
Done

00:23:03: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.30.1 on Serial2/0 from LOADING to FULL, Loading
Done
```

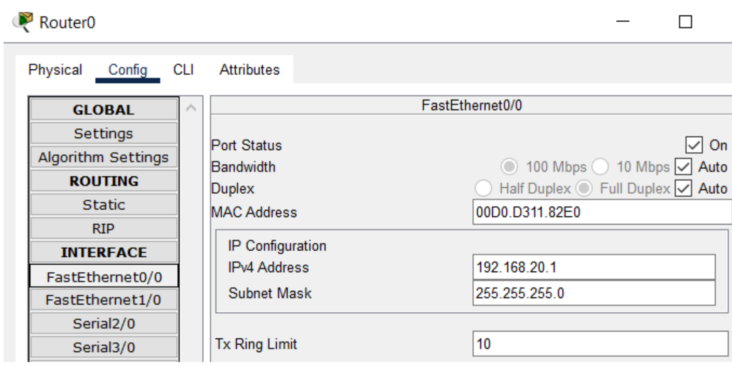

Router1
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
INTERFACE
FastEthernet0/0
FastEthernet1/0
Serial2/0
Serial3/0
FastEthernet0/0
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
0010.1176.76E0
IP Configuration
IPv4 Address
192.168.10.1
Subnet Mask
255.255.255.0
Tx Ring Limit
10

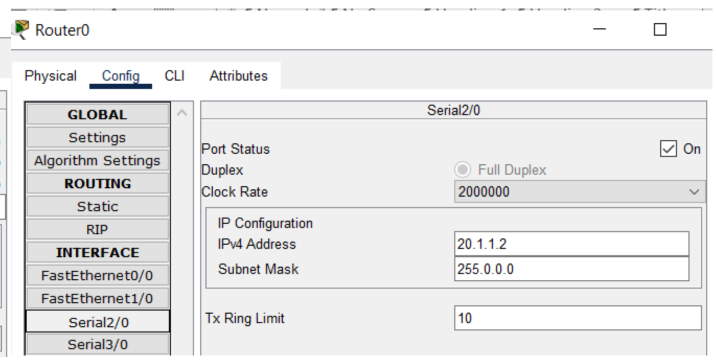

Router1
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
INTERFACE
FastEthernet0/0
FastEthernet1/0
Serial2/0
Serial3/0
FastEthernet4/0
FastEthernet5/0
Serial2/0
Port Status
On
Duplex
Full Duplex
Clock Rate
2000000
IP Configuration
IPv4 Address
10.1.1.1
Subnet Mask
255.0.0.0
Tx Ring Limit
10

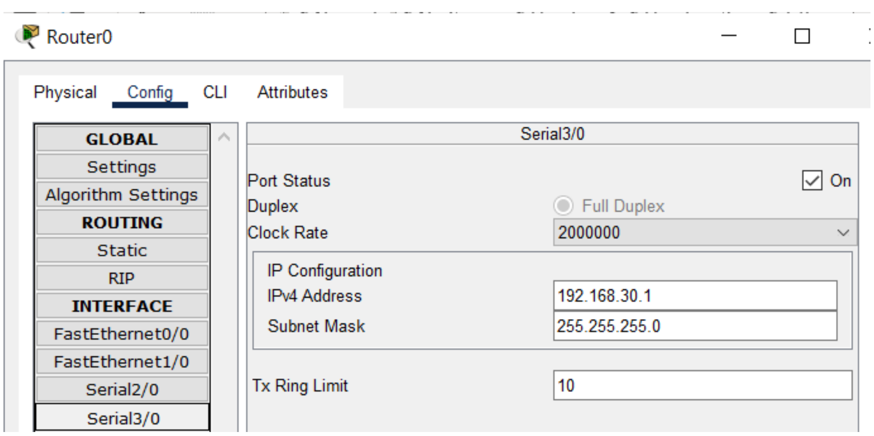

Router1
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
INTERFACE
FastEthernet0/0
FastEthernet1/0
Serial2/0
Serial3/0
Serial3/0
Port Status
On
Duplex
Full Duplex
Clock Rate
2000000
IP Configuration
IPv4 Address
20.1.1.1
Subnet Mask
255.0.0.0
Tx Ring Limit
10

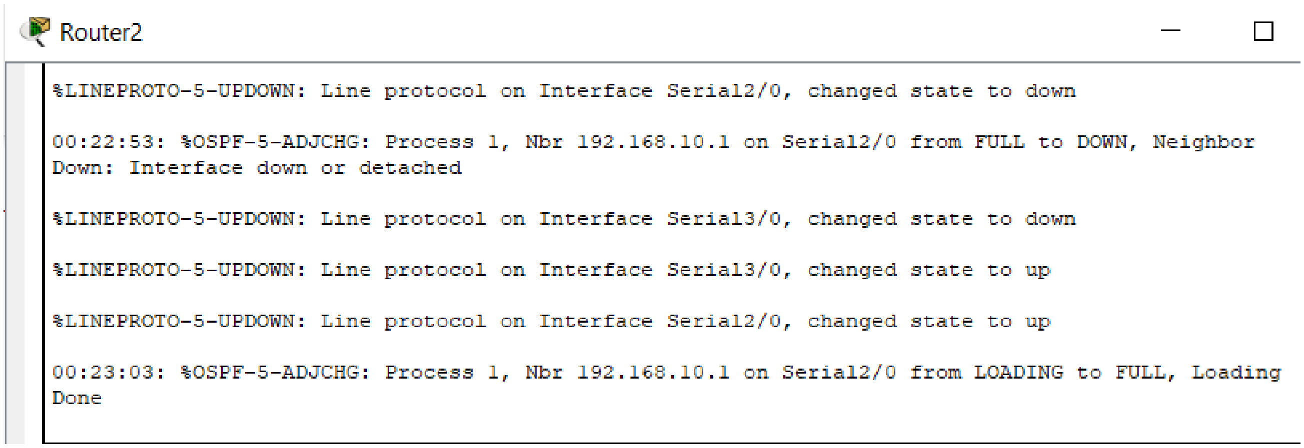


```
00:22:53: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.10.1 on Serial2/0 from FULL to DOWN, Neighbor
Down: Interface down or detached

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up

00:23:03: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.10.1 on Serial2/0 from LOADING to FULL, Loading
Done
```

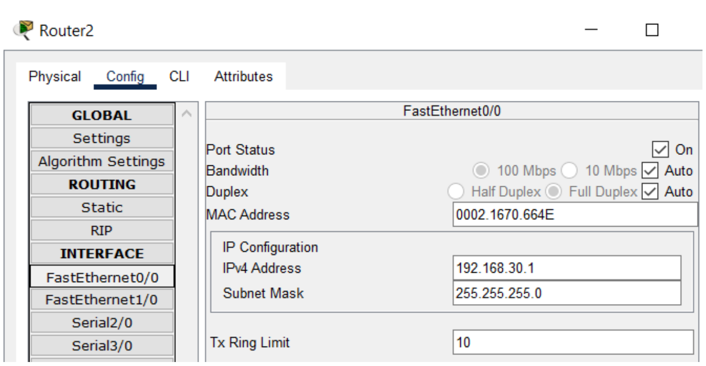


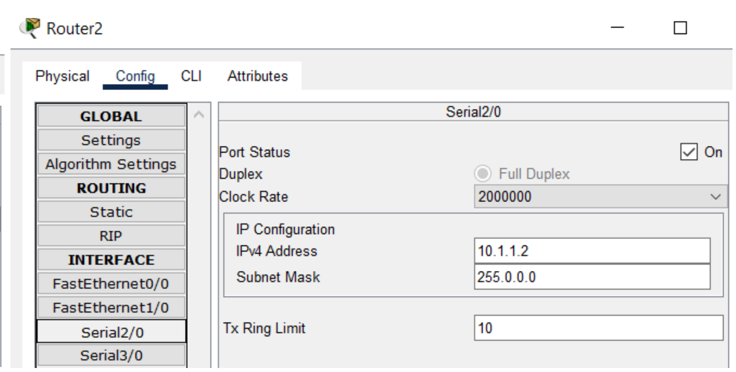


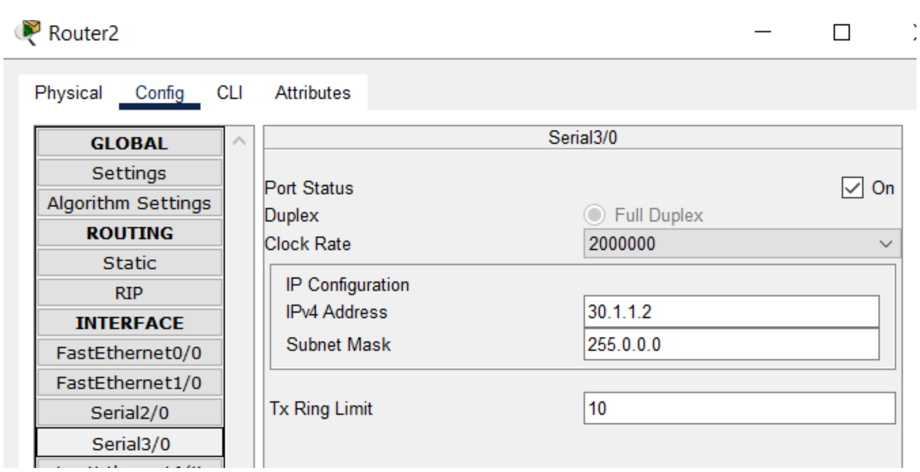





```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to down

00:22:53: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.10.1 on Serial2/0 from FULL to DOWN, Neighbor
Down: Interface down or detached

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up

00:23:03: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.10.1 on Serial2/0 from LOADING to FULL, Loading
Done
```







Wireless Sensor Networks & Mobile Communication
Practical No. 6

DEPARTMENT OF COMPUTER SCIENCE

Name:	Kimaya Naik	**Roll Number**	TCS232048
Paper Code:	SIUSCS61	**Class**	T.Y. B.Sc. (Computer Science)
Topic:	Wireless network	**Batch**	I
Date:	05/02/2024	**Practical No**	6

A) AIM: Simulate a network for simple wireless network.

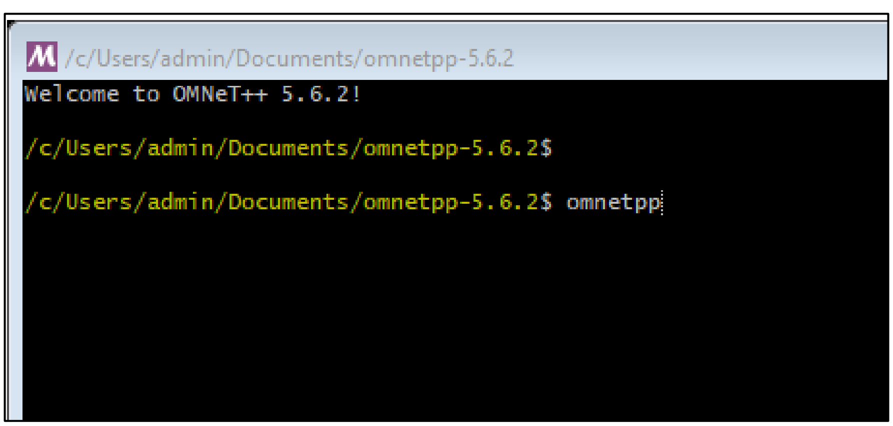
B) DESCRIPTION:

Computer networks that are not connected by cables are called wireless networks. They generally use radio waves for communication between the network nodes. They allow devices to be connected to the network while roaming around within the network coverage. It increases the mobility of network devices connected to the system since the devices need not be connected to each other. Types of Wireless Networks: Wireless LANs, Wireless MANs and Wireless WANs. Wireless networks require very limited or no wires. Thus, it reduces the equipment and setup costs.

