Name of Instructor: Jescia D’cruz

Index Page

Name of Instructor: Jescia D’cruz

Wireless Sensor Networks & Mobile Communication

Practical No.1

DEPARTMENT OF COMPUTER SCIENCE

Name:	Ajay Kumar Uthaya Kumar	**Roll Number**	TCS2324002
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. (For Eg. Sensors, Node(Sensor mote), Base Station, Graphical User Interface

B) DESCRIPTION:

A sensor is a device or module that detects and responds to some type of input from the physical environment. It translates physical phenomena like light, heat, sound, motion, etc., into measurable signals, often electrical, which can be further processed and analysed.

C) NETWORK TOPOLOGY, CONFIGURATIONS AND OUTPUT:

Theory

A sensor node is a self-contained unit equipped with sensors, processing capabilities, and communication interfaces. It's typically part of a larger sensor network where multiple sensor nodes collaborate to collect and transmit data. The hardware components of a sensor node include:

a. Sensors: These are the primary components responsible for detecting environmental parameters such as temperature, humidity, pressure, light intensity, etc. Sensors come in various types and technologies, each suited for specific applications.

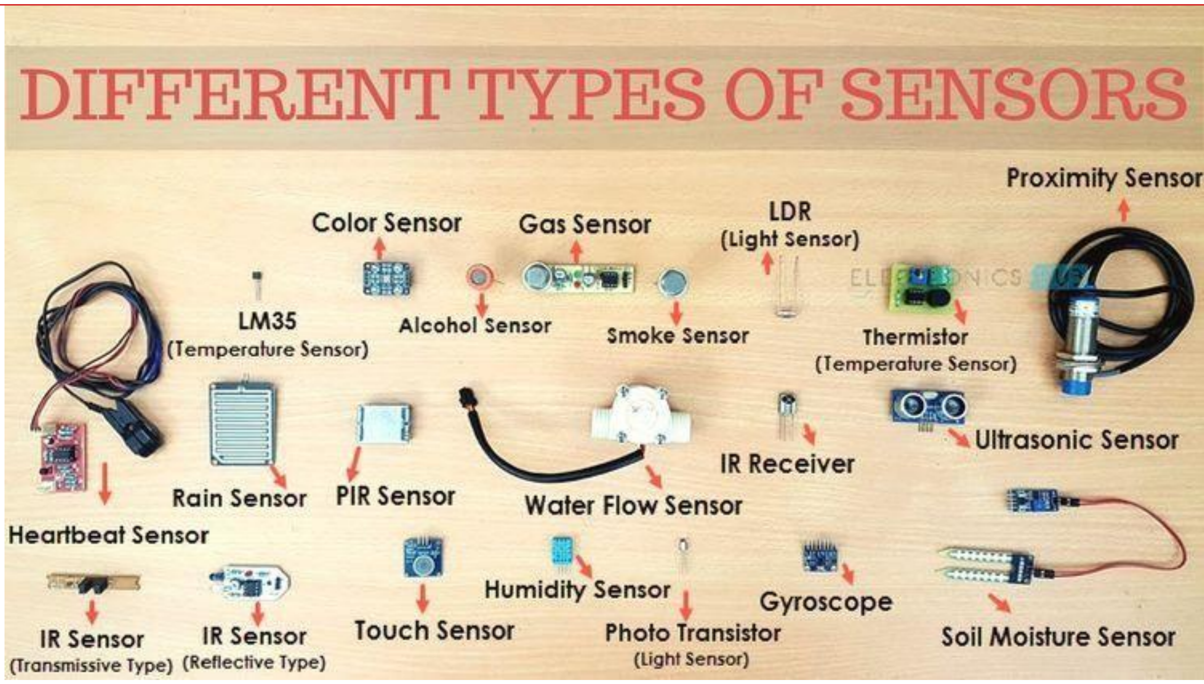
b. Node (Sensor Mote): The node or sensor mote is the central component housing the sensors, processing unit, memory, and communication interfaces. It's often a compact and energy-efficient device designed for deployment in diverse environments, including remote and harsh locations.

c. Base Station: The base station serves as the central hub or gateway for collecting data from multiple sensor nodes. It typically has more processing power and storage capacity compared to individual sensor nodes. The base station aggregates data, performs further analysis if required, and may also facilitate communication with external networks or systems.

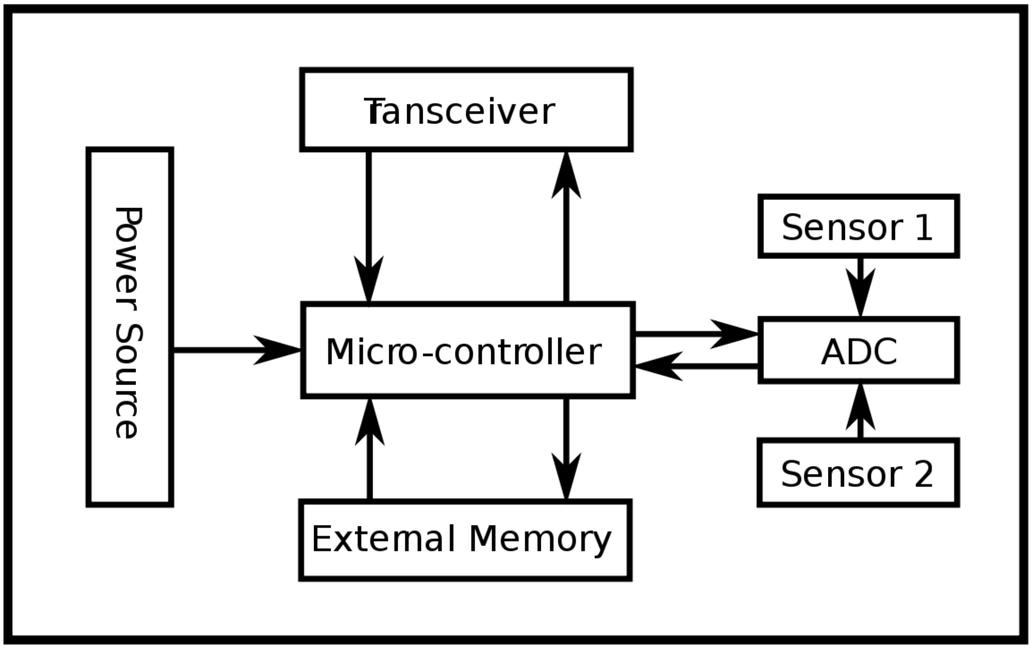
d. Graphical User Interface (GUI): The GUI provides a user-friendly interface for interacting with the sensor network. It allows users to visualize real-time data, configure settings, monitor sensor statuses, and analyze collected data through intuitive graphical representations. GUIs can be desktop applications, web-based dashboards, or mobile apps, depending on the requirements and accessibility preferences.

Output

1. Sensors

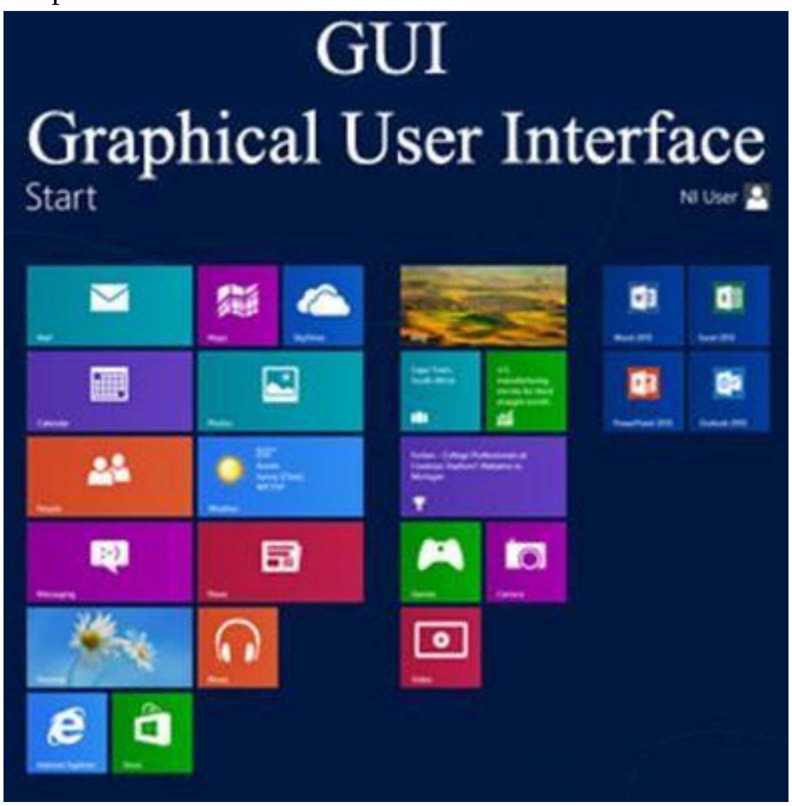


2. Node(Sensor Mote)



3. Base Station

4. Graphical User Interface



Wireless Sensor Networks & Mobile Communication

Practical No.2A

DEPARTMENT OF COMPUTER SCIENCE

Name:	Ajay Kumar Uthaya Kumar	**Roll Number**	TCS2324002
Paper Code:	SIUSCS61	**Class**	TYBSc(Computer Science)
Topic:	TinyOS computational concept	**Batch**	I
Date:	08/1/24	**Practical No**	2A

A) AIM: Exploring and Understanding TinyOS computational concepts – Events, Commands and Task.

- nesC model
- nesC Components

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. It began as a collaboration between the University of California, Berkeley, Intel Research, and Crossbow Technology, was released as free and open-source software under a BSD license, and has since grown into an international consortium, the TinyOS Alliance. TinyOS has been used in space, being implemented in ESTCube-1.

C) NETWORK TOPOLOGY, CONFIGURATIONS AND OUTPUT:

Theory

TinyOS is an open-source operating system designed specifically for embedded systems and wireless sensor networks. It follows a component-based programming model and is primarily programmed using the nesC (network embedded systems C) language. Understanding the computational concepts in TinyOS, including Events, Commands, and Tasks, is crucial for effectively developing applications for resource-constrained devices.

1. Events
    - In TinyOS, Events are asynchronous signals or notifications that indicate the occurrence of specific conditions or events within the system.
    - Events are typically generated by hardware interrupts, software components, or other external stimuli.
    - They serve as triggers for initiating actions or executing code in response to certain events.
    - Event handlers, also known as event-driven routines or functions, are registered to handle specific events.
    - When an event occurs, the corresponding event handler is invoked to process the event.
    - Events are essential for implementing reactive behavior and event-driven programming paradigms in TinyOS applications.
    - Example: An event might be generated when a sensor reading is available, when a packet is received over the wireless network, or when a timer expires.
2. Commands
    - Commands in TinyOS represent synchronous operations or actions that can be invoked by software components to perform specific tasks or interact with hardware peripherals.
    - Unlike events, commands are typically executed in a synchronous manner, meaning that the caller may block until the command execution completes.
    - Commands encapsulate functionality and behavior that can be reused across different parts of the application.
    - Components expose a set of commands that can be invoked by other components or application code to perform various operations.
    - Example: Commands might include functions to configure sensor parameters, transmit data over the network, or control actuators in the environment.
3. Tasks
    - Tasks in TinyOS represent units of concurrent execution or lightweight threads of control within the system.
    - Tasks are used to perform background processing, handle asynchronous events, and execute command sequences in a sequential manner.
    - Each task typically corresponds to a specific functionality or operation within the application. Tasks are scheduled and managed by the TinyOS scheduler, which ensures that tasks are

executed in a timely and efficient manner, considering resource constraints and system priorities.

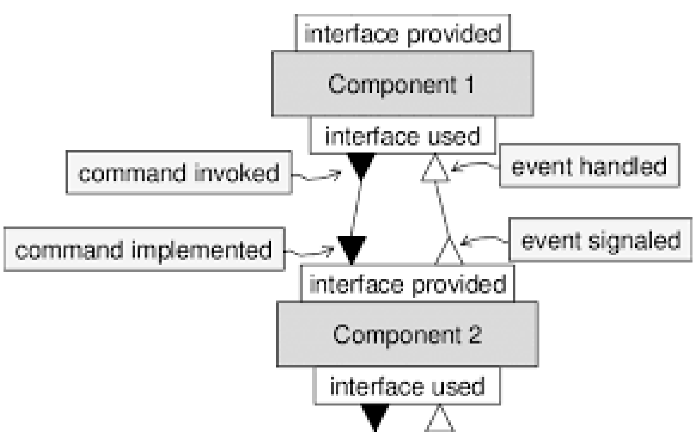
- Tasks can be initiated by events, commands, or other tasks, and they may communicate with each other through shared data structures or message passing.
- Example: A task might be responsible for periodically sampling sensor data, processing incoming packets from the network, or performing periodic maintenance tasks.

nesC Model: The nesC programming language is specifically designed for programming embedded systems and sensor networks. It follows a component-based model, where software functionality is organized into reusable and composable components. Components encapsulate related functionality, including event handlers, command interfaces, task implementations, and internal state. Components can interact with each other through well-defined interfaces, exchanging events, commands, and data. The nesC compiler translates nesC code into efficient C code, which is then compiled and executed on the target platform. nesC promotes modular design, code reuse, and maintainability, making it well-suited for developing complex applications for resource-constrained devices.

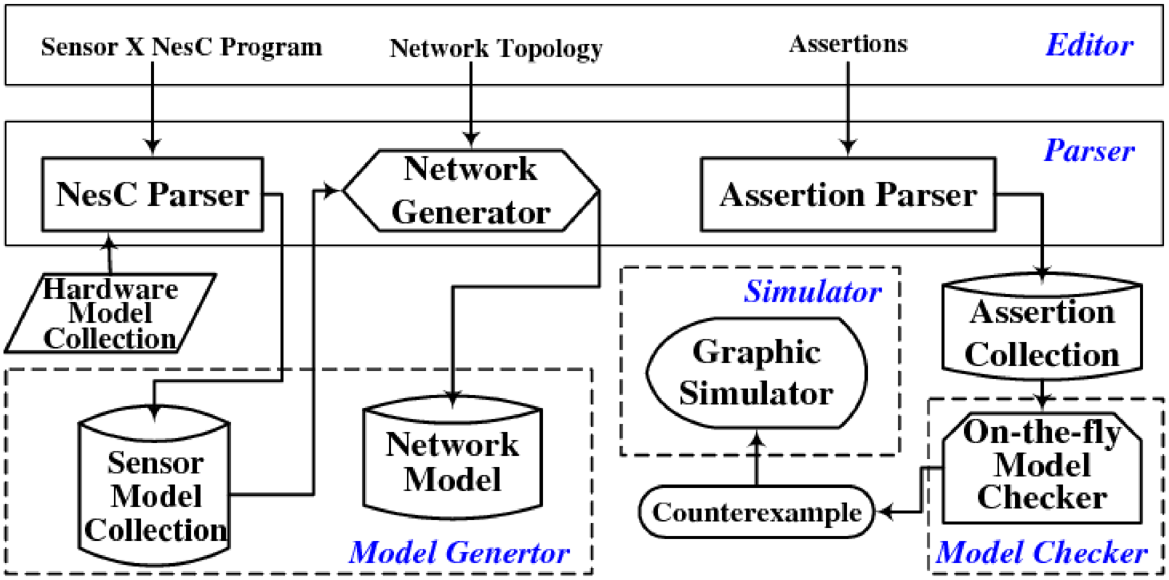
nesC Components: Components in nesC represent modular units of functionality that can be composed and interconnected to build larger applications. Each component encapsulates a coherent set of functionality, including event handlers, command interfaces, tasks, and internal state. Components define interfaces that specify the events they emit, the commands they provide, and the configuration parameters they accept. Components can be instantiated multiple times with different configurations, allowing for flexible and configurable application design. Components can communicate with each other through interface connections, enabling message passing, event propagation, and command invocation. Components can be developed independently, tested in isolation, and then integrated into larger systems, promoting code modularity and reusability. Example: A sensor component might expose events for new sensor readings, commands for configuring sensor parameters, and tasks for periodic data sampling.

Output

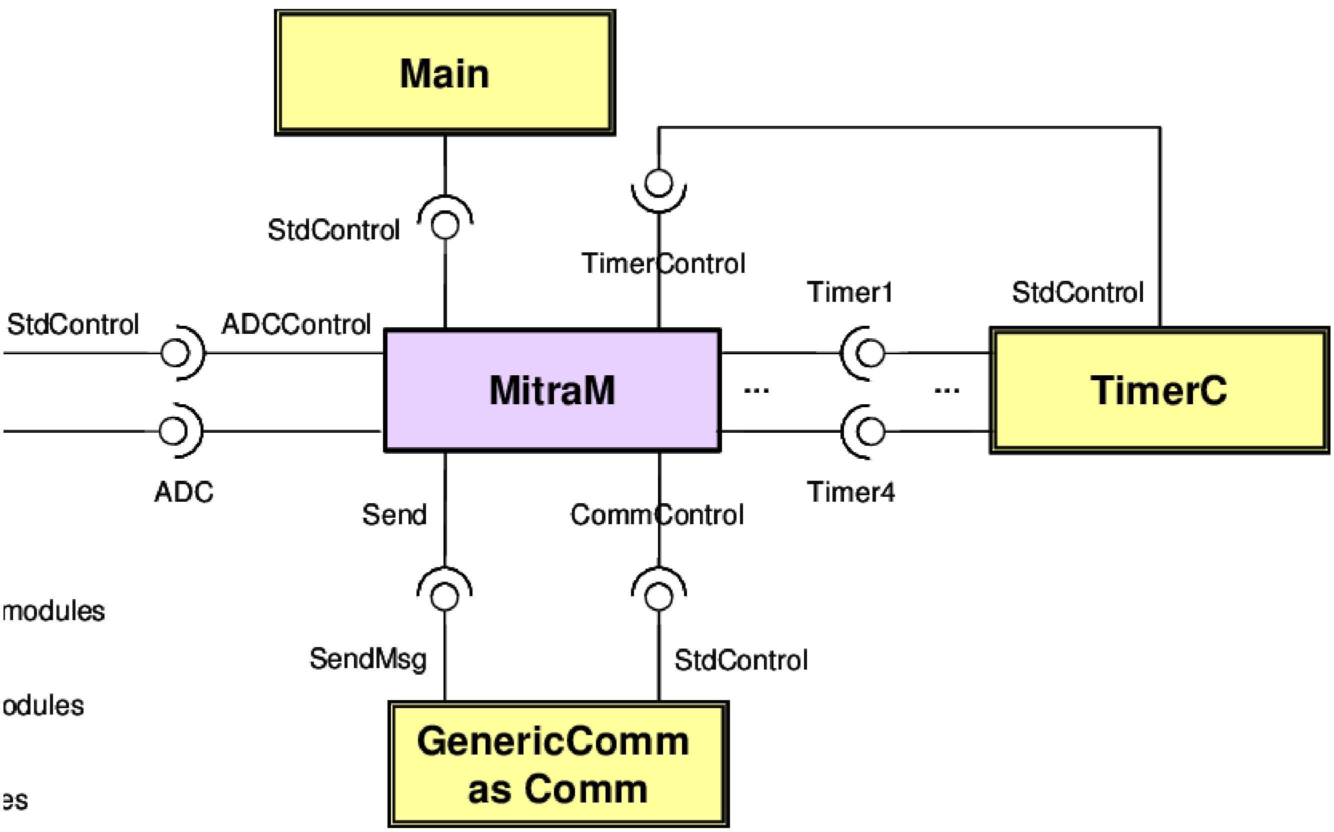
TinyOS



nesC model



nesC Component



Wireless Sensor Networks & Mobile Communication

Practical No.2B

DEPARTMENT OF COMPUTER SCIENCE

Name:	Ajay Kumar Uthaya Kumar	**Roll Number**	TCS2324002
Paper Code:	SIUSCS61	**Class**	TYBSc(Computer Science)
Topic:	Tossim computational concept	**Batch**	I
Date:	08/1/24	**Practical No**	2B

A) AIM: Exploring and Understanding Tossim computational concepts – Events, Commands and Task.

B) DESCRIPTION:

Tossim Simulator is a discrete event simulator framework for TinyOS wireless sensor network. Instead of NS2, we implement Tossim, which captures the network behavior and attraction based on bit granularity, not on the packet level. It operate in a sensor network called motes, which deployed from IEEE based papers. Motes are referred to as tiny sensing and computation device for limited communication, computation, and also energy resources.

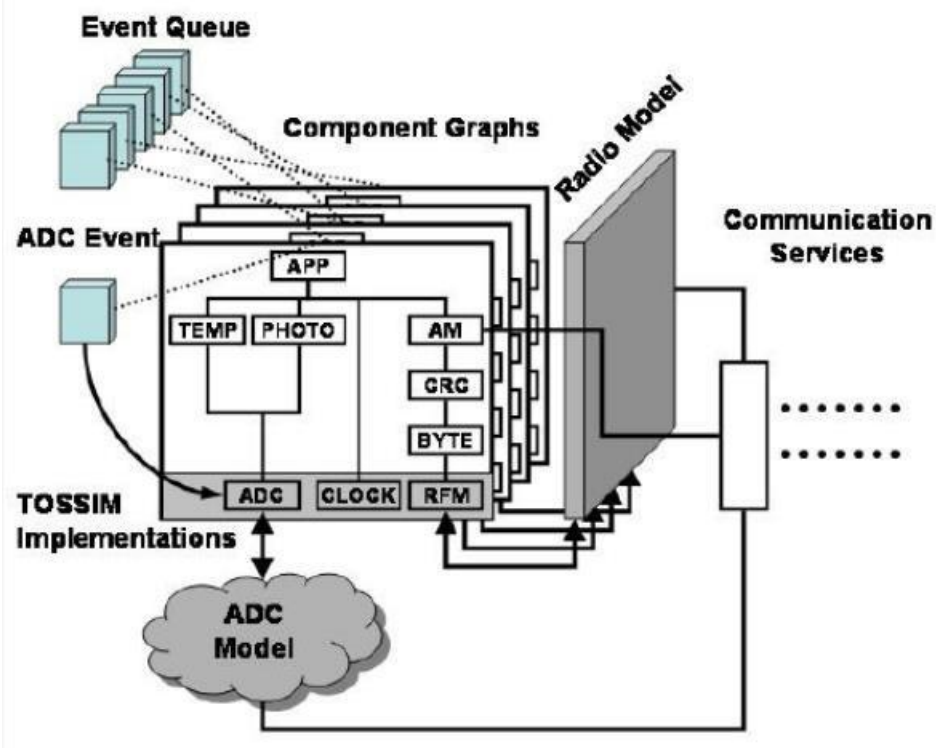
C) NETWORK TOPOLOGY, CONFIGURATIONS AND OUTPUT:

Theory

TinyOS is an open-source operating system designed specifically for embedded systems and wireless sensor networks. It follows a component-based programming model and is primarily programmed using the nesC (network embedded systems C) language. Understanding the computational concepts in TinyOS, including Events, Commands, and Tasks, is crucial for effectively developing applications for resource-constrained devices.

1. Events
    - In Tossim, events represent occurrences or incidents that trigger some action or response in the simulation.
    - These events could be packet transmissions, receptions, node movements, changes in network topology, etc.
    - Events are typically scheduled to occur at specific simulation times and can influence the behavior of the simulated network.
2. Commands
    - Commands in Tossim are instructions or actions issued to control various aspects of the simulation.
    - These commands can be used to start, pause, resume, or stop the simulation, modify simulation parameters, observe simulation state, etc.
    - Users can issue commands through a command-line interface or scripting environment to interact with the simulation dynamically.
3. Tasks
    - Tasks in Tossim are units of computation or activities that are executed by individual nodes in the simulated network.
    - These tasks represent the functionality or operations performed by nodes, such as processing received packets, executing application logic, updating routing tables, etc.
    - Tasks are typically scheduled and executed asynchronously based on events or timers within the simulation environment.

Output


Event Queue
Component Graphs
Radio Model
ADC Event
Communication
Services
APP
TEMP
PHOTO
AM
CRC
BYTE
ADC
CLOCK
RFM
TOSSIM
Implementations
ADC
Model

Wireless Sensor Networks & Mobile Communication

Practical No.3A

DEPARTMENT OF COMPUTER SCIENCE

Name:	Ajay Kumar Uthaya Kumar	**Roll Number**	TCS2324002
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:

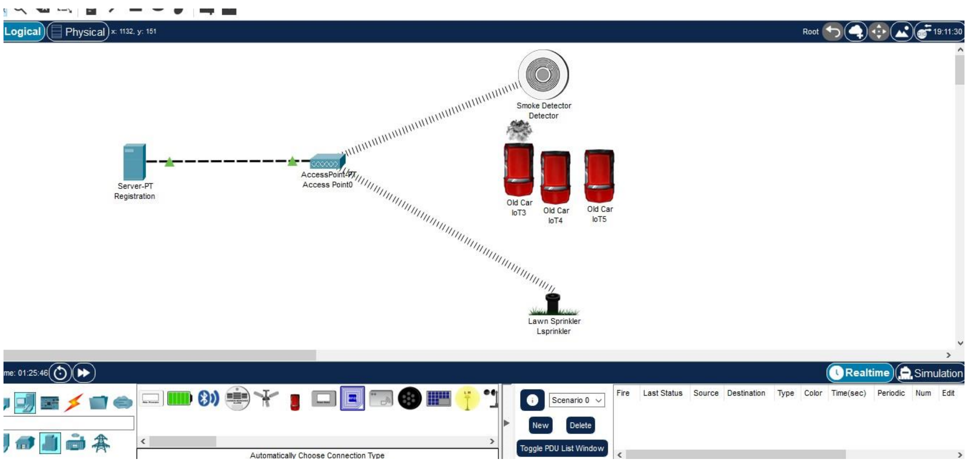
A smoke detection and fire prevention system is a crucial component of building safety infrastructure designed to detect the presence of smoke or fire and take appropriate action to mitigate the risk of fire-related incidents. Below is a detailed description outlining its components:

Components:

- Smoke Detectors: These devices are equipped with sensors that detect the presence of smoke particles in the air. There are different types of smoke detectors, including ionization, photoelectric, and dual-sensor detectors.
- Fire Alarms: Fire alarms are audible and visual alert systems triggered by smoke detectors or heat sensors. They notify occupants of the building about the presence of smoke or fire, allowing them to evacuate safely.
- Heat Sensors: Heat sensors detect changes in temperature, indicating the presence of fire. They complement smoke detectors by providing an additional layer of fire detection.
- Sprinkler Systems: Automatic sprinkler systems are activated in response to a fire alarm or heat detection, releasing water to suppress the fire and prevent its spread.

