**Practical No. 01**

**AIM:** Create network with three routers with RIPv2 and each router associated network will have minimum three PC. Show connectivity.

**Steps to perform practical:**

Step 1 : Add 3 routers, 3 switches, end user clients to the switches.

Step 2 : Connect all the devices with straight through cable.

Step 3 : Add serial port devices in the routers in the router.

Step 4 : Give ip to each port of router

enable

configure terminal

interface gig0/0/0

ip address 192.168.10.1 255.255.255.0

no shutdown

exit

Step 5 : Set IP address, Subnet & Gateway for End User Client (manually in input boxes). Note that the gateway will be same as the routers IP.

Step 6 : Apply RIP to each of the router

configure terminal

router rip

network 192.168.10.0

network 192.168.40.0

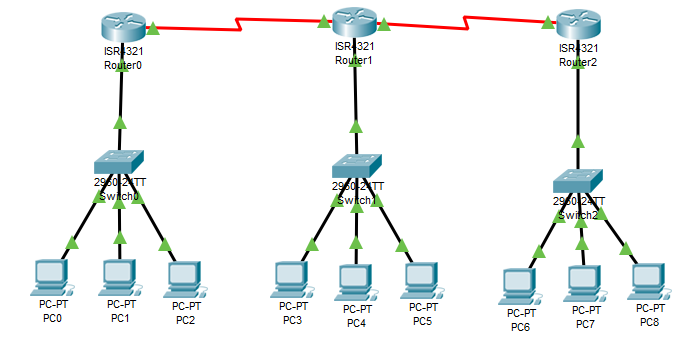
exit

Step 7 : Go for (show ip route) command on each router.

Step 8 : Show connection between PC6 and PC0.

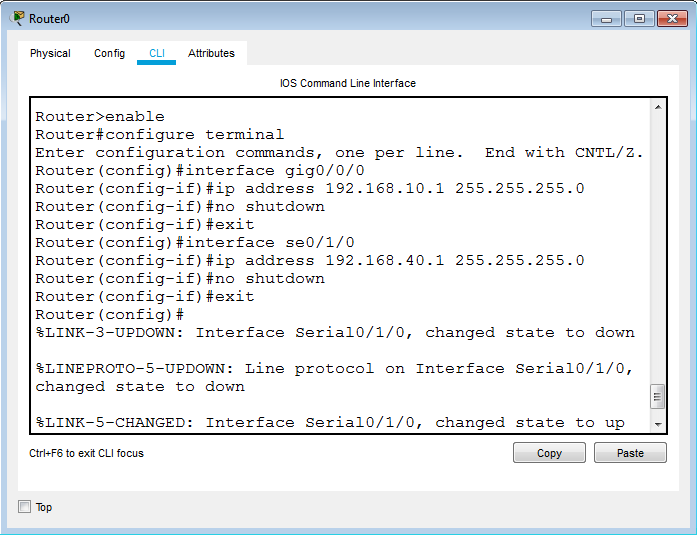
System > Desktop tab > Command Prompt > ping 192.168.10.2

**NETWORK TOPOLOGY:**

****

**i)Router Configuration:-**

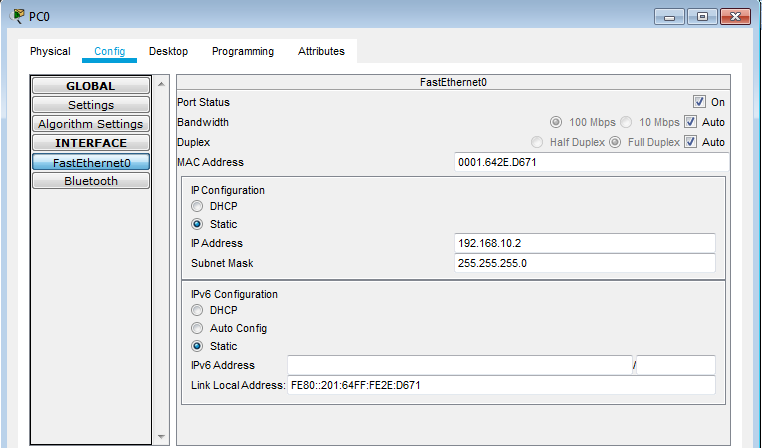
**Set IP addresses to routers :-**

****

Set IP addresses to all remaining routers as shown above.