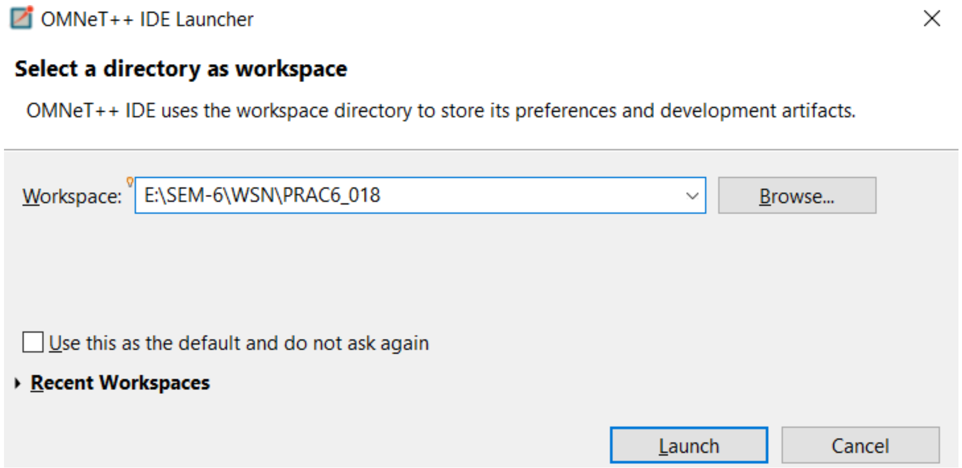
C) NETWORK TOPOLOGY, CONFIGURATIONS AND OUTPUT:

Configurations:

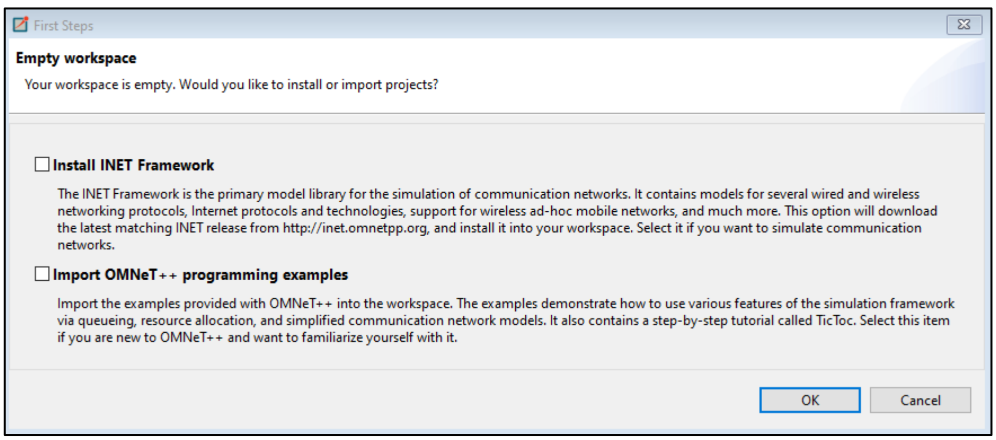
Step 1: Open omnet folder → Open mingwenv → Type omnetpp in console



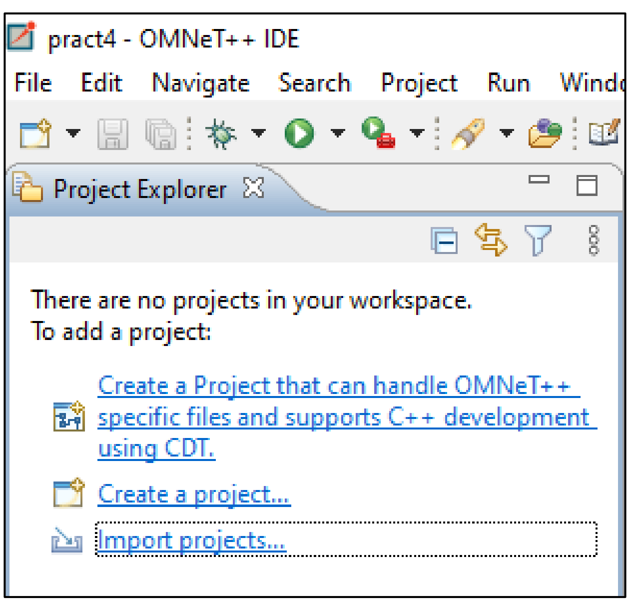
Step 2: Create workspace outside omnet and inet folder → Launch

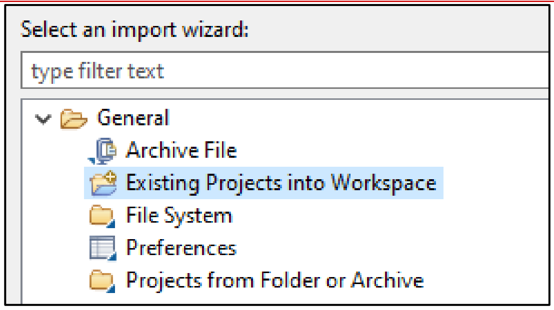


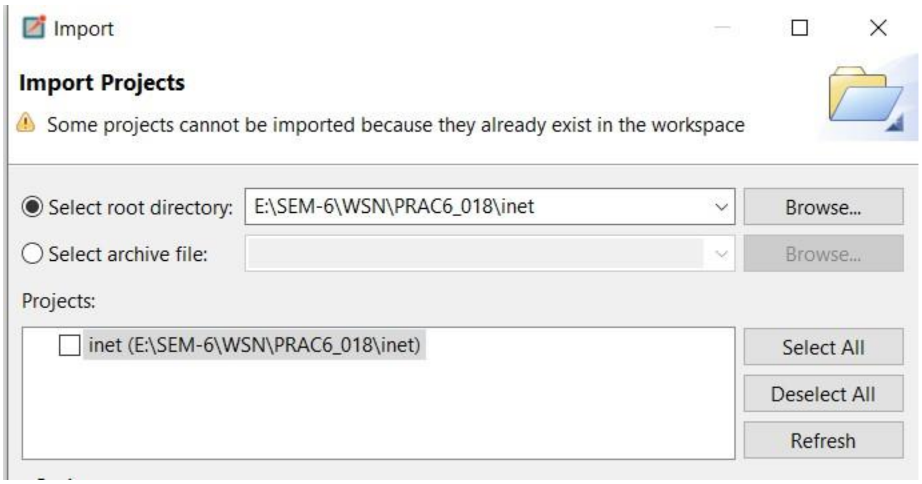
Step 3: Uncheck the boxes and click ok



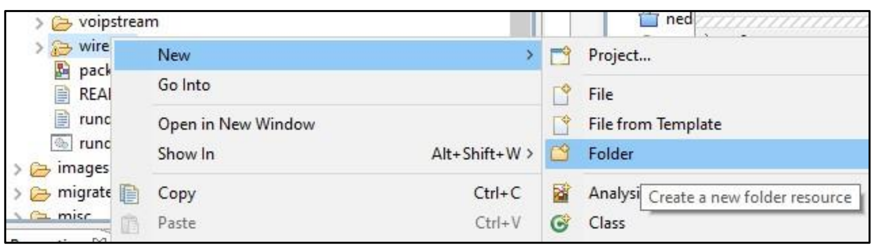
Step 4: Click import projects → General → Existing Projects into Workspace → Next → Browse inet folder location (under "select root directory") → Check inet → Finish

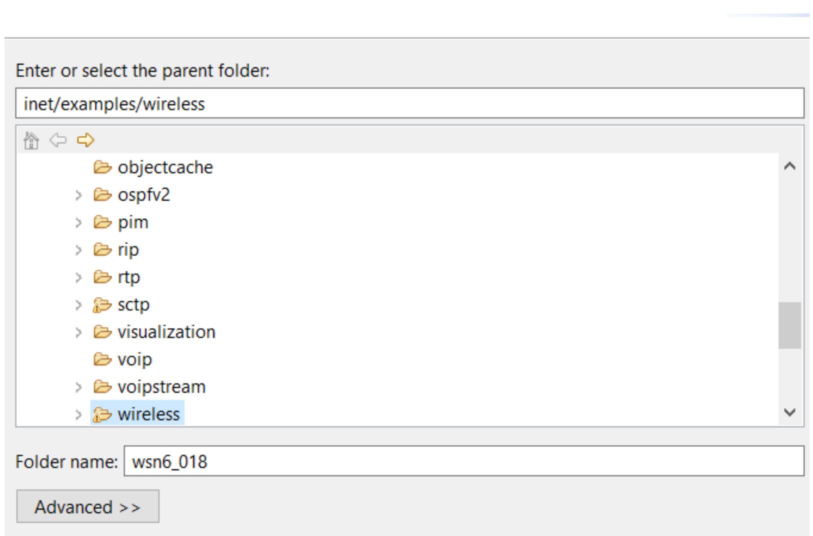




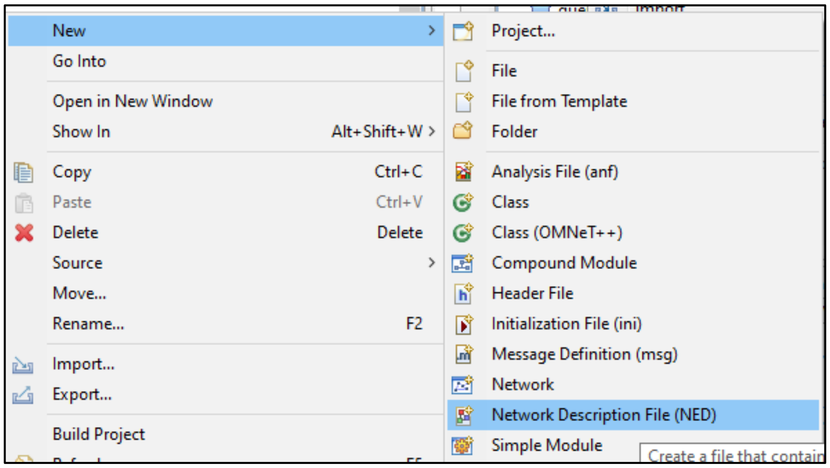


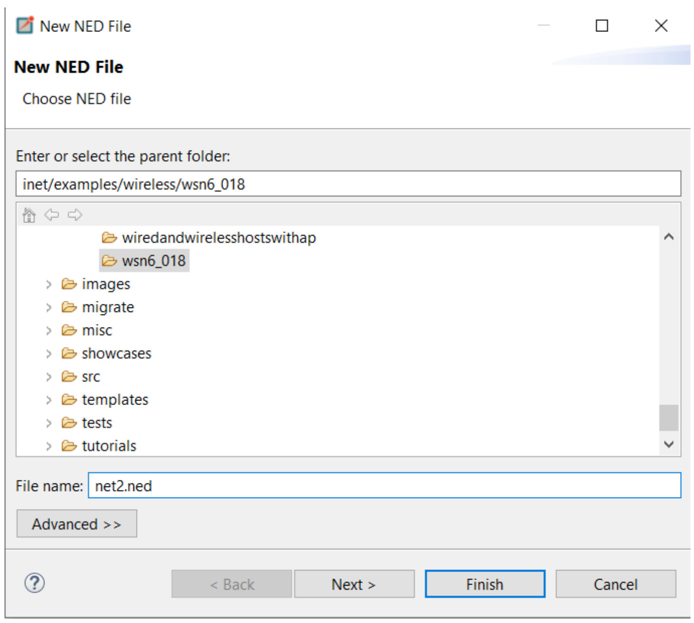
Step 5: Click inet folder → Examples → wireless → Right click → New → Folder → Provide folder name → Finish

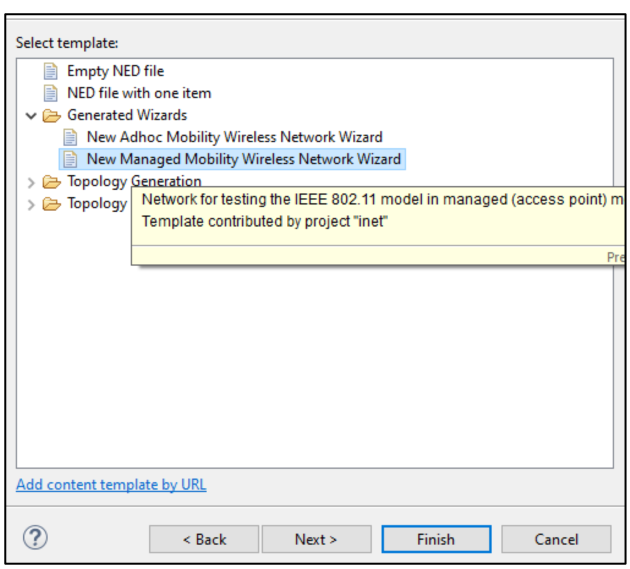


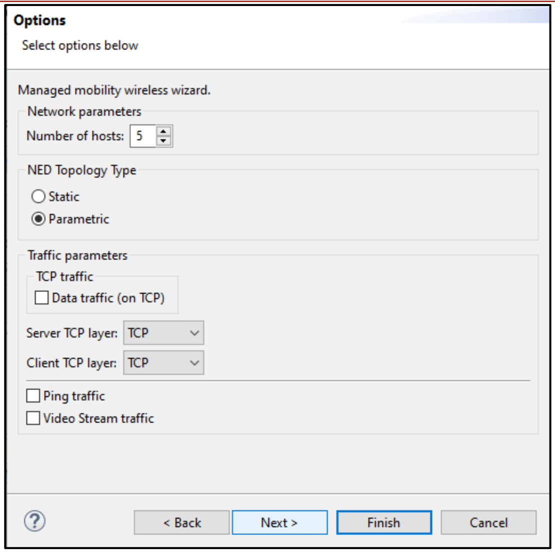


Step 6: Right click on the created folder → New → Select NED → Provide a name → Next → Click “generated wizards” → Select “New managed mobility wireless network wizard” → Next → Finish



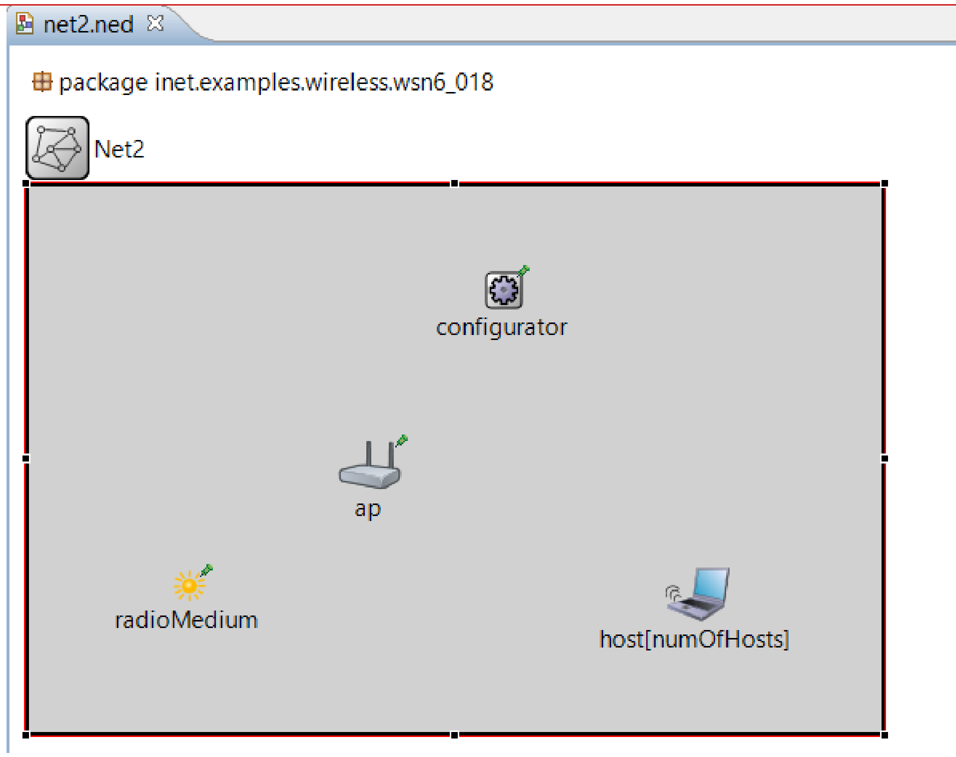




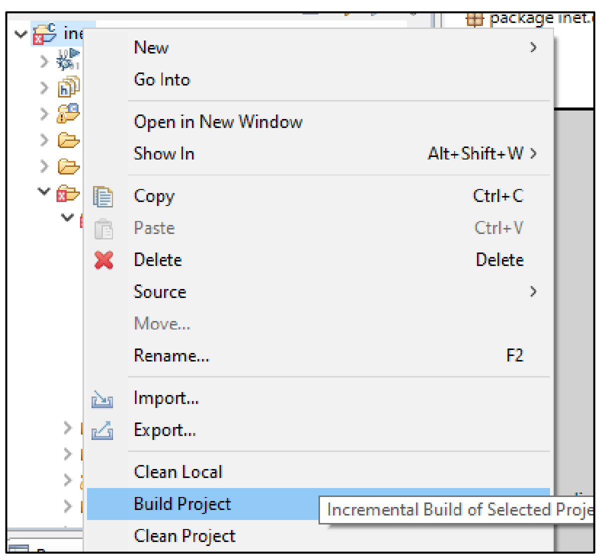


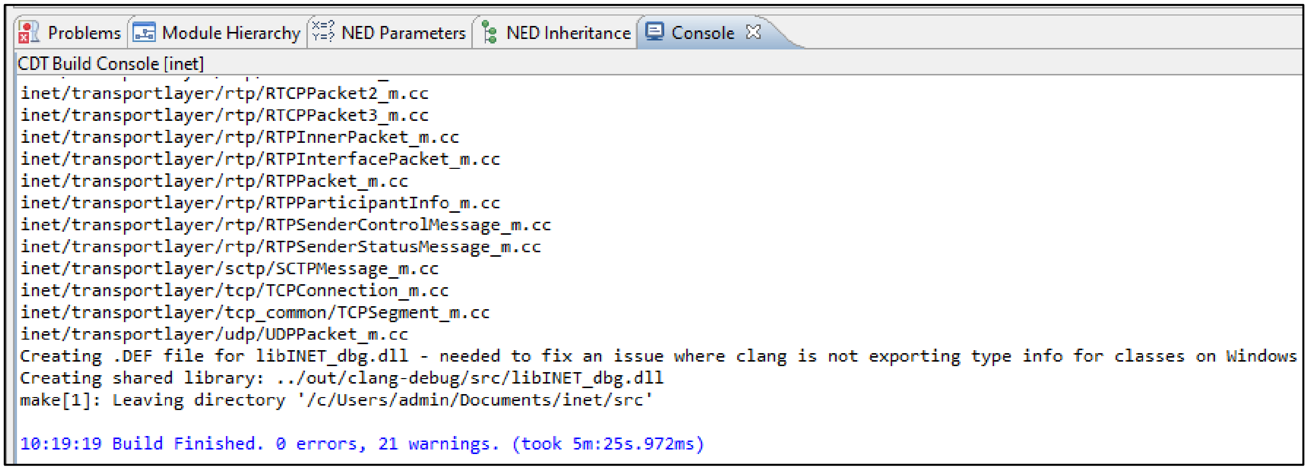
Step 7: Go to wireless.ned (of your created practical folder; under wireless folder) → Source → Do the following changes as mentioned below → Save the changes

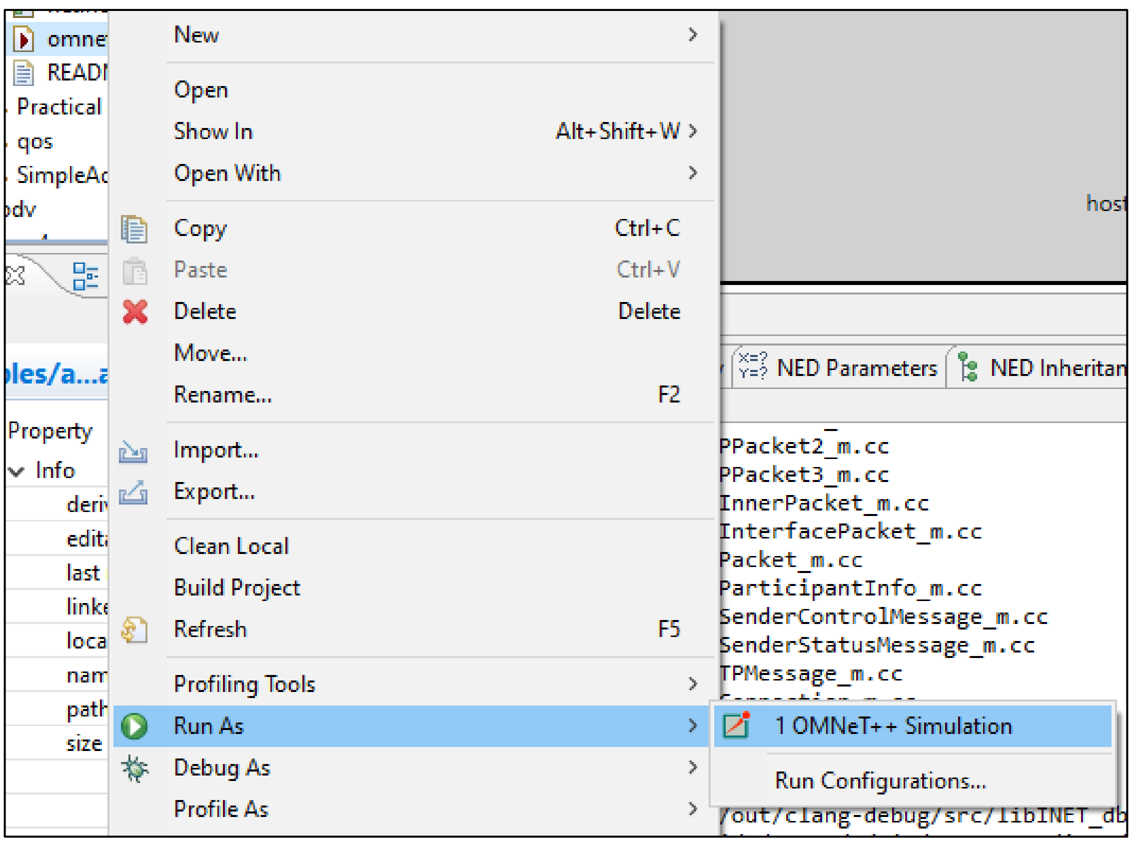
```
package inet.examples.wireless.wsn6_018;

// numOfHosts: 5

import inet.examples.adhoc.hostautoconf.Host;
import inet.networklayer.configurator.ipv4.IPv4NetworkConfigurator;
import inet.node.inet.WirelessHost;
import inet.node.wireless.AccessPoint;
import inet.physicallayer.ieee80211.packetlevel.Ieee80211ScalarRadioMedium;


network Net2
{
    parameters:
        int numOfHosts;

    submodules:
        host[numOfHosts]: WirelessHost {
            @display("r=,,#707070");
        }

        ap: AccessPoint {
            @display("p=213,174;r=,,#707070");
        }

        configurator: IPv4NetworkConfigurator {
            @display("p=297,65");
        }
        radioMedium: Ieee80211ScalarRadioMedium {
            parameters:
                @display("p=100,250");
        }
}
```

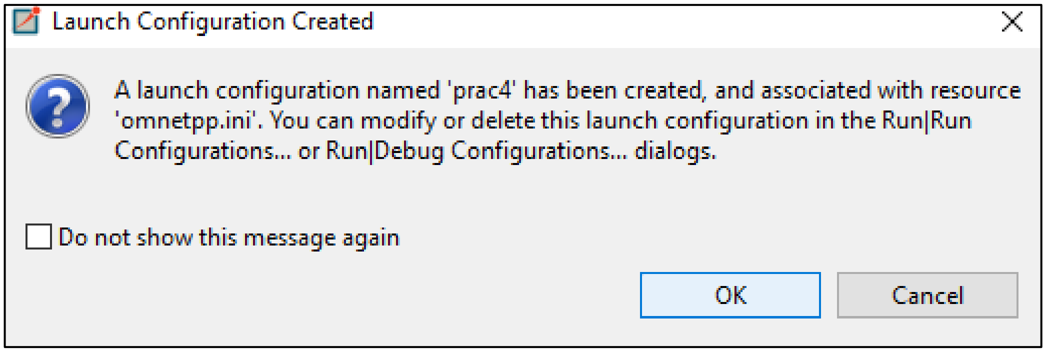


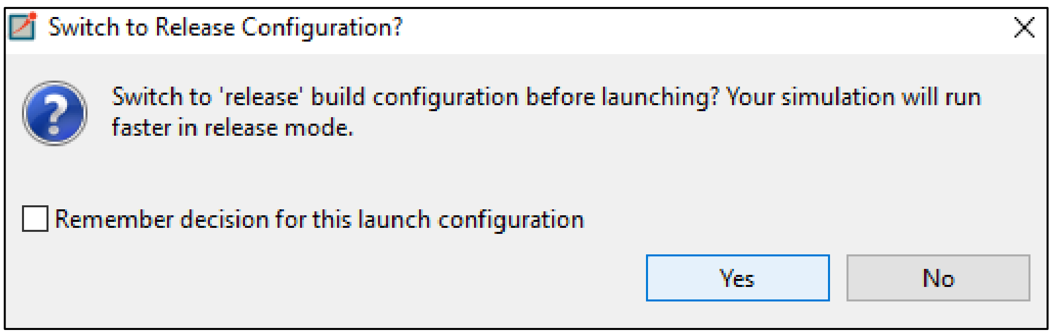
Step 8: Now, for testing our simulation: Right click on inet → Select build project → Right click on omnetpp.ini (present in your created practical folder) → Select "run as" → Select "omnet++ simulation → Click OK → Click Yes → Done

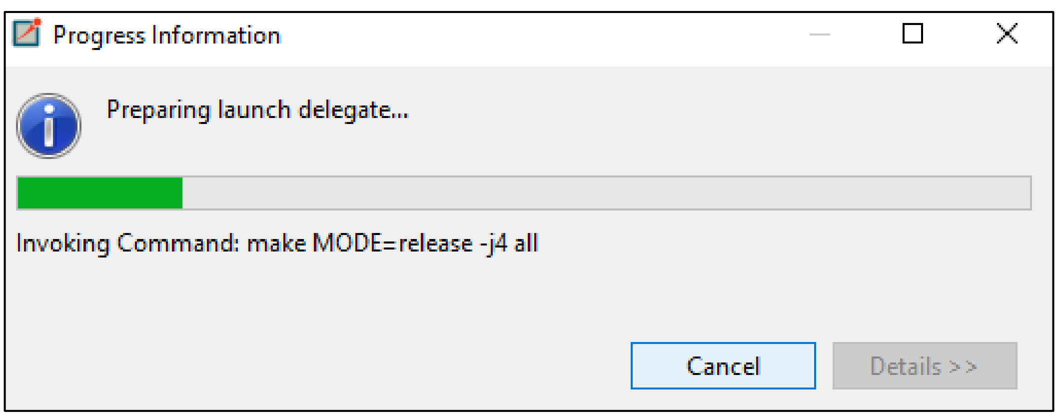



Problems
Module Hierarchy
NED Parameters
NED Inheritance
Console
CDT Build Console [inet]
inet/transportlayer/rtp/RTCPPacket2_m.cc
inet/transportlayer/rtp/RTCPPacket3_m.cc
inet/transportlayer/rtp/RTPInnerPacket_m.cc
inet/transportlayer/rtp/RTPInterfacePacket_m.cc
inet/transportlayer/rtp/RTPPacket_m.cc
inet/transportlayer/rtp/RTPParticipantInfo_m.cc
inet/transportlayer/rtp/RTPSenderControlMessage_m.cc
inet/transportlayer/rtp/RTPSenderStatusMessage_m.cc
inet/transportlayer/sctp/SCTPMessage_m.cc
inet/transportlayer/tcp/TCPConnection_m.cc
inet/transportlayer/tcp_common/TCPSegment_m.cc
inet/transportlayer/udp/UDPPacket_m.cc
Creating .DEF file for libINET_dbg.dll - needed to fix an issue where clang is not exporting type info for classes on Windows
Creating shared library: ../out/clang-debug/src/libINET_dbg.dll
make[1]: Leaving directory '/c/Users/admin/Documents/inet/src'
10:19:19 Build Finished. 0 errors, 21 warnings. (took 5m:25s.972ms)

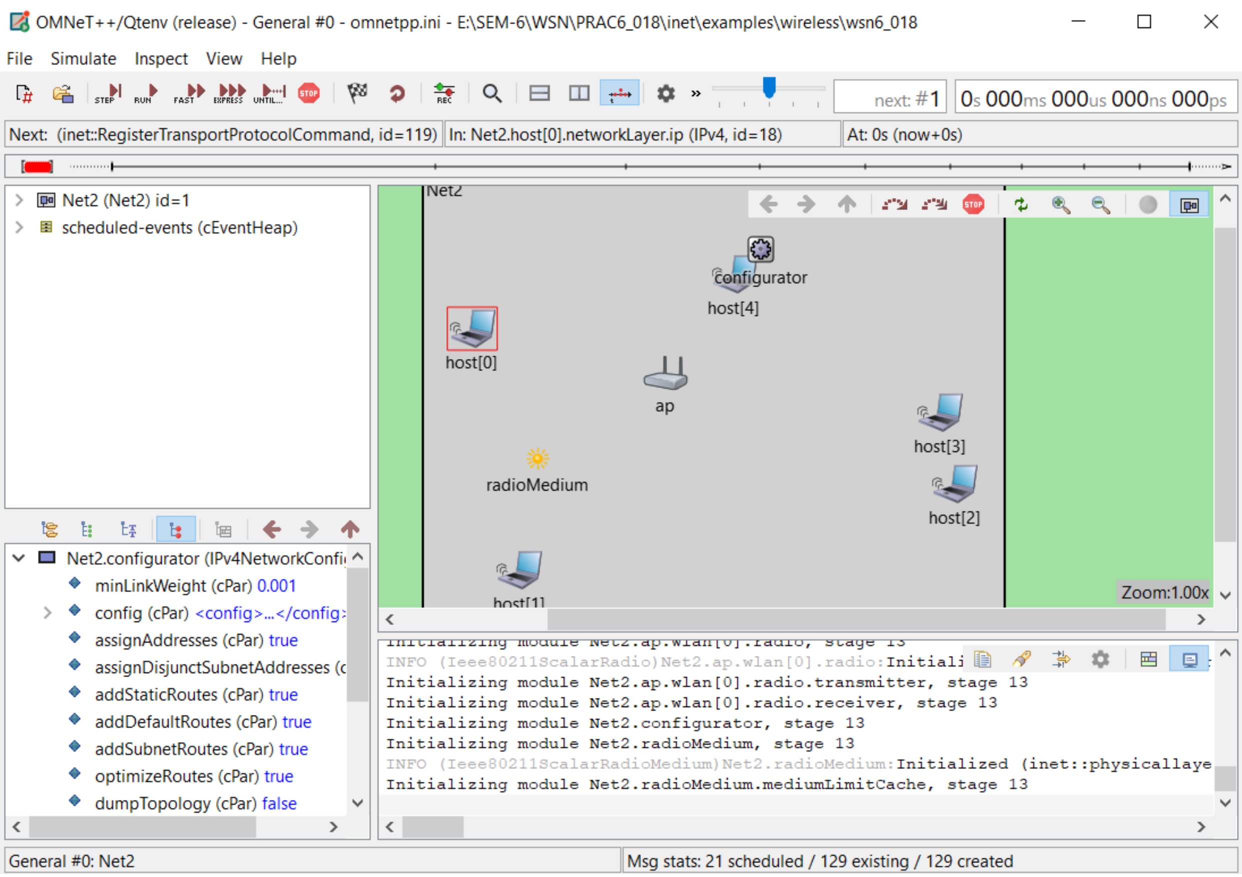

New
Open
Show In Alt+Shift+W
Open With
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Move...