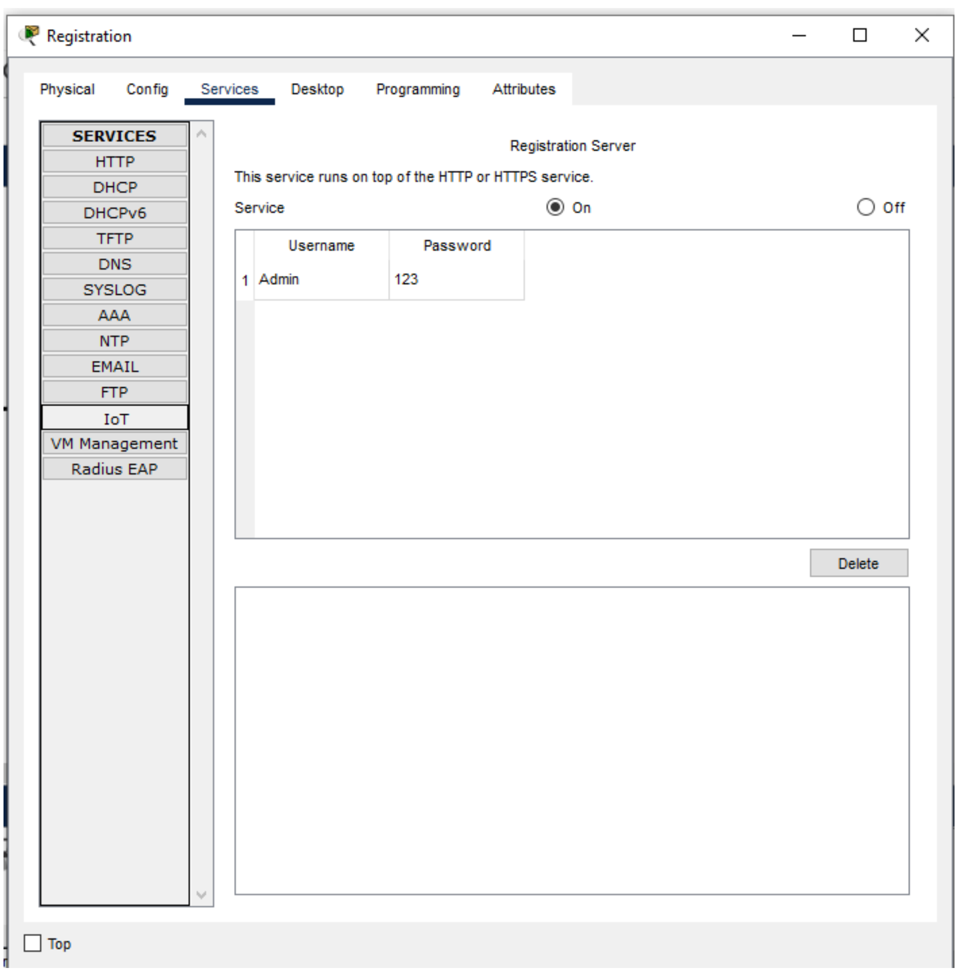
C) NETWORK TOPOLOGY, CONFIGURATIONS AND OUTPUT:

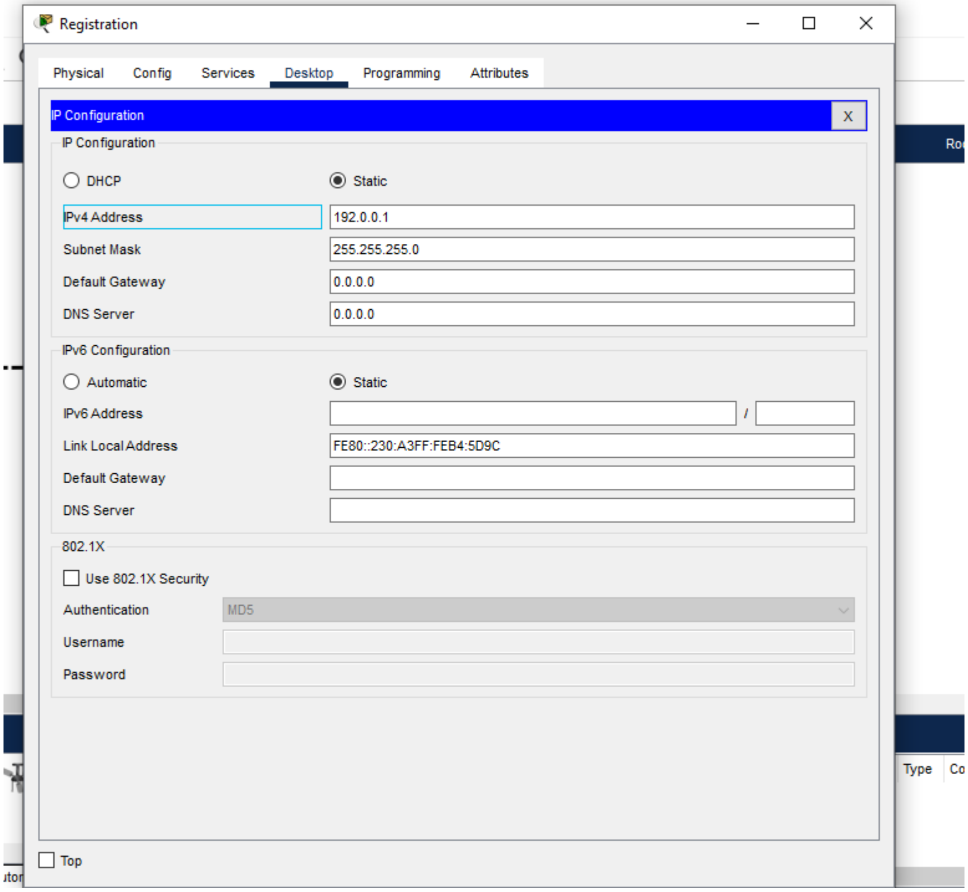
Network topology(only for cisco packet tracer practical's):

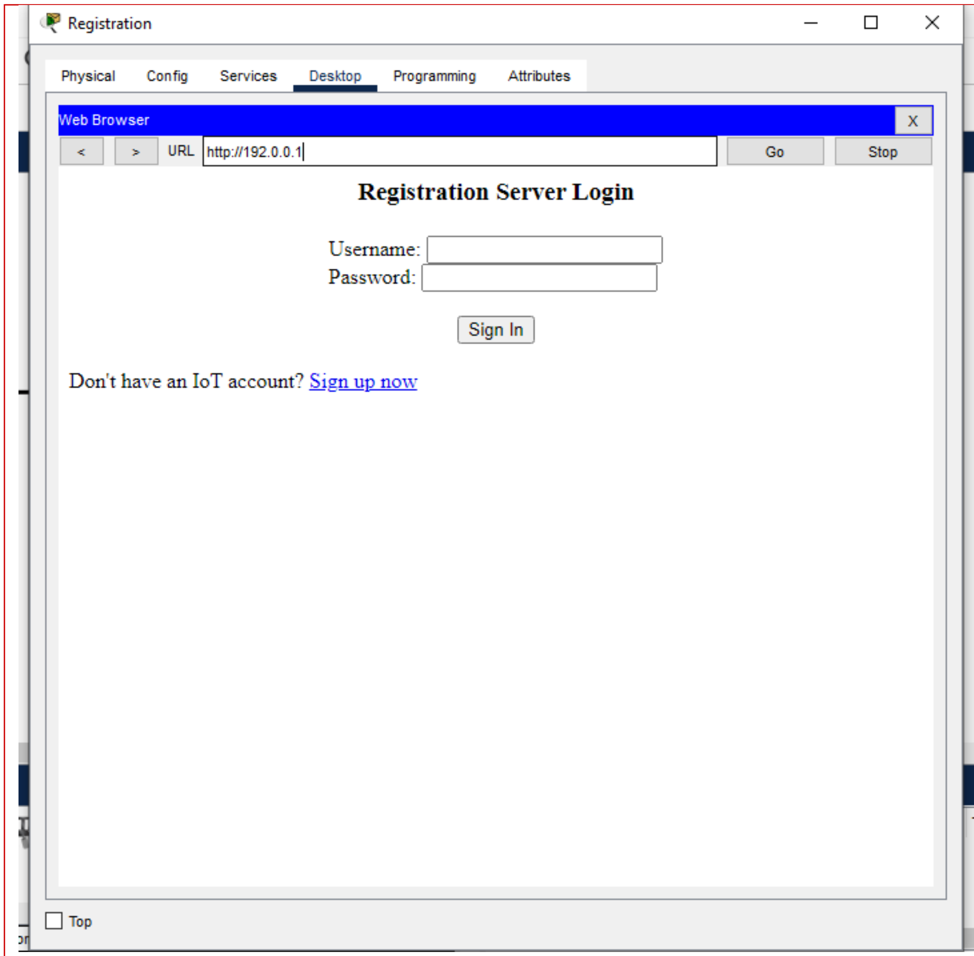


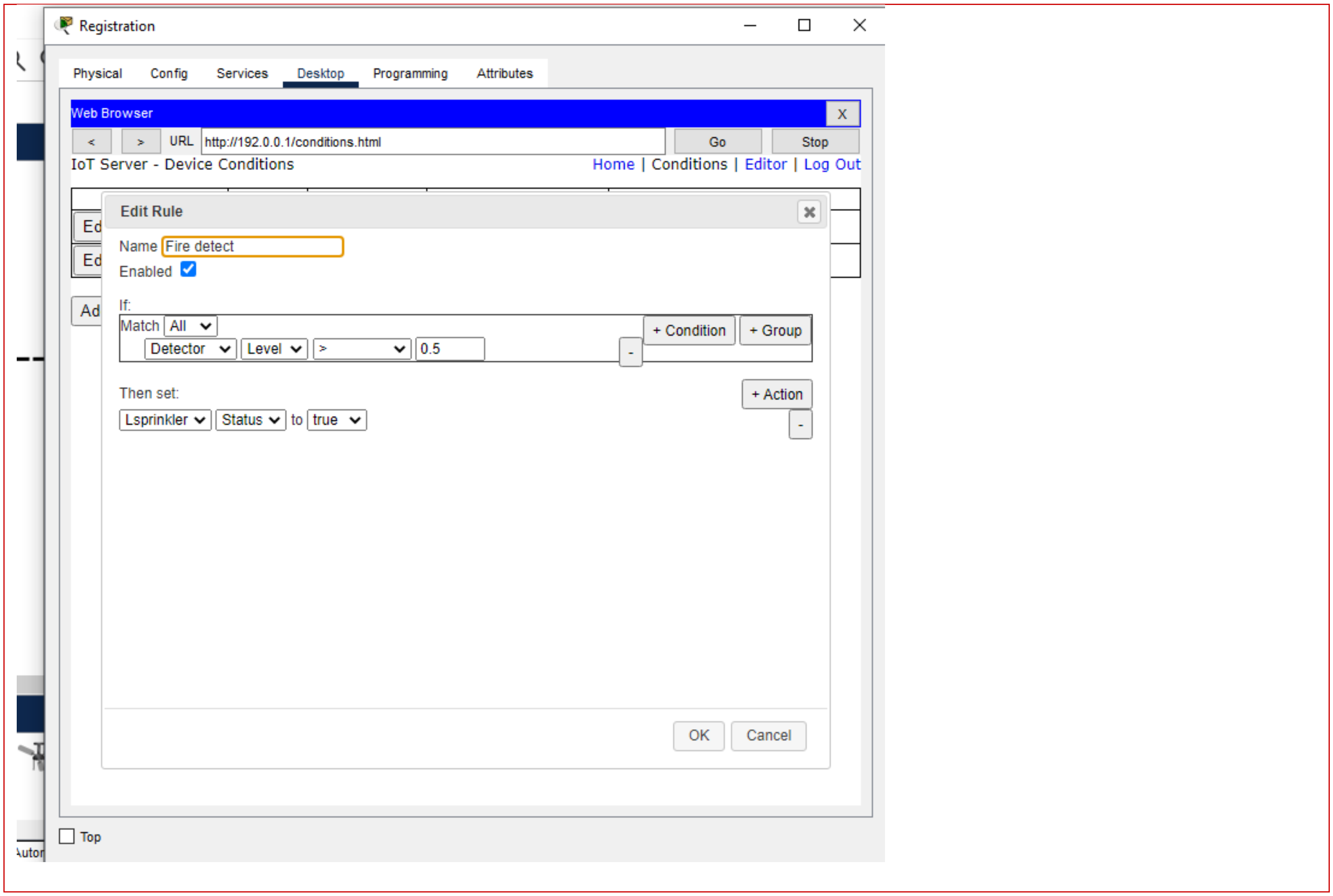
Configurations:

For Server:

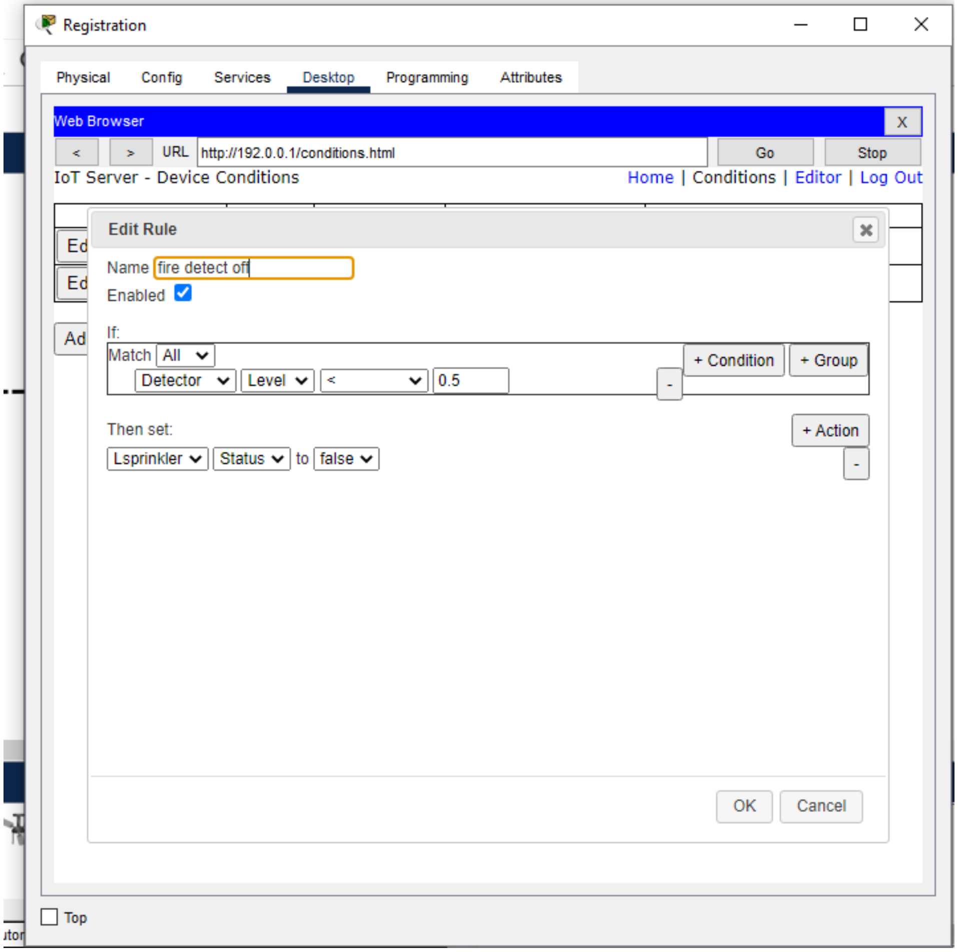



Registration
Physical
Config
Services
Desktop
Programming
Attributes
IP Configuration
X
IP Configuration
DHCP
Static
IPv4 Address
192.0.0.1
Subnet Mask
255.255.255.0
Default Gateway
0.0.0.0
DNS Server
0.0.0.0
IPv6 Configuration
Automatic
Static
IPv6 Address
/
Link Local Address
FE80::230:A3FF:FEB4:5D9C
Default Gateway
DNS Server
802.1X
Use 802.1X Security
Authentication
MD5
Username
Password
Top
Type
Co

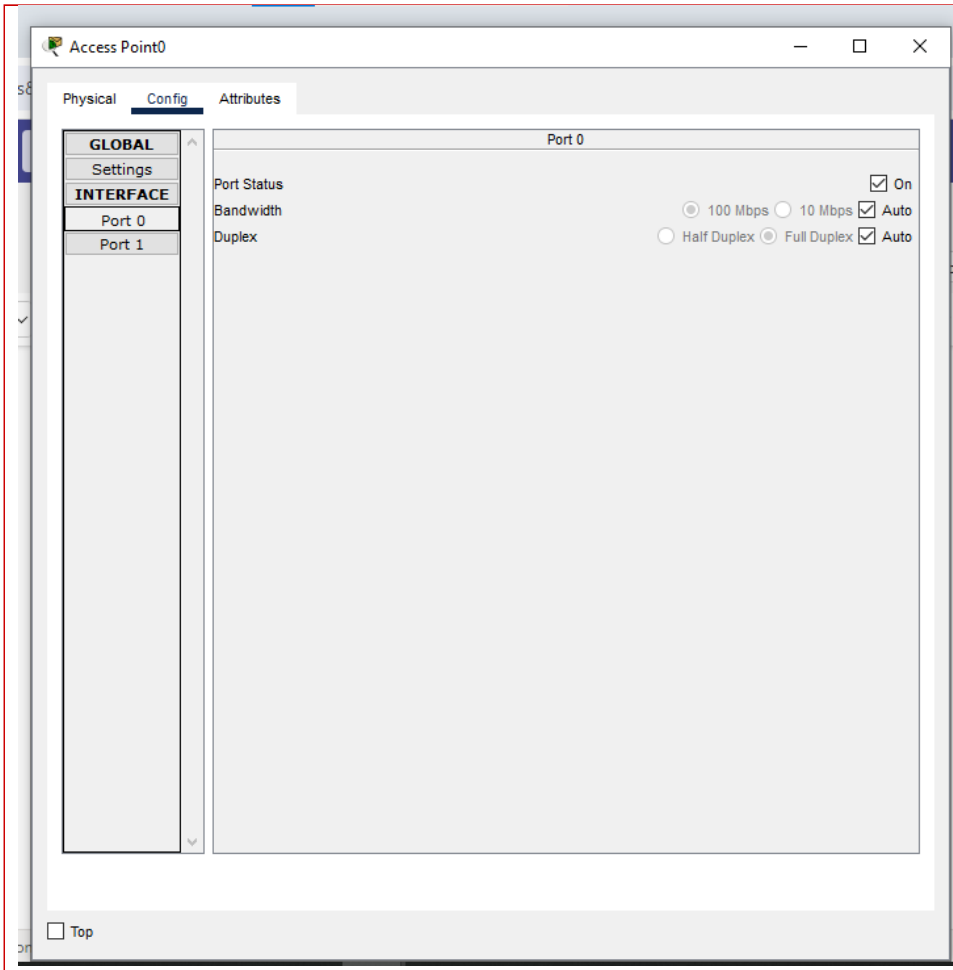

Registration
Physical
Config
Services
Desktop
Programming
Attributes
Web Browser
X
<
>
URL
http://192.0.0.1
Go
Stop
Registration Server Login
Username:
Password:
Sign In
Don't have an IoT account? Sign up now
Top

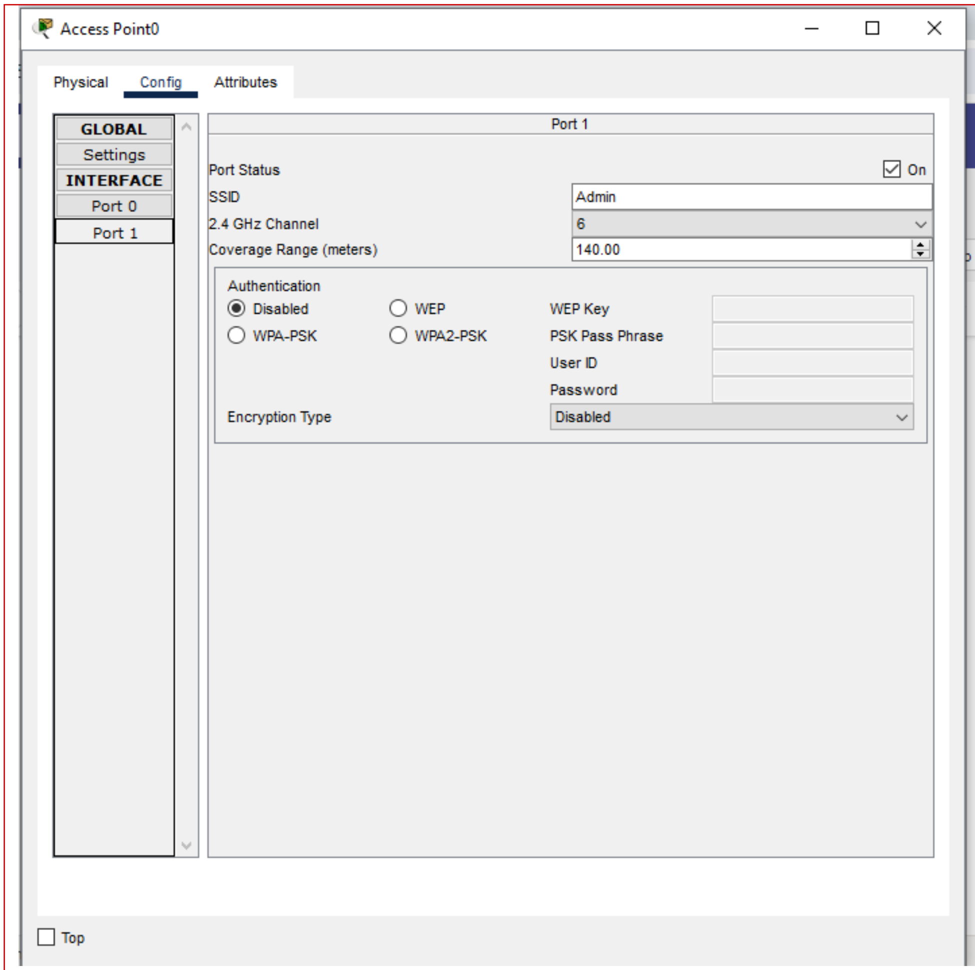


Name of Instructor: Jescia D'cruz

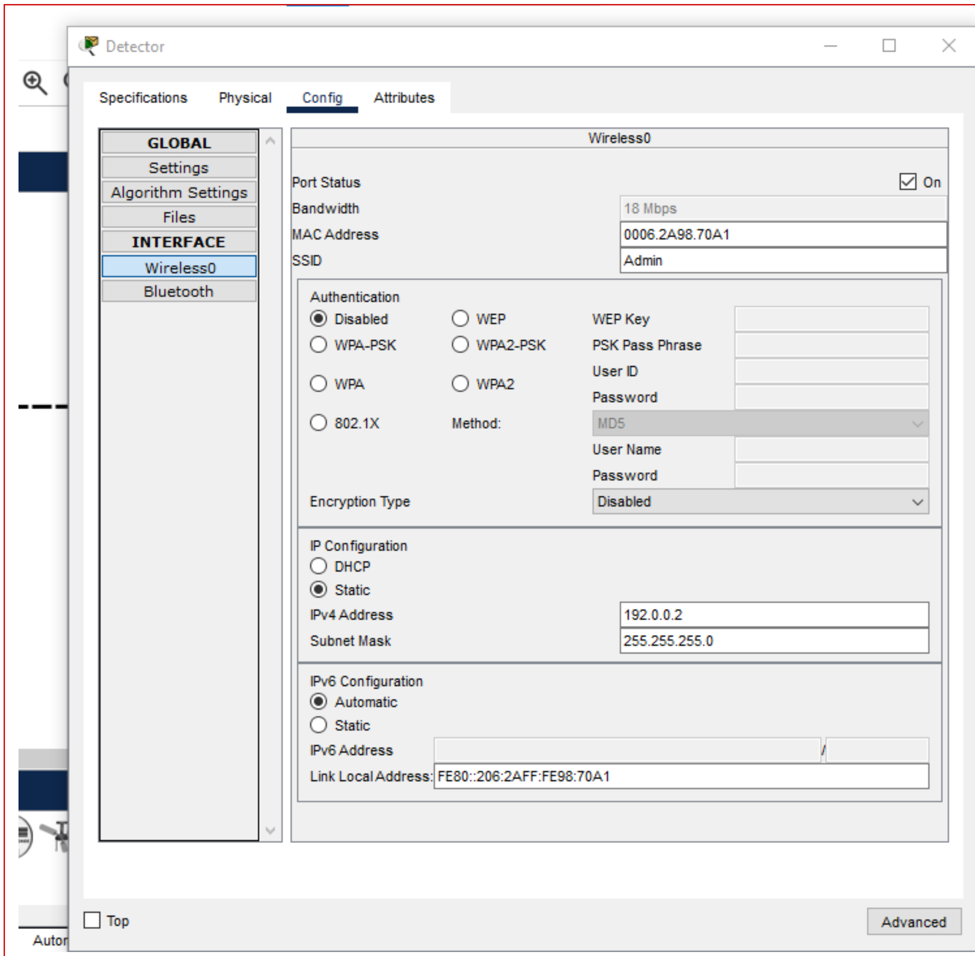


For Access-Point

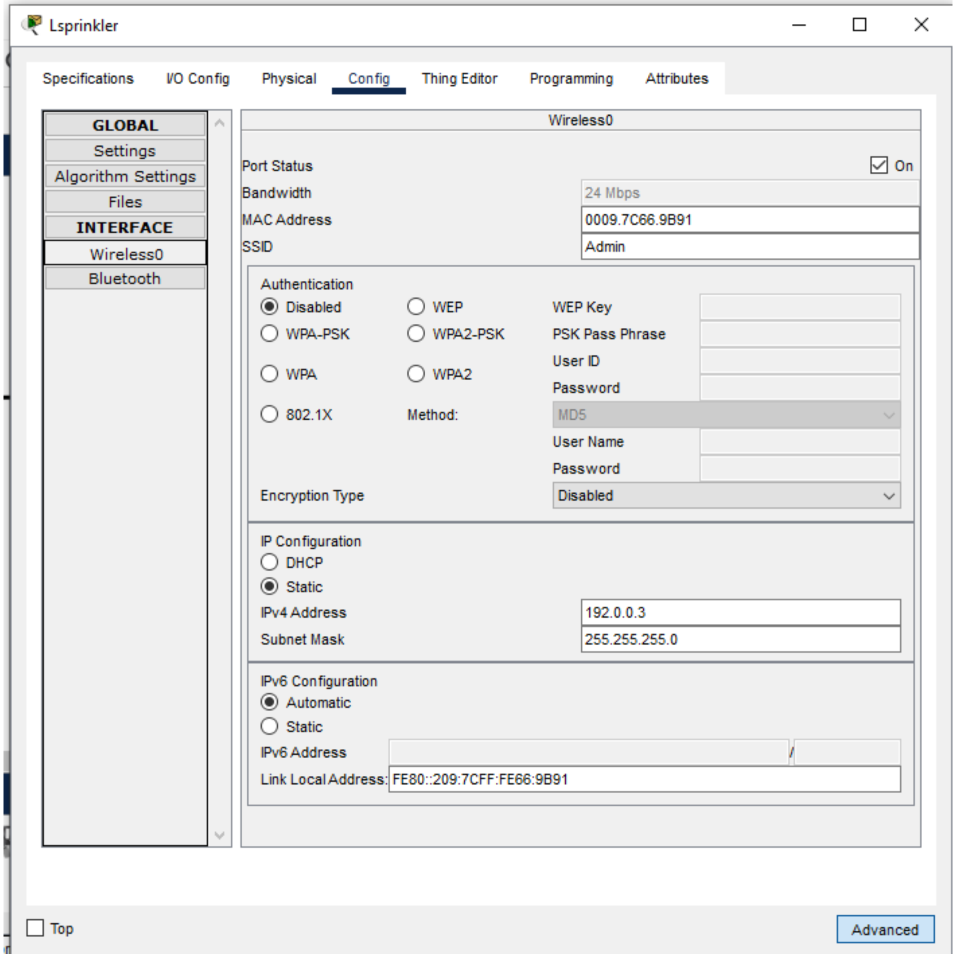

Access Point0
Physical
Config
Attributes
GLOBAL
Settings
INTERFACE
Port 0
Port 1
Port 0
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
Top