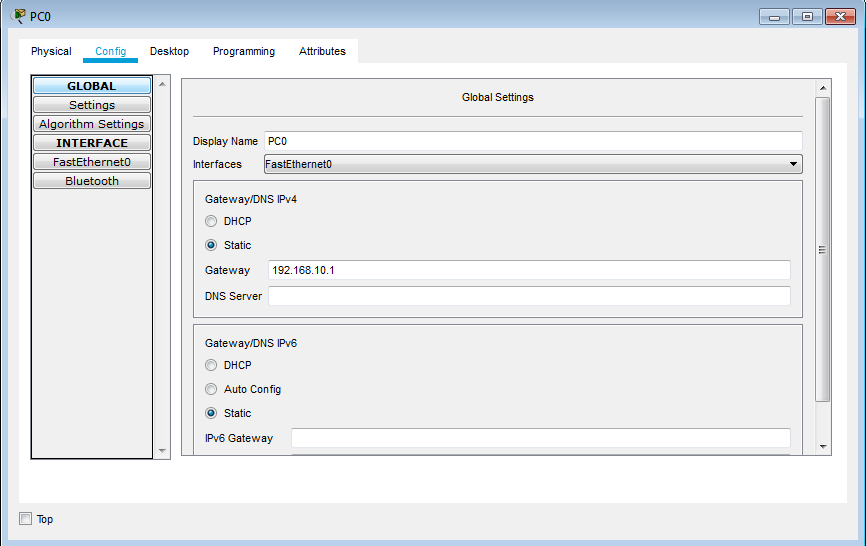
**ii)End User Client Configuration :**

**a)Set IP address to end user client :**

****

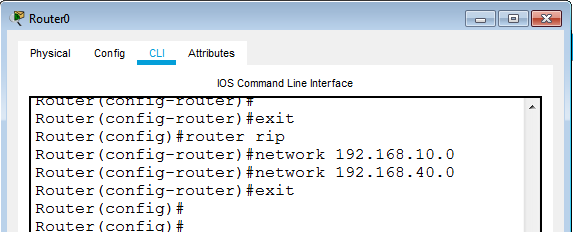
Set IP addresses to all remaining clients as shown above.

**b)Set Gateway to end user client:**

****

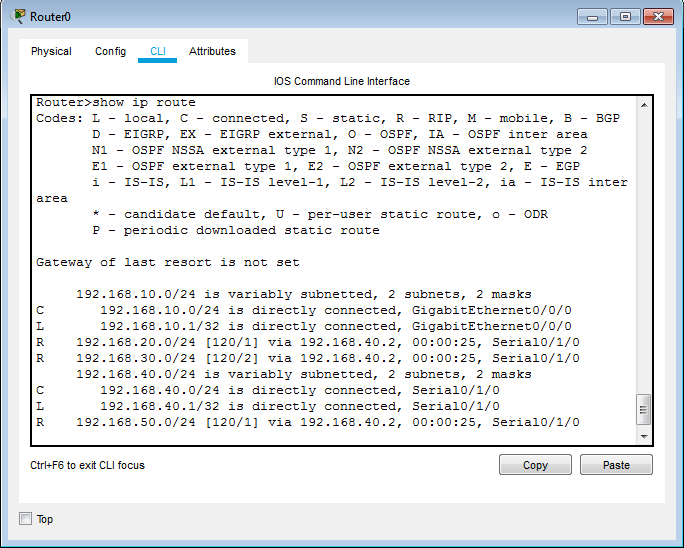
Set Gateways to all remaining end user client.

**Set RIP protocol to routers :**



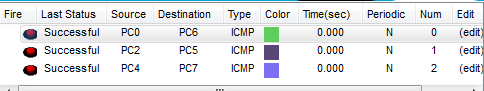
Set RIP protocol to all remaining routers as shown above.