Rename... F2
Import...
Export...
Clean Local
Build Project
Refresh F5
Profiling Tools
Run As
Debug As
Profile As
1 OMNeT++ Simulation
Run Configurations...
NED Parameters
Practical
qos
Property
Info


Launch Configuration Created
A launch configuration named 'prac4' has been created, and associated with resource 'omnetpp.ini'. You can modify or delete this launch configuration in the Run|Run Configurations... or Run|Debug Configurations... dialogs.
Do not show this message again
OK
Cancel

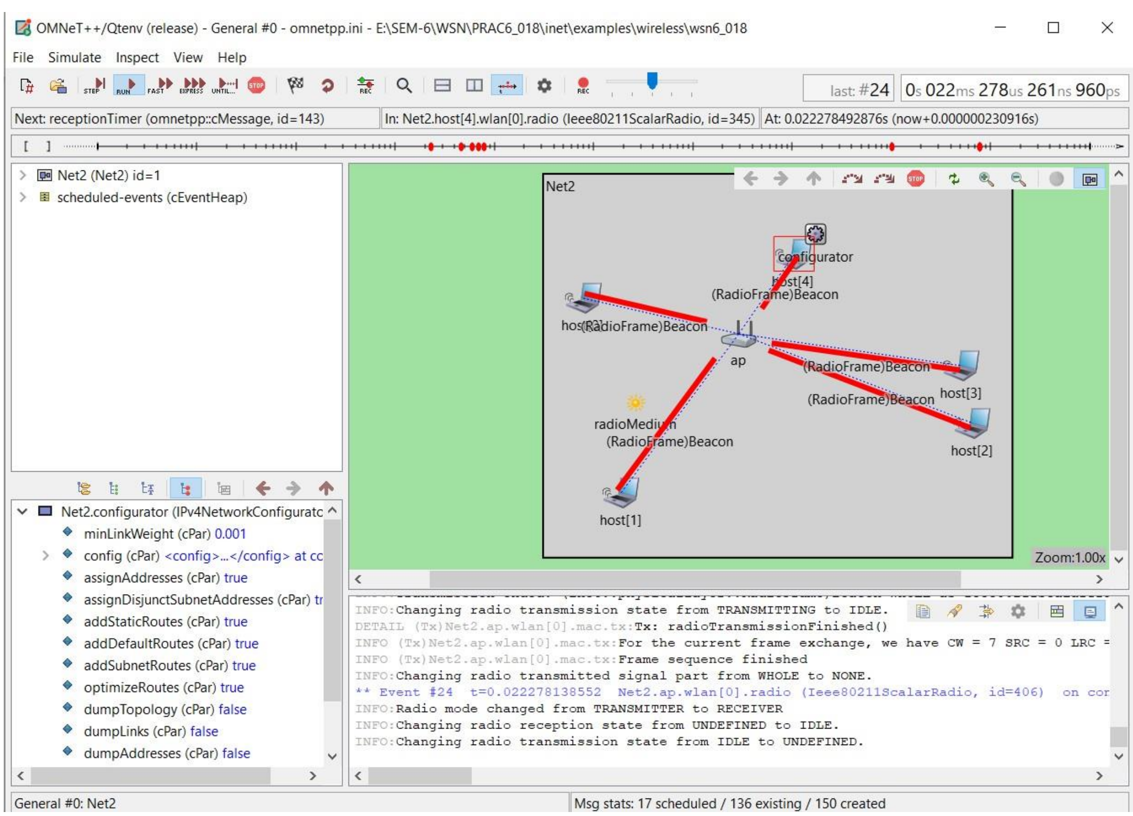

Switch to Release Configuration?
Switch to 'release' build configuration before launching? Your simulation will run faster in release mode.
Remember decision for this launch configuration
Yes
No


Progress Information
Preparing launch delegate...
Invoking Command: make MODE=release -j4 all
Cancel
Details >>

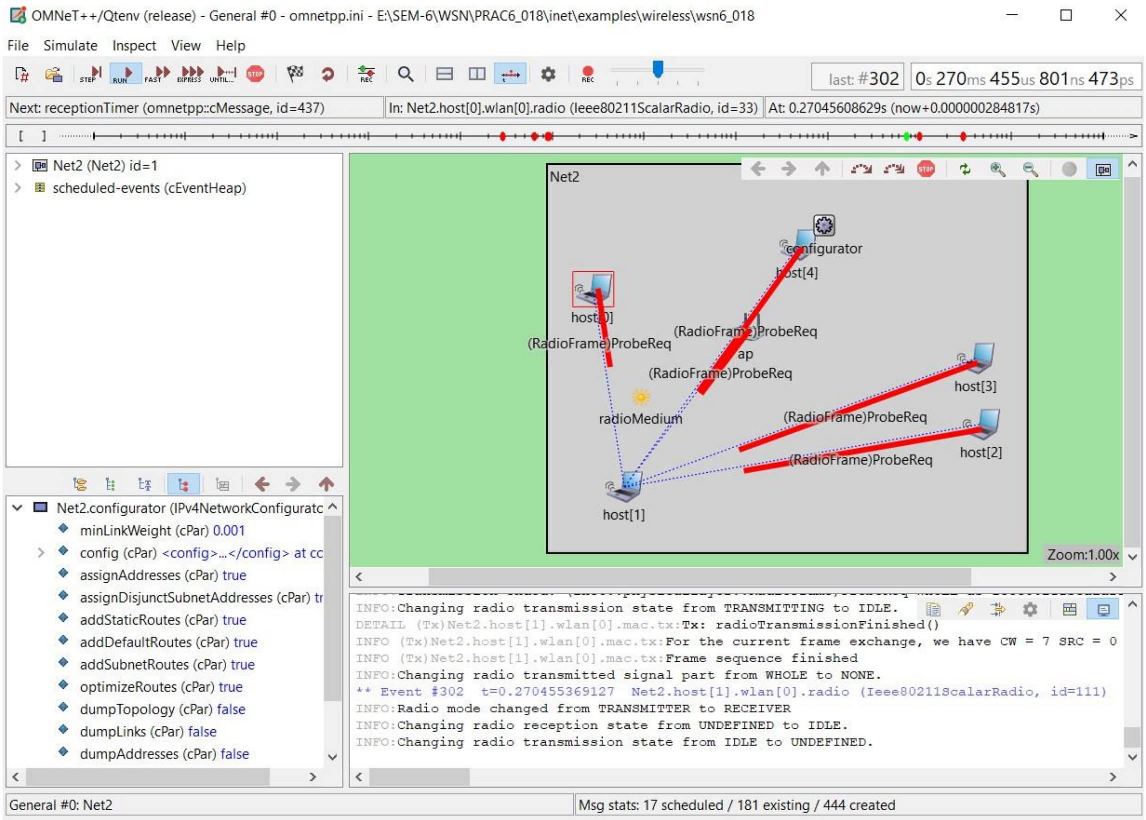
Output:

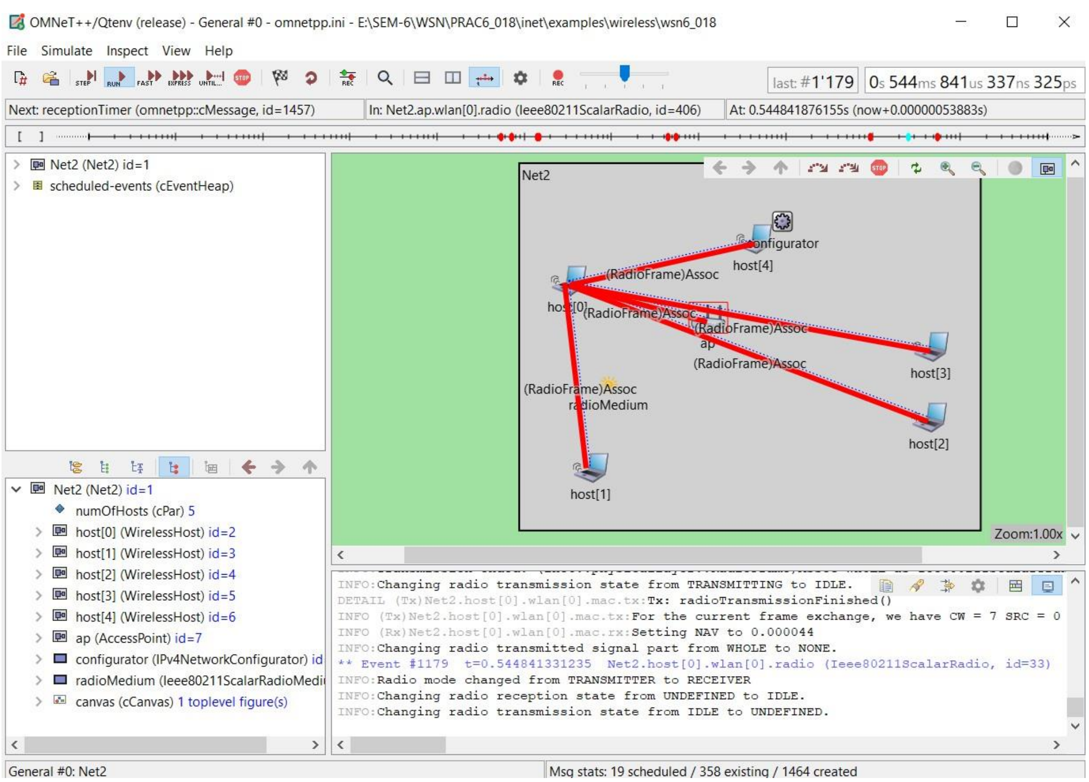


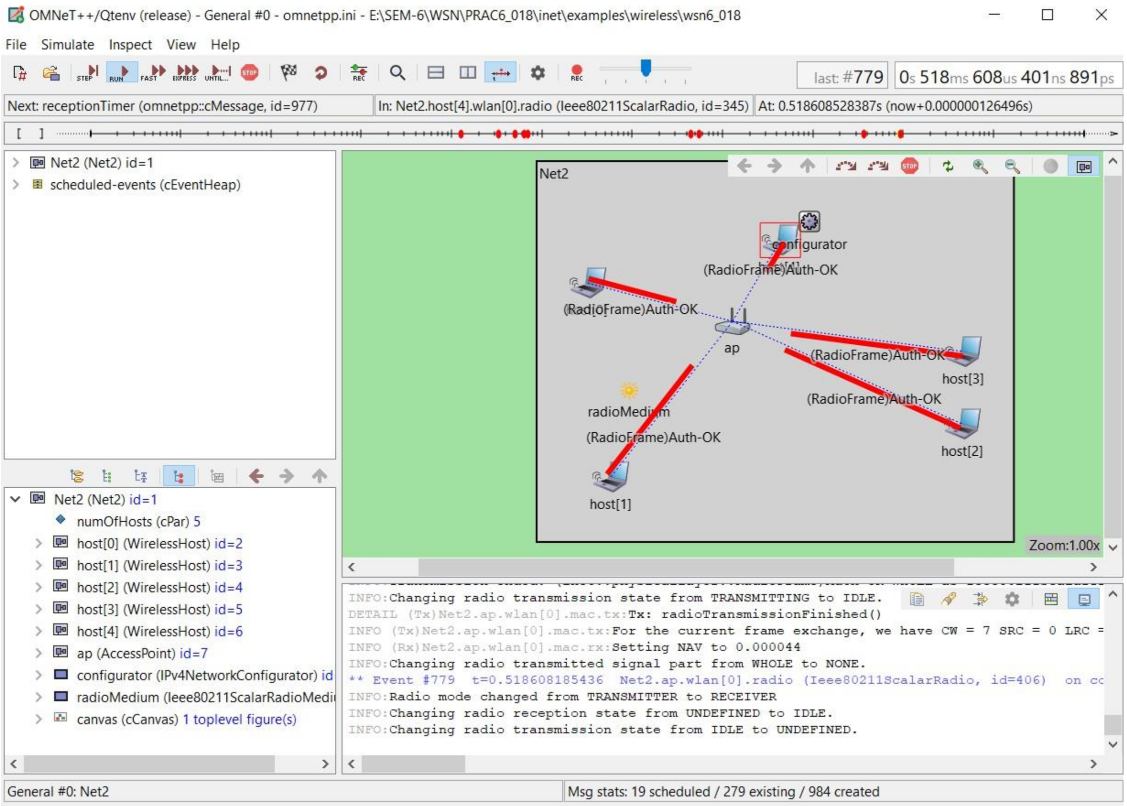
OMNeT++/Qtenv (release) - General #0 - omnetpp.ini - E:\SEM-6\WSN\PRAC6_018\inet\examples\wireless\wsn6_018
File Simulate Inspect View Help
next: #1
0s 000ms 000us 000ns 000ps
Next: (inet::RegisterTransportProtocolCommand, id=119)
In: Net2.host[0].networkLayer.ip (IPv4, id=18)
At: 0s (now+0s)
Net2 (Net2) id=1
scheduled-events (cEventHeap)
Net2.configurator (IPv4NetworkConfi
minLinkWeight (cPar) 0.001
config (cPar) <config>...</config>
assignAddresses (cPar) true
addStaticRoutes (cPar) true
addDefaultRoutes (cPar) true
addSubnetRoutes (cPar) true
optimizeRoutes (cPar) true
dumpTopology (cPar) false
Net2
configurator
host[4]
host[0]
ap
host[3]
radioMedium
host[2]
host[1]
Zoom:1.00x
Initializing module Net2.ap.wlan[0].radio.transmitter, stage 13
Initializing module Net2.ap.wlan[0].radio.receiver, stage 13
Initializing module Net2.configurator, stage 13
Initializing module Net2.radioMedium, stage 13
INFO (Ieee80211ScalarRadioMedium)Net2.radioMedium:Initialized (inet::physicallaye
Initializing module Net2.radioMedium.mediumLimitCache, stage 13
General #0: Net2
Msg stats: 21 scheduled / 129 existing / 129 created



OMNeT++/Qtenv (release) - General #0 - omnetpp.ini - E:\SEM-6\WSN\PRAC6_018\inet\examples\wireless\wsn6_018
File Simulate Inspect View Help
last: #24
0s 022ms 278us 261ns 960ps
Next: receptionTimer (omnetpp::cMessage, id=143)
In: Net2.host[4].wlan[0].radio (Ieee80211ScalarRadio, id=345)
At: 0.022278492876s (now+0.000000230916s)
Net2 (Net2) id=1
scheduled-events (cEventHeap)
Net2.configurator (IPv4NetworkConfigurato
minLinkWeight (cPar) 0.001
config (cPar) <config>...</config> at cc
assignAddresses (cPar) true
assignDisjunctSubnetAddresses (cPar) tr
addStaticRoutes (cPar) true
addDefaultRoutes (cPar) true
addSubnetRoutes (cPar) true
optimizeRoutes (cPar) true
dumpTopology (cPar) false
dumpLinks (cPar) false
dumpAddresses (cPar) false
Net2
configurator
(RadioFrame)Beacon
ap
host[3]
host[2]
radioMedium
host[1]
Zoom:1.00x
INFO:Changing radio transmission state from TRANSMITTING to IDLE.
DETAIL (Tx)Net2.ap.wlan[0].mac.tx:Tx: radioTransmissionFinished()
INFO (Tx)Net2.ap.wlan[0].mac.tx:For the current frame exchange, we have CW = 7 SRC = 0 LRC =
INFO (Tx)Net2.ap.wlan[0].mac.tx:Frame sequence finished
INFO:Changing radio transmitted signal part from WHOLE to NONE.
** Event #24 t=0.022278138552 Net2.ap.wlan[0].radio (Ieee80211ScalarRadio, id=406) on cor
INFO:Radio mode changed from TRANSMITTER to RECEIVER
INFO:Changing radio reception state from UNDEFINED to IDLE.
INFO:Changing radio transmission state from IDLE to UNDEFINED.
General #0: Net2
Msg stats: 17 scheduled / 136 existing / 150 created


OMNeT++/Qtenv (release) - General #0 - omnetpp.ini - E:\SEM-6\WSN\PRAC6_018\inet\examples\wireless\wsn6_018
File Simulate Inspect View Help
last: #302
0s 270ms 455us 801ns 473ps
Next: receptionTimer (omnetpp::cMessage, id=437)
In: Net2.host[0].wlan[0].radio (Ieee80211ScalarRadio, id=33)
At: 0.27045608629s (now+0.000000284817s)
Net2 (Net2) id=1
scheduled-events (cEventHeap)
Net2
configurator
host[4]
host[0]
(RadioFrame)ProbeReq
ap
host[3]
radioMedium
host[2]
host[1]
Zoom:1.00x
Net2.configurator (IPv4NetworkConfigurator)
minLinkWeight (cPar) 0.001
config (cPar) <config>...</config> at cc
assignAddresses (cPar) true
assignDisjunctSubnetAddresses (cPar) tr
addStaticRoutes (cPar) true
addDefaultRoutes (cPar) true
addSubnetRoutes (cPar) true
optimizeRoutes (cPar) true
dumpTopology (cPar) false