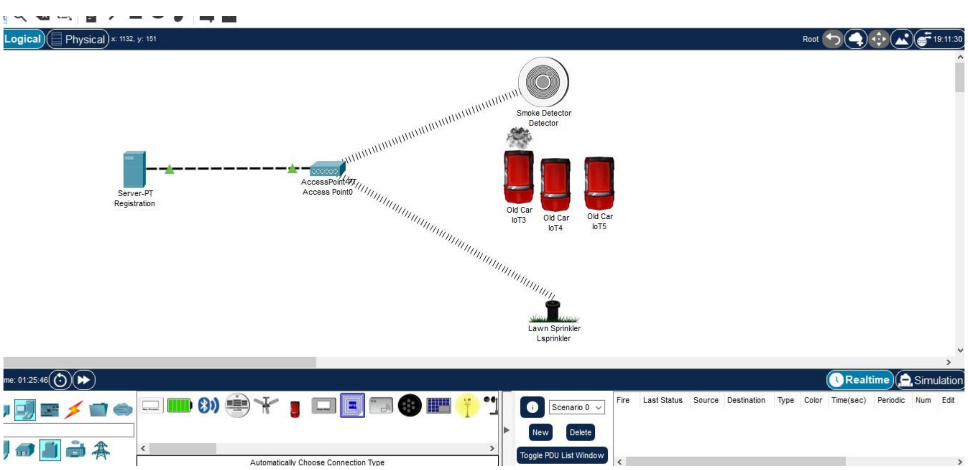
For Detector



For LSprinkler


Lsprinkler
Specifications
I/O Config
Physical
Config
Thing Editor
Programming
Attributes
GLOBAL
Settings
Algorithm Settings
Files
INTERFACE
Wireless0
Bluetooth
Wireless0
Port Status
On
Bandwidth
24 Mbps
MAC Address
0009.7C66.9B91
SSID
Admin
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
192.0.0.3
Subnet Mask
255.255.255.0
IPv6 Configuration
Automatic
Static
IPv6 Address
Link Local Address:
FE80::209:7CFF:FE66:9B91
Top
Advanced

Output


Logical
Physical
Smoke Detector
Detector
Server-PT
Registration
AccessPoint-PT
Access Point0
Old Car
IoT3
Old Car
IoT4
Old Car
IoT5
Lawn Sprinkler
Lsprinkler
Realtime
Simulation
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num
Edit
Automatically Choose Connection Type

Wireless Sensor Networks & Mobile Communication

Practical No.3B

DEPARTMENT OF COMPUTER SCIENCE

Name:	Ajay Kumar Uthaya Kumar	**Roll Number**	TCS2324002
Paper Code:	SIUSCS61	**Class**	TYBSc(Computer Science)
Topic:	Garden Sprinkler	**Batch**	I
Date:	15/1/24	**Practical No**	3B

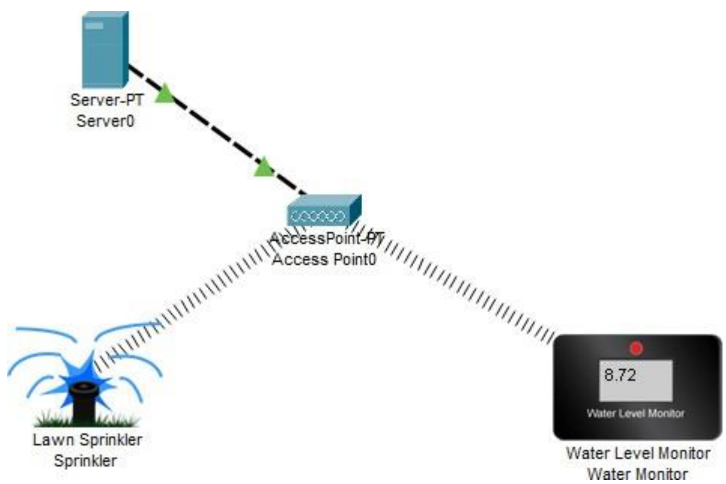
A) AIM: Design garden sprinkler system using Cisco Packet Tracer

B) DESCRIPTION:

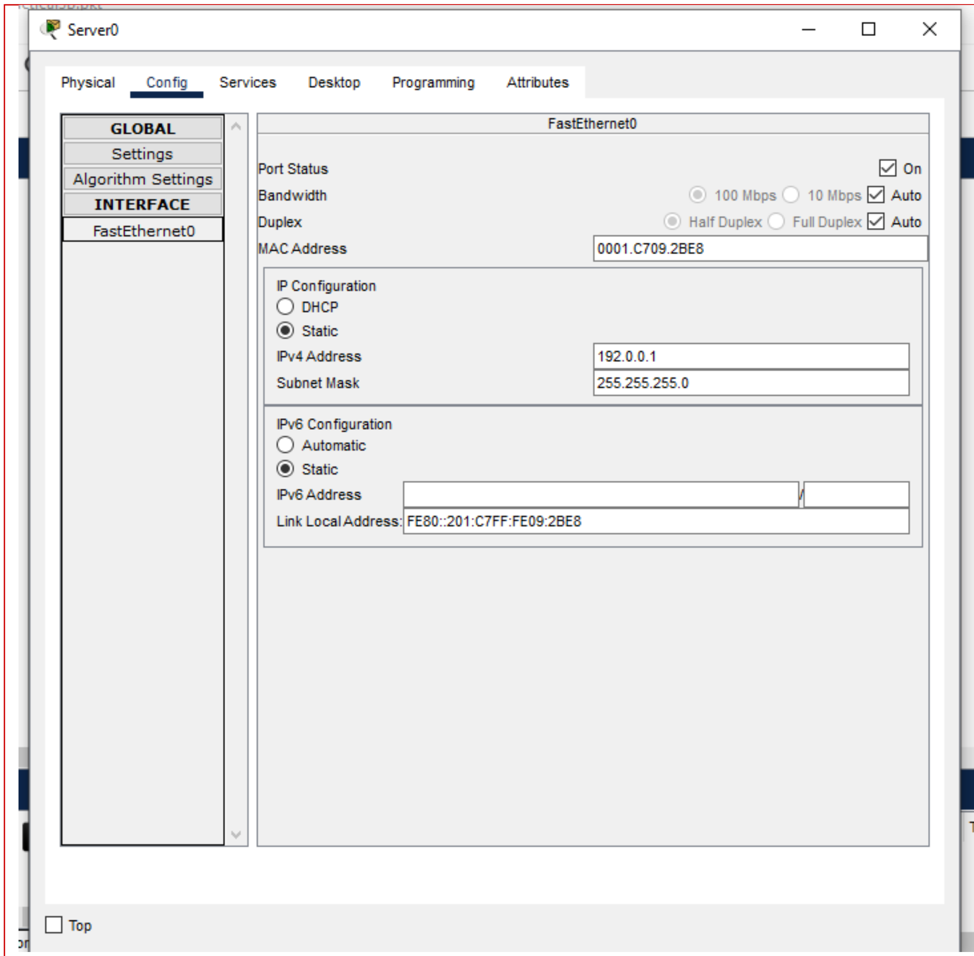
An irrigation sprinkler is a device used to irrigate (water) agricultural crops, lawns, landscapes, golf courses, and other areas. They are also used for cooling and for the control of airborne dust. Sprinkler irrigation is the method of applying water in a controlled manner in way similar to rainfall. The water is distributed through a network that may consist of pumps, valves, pipes, and sprinklers. Irrigation sprinklers can be used for residential, industrial, and agricultural usage. It is useful on uneven land where sufficient water is not available as well as on sandy soil. The perpendicular pipes, having rotating nozzles on top, are joined to the main pipeline at regular intervals. When water is pressurized through the main pipe it escapes from the rotating nozzles. It gets sprinkled on the crop. In sprinkler or overhead irrigation, water is piped to one more central locations within the field and distributed by overhead high pressure sprinklers or guns.

C) NETWORK TOPOLOGY, CONFIGURATIONS AND OUTPUT:

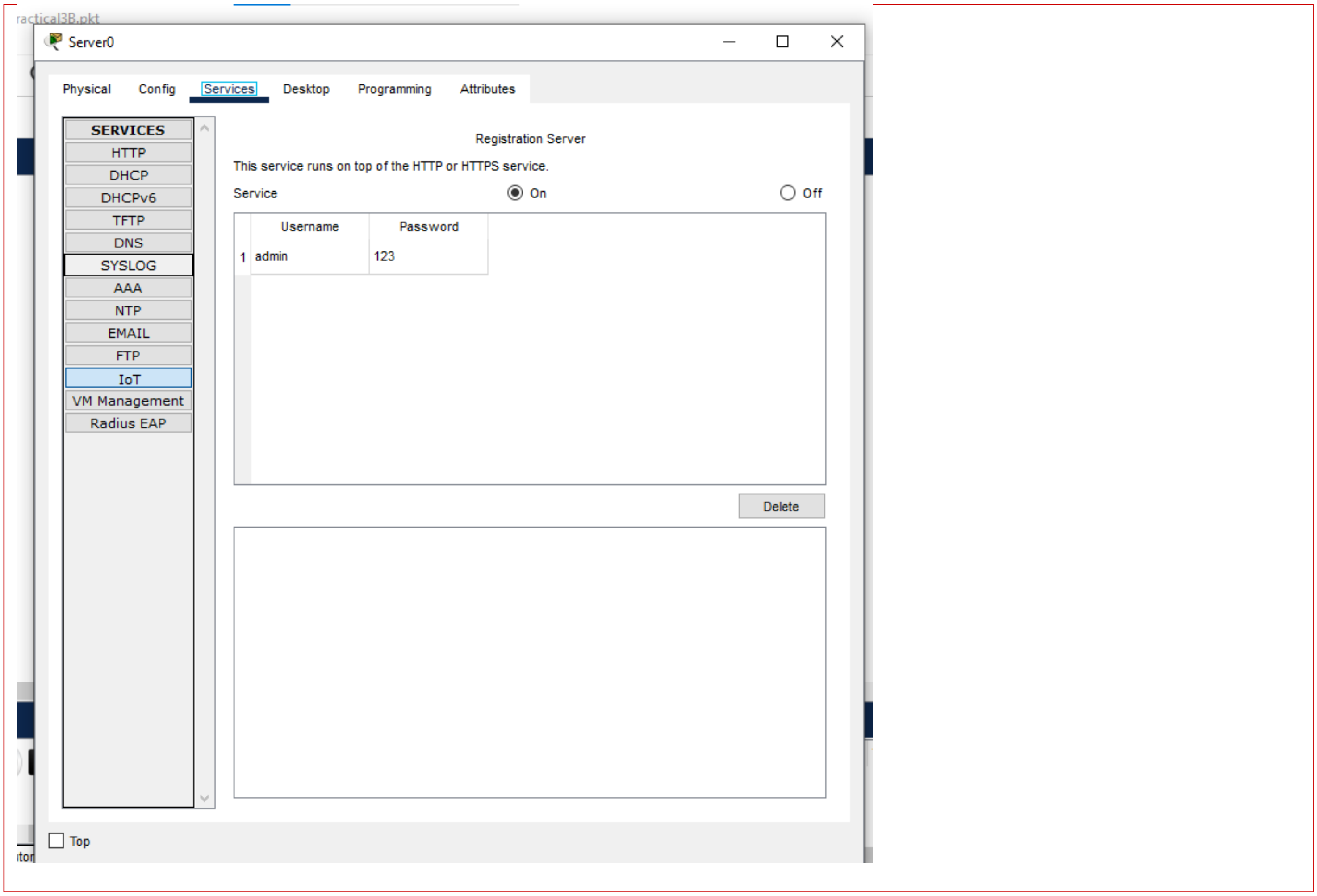
Network topology(only for cisco packet tracer practical's):

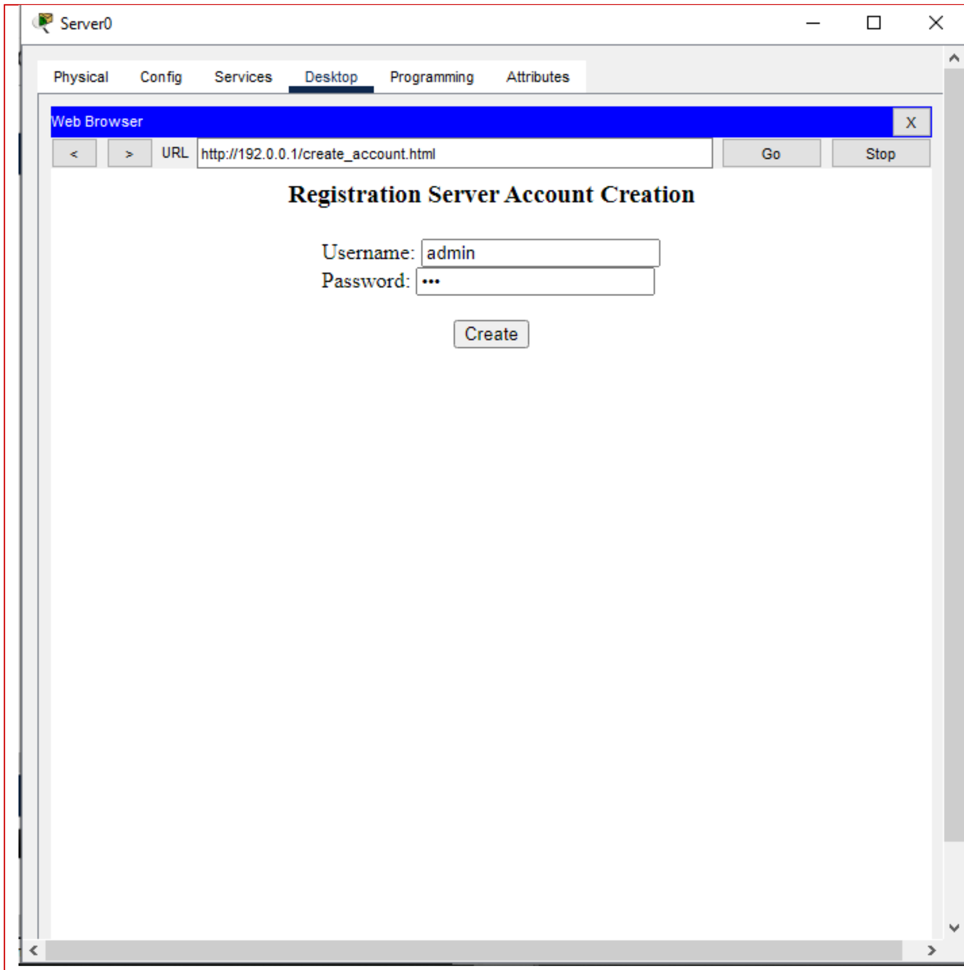


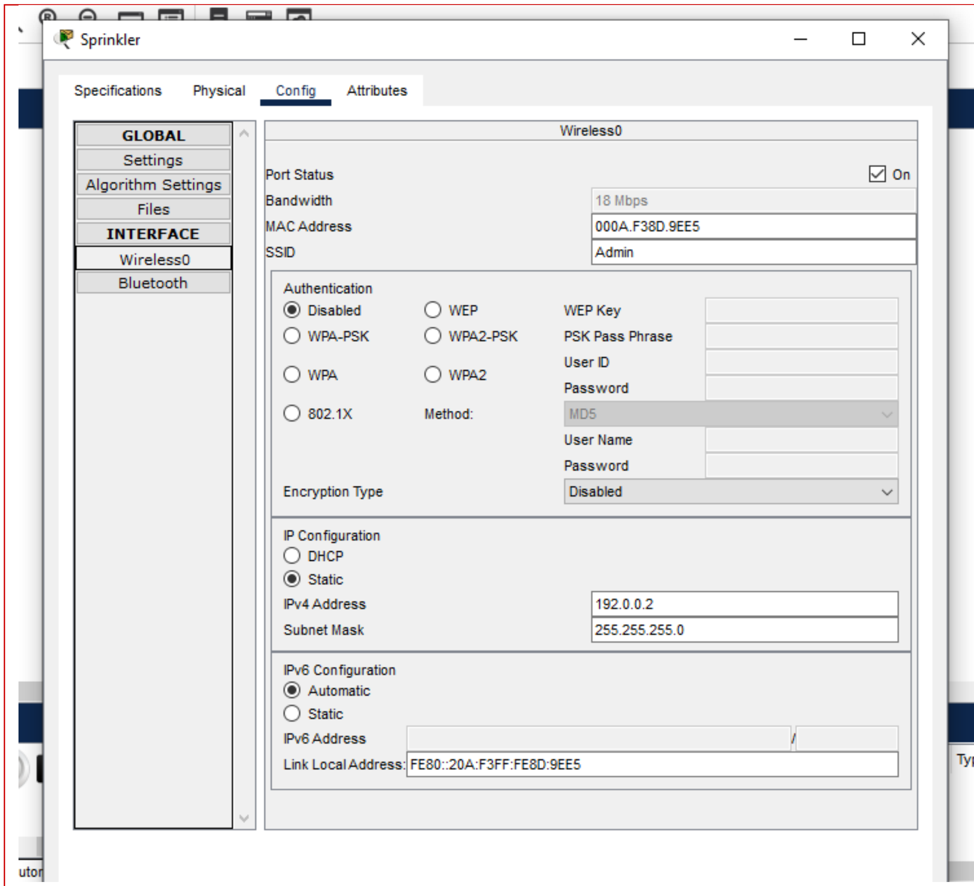
Configurations:

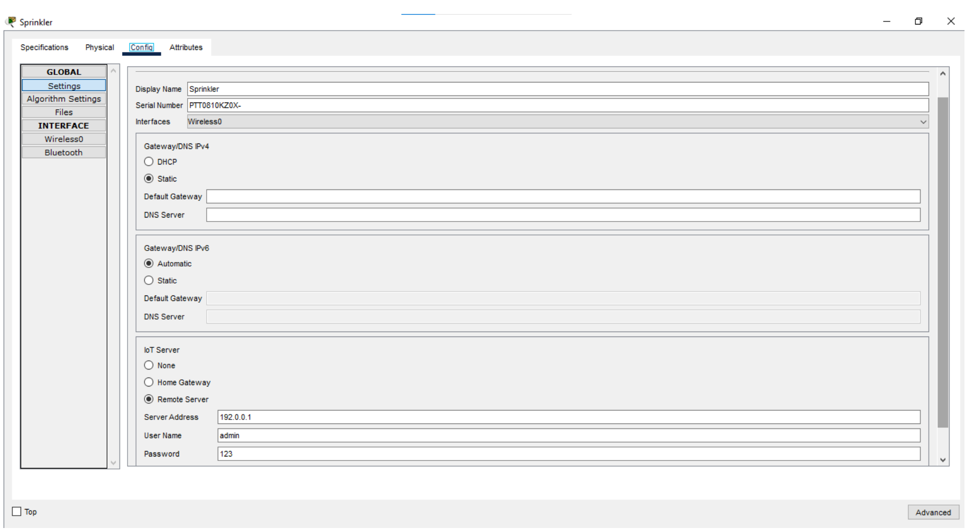


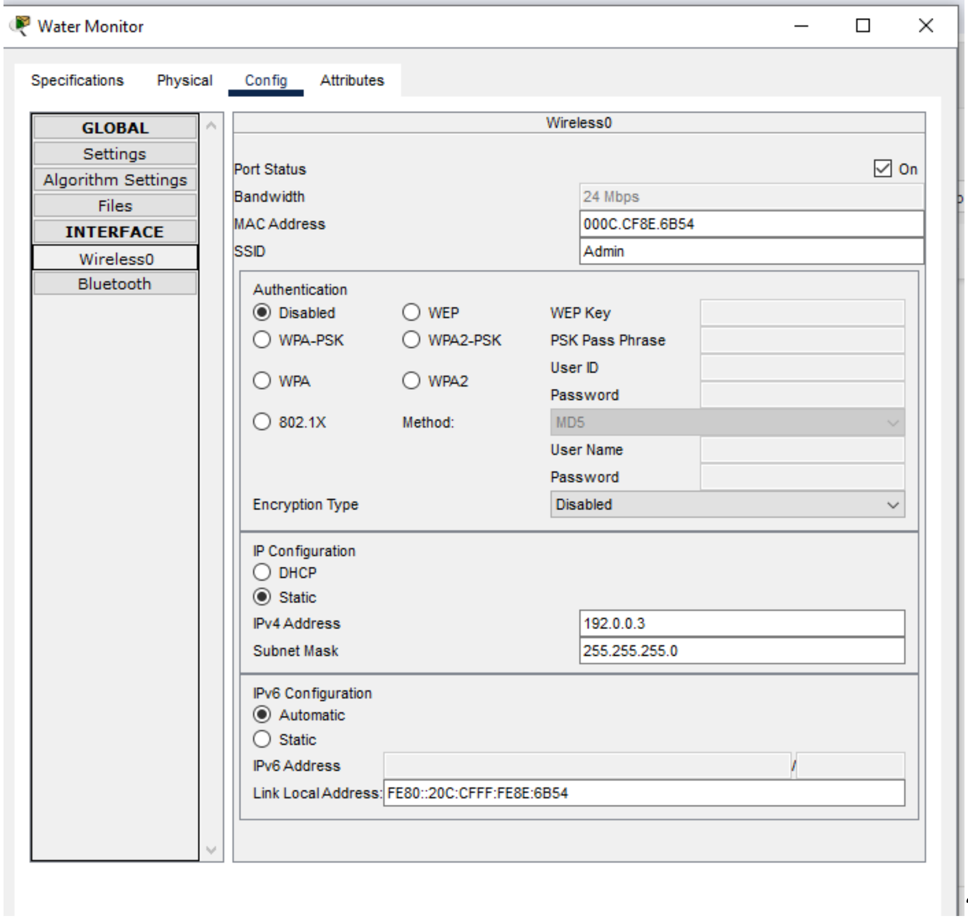
Name of Instructor: Jescia D’cruz

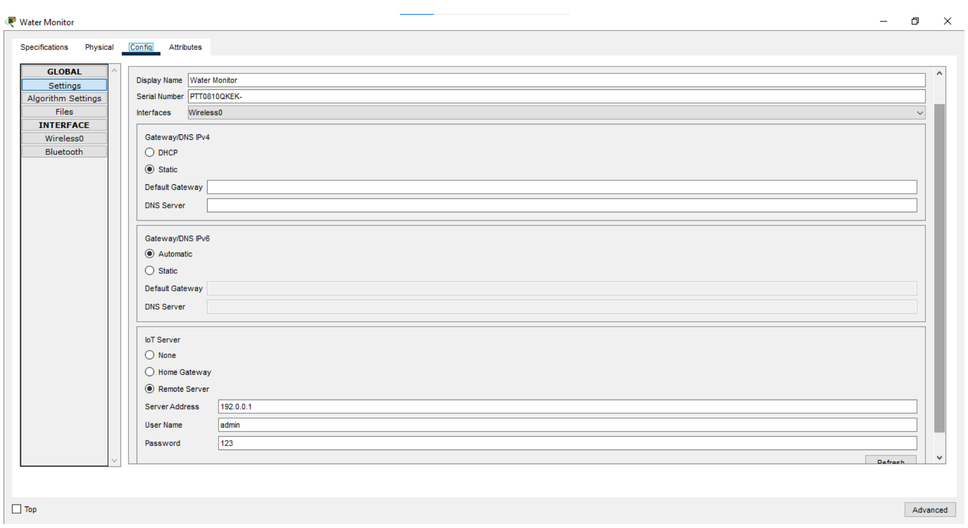

ractical3B.pkt
Server0
Physical
Config
Services
Desktop
Programming
Attributes
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
IoT
VM Management
Radius EAP
Registration Server
This service runs on top of the HTTP or HTTPS service.
Service
On
Off
Username
Password
1
admin
123
Delete
Top


Server0
Physical
Config
Services
Desktop
Programming
Attributes
Web Browser
X
<
>
URL
http://192.0.0.1/create_account.html
Go
Stop
Registration Server Account Creation
Username:
admin
Password:
Create