dumpLinks (cPar) false
dumpAddresses (cPar) false
INFO:Changing radio transmission state from TRANSMITTING to IDLE.
DETAIL (Tx)Net2.host[1].wlan[0].mac.tx:Tx: radioTransmissionFinished()
INFO (Tx)Net2.host[1].wlan[0].mac.tx:For the current frame exchange, we have CW = 7 SRC = 0
INFO (Tx)Net2.host[1].wlan[0].mac.tx:Frame sequence finished
INFO:Changing radio transmitted signal part from WHOLE to NONE.
** Event #302 t=0.270455369127 Net2.host[1].wlan[0].radio (Ieee80211ScalarRadio, id=111)
INFO:Radio mode changed from TRANSMITTER to RECEIVER
INFO:Changing radio reception state from UNDEFINED to IDLE.
INFO:Changing radio transmission state from IDLE to UNDEFINED.
General #0: Net2
Msg stats: 17 scheduled / 181 existing / 444 created


OMNeT++/Qtenv (release) - General #0 - omnetpp.ini - E:\SEM-6\WSN\PRAC6_018\inet\examples\wireless\wsn6_018
File Simulate Inspect View Help
last: #1'179
0s 544ms 841us 337ns 325ps
Next: receptionTimer (omnetpp::cMessage, id=1457)
In: Net2.ap.wlan[0].radio (Ieee80211ScalarRadio, id=406)
At: 0.544841876155s (now+0.00000053883s)
Net2 (Net2) id=1
scheduled-events (cEventHeap)
Net2
configurator
host[4]
(RadioFrame)Assoc
host[0]
ap
host[3]
radioMedium
host[2]
host[1]
Zoom:1.00x
numOfHosts (cPar) 5
host[0] (WirelessHost) id=2
host[1] (WirelessHost) id=3
host[2] (WirelessHost) id=4
host[3] (WirelessHost) id=5
host[4] (WirelessHost) id=6
ap (AccessPoint) id=7
configurator (IPv4NetworkConfigurator) id
radioMedium (Ieee80211ScalarRadioMedi
canvas (cCanvas) 1 toplevel figure(s)
INFO:Changing radio transmission state from TRANSMITTING to IDLE.
DETAIL (Tx)Net2.host[0].wlan[0].mac.tx:Tx: radioTransmissionFinished()
INFO (Tx)Net2.host[0].wlan[0].mac.tx:For the current frame exchange, we have CW = 7 SRC = 0
INFO (Rx)Net2.host[0].wlan[0].mac.rx:Setting NAV to 0.000044
INFO:Changing radio transmitted signal part from WHOLE to NONE.
** Event #1179 t=0.544841331235 Net2.host[0].wlan[0].radio (Ieee80211ScalarRadio, id=33)
INFO:Radio mode changed from TRANSMITTER to RECEIVER
INFO:Changing radio reception state from UNDEFINED to IDLE.
INFO:Changing radio transmission state from IDLE to UNDEFINED.
General #0: Net2
Msg stats: 19 scheduled / 358 existing / 1464 created



OMNeT++/Qtenv (release) - General #0 - omnetpp.ini - E:\SEM-6\WSN\PRAC6_018\inet\examples\wireless\wsn6_018
File Simulate Inspect View Help
STEP RUN FAST EXPRESS UNTIL... STOP
last: #779
0s 518ms 608us 401ns 891ps
Next: receptionTimer (omnetpp::cMessage, id=977)
In: Net2.host[4].wlan[0].radio (Ieee80211ScalarRadio, id=345)
At: 0.518608528387s (now+0.000000126496s)
Net2 (Net2) id=1
scheduled-events (cEventHeap)
Net2
configurator
(RadioFrame)Auth-OK
ap
host[3]
host[2]
host[1]
radioMedium
Zoom:1.00x
Net2 (Net2) id=1
numOfHosts (cPar) 5
host[0] (WirelessHost) id=2
host[1] (WirelessHost) id=3
host[2] (WirelessHost) id=4
host[3] (WirelessHost) id=5
host[4] (WirelessHost) id=6
ap (AccessPoint) id=7
configurator (IPv4NetworkConfigurator) id
radioMedium (Ieee80211ScalarRadioMedi
canvas (cCanvas) 1 toplevel figure(s)
INFO:Changing radio transmission state from TRANSMITTING to IDLE.
DETAIL (Tx)Net2.ap.wlan[0].mac.tx:Tx: radioTransmissionFinished()
INFO (Tx)Net2.ap.wlan[0].mac.tx:For the current frame exchange, we have CW = 7 SRC = 0 LRC =
INFO (Rx)Net2.ap.wlan[0].mac.rx:Setting NAV to 0.000044
INFO:Changing radio transmitted signal part from WHOLE to NONE.
** Event #779 t=0.518608185436 Net2.ap.wlan[0].radio (Ieee80211ScalarRadio, id=406) on cc
INFO:Radio mode changed from TRANSMITTER to RECEIVER
INFO:Changing radio reception state from UNDEFINED to IDLE.
INFO:Changing radio transmission state from IDLE to UNDEFINED.
General #0: Net2
Msg stats: 19 scheduled / 279 existing / 984 created

Wireless Sensor Networks & Mobile Communication

Practical No.7

DEPARTMENT OF COMPUTER SCIENCE

Name:	Kimaya Naik	**Roll Number**	TCS2324048
Paper Code:	SIUSCS61	**Class**	TYBSc(Computer Science)
Topic:	Mac Protocol	**Batch**	I
Date:	06/2/24	**Practical No**	7

A) AIM: Create a Mac protocol simulation implementation for wireless Sensor Networks

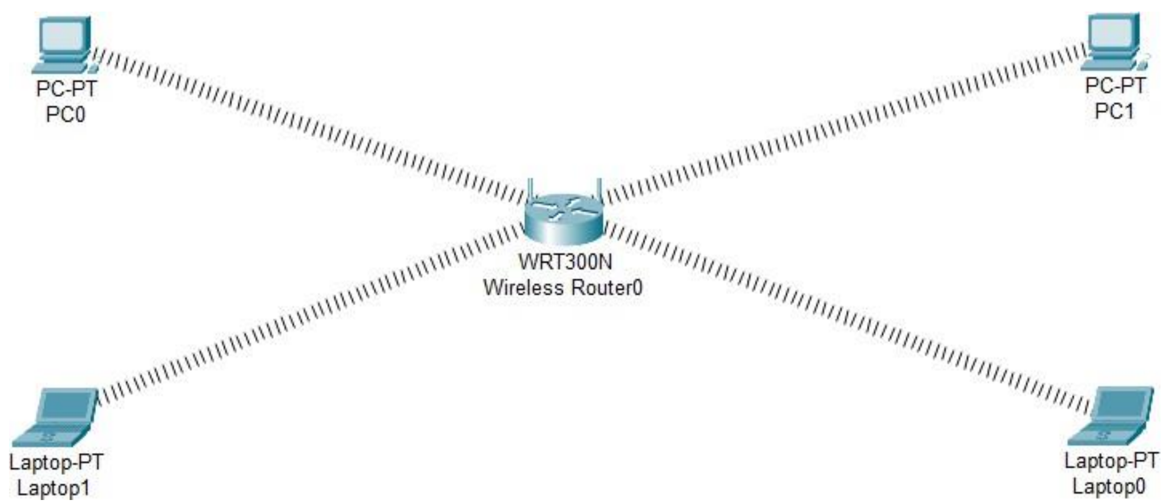
B) DESCRIPTION:

The Medium Access Control (MAC) layer is a sublayer of the data link layer that deals with the protocol access to the physical network medium. MAC protocols control how devices in a network access and use the shared communication medium to avoid collisions and ensure efficient communication.