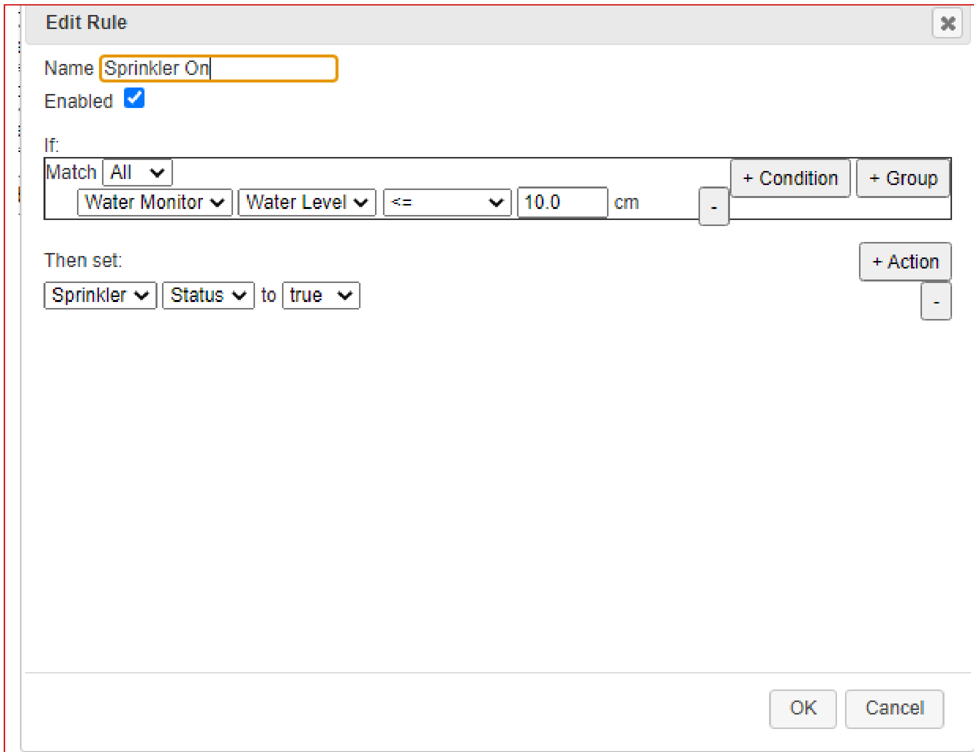

Sprinkler
Specifications
Physical
Config
Attributes
GLOBAL
Settings
Algorithm Settings
Files
INTERFACE
Wireless0
Bluetooth
Wireless0
Port Status
On
Bandwidth
18 Mbps
MAC Address
000A.F38D.9EE5
SSID
Admin
Authentication
Disabled
WEP
WEP Key
WPA-PSK
WPA2-PSK
PSK Pass Phrase
User ID
WPA
WPA2
Password
802.1X
Method:
MD5
User Name
Password
Encryption Type
Disabled
IP Configuration
DHCP
Static
IPv4 Address
192.0.0.2
Subnet Mask
255.255.255.0
IPv6 Configuration
Automatic
Static
IPv6 Address
Link Local Address:
FE80::20A:F3FF:FE8D:9EE5

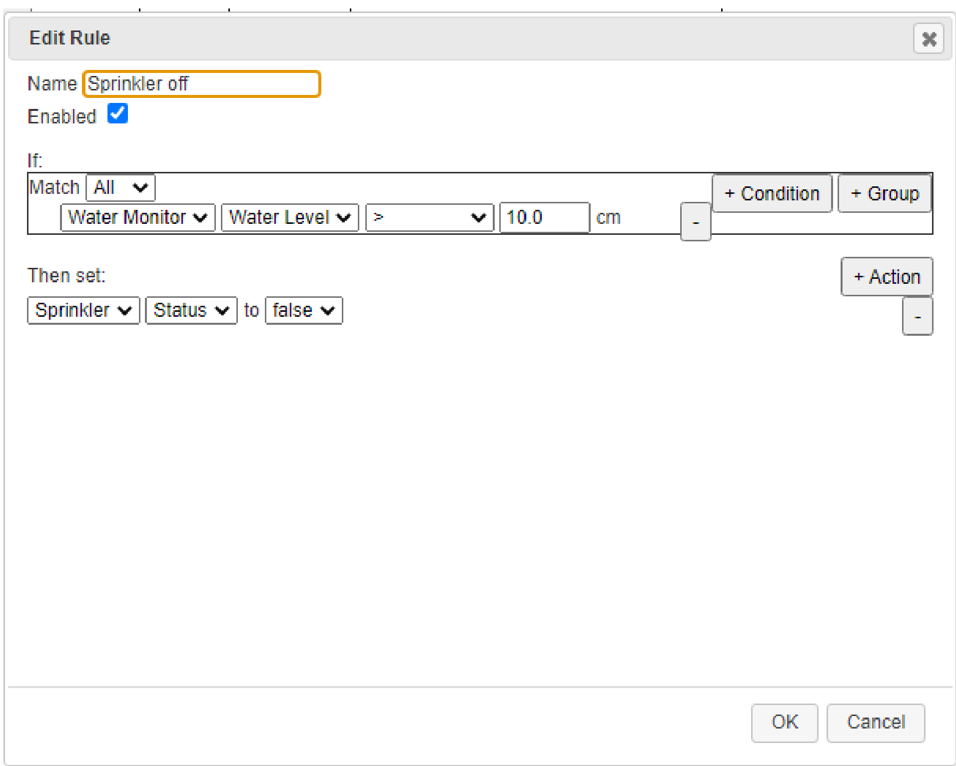

Sprinkler
Specifications
Physical
Config
Attributes
GLOBAL
Settings
Algorithm Settings
Files
INTERFACE
Wireless0
Bluetooth
Display Name
Sprinkler
Serial Number
PTT0810KZ0X-
Interfaces
Wireless0
Gateway/DNS IPv4
DHCP
Static
Default Gateway
DNS Server
Gateway/DNS IPv6
Automatic
Static
Default Gateway
DNS Server
IoT Server
None
Home Gateway
Remote Server
Server Address
192.0.0.1
User Name
admin
Password
123
Top
Advanced


Water Monitor
Specifications
Physical
Config
Attributes
GLOBAL
Settings
Algorithm Settings
Files
INTERFACE
Wireless0
Bluetooth
Wireless0
Port Status
On
Bandwidth
24 Mbps
MAC Address
000C.CF8E.6B54
SSID
Admin
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
192.0.0.3
Subnet Mask
255.255.255.0
IPv6 Configuration
Automatic
Static
IPv6 Address
Link Local Address:
FE80::20C:CFFF:FE8E:6B54

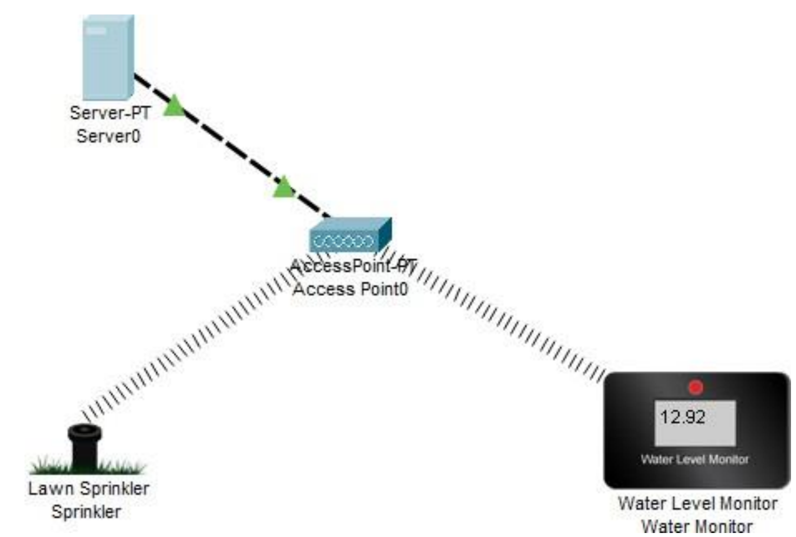

Water Monitor
Specifications
Physical
Config
Attributes
GLOBAL
Settings
Algorithm Settings
Files
INTERFACE
Wireless0
Bluetooth
Display Name
Water Monitor
Serial Number
PTT0810QKEK-
Interfaces
Wireless0
Gateway/DNS IPv4
DHCP
Static
Default Gateway
DNS Server
Gateway/DNS IPv6
Automatic
Static
Default Gateway
DNS Server
IoT Server
None
Home Gateway
Remote Server
Server Address
192.0.0.1
User Name
admin
Password
123
Top
Advanced

Condition


Edit Rule
Name Sprinkler On
Enabled
If:
Match All
Water Monitor
Water Level
<=
10.0
cm
+ Condition
+ Group
Then set:
+ Action
Sprinkler
Status
to
true
OK
Cancel


Edit Rule
Name Sprinkler off
Enabled
If:
Match All
Water Monitor
Water Level
>
10.0
cm
+ Condition
+ Group
Then set:
+ Action
Sprinkler
Status
to
false
OK
Cancel

Output



Name of Instructor: Jescia D'cruz

Wireless Sensor Networks & Mobile Communication

Practical No.4

DEPARTMENT OF COMPUTER SCIENCE

Name:	Ajay Kumar Uthaya Kumar	**Roll Number**	TCS2324002
Paper Code:	SIUSCS61	**Class**	TYBSc(Computer Science)
Topic:	Adhoc Network	**Batch**	I
Date:	05/2/24	**Practical No**	4

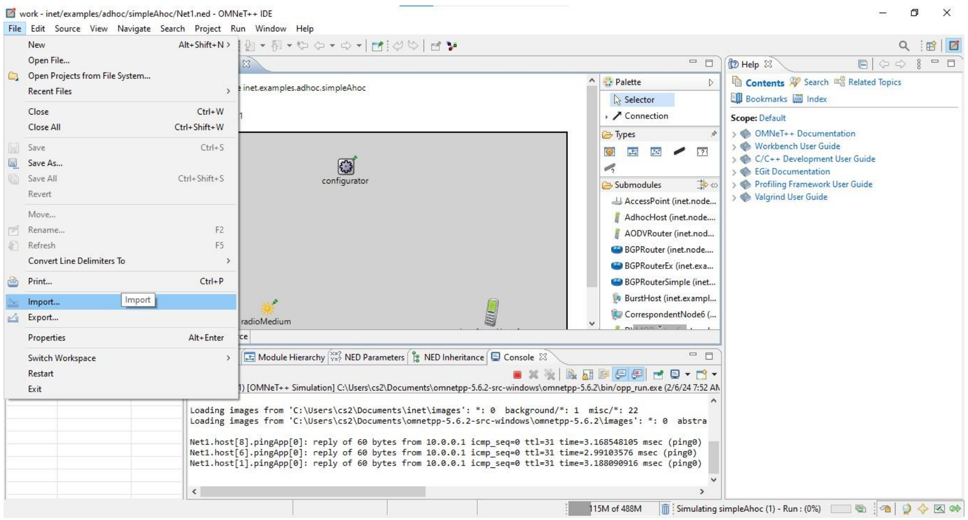
A) AIM: Create and simulate a simple adhoc network

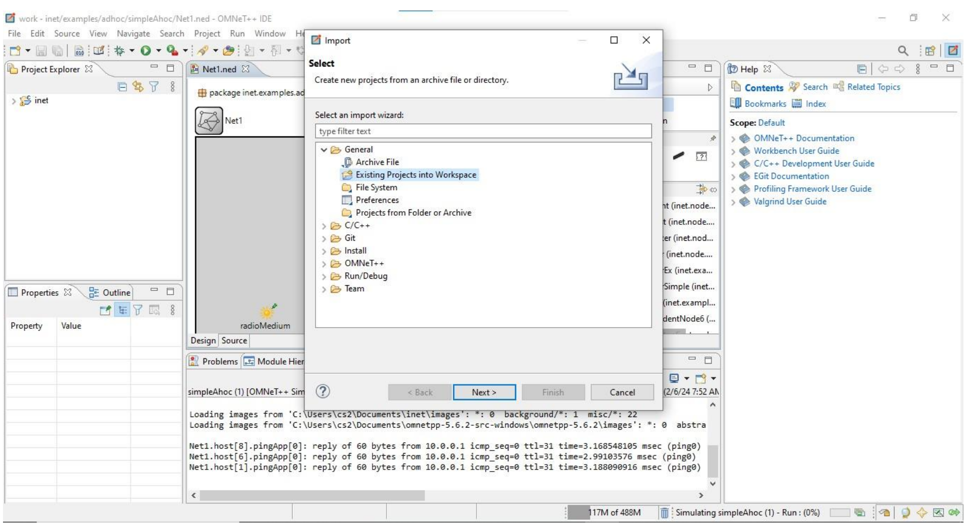
B) DESCRIPTION:

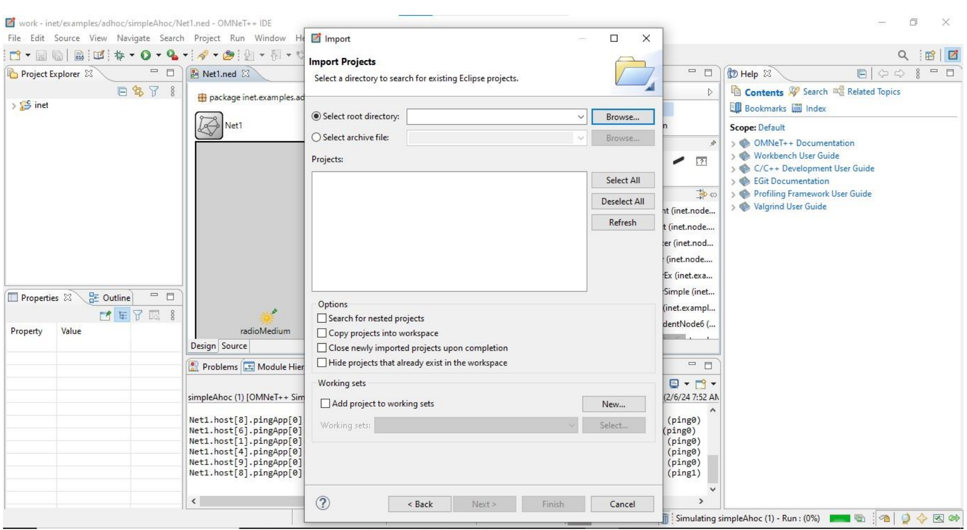
An ad hoc network is one that is spontaneously formed when devices connect and communicate with each other. The term ad hoc is a Latin word that literally means "for this," implying improvised or impromptu. Ad hoc networks are mostly wireless local area networks (WLANs). The devices communicate with 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.

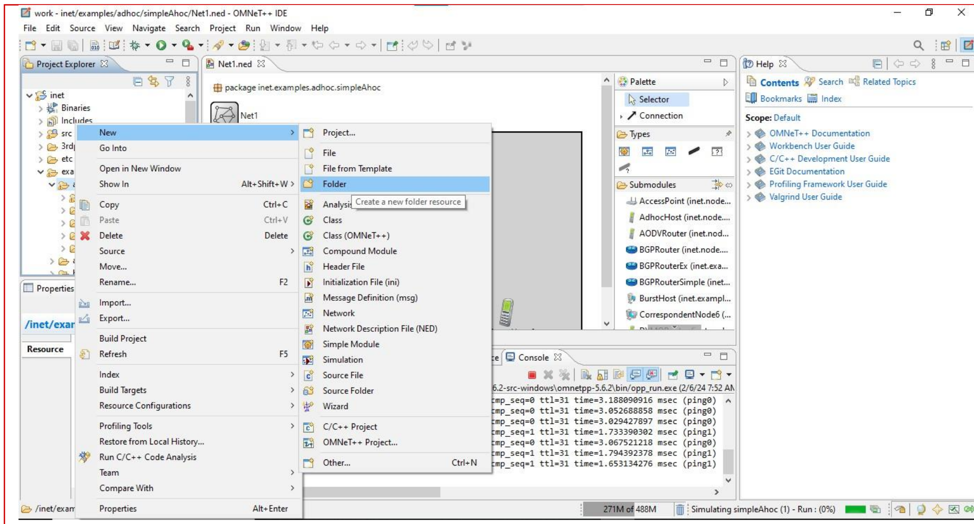
C) NETWORK TOPOLOGY, CONFIGURATIONS AND OUTPUT:

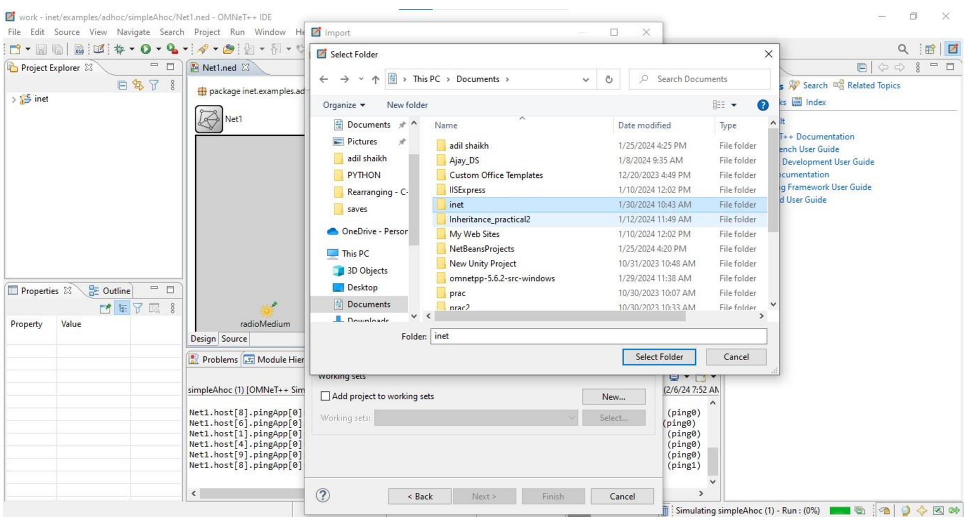
Configurations:

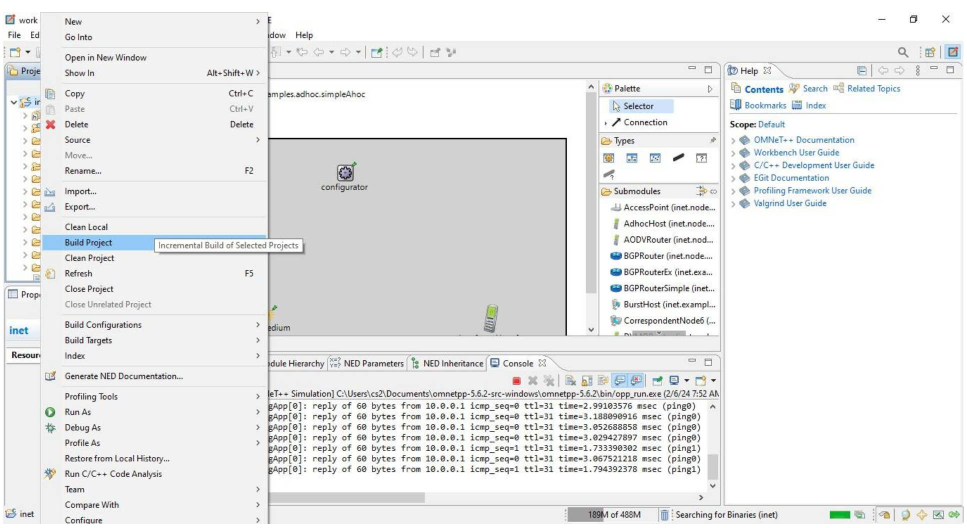




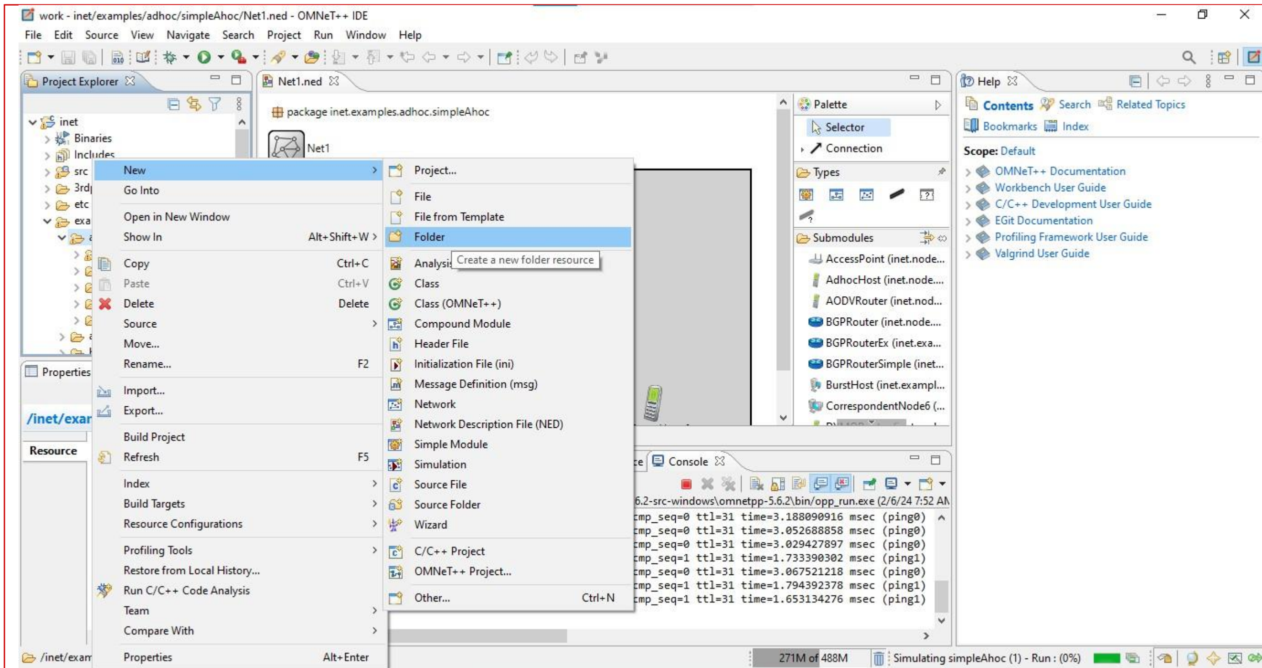


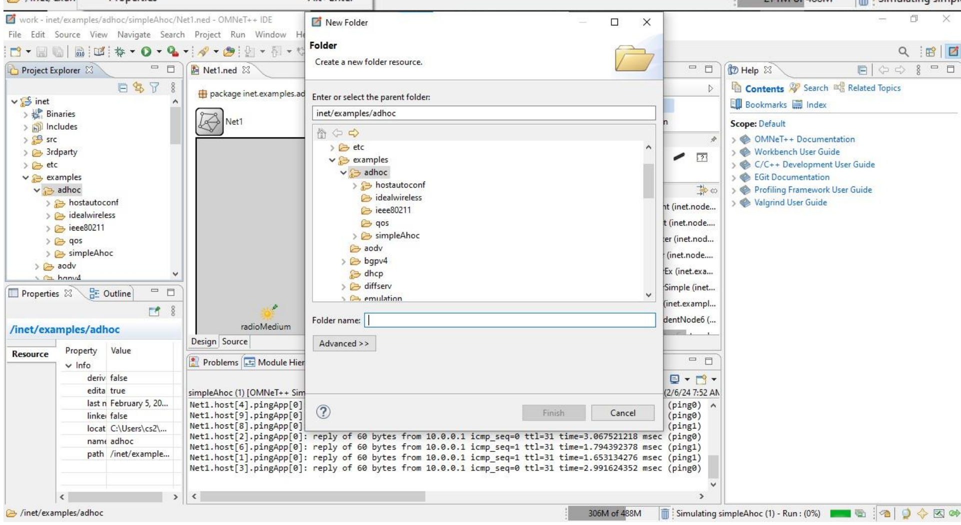


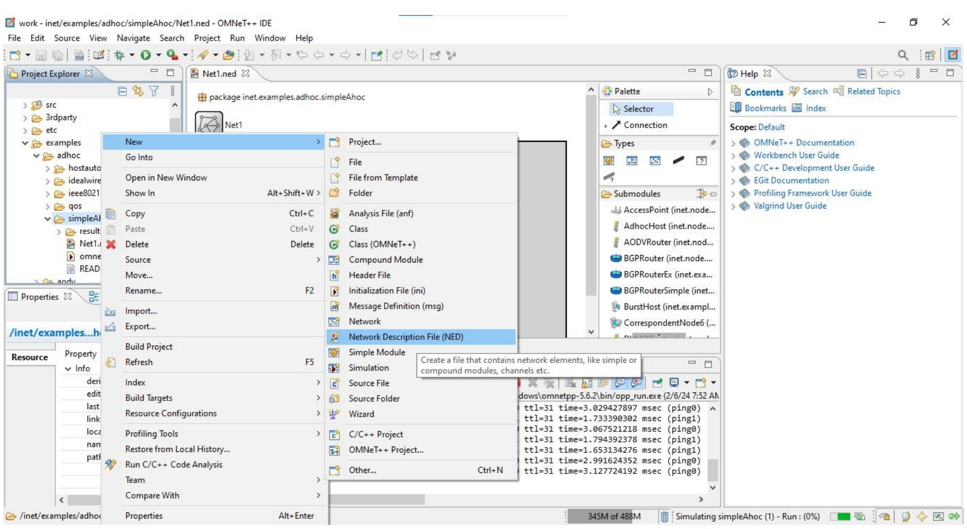


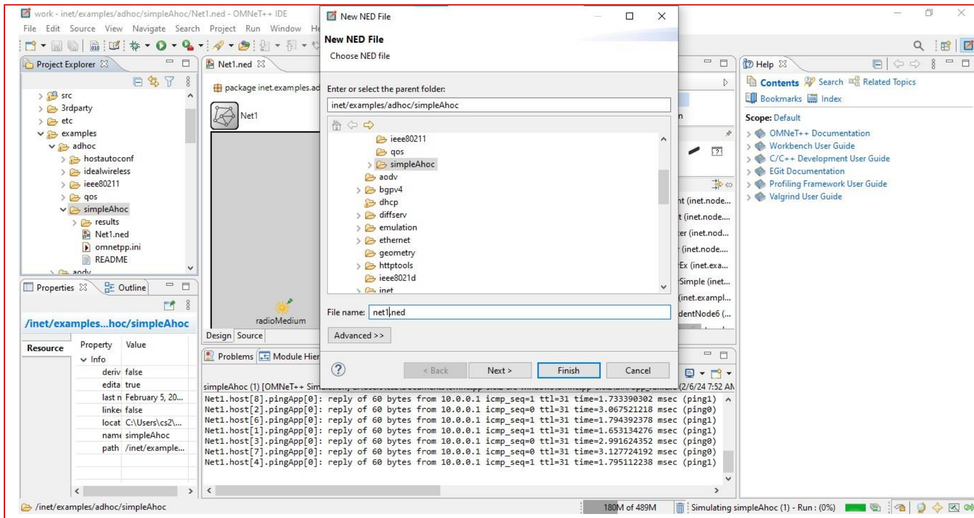


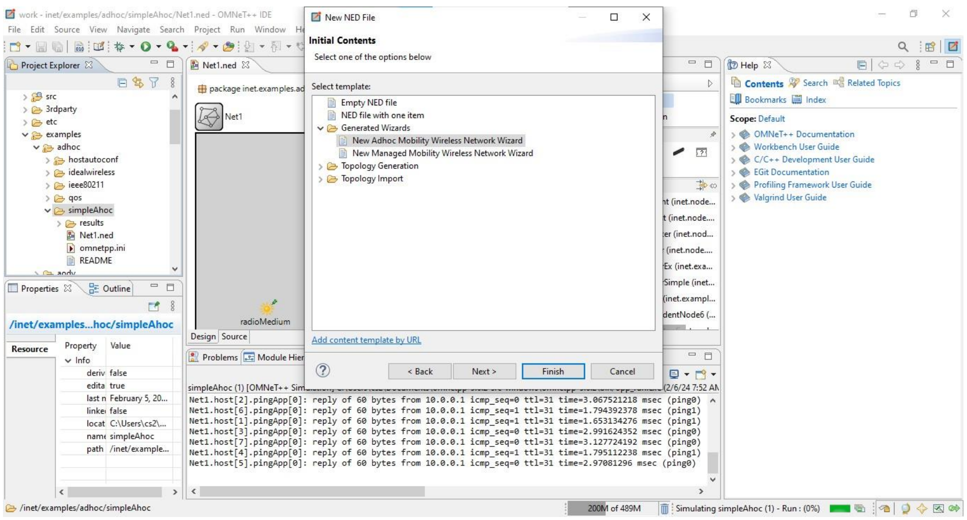
Name of Instructor: Jescia D’cruz

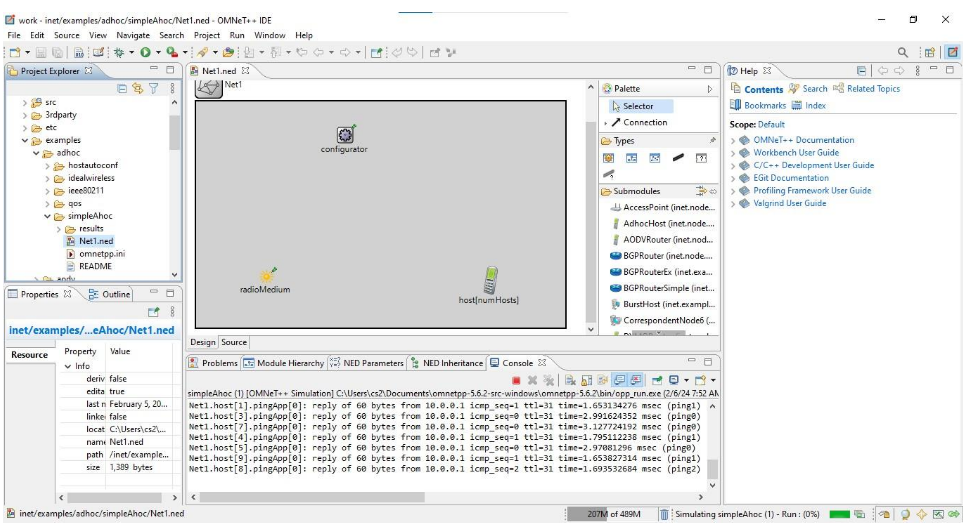

work - inet/examples/adhoc/simpleAhoc/Net1.ned - OMNeT++ IDE
File Edit Source View Navigate Search Project Run Window Help
Project Explorer
Net1.ned
package inet.examples.adhoc.simpleAhoc
New
Go Into
Open in New Window
Show In Alt+Shift+W
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Source
Move...
Rename... F2
Import...
Export...
Build Project
Refresh F5
Index
Build Targets
Resource Configurations
Profiling Tools
Restore from Local History...
Run C/C++ Code Analysis
Team
Compare With
Properties Alt+Enter
Project...
File
File from Template
Folder
Create a new folder resource
Class
Class (OMNeT++)
Compound Module
Header File
Initialization File (ini)
Message Definition (msg)
Network
Network Description File (NED)
Simple Module
Simulation
Source File
Source Folder
Wizard
C/C++ Project
OMNeT++ Project...
Other... Ctrl+N
Palette
Selector
Connection
Types
Submodules
Console
Help
Contents
Search
Related Topics
Bookmarks
Index
Scope: Default
OMNeT++ Documentation
Workbench User Guide
C/C++ Development User Guide
EGit Documentation
Profiling Framework User Guide
Valgrind User Guide
271M of 488M
Simulating simpleAhoc (1) - Run : (0%)


New Folder
Folder
Create a new folder resource.
Enter or select the parent folder:
inet/examples/adhoc
etc
examples
adhoc
hostautoconf
idealwireless
ieee80211
qos
simpleAhoc
aodv
bgpv4
dhcp
diffserv
Folder name:
Advanced >>
Finish
Cancel
/inet/examples/adhoc
306M of 488M

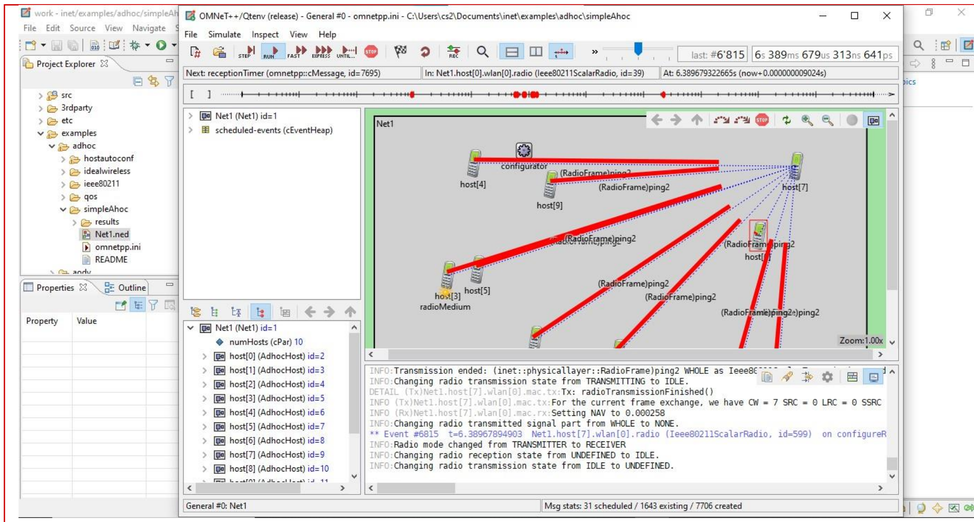

work - inet/examples/adhoc/simpleAhoc/Net1.ned - OMNeT++ IDE
New
Project...
File
File from Template
Folder
Analysis File (anf)
Class
Class (OMNeT++)
Compound Module
Header File
Initialization File (ini)
Message Definition (msg)
Network
Network Description File (NED)
Simple Module
Create a file that contains network elements, like simple or compound modules, channels etc.
Simulation
Source File
Source Folder
Wizard
C/C++ Project
OMNeT++ Project...
Other... Ctrl+N
345M of 488M


New NED File
Choose NED file
Enter or select the parent folder:
inet/examples/adhoc/simpleAhoc
File name: net1.ned
Advanced >>
< Back
Next >
Finish
Cancel


New NED File
Initial Contents
Select one of the options below
Select template:
Empty NED file
NED file with one item
Generated Wizards
New Adhoc Mobility Wireless Network Wizard
New Managed Mobility Wireless Network Wizard
Topology Generation
Topology Import
Add content template by URL


work - inet/examples/adhoc/simpleAhoc/Net1.ned - OMNeT++ IDE
Net1
configurator
radioMedium
host[numHosts]

Output



Name of Instructor: Jescia D’cruz

Wireless Sensor Networks & Mobile Communication

Practical No.5

DEPARTMENT OF COMPUTER SCIENCE

Name:	Ajay Kumar Uthaya Kumar	**Roll Number**	TCS2324002
Paper Code:	SIUSCS61	**Class**	TYBSc(Computer Science)
Topic:	Routing	**Batch**	I
Date:	29/1/24	**Practical No**	5

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

B) DESCRIPTION:

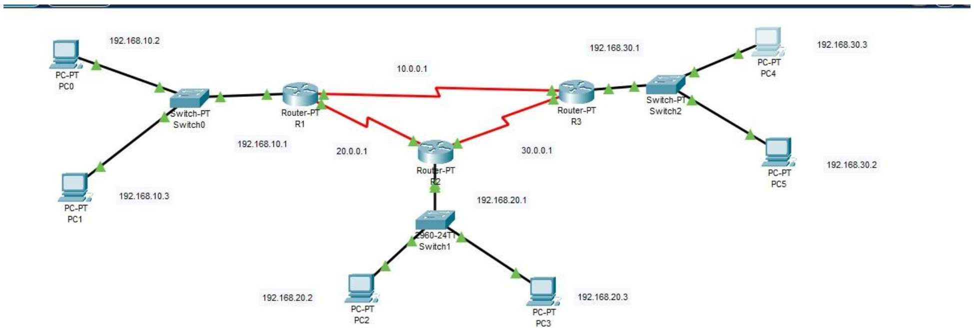
Network routing is the process of selecting a path across one or more networks. The principles of routing can apply to any type of network, from telephone networks to public transportation. In packet-switching networks, such as the Internet, routing selects the paths for Internet Protocol (IP) packets to travel from their origin to their destination. These Internet routing decisions are made by specialized pieces of network hardware called routers.

Name of Instructor: Jescia D'cruz

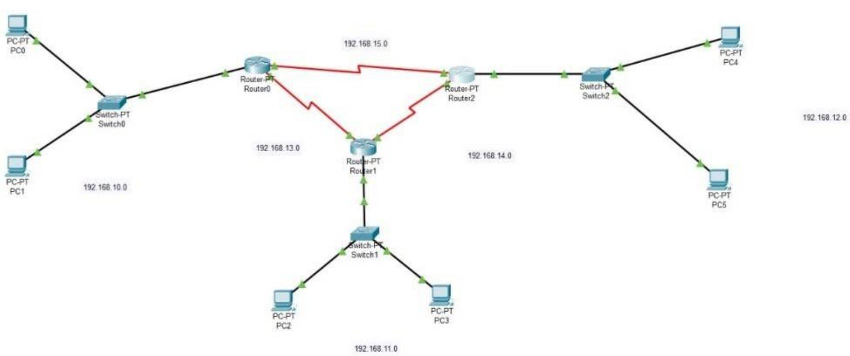
C) NETWORK TOPOLOGY, CONFIGURATIONS AND OUTPUT:

Network topology(only for cisco packet tracer practical's):

1) Using RIP

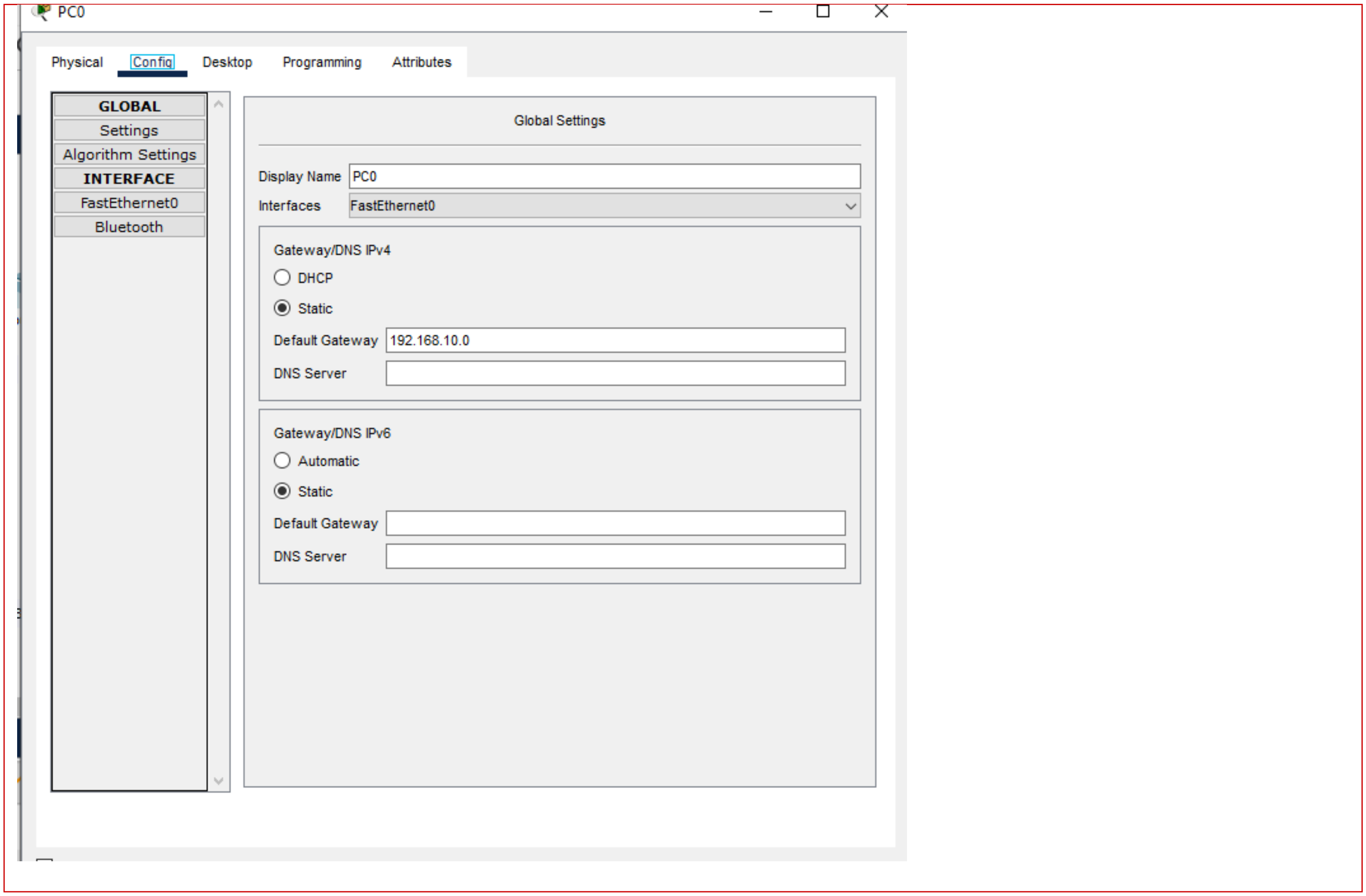


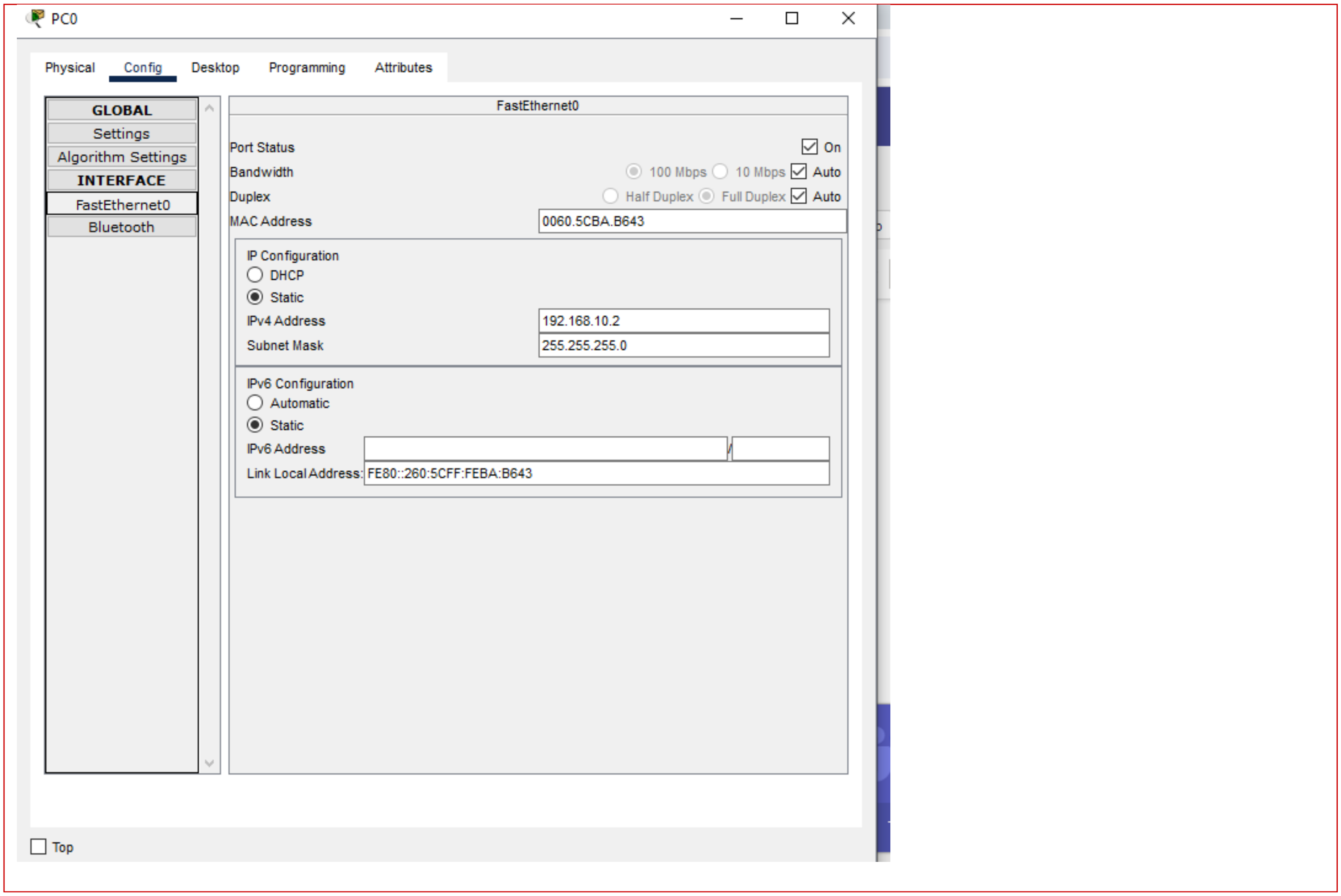
2) Using OSPF

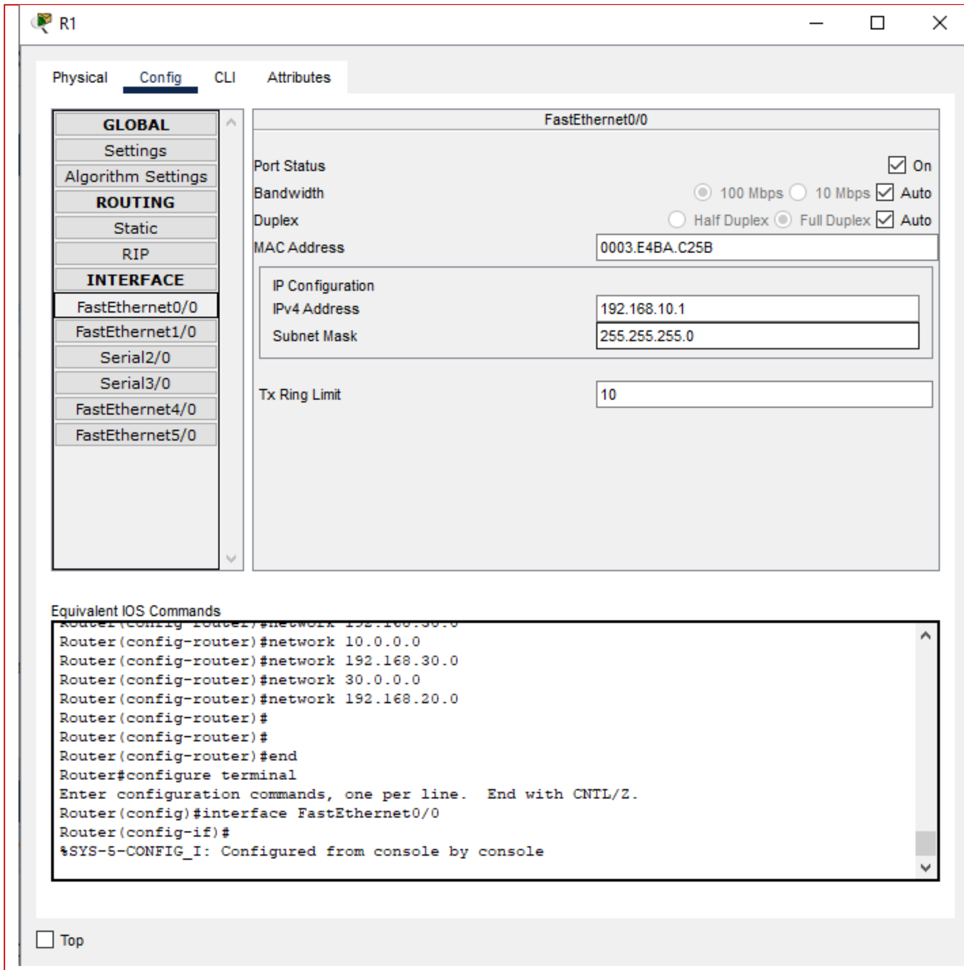


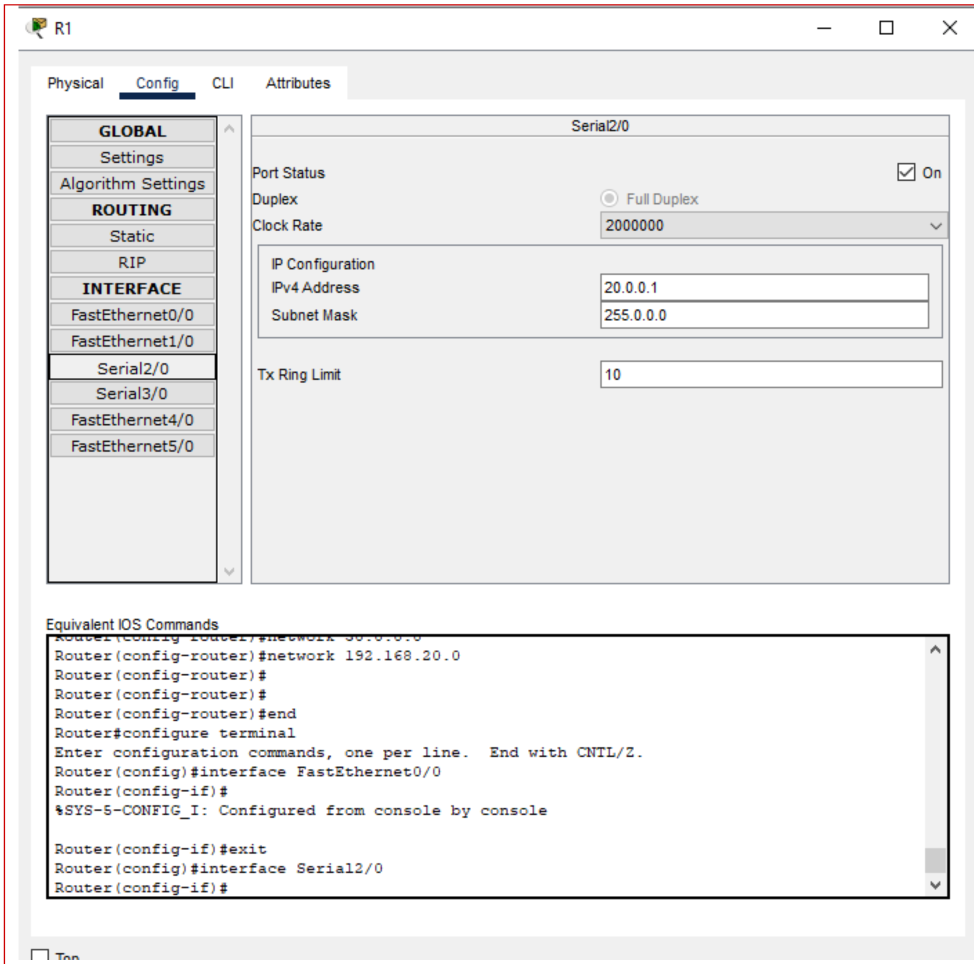
Configurations:

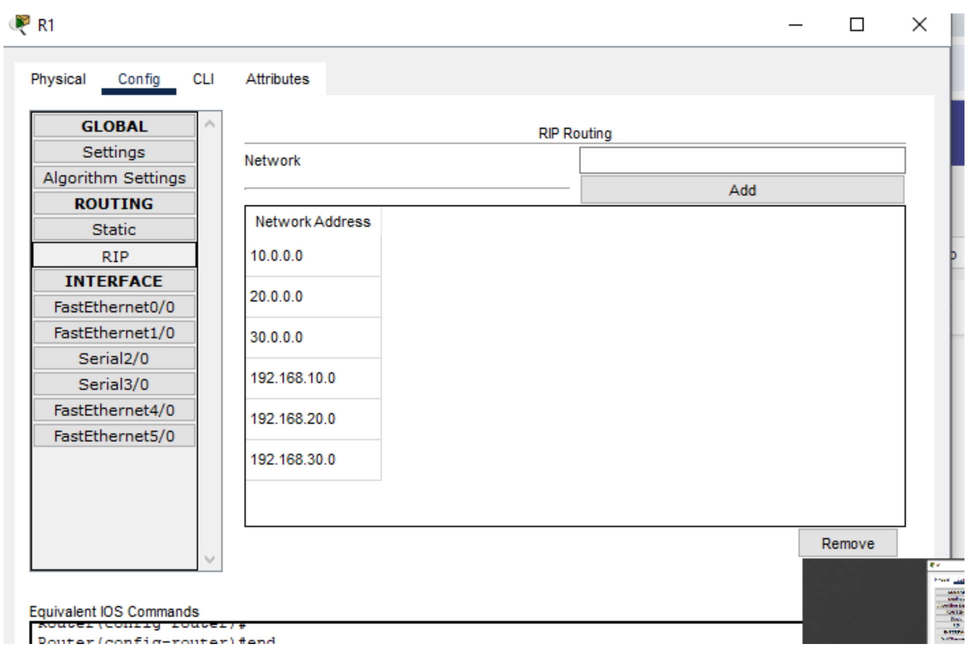
1) Using RIP


PC0
Physical
Config
Desktop
Programming
Attributes
GLOBAL
Settings
Algorithm Settings
INTERFACE
FastEthernet0
Bluetooth
Global Settings
Display Name
PC0
Interfaces
FastEthernet0
Gateway/DNS IPv4
DHCP
Static
Default Gateway
192.168.10.0
DNS Server
Gateway/DNS IPv6
Automatic
Static
Default Gateway
DNS Server