1. **CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance):** Commonly used in wireless networks, CSMA/CA involves nodes checking for the channel's availability before transmitting to avoid collisions.
2. **CSMA/CD (Carrier Sense Multiple Access with Collision Detection):** Historically used in wired networks like Ethernet, CSMA/CD detects collisions and takes steps to resolve them.
3. **TDMA (Time Division Multiple Access):** In TDMA, time is divided into slots, and each node is assigned a specific time slot for transmission. This approach is often used in satellite communication and cellular networks.
4. **FDMA (Frequency Division Multiple Access):** FDMA allocates different frequency bands to different nodes for simultaneous transmission, commonly used in analog communication systems.
5. **CDMA (Code Division Multiple Access):** CDMA assigns a unique code to each node, allowing multiple nodes to transmit simultaneously on the same frequency band. It's widely used in digital cellular networks.

C) NETWORK TOPOLOGY, CONFIGURATIONS AND OUTPUT:

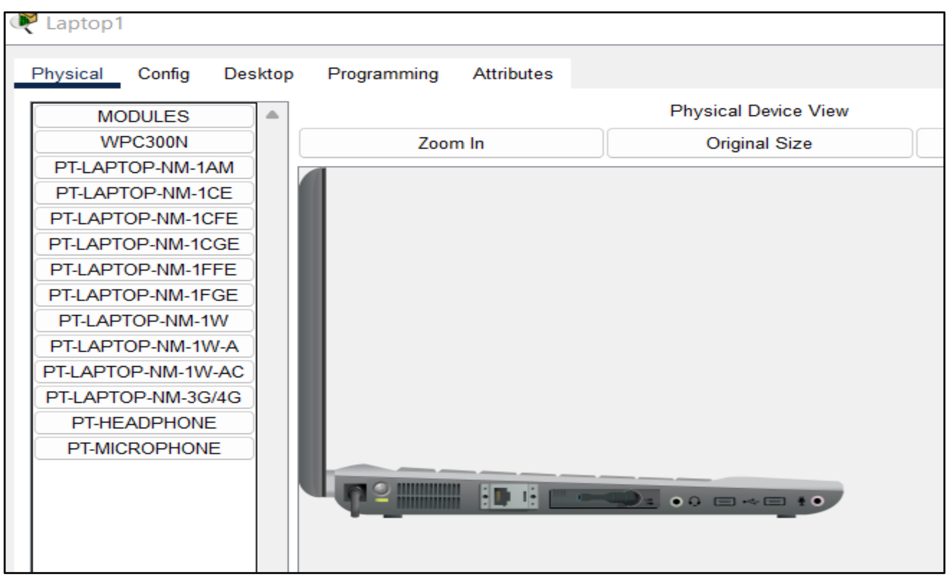
Network topology:



Configurations:

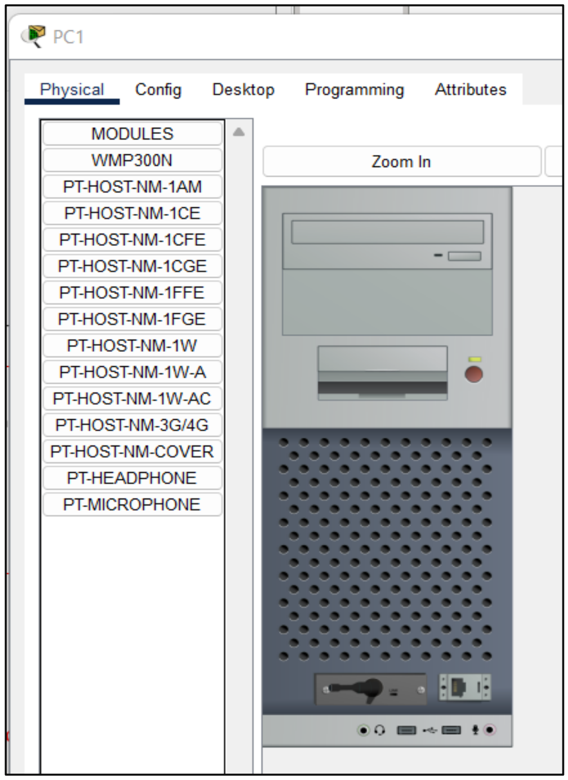
Configurations done on Laptops:

Physical configuration: Adding the wireless interface



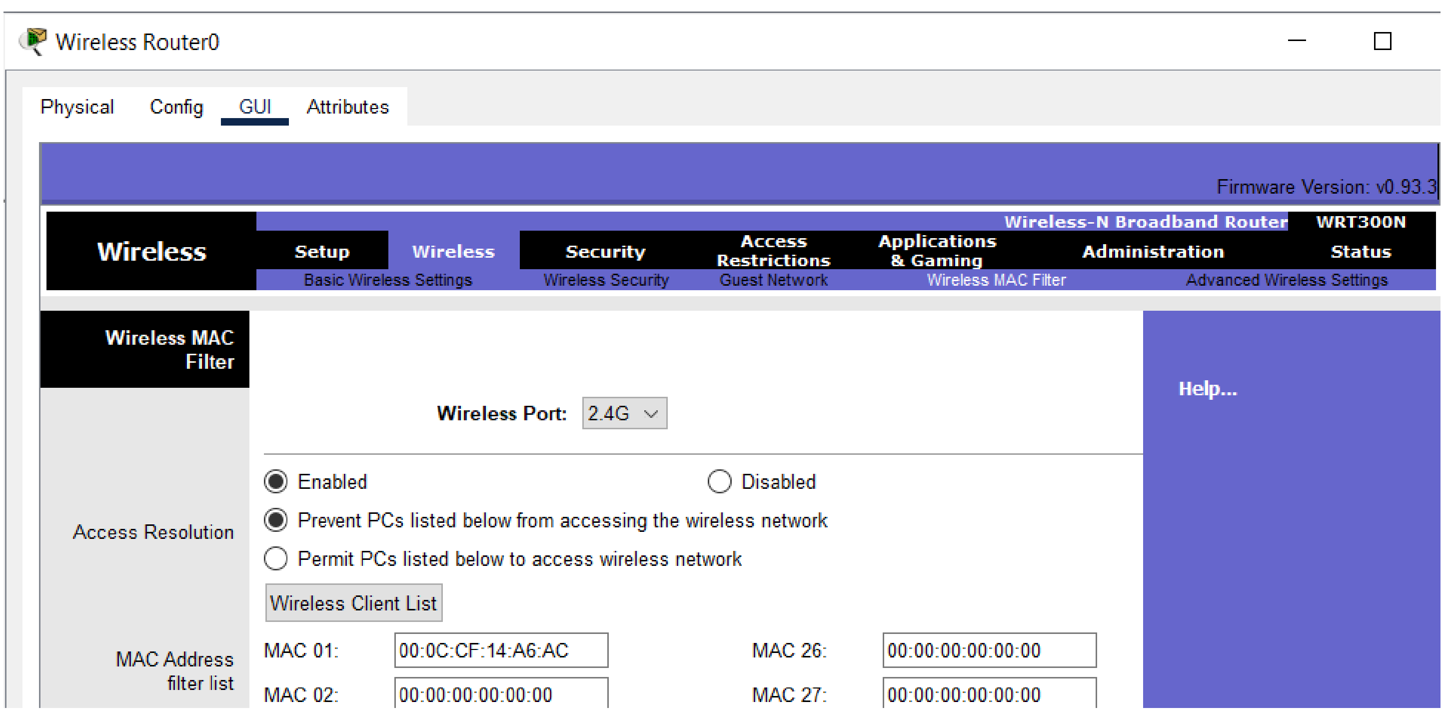
Configuration done on PCs:

Physical configuration: Adding the wireless interface

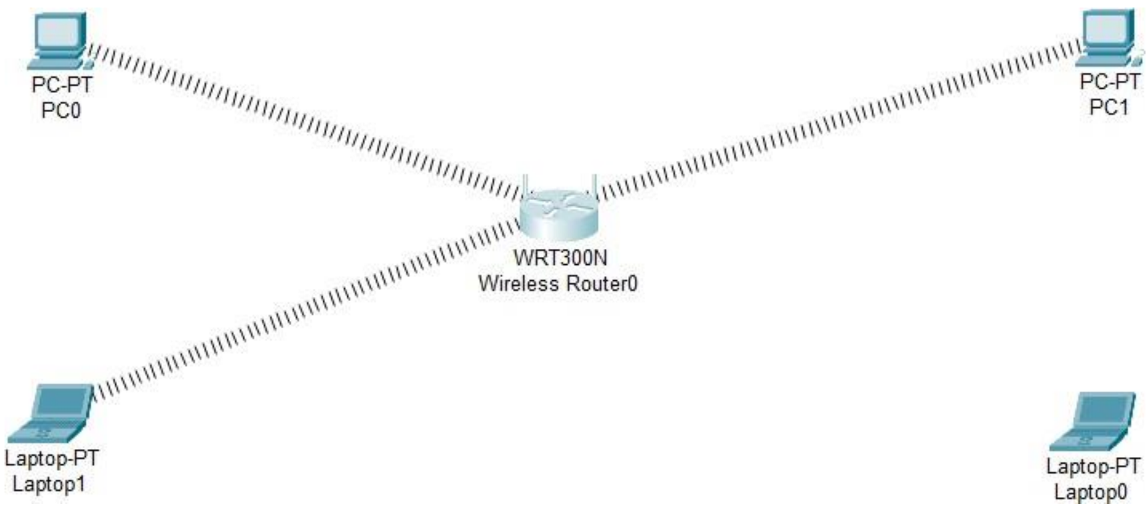


Configuration done on Router:

Adding the MAC Address of Pcs in the MAC Address filter list(present in the GUI)



Output



PC-PT
PC0
PC-PT
PC1
WRT300N
Wireless Router0
Laptop-PT
Laptop1
Laptop-PT
Laptop0

Wireless Sensor Networks & Mobile Communication

Practical No.8

DEPARTMENT OF COMPUTER SCIENCE

Name:	Kimaya Naik	Roll Number	TCS2324048
Paper Code:	SIUSCS61	Class	TYBSc(Computer Science)
Topic:	Mobile Adhoc Network	Batch	I
Date:	06/2/24	Practical No	8

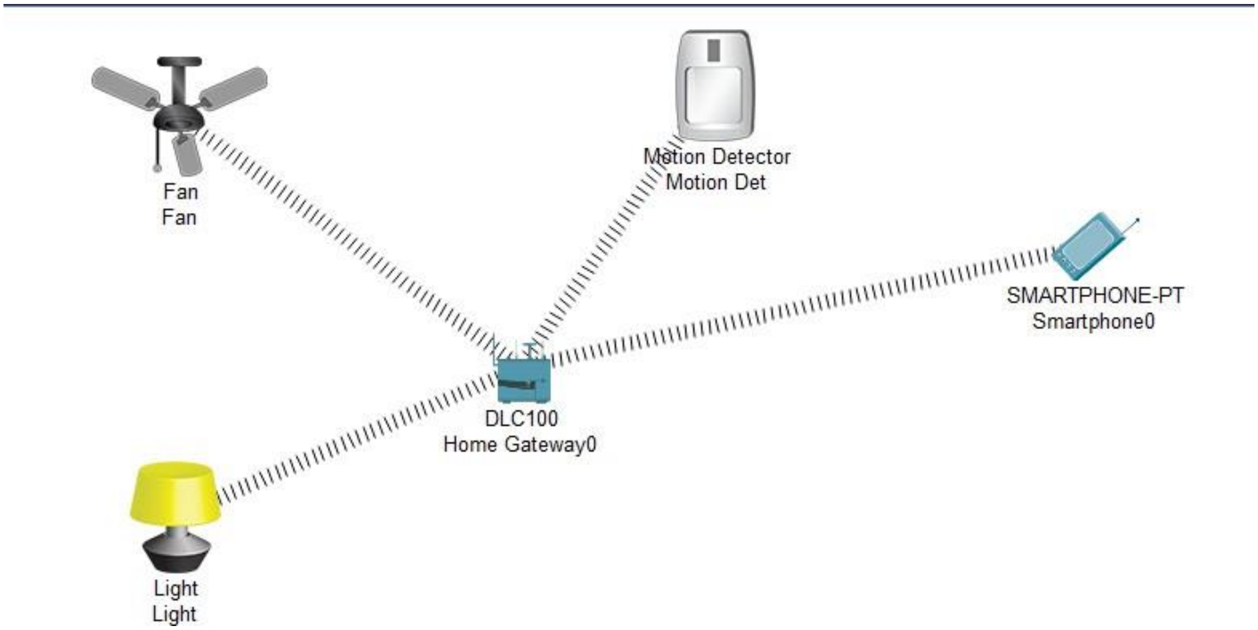
A) AIM: Simulate Mobile Adhoc Network with Directional Antenna

B) DESCRIPTION:

MANET stands for Mobile Adhoc Network also called a wireless Adhoc network or Adhoc wireless network that usually has a routable networking environment on top of a Link Layer ad hoc network.. They consist of a set of mobile nodes connected wirelessly in a self-configured, self-healing network without having a fixed infrastructure. MANET nodes are free to move randomly as the network topology changes frequently. Each node behaves as a router as they forward traffic to other specified nodes in the network.

C) NETWORK TOPOLOGY, CONFIGURATIONS AND OUTPUT:

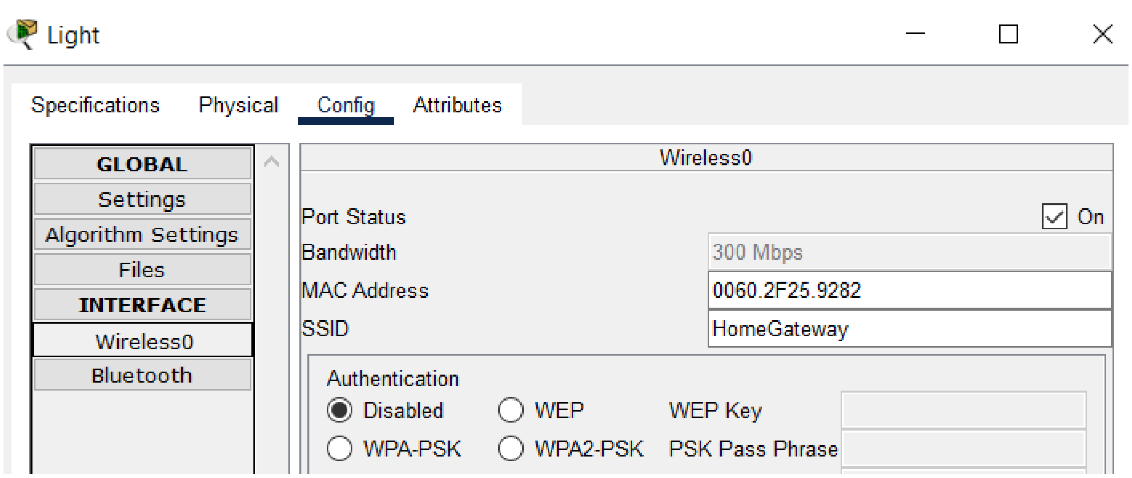
Network topology:



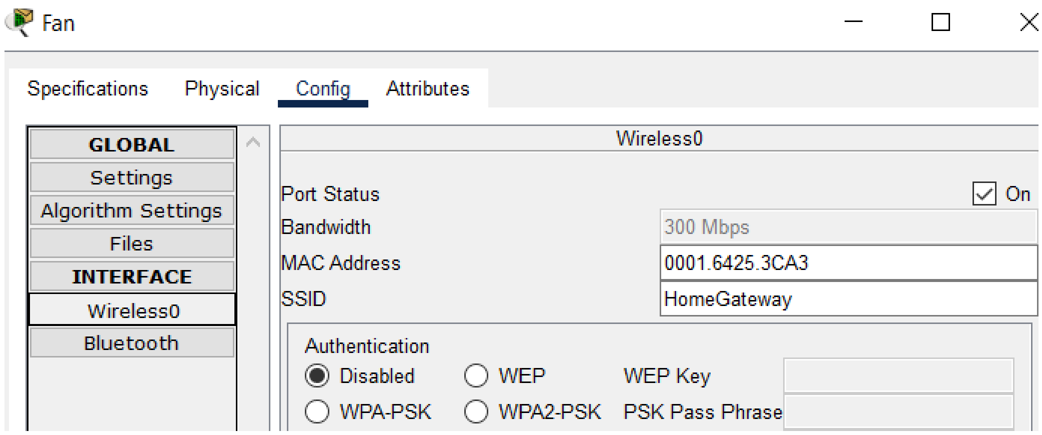
Configurations:

Copy the SSID of Homegateway and paste it to the ssid on fan, light and motion detector.

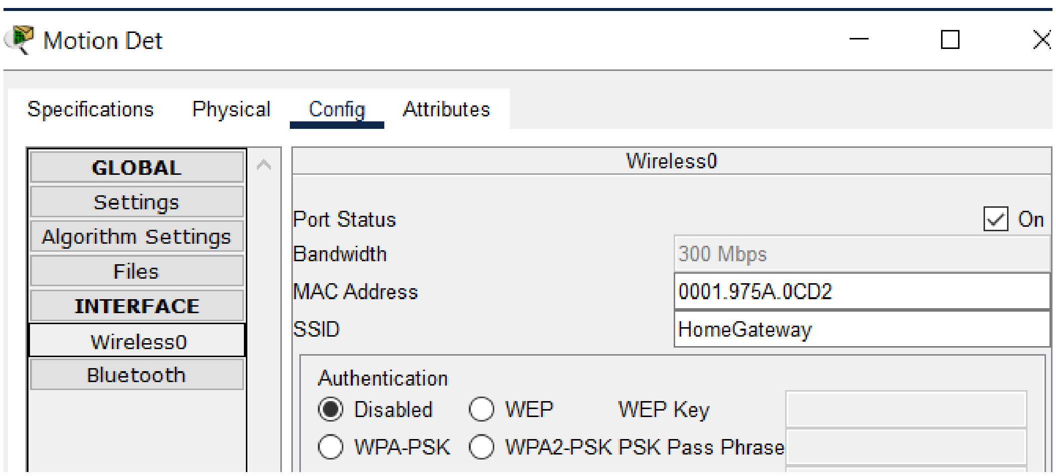
For Light:



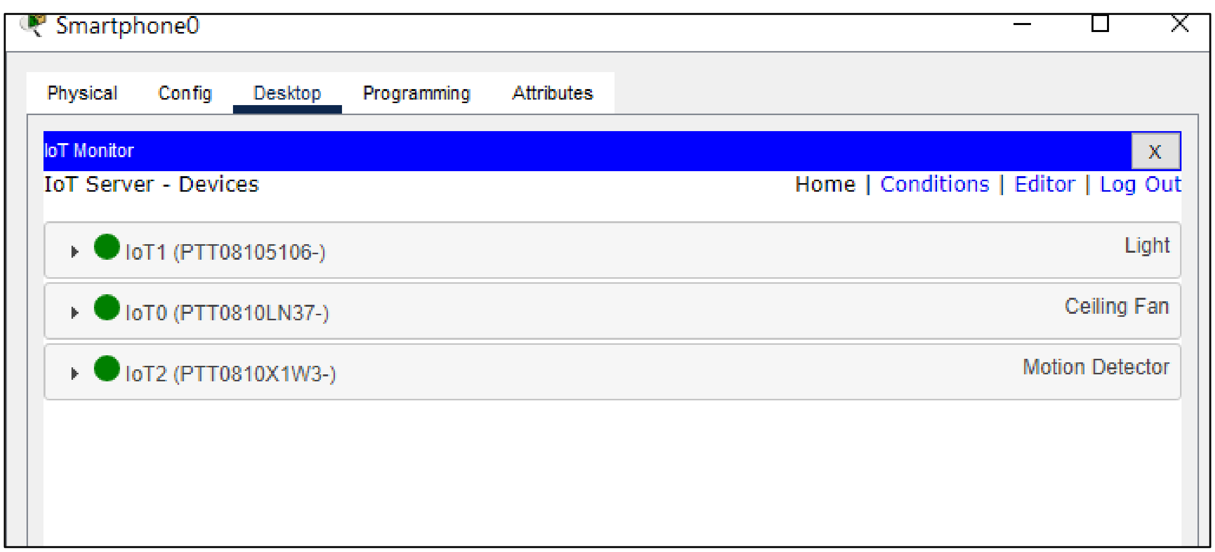
For fan:

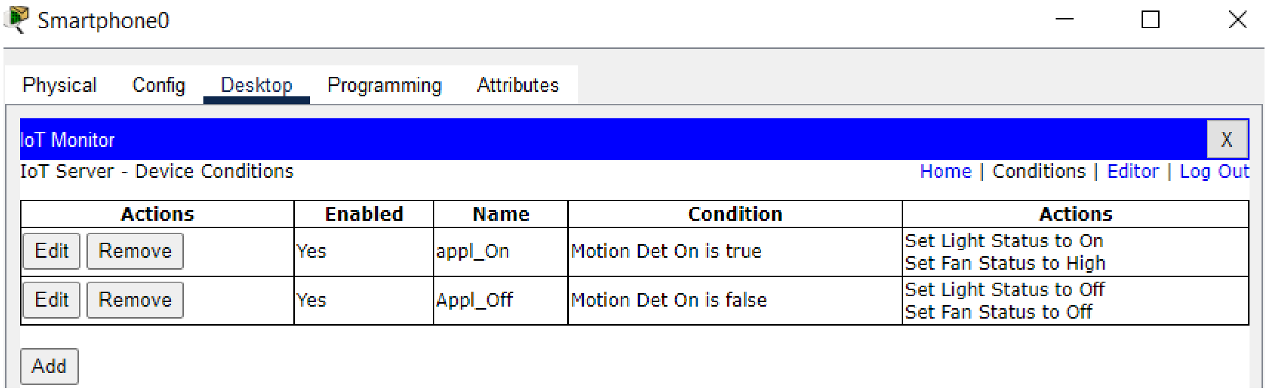


For Motion detector:



For smartphone(Go to Desktop>>IOT monitor)

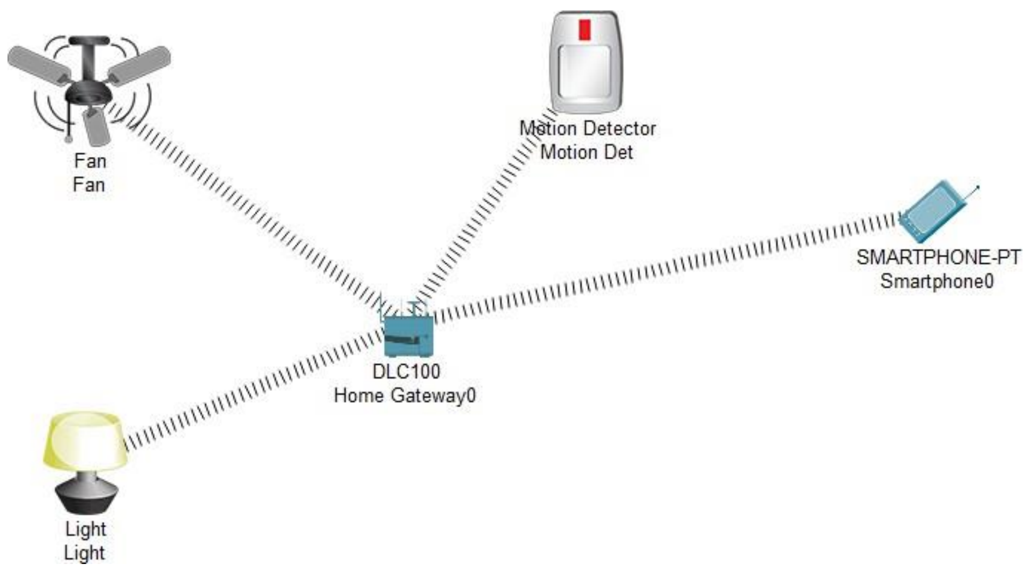


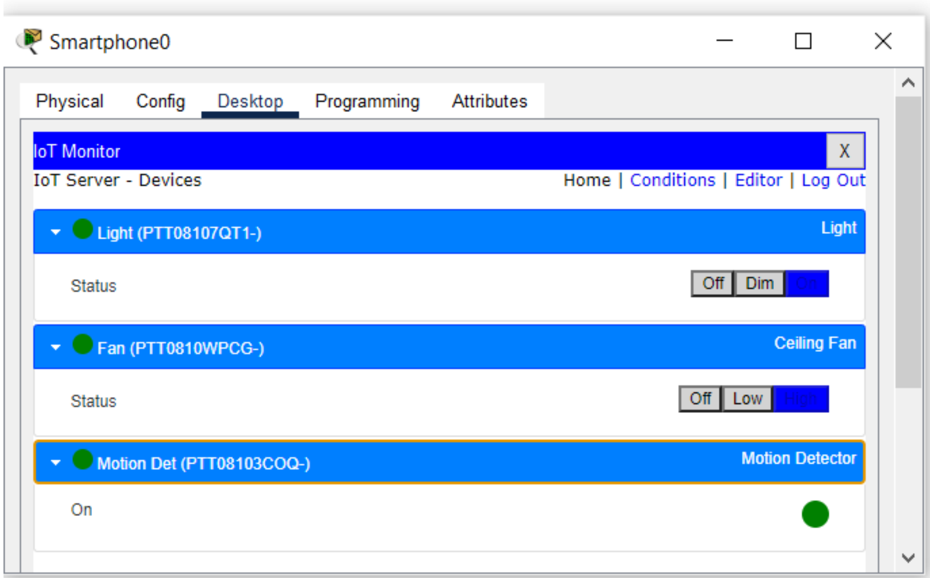


Actions	Enabled	Name	Condition	Actions
Edit Remove	Yes	appl_On	Motion Det On is true	Set Light Status to On Set Fan Status to High
Edit Remove	Yes	Appl_Off	Motion Det On is false	Set Light Status to Off Set Fan Status to Off

Output :

When motion detector is on(press alt with motion detector):





Wireless Sensor Networks & Mobile Communication

Practical No.9

DEPARTMENT OF COMPUTER SCIENCE

Name:	Kimaya Naik	**Roll Number**	TCS2324048
Paper Code:	SIUSCS61	**Class**	TYBSc(Computer Science)
Topic:	Mobile Network	**Batch**	I
Date:	06/2/24	**Practical No**	9

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

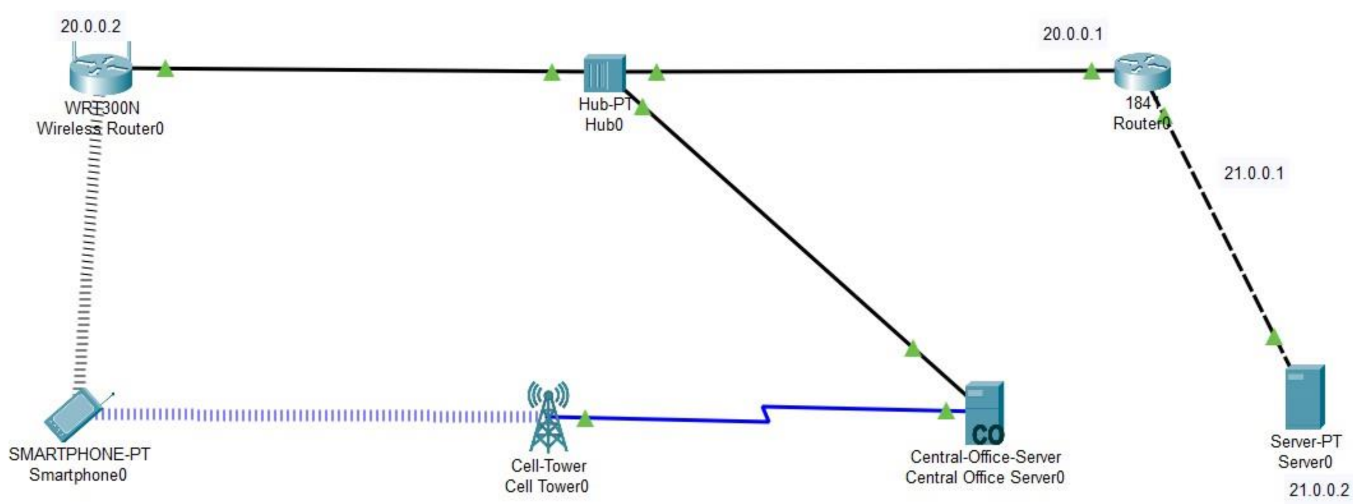
B) DESCRIPTION:

An ad hoc network is one that is spontaneously formed when devices connect and communicate with each other. The term ad hoc is a Latin word that literally means "for this," implying improvised or impromptu.