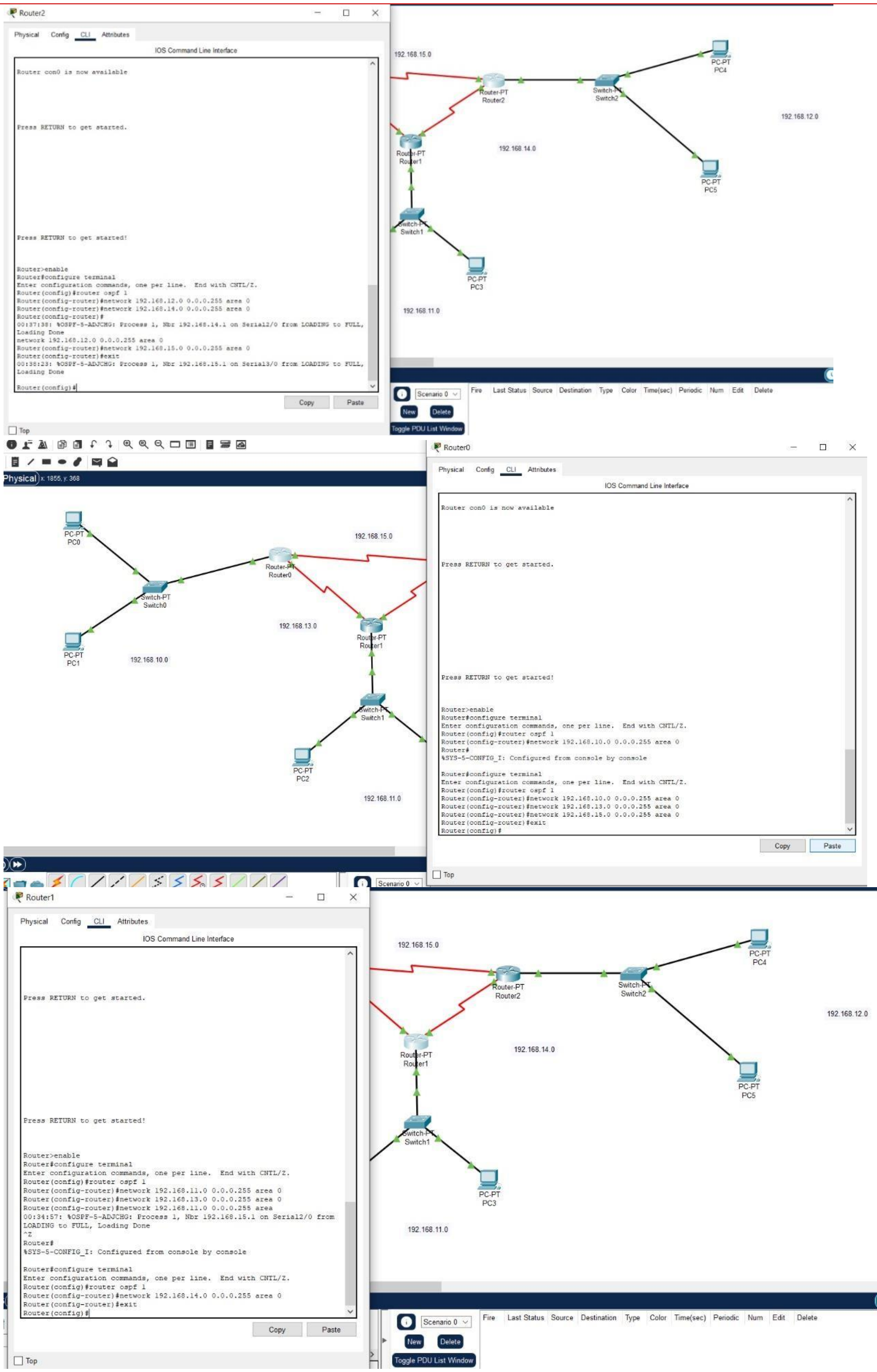

PC0
Physical
Config
Desktop
Programming
Attributes
GLOBAL
Settings
Algorithm Settings
INTERFACE
FastEthernet0
Bluetooth
FastEthernet0
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
0060.5CBA.B643
IP Configuration
DHCP
Static
IPv4 Address
192.168.10.2
Subnet Mask
255.255.255.0
IPv6 Configuration
Automatic
Static
IPv6 Address
Link Local Address:
FE80::260:5CFF:FEBA:B643
Top


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
FastEthernet0/0
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
0003.E4BA.C25B
IP Configuration
IPv4 Address
192.168.10.1
Subnet Mask
255.255.255.0
Tx Ring Limit
10
Equivalent IOS Commands
Router(config-router)#network 10.0.0.0
Router(config-router)#network 192.168.30.0
Router(config-router)#network 30.0.0.0
Router(config-router)#network 192.168.20.0
Router(config-router)#
Router(config-router)#
Router(config-router)#end
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#
%SYS-5-CONFIG_I: Configured from console by console
Top


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
Serial2/0
Port Status
On
Duplex
Full Duplex
Clock Rate
2000000
IP Configuration
IPv4 Address
20.0.0.1
Subnet Mask
255.0.0.0
Tx Ring Limit
10
Equivalent IOS Commands
Router(config-router)#network 192.168.20.0
Router(config-router)#
Router(config-router)#
Router(config-router)#end
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#
%SYS-5-CONFIG_I: Configured from console by console
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#


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

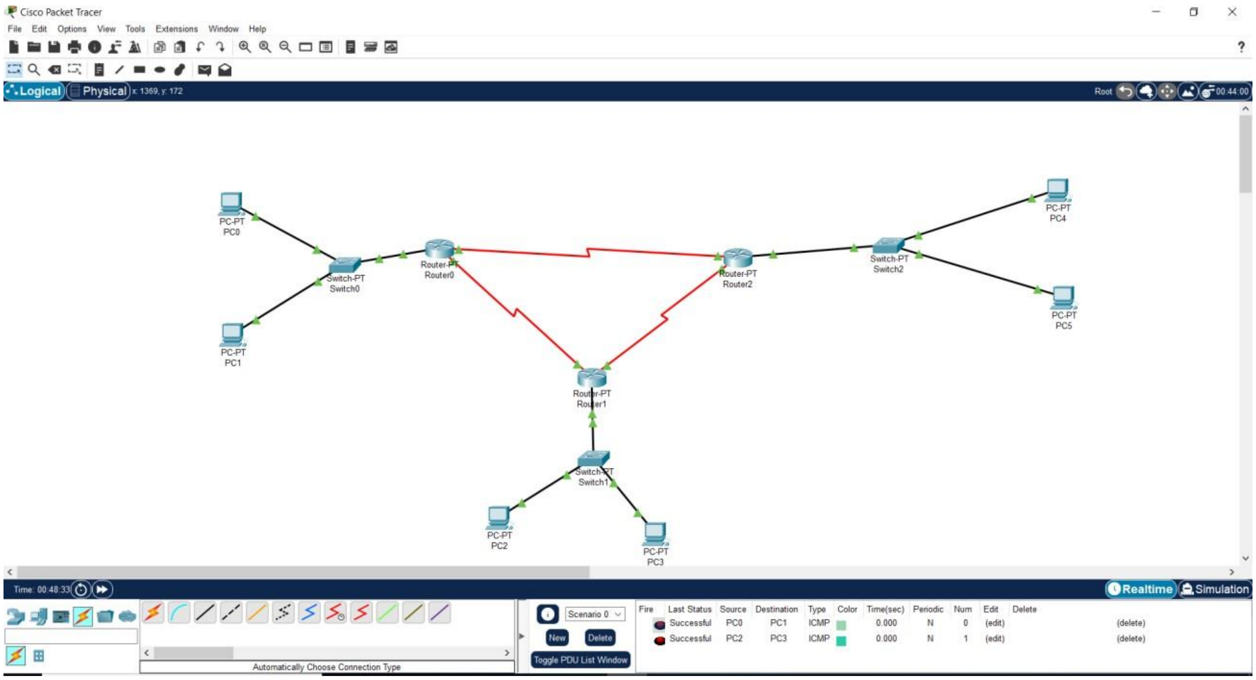
2) Using OSPF



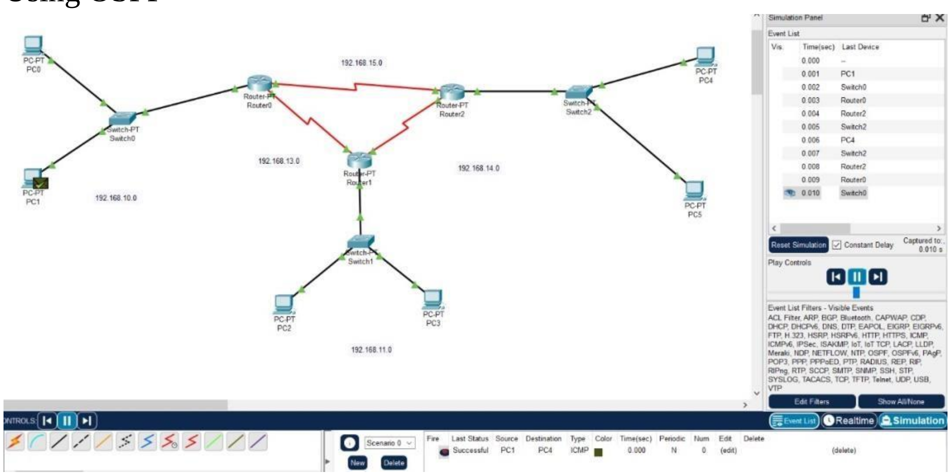
Router2
Physical
Config
CLI
Attributes
IOS Command Line Interface
Router con0 is now available
Press RETURN to get started.
Press RETURN to get started!
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#network 192.168.12.0 0.0.0.255 area 0
Router(config-router)#network 192.168.14.0 0.0.0.255 area 0
Router(config-router)#
00:37:38: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.14.1 on Serial2/0 from LOADING to FULL, Loading Done
network 192.168.12.0 0.0.0.255 area 0
Router(config-router)#network 192.168.15.0 0.0.0.255 area 0
Router(config-router)#exit
00:38:23: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.15.1 on Serial3/0 from LOADING to FULL, Loading Done
Router(config)#
Copy
Paste
Top
192.168.15.0
Router-PT
Router2
Switch-PT
Switch2
PC-PT
PC4
192.168.12.0
192.168.14.0
Router-PT
Router1
PC-PT
PC5
Switch-PT
Switch1
PC-PT
PC3
192.168.11.0
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num
Edit
Delete
Physical
x: 1855, y: 368
Router0
Physical
Config
CLI
Attributes
IOS Command Line Interface
Router con0 is now available
Press RETURN to get started.
Press RETURN to get started!
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#network 192.168.10.0 0.0.0.255 area 0
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#network 192.168.10.0 0.0.0.255 area 0
Router(config-router)#network 192.168.13.0 0.0.0.255 area 0
Router(config-router)#network 192.168.15.0 0.0.0.255 area 0
Router(config-router)#exit
Router(config)#
Copy
Paste
Top
PC-PT
PC0
192.168.15.0
Router-PT
Router0
Switch-PT
Switch0
192.168.13.0
PC-PT
PC1
192.168.10.0
Router-PT
Router1
Switch-PT
Switch1
PC-PT
PC2
192.168.11.0
Scenario 0
Router1
Physical
Config
CLI
Attributes
IOS Command Line Interface
Press RETURN to get started.
Press RETURN to get started!
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#network 192.168.11.0 0.0.0.255 area 0
Router(config-router)#network 192.168.13.0 0.0.0.255 area 0
Router(config-router)#network 192.168.11.0 0.0.0.255 area
00:34:57: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.15.1 on Serial2/0 from LOADING to FULL, Loading Done
^Z
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#network 192.168.14.0 0.0.0.255 area 0
Router(config-router)#exit
Router(config)#
Copy
Paste
Top
192.168.15.0
Router-PT
Router2
Switch-PT
Switch2
PC-PT
PC4
192.168.12.0
192.168.14.0
Router-PT
Router1
PC-PT
PC5
Switch-PT
Switch1
PC-PT
PC3
192.168.11.0
Scenario 0
New
Delete
Toggle PDU List Window
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num
Edit
Delete

Output

1) Using RIP



2) Using OSPF



Name of Instructor: Jescia D'cruz

Wireless Sensor Networks & Mobile Communication

Practical No.6

DEPARTMENT OF COMPUTER SCIENCE

Name:	Ajay Kumar Uthaya Kumar	Roll Number	TCS2324002
Paper Code:	SIUSCS61	Class	TYBSc(Computer Science)
Topic:	Wireless sensor network	Batch	I
Date:	05/1/24	Practical No	6

A) AIM: Implement a Wireless sensor network simulation

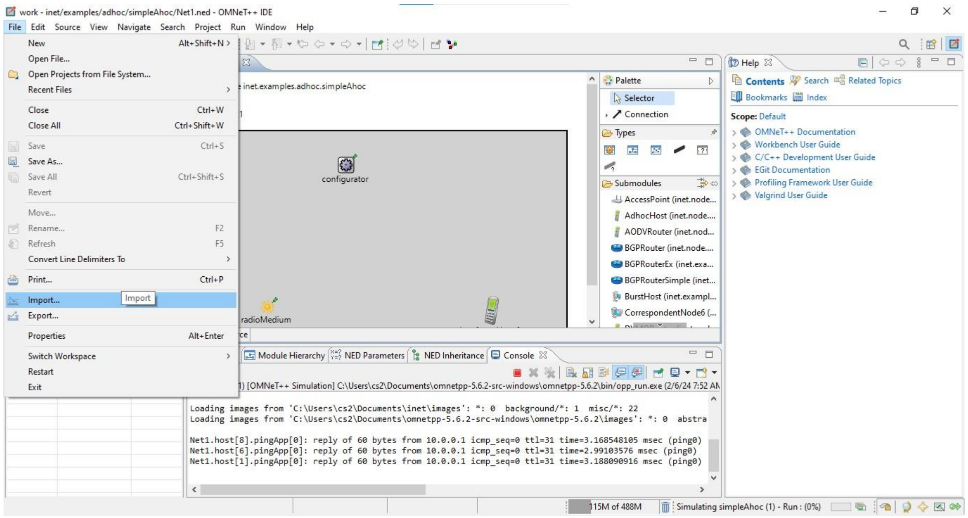
B) DESCRIPTION:

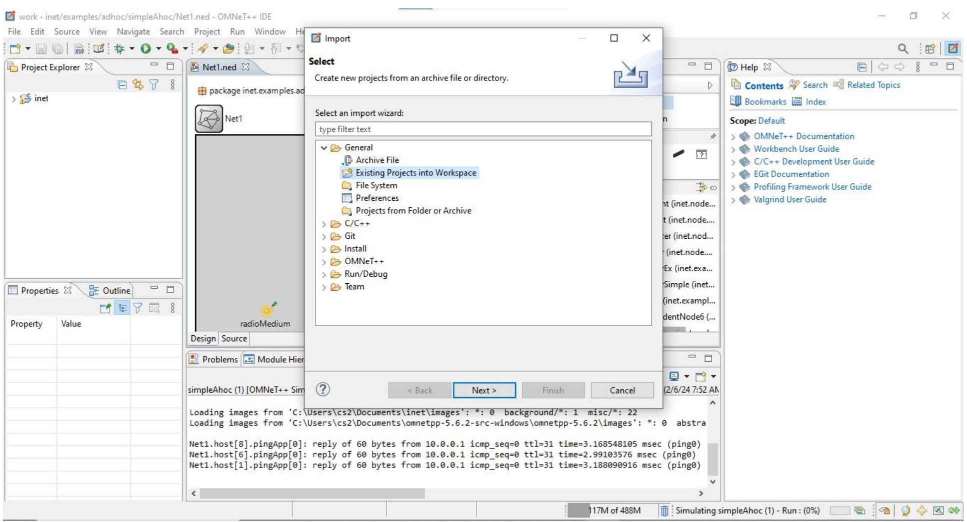
Wireless Sensor Network (WSN) is an infrastructure-less wireless network that is deployed in a large number of wireless sensors in an ad-hoc manner that is used to monitor the system, physical or environmental conditions. Sensor nodes are used in WSN with the onboard processor that manages and monitors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

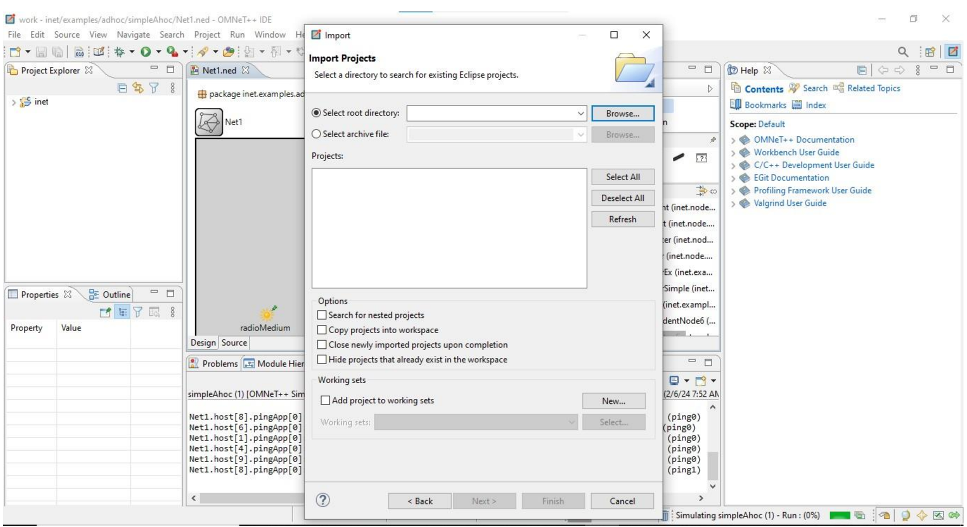
Name of Instructor: Jescia D'cruz

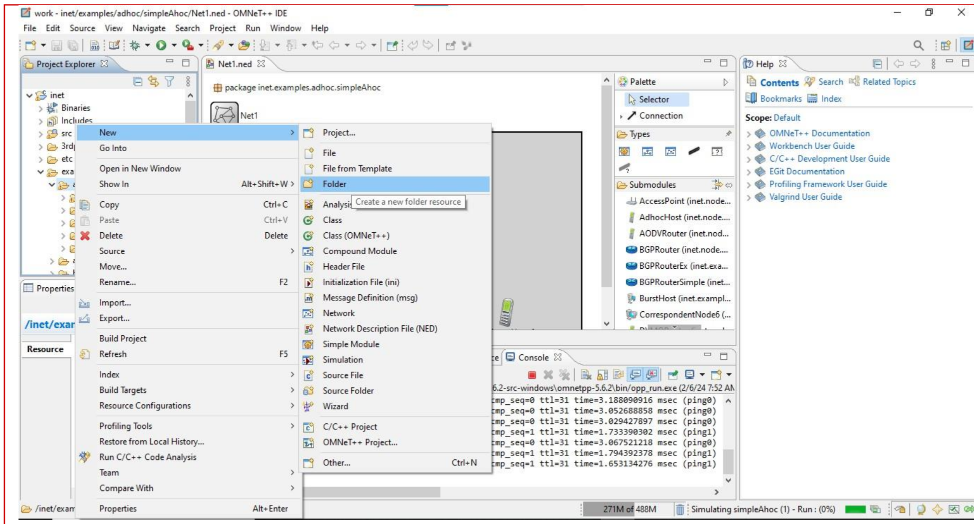
C) NETWORK TOPOLOGY, CONFIGURATIONS AND OUTPUT:

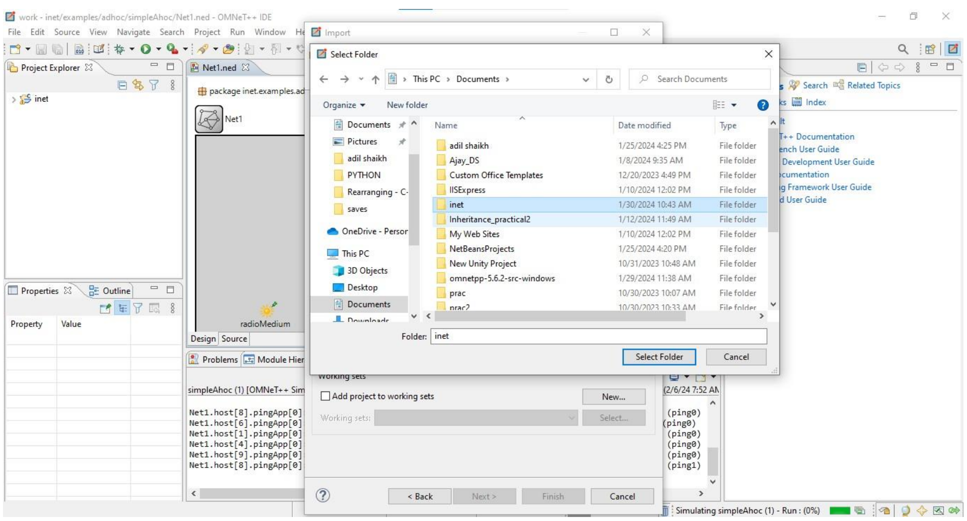
Configurations:

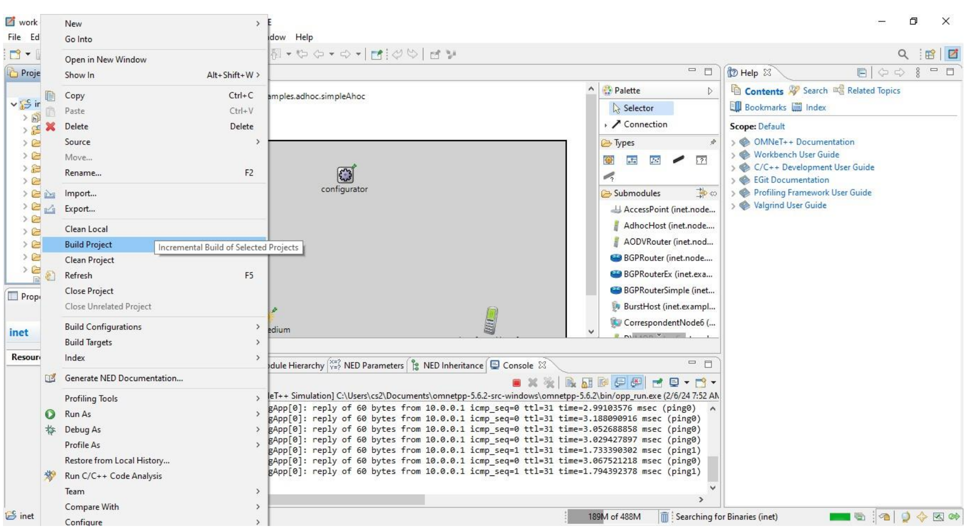




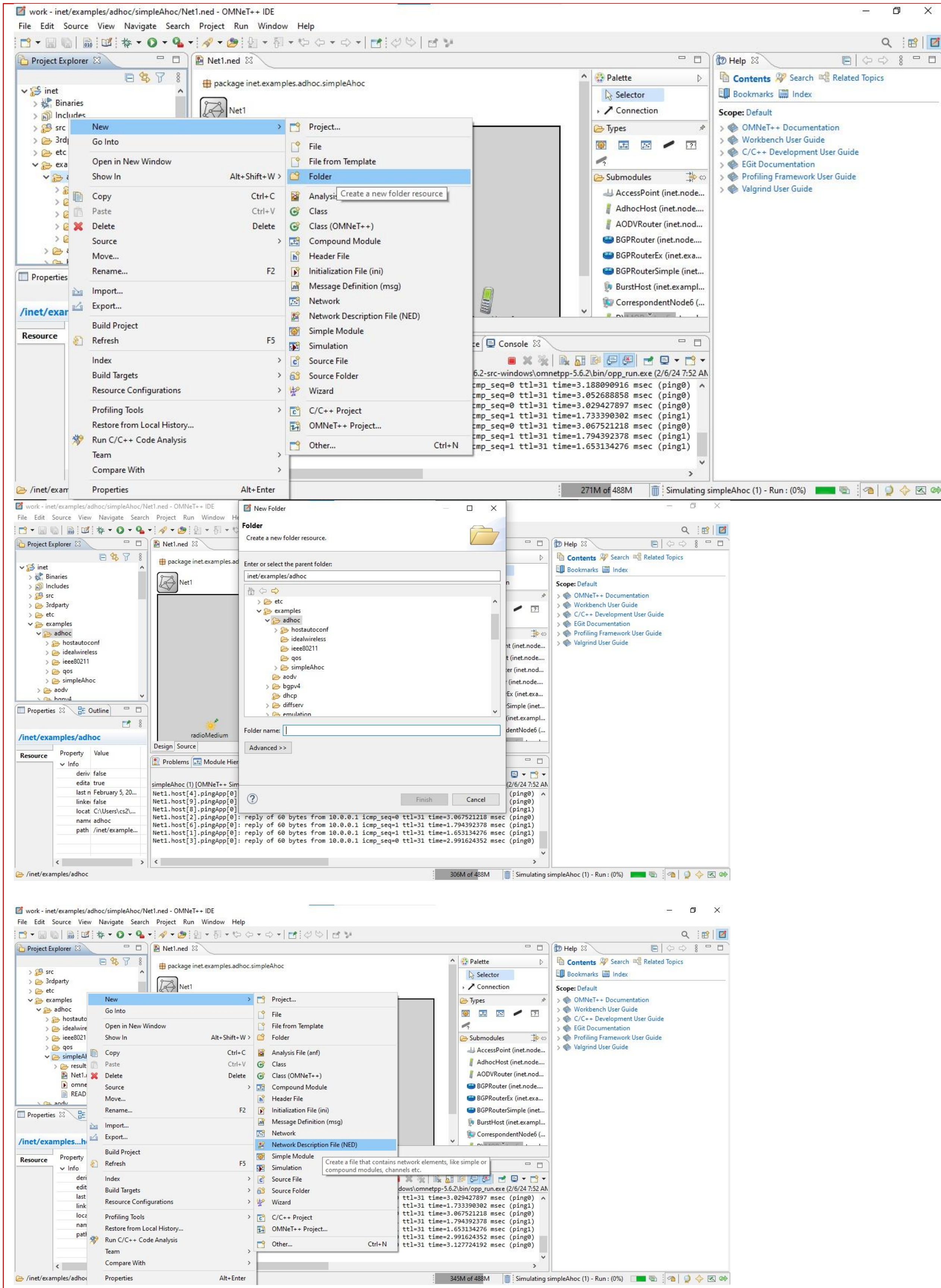


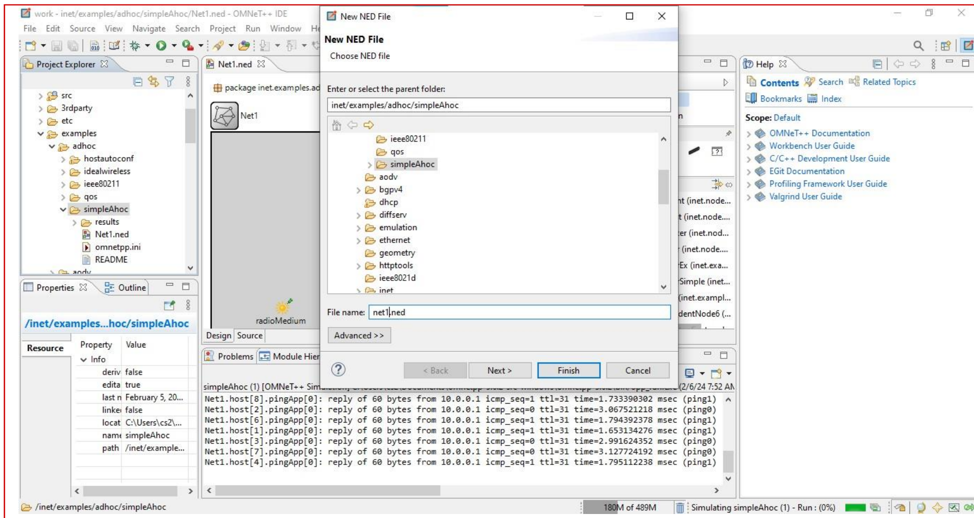


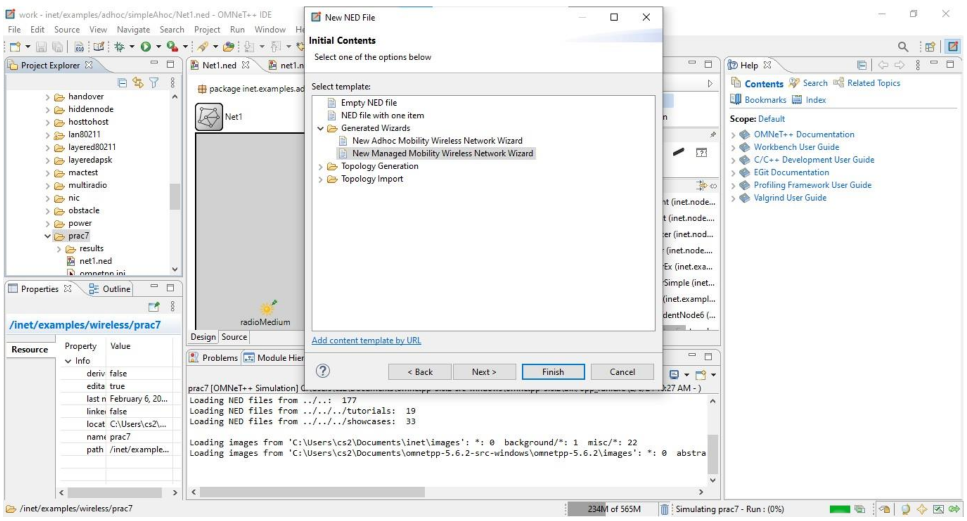


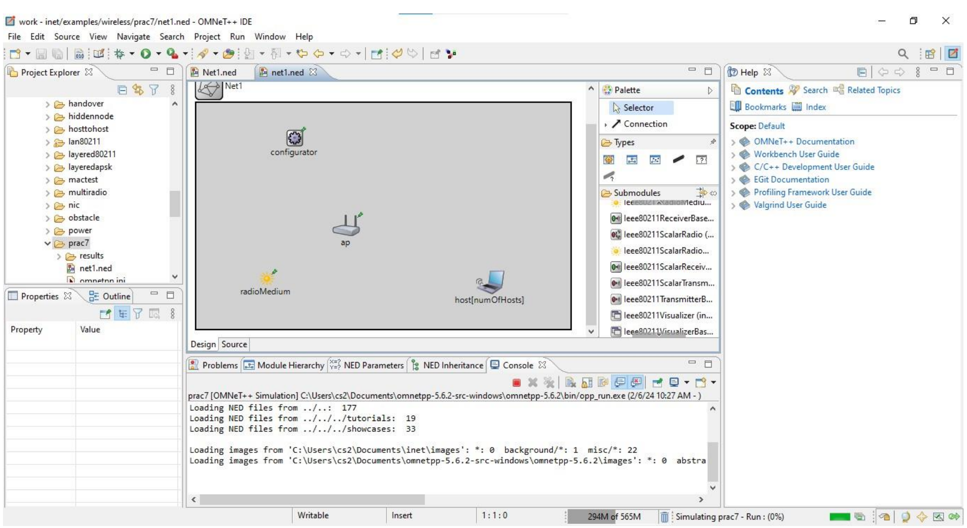


Name of Instructor: Jescia D’cruz


work - inet/examples/adhoc/simpleAhoc/Net1.ned - OMNeT++ IDE
File Edit Source View Navigate Search Project Run Window Help
Project Explorer
Net1.ned
package inet.examples.adhoc.simpleAhoc
New
Go Into
Open in New Window
Show In Alt+Shift+W
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Source
Move...
Rename... F2
Import...
Export...
Build Project
Refresh F5
Index
Build Targets
Resource Configurations
Profiling Tools
Restore from Local History...
Run C/C++ Code Analysis
Team
Compare With
Properties Alt+Enter
Project...
File
File from Template
Folder
Create a new folder resource
Class
Class (OMNeT++)
Compound Module
Header File
Initialization File (ini)
Message Definition (msg)
Network
Network Description File (NED)
Simple Module
Simulation
Source File
Source Folder
Wizard
C/C++ Project
OMNeT++ Project...
Other... Ctrl+N
Palette
Selector
Connection
Types
Submodules
Console
Help
Contents
Search
Related Topics
Bookmarks
Index
Scope: Default
OMNeT++ Documentation
Workbench User Guide
C/C++ Development User Guide
EGit Documentation
Profiling Framework User Guide
Valgrind User Guide
Simulating simpleAhoc (1) - Run : (0%)
New Folder
Folder
Create a new folder resource.
Enter or select the parent folder:
inet/examples/adhoc
Folder name:
Advanced >>
Finish
Cancel
Create a file that contains network elements, like simple or compound modules, channels etc.







Code:

//

// This program is free software: you can redistribute it and/or modify

```
// it under the terms of the GNU Lesser General Public License as published by
// the Free Software Foundation, either version 3 of the License, or
// (at your option) any later version.
//
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
// GNU Lesser General Public License for more details.
//
// You should have received a copy of the GNU Lesser General Public License
// along with this program.  If not, see http://www.gnu.org/licenses/.
//

package inet.examples.wireless.prac7;

// numOfHosts: 5

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

network Net1
{
   parameters:
```

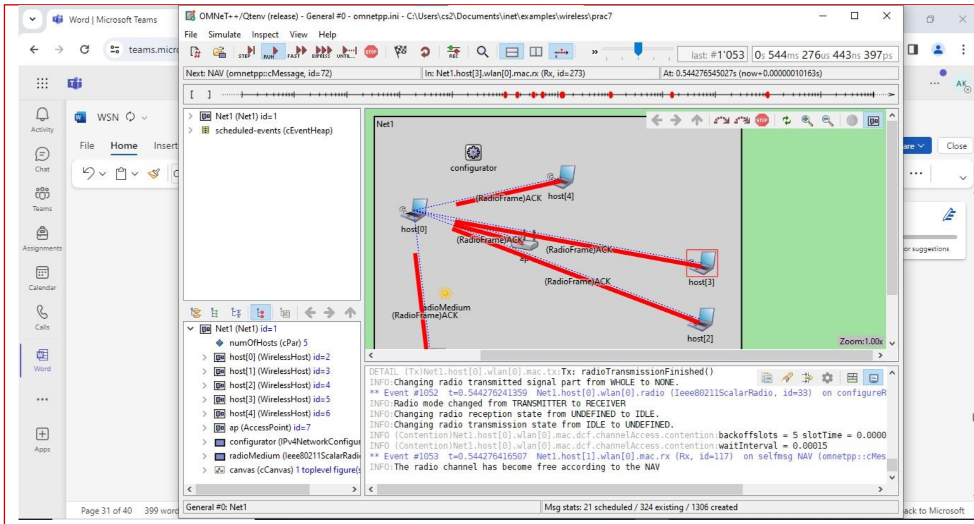
```
    int numOfHosts;


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


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


    configurator: IPv4NetworkConfigurator {
      @display("p=140,50");
    }
    radioMedium: Ieee80211ScalarRadioMedium {
      parameters:
        @display("p=100,250");
    }
}
```

Output



Name of Instructor: Jescia D'cruz

Wireless Sensor Networks & Mobile Communication

Practical No.7

DEPARTMENT OF COMPUTER SCIENCE

Name:	Ajay Kumar Uthaya Kumar	**Roll Number**	TCS2324002
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 network

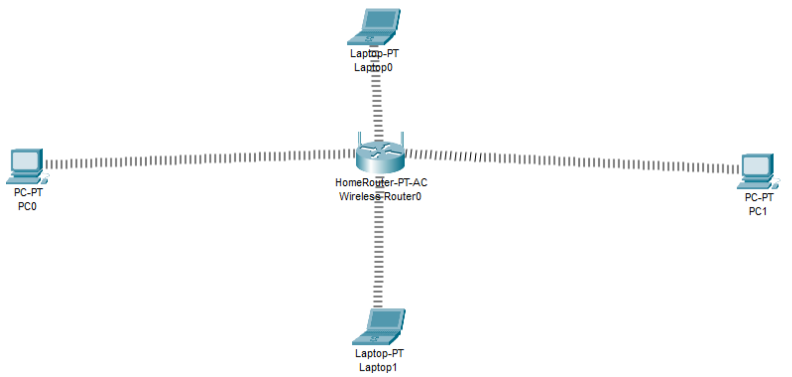
B) DESCRIPTION:

Media access control (MAC) protocols enforce a methodology to allow multiple devices access to a shared media network. Before LANs, communication between computing devices had been point-to-point. That is, two devices were connected by a dedicated channel. LANs are shared media networks, in which all devices attached to the network receive each transmission and must recognize which frames they should accept. Media sharing reduced the cost of the network, but also meant that MAG protocols were needed to coordinate use of the medium. There are two approaches to media access control in LANs: contention and token-passing.

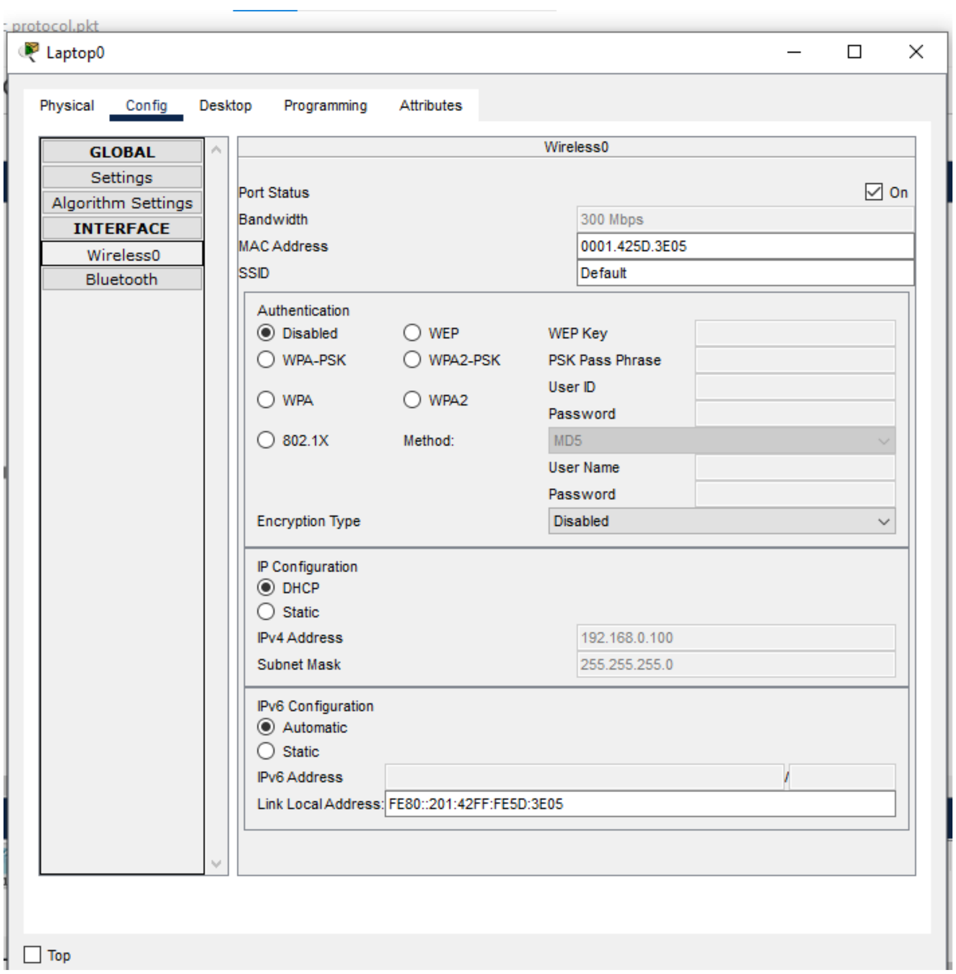
C) NETWORK TOPOLOGY, CONFIGURATIONS AND OUTPUT:

C) NETWORK TOPOLOGY, CONFIGURATIONS AND OUTPUT:

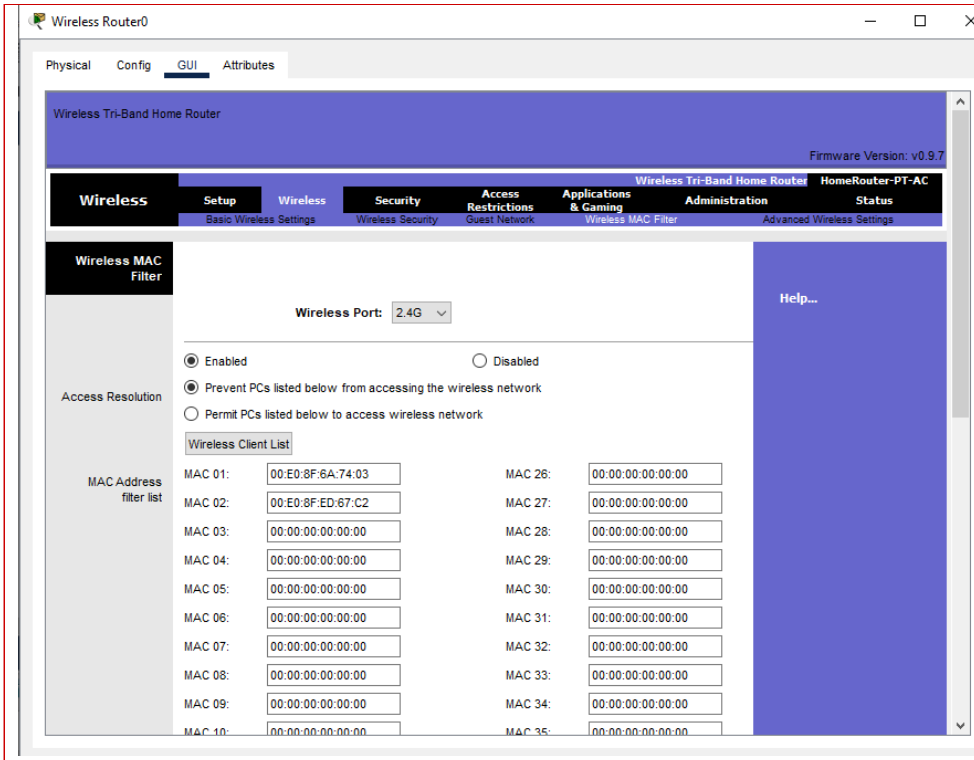
Network topology(only for cisco packet tracer practical’s):



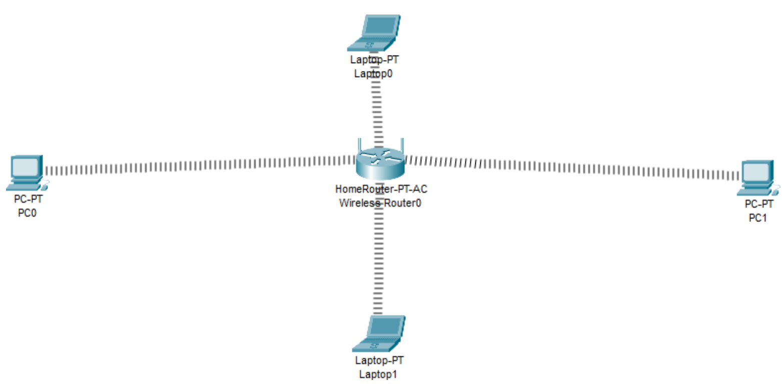
Configurations:



Name of Instructor: Jescia D’cruz



Output



Name of Instructor: Jescia D'cruz

Wireless Sensor Networks & Mobile Communication

Practical No.8

DEPARTMENT OF COMPUTER SCIENCE

Name:	Ajay Kumar Uthaya Kumar	**Roll Number**	TCS2324002
Paper Code:	SIUSCS61	**Class**	TYBSc(Computer Science)
Topic:	Mobile Adhoc	**Batch**	I
Date:	06/2/24	**Practical No**	8

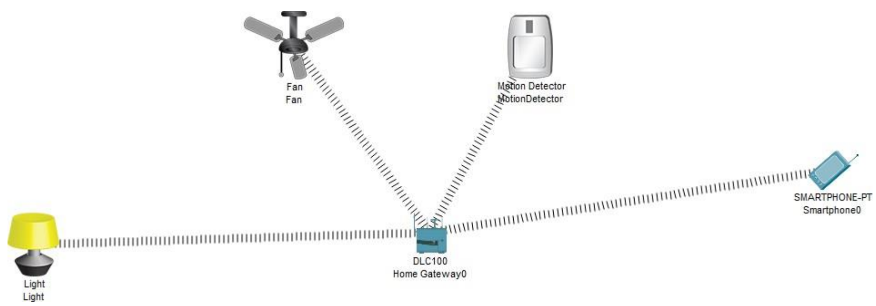
A) AIM: Stimulate a Mobile Adhoc network with directional antennas

B) DESCRIPTION:

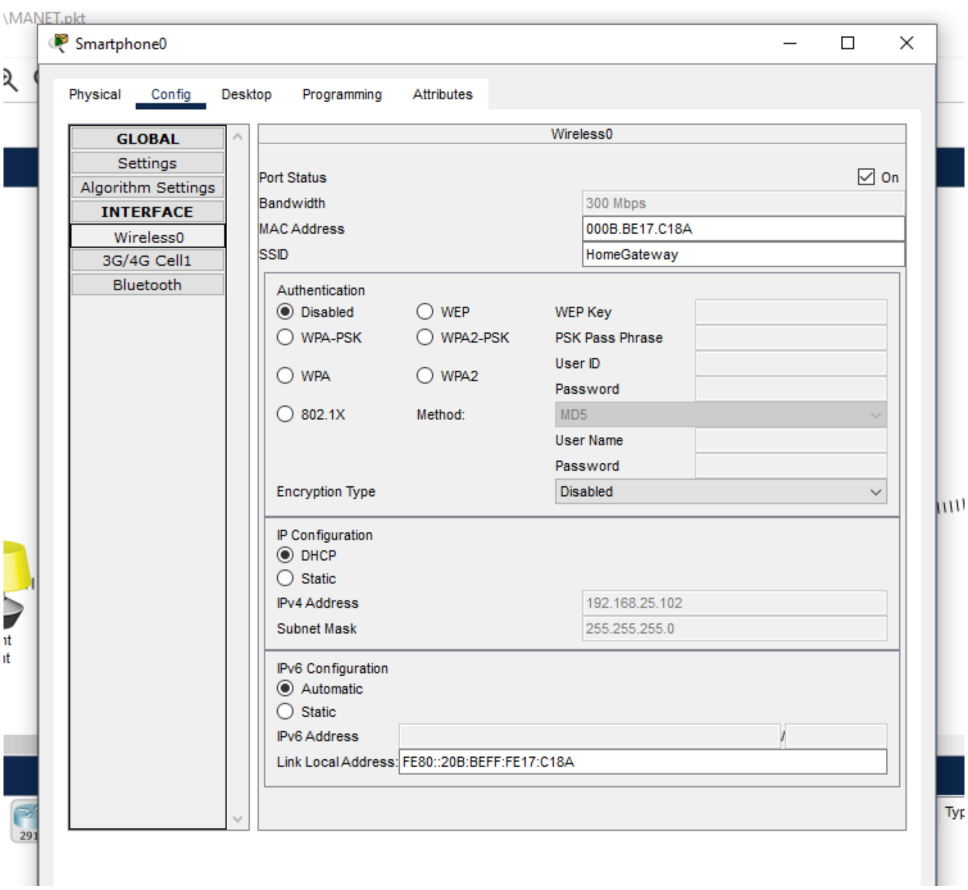
MANET stands for Mobile Adhoc Network also called a wireless Adhoc network or Adhoc wireless network that usually has a routable networking environment on top of a Link Layer ad hoc network.. They consist of a set of mobile nodes connected wirelessly in a self-configured, self-healing network without 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.