Ad hoc networks are mostly **wireless local area networks** (WLANs). The devices communicate with each other directly instead of relying on a base station or access points as in wireless **LANs** for data transfer co-ordination. Each device participates in routing activity, by determining the route using the routing algorithm and forwarding data to other devices via this route.

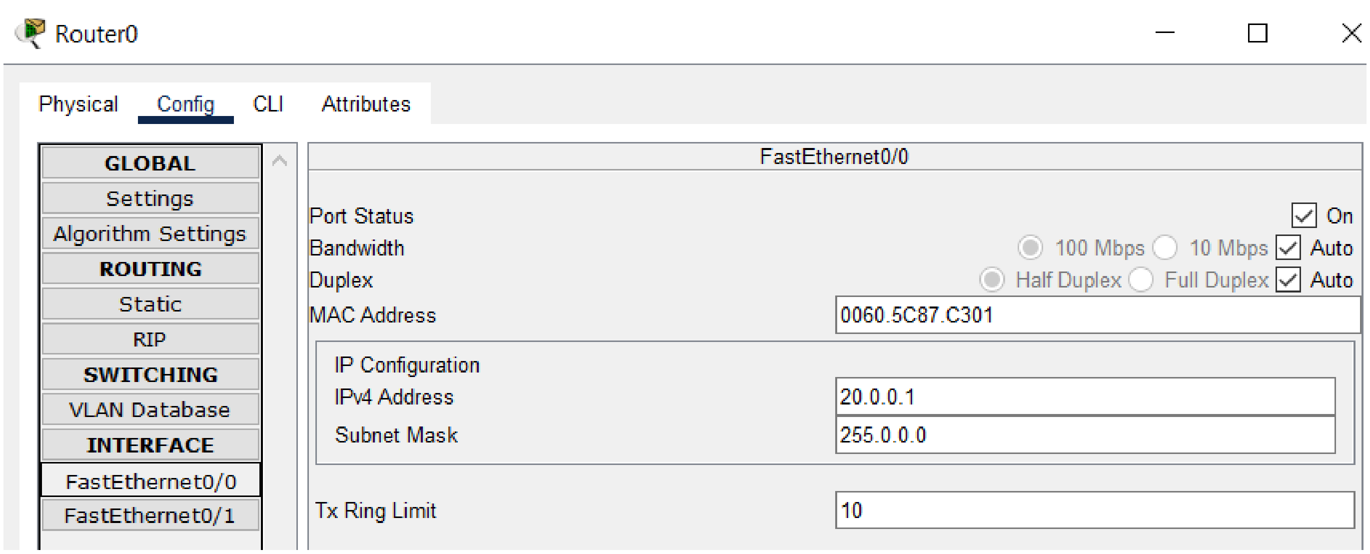
C) NETWORK TOPOLOGY, CONFIGURATIONS AND OUTPUT:

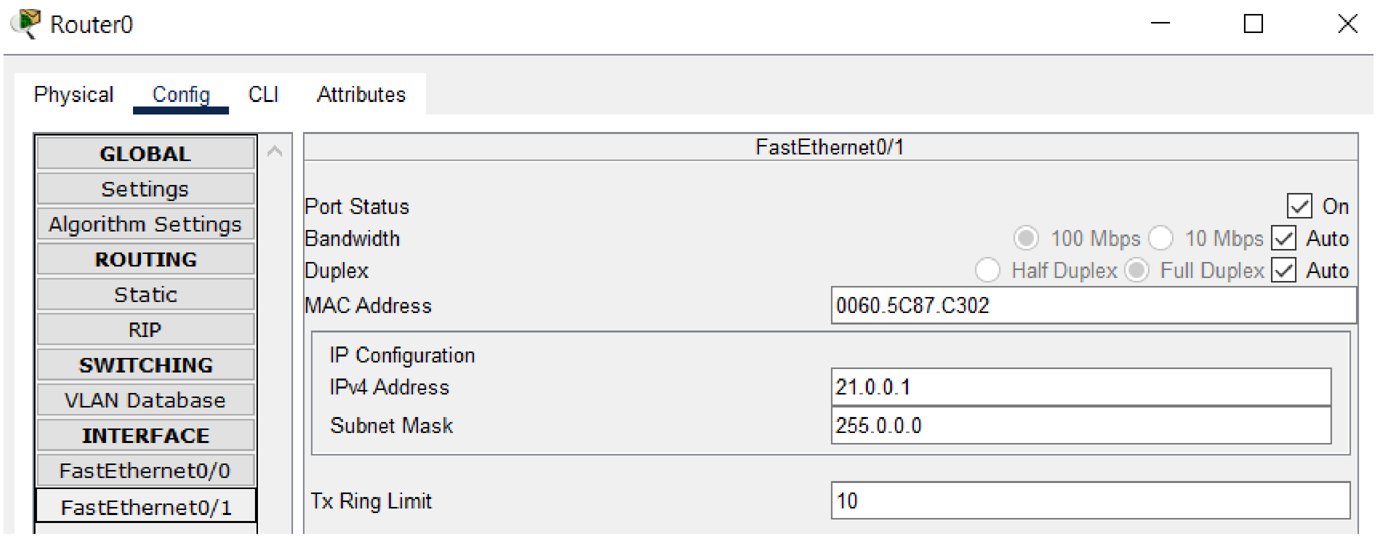
Network topology



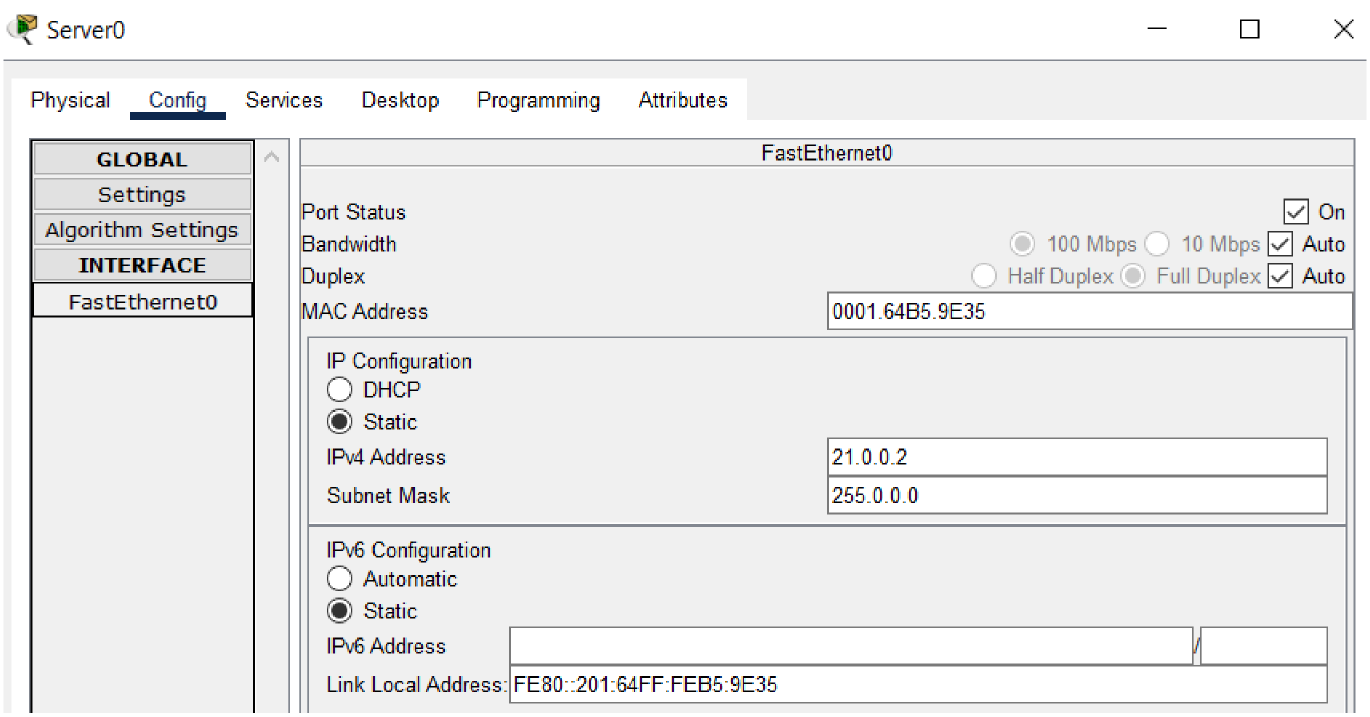
Configurations:

For 1841 Router:

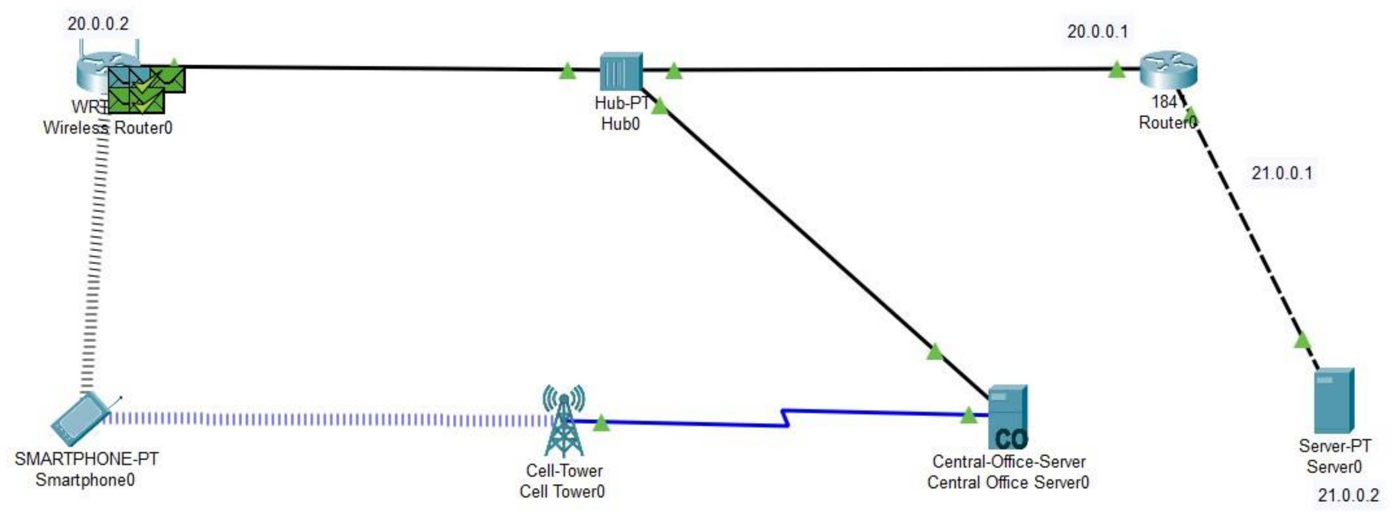




For Server-Pt:



Output



20.0.0.2
WRT
Wireless Router0
Hub-PT
Hub0
20.0.0.1
1841
Router0
21.0.0.1
SMARTPHONE-PT
Smartphone0
Cell-Tower
Cell Tower0
CO
Central-Office-Server
Central Office Server0
Server-PT
Server0
21.0.0.2