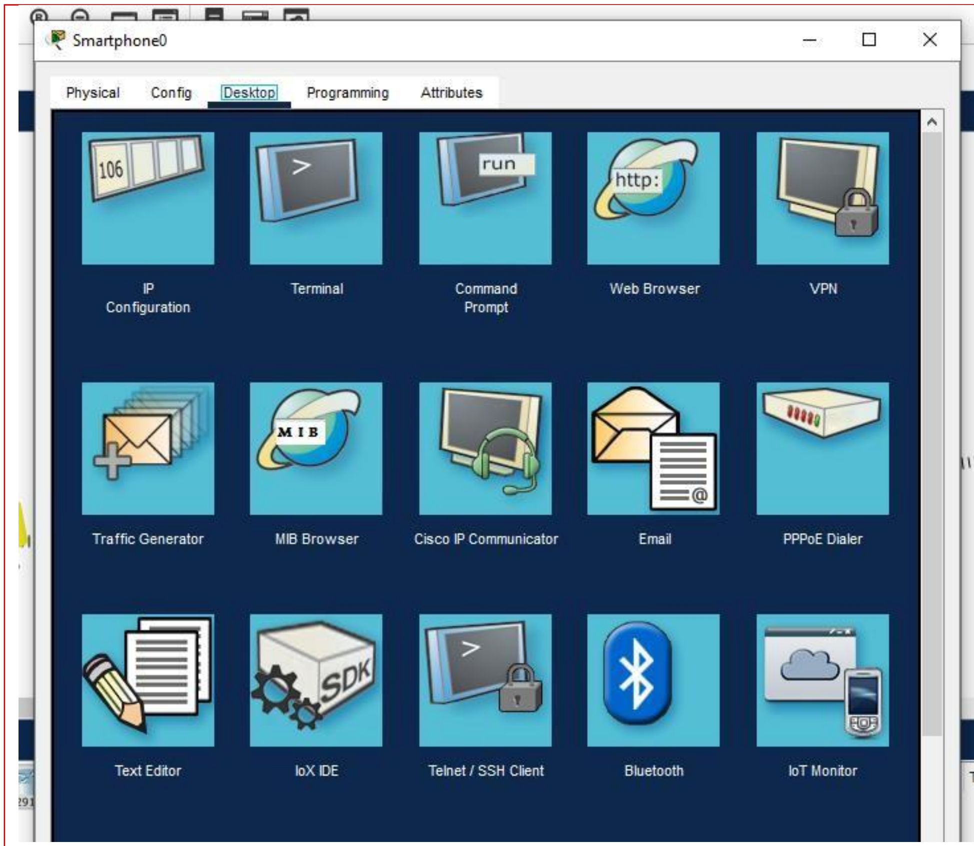
C) NETWORK TOPOLOGY, CONFIGURATIONS AND OUTPUT:

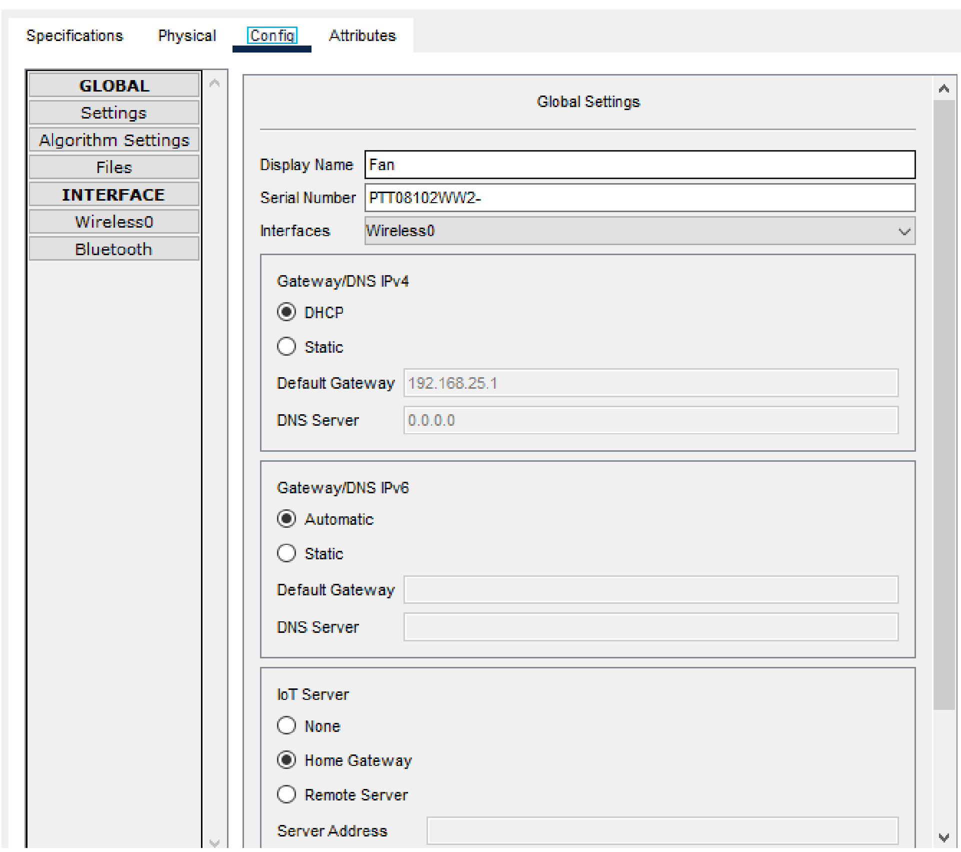
Network topology(only for cisco packet tracer practical's):

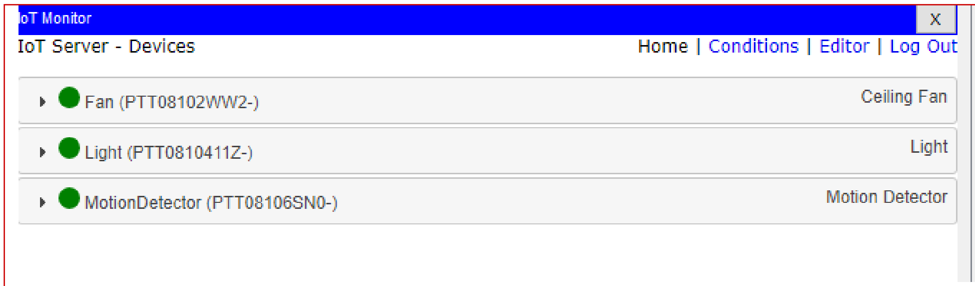


Configurations:

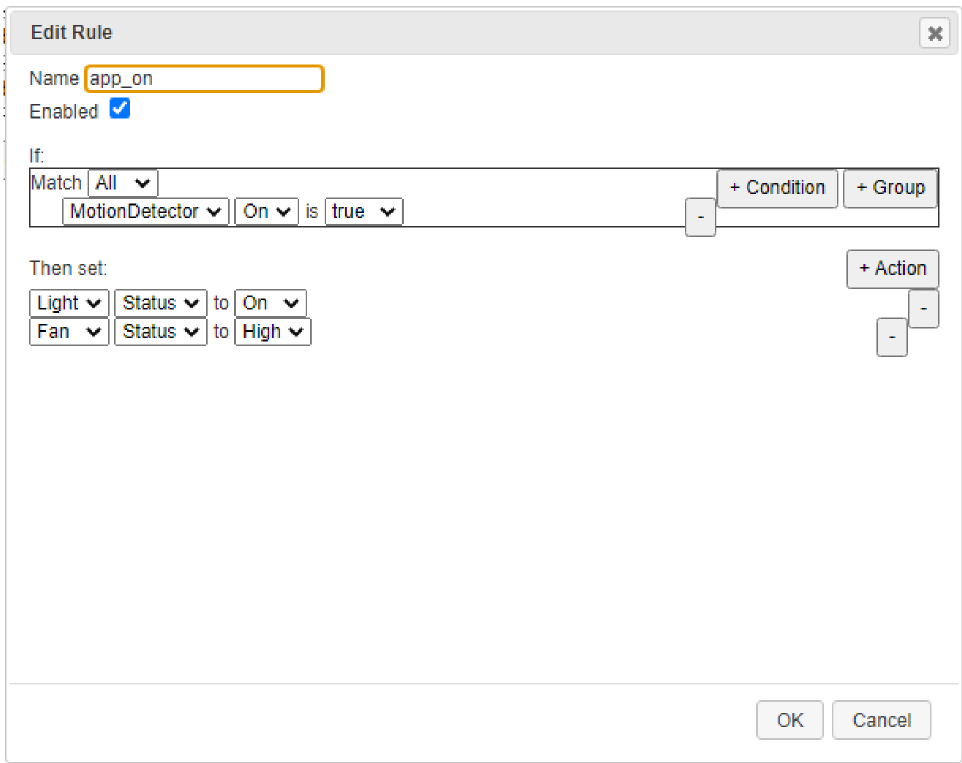



Smartphone0
Physical
Config
Desktop
Programming
Attributes
IP Configuration
Terminal
Command Prompt
Web Browser
VPN
Traffic Generator
MIB Browser
Cisco IP Communicator
Email
PPPoE Dialer
Text Editor
IoX IDE
Telnet / SSH Client
Bluetooth
IoT Monitor

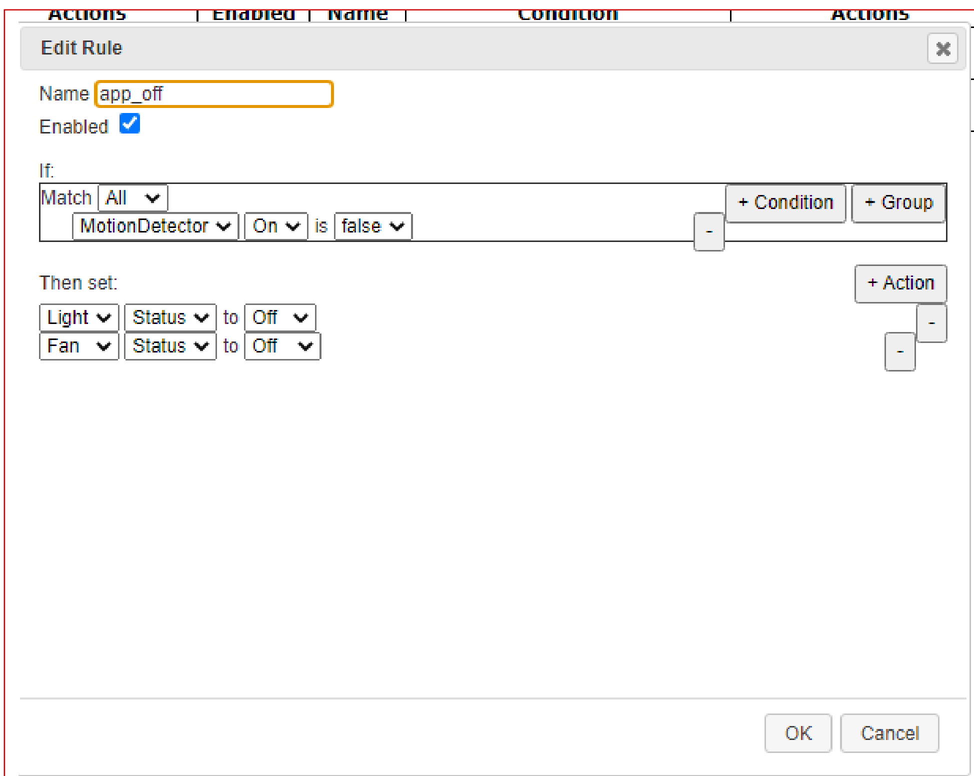

Specifications
Physical
Config
Attributes
GLOBAL
Settings
Algorithm Settings
Files
INTERFACE
Wireless0
Bluetooth
Global Settings
Display Name Fan
Serial Number PTT08102WW2-
Interfaces Wireless0
Gateway/DNS IPv4
DHCP
Static
Default Gateway 192.168.25.1
DNS Server 0.0.0.0
Gateway/DNS IPv6
Automatic
Static
Default Gateway
DNS Server
IoT Server
None
Home Gateway
Remote Server
Server Address



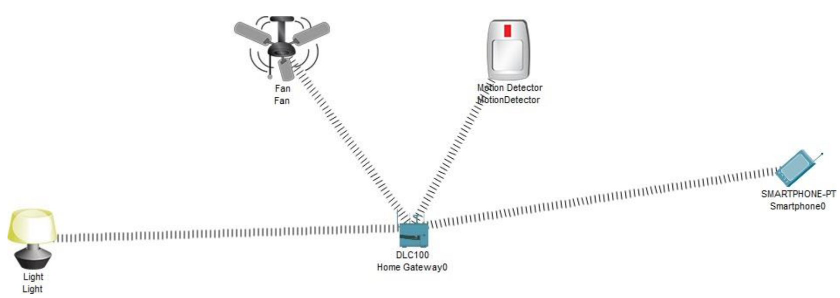
Conditions



Name of Instructor: Jescia D'cruz


Actions | Enabled | Name | Condition | Actions
Edit Rule
Name app_off
Enabled
If:
Match All
MotionDetector On is false
+ Condition
+ Group
Then set:
+ Action
Light Status to Off
Fan Status to Off
OK
Cancel

Output


Fan
Fan
Motion Detector
MotionDetector
SMARTPHONE-PT
Smartphone0
DLC100
Home Gateway0
Light
Light

Wireless Sensor Networks & Mobile Communication

Practical No.9

DEPARTMENT OF COMPUTER SCIENCE

Name:	Ajay Kumar Uthaya Kumar	**Roll Number**	TCS2324002
Paper Code:	SIUSCS61	**Class**	TYBSc(Computer Science)
Topic:	Mobile Network	**Batch**	I
Date:	06/2/24	**Practical No**	9

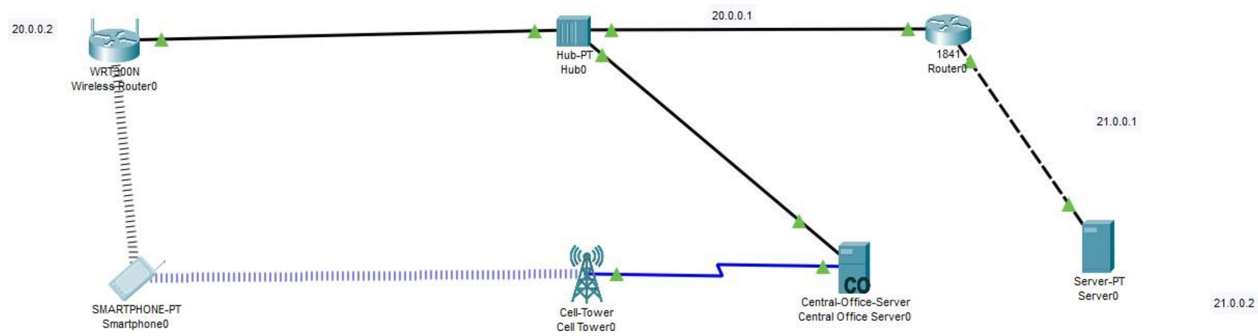
A) AIM: Create a mobile network using Cell Tower, Central office server, Web Browser and Web Server, Stimulate connection between them.

B) DESCRIPTION:

A mobile network (also wireless network) route's communications in the form of radio waves to and from users. It is composed of base stations that each cover a delimited area or "cell." When joined together these cells provide radio coverage over a wide geographic area. This enables a large number of portable transceivers (e.g., mobile phones, pagers, etc.) to communicate with each other and with fixed transceivers and telephones anywhere in the network, even if some of the transceivers are moving through more than one cell during transmission. Mobile networks are rapidly becoming the universal service delivery vehicle for all applications. The key question is whether they can manage to keep up with the underlying bandwidth demands. The rising demand for mobile broadband services has accelerated the move to LTE and LTE-Advanced. This latest mobile technology further increases not only bandwidth but also quality requirements of the backhaul network.

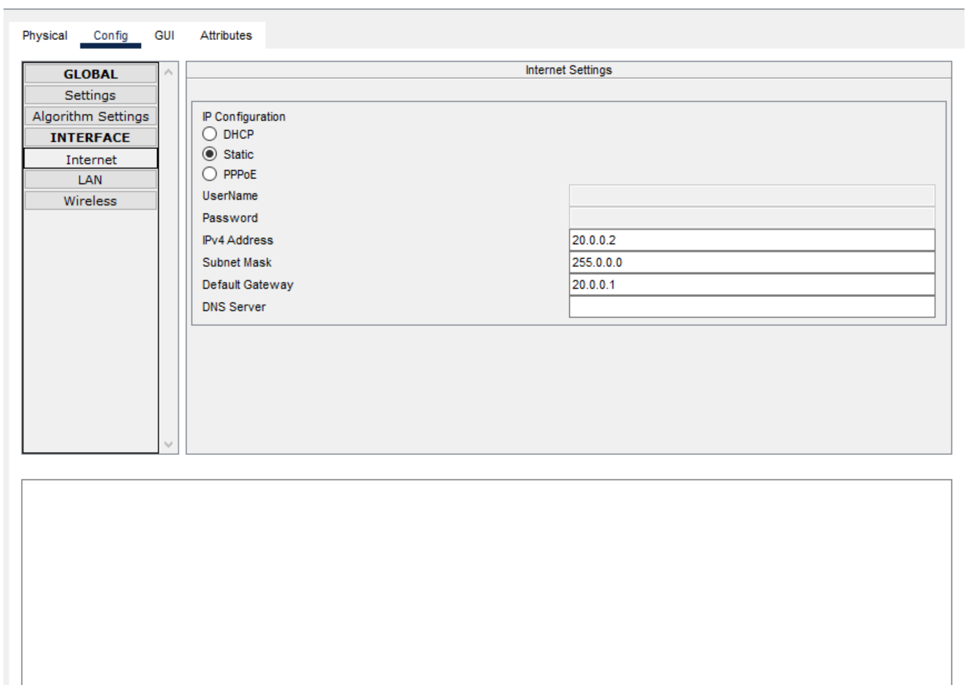
C) NETWORK TOPOLOGY, CONFIGURATIONS AND OUTPUT:

Network topology(only for cisco packet tracer practical's):



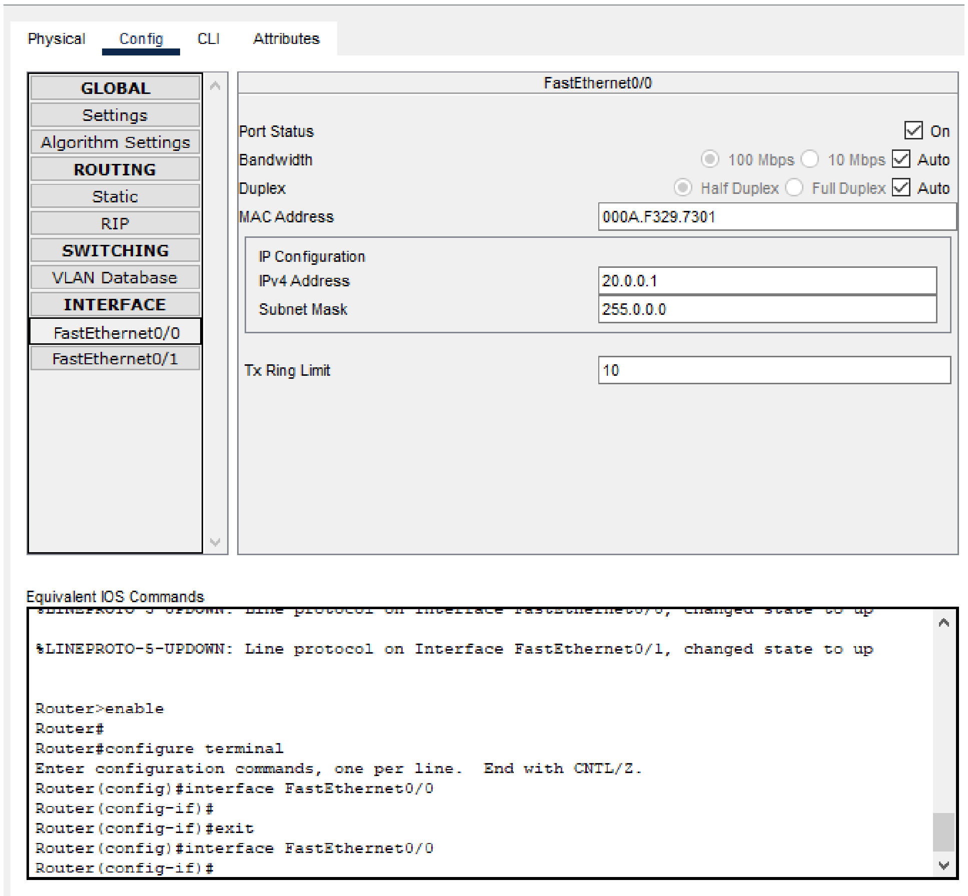
Configurations:

Wireless router configuration

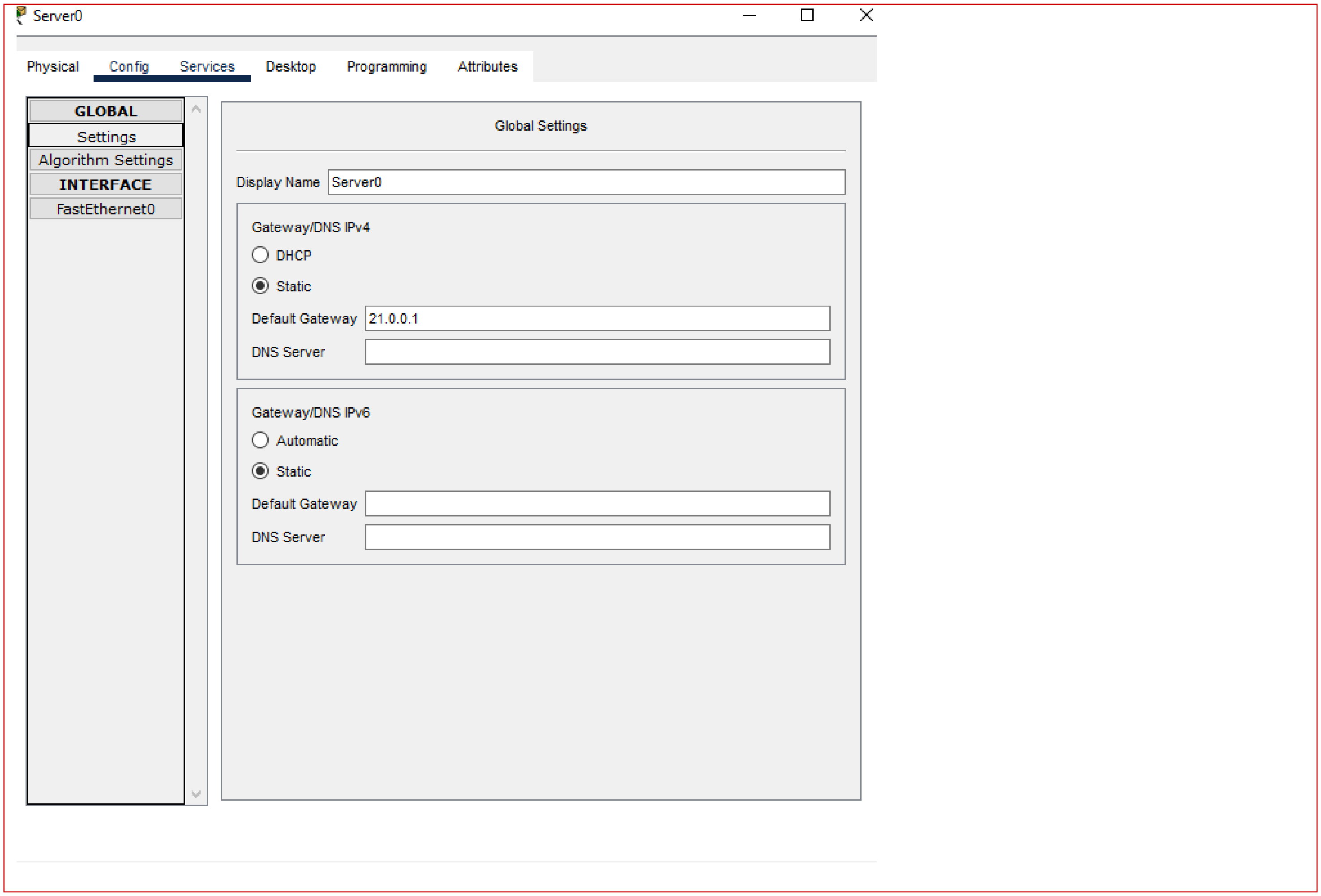


1841 router configuration

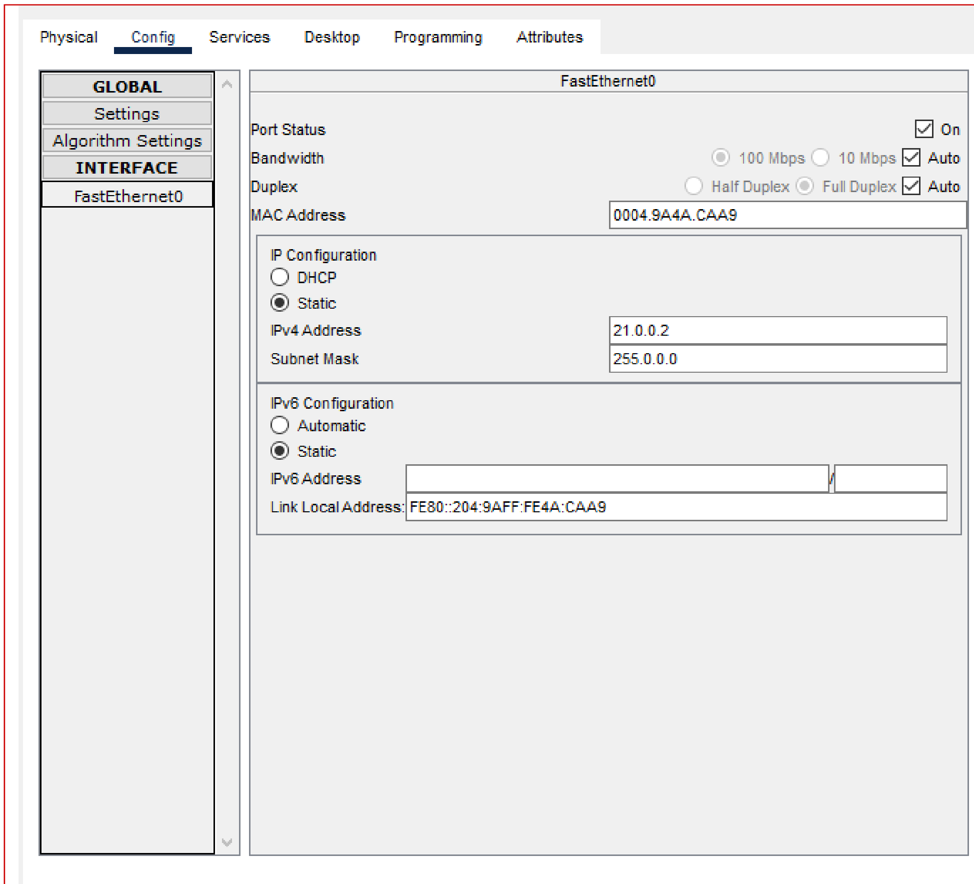
Name of Instructor: Jescia D'cruz



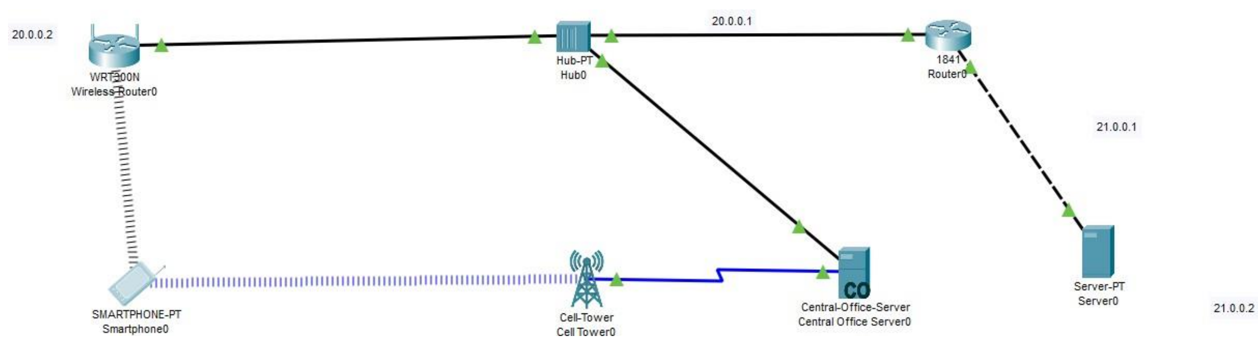
Server configuration



Name of Instructor: Jescia D'cruz


Physical
Config
Services
Desktop
Programming
Attributes
GLOBAL
Settings
Algorithm Settings
INTERFACE
FastEthernet0
FastEthernet0
Port Status
On
Bandwidth
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
0004.9A4A.CAA9
IP Configuration
DHCP
Static
IPv4 Address
21.0.0.2
Subnet Mask
255.0.0.0
IPv6 Configuration
Automatic
Static
IPv6 Address
Link Local Address:
FE80::204:9AFF:FE4A:CAA9

Output


20.0.0.2
WRT300N
Wireless Router0
Hub-PT
Hub0
20.0.0.1
1841
Router0
21.0.0.1
SMARTPHONE-PT
Smartphone0
Cell-Tower
Cell Tower0
Central-Office-Server
Central Office Server0
Server-PT
Server0
21.0.0.2