**Go for “show ip route” command for each router:**

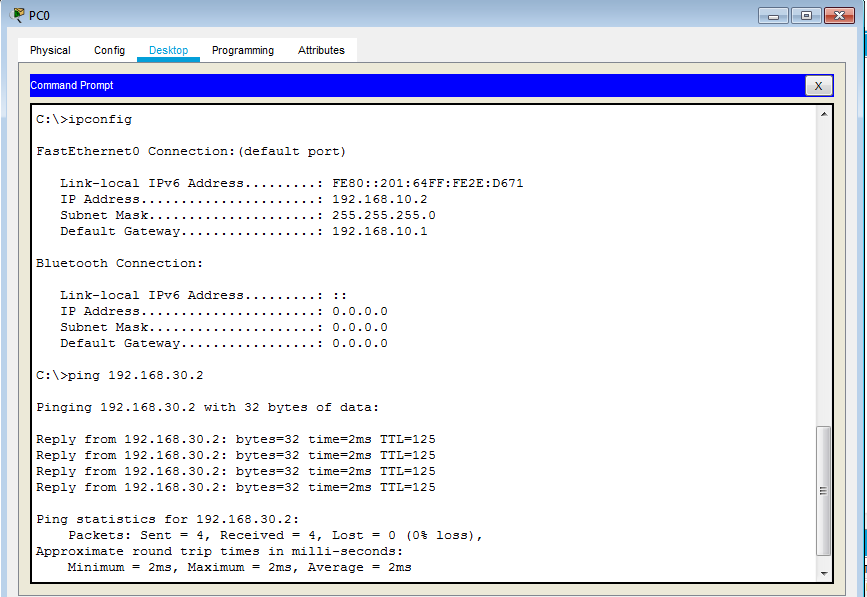


Show “show ip route” command for all remaining routers.

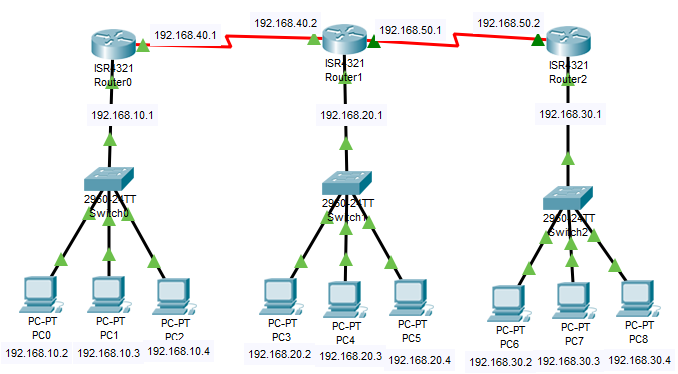
**Show Packet Tracing :**

****

**Show Connection between PC0 to PC6:**

****

**FINAL TOPOLOGY:**

****

**Conclusion:** We have learnt to create RIPv2 with three routers as shown above.

**Practical No. 02**

**AIM:** Create network with three routers with OSPF and each router associated network will have minimum three PC. Show connectivity.

**Steps to perform practical:**

Step 1 : Add 3 routers, 3 switches, end user clients to the switches.

Step 2 : Connect all the devices with straight through cable.

Step 3 : Add serial port devices in the routers in the router.

Step 4 : Give IP to each port of router

enable

configure terminal

interface gig0/0/0

ip address 192.168.10.1 255.255.255.0

no shutdown

exit

Step 5 : Set IP address, Subnet & Gateway for End User Client (manually in input boxes). Note that the gateway will be same as the routers IP.

Step 6 : Apply OSPF to each of the router

configure terminal

router ospf 100

network 192.168.10.0 0.255.255.255 area 0

network 192.168.40.0 0.255.255.255 area 0

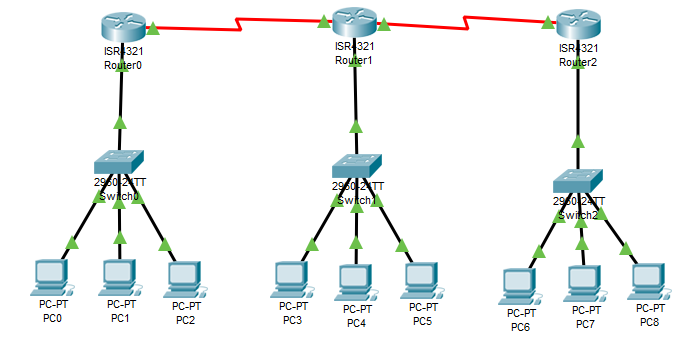
exit

Step 7 : Go for (show ip route) command on each router.

Step 8 : Show connection between PC6 and PC0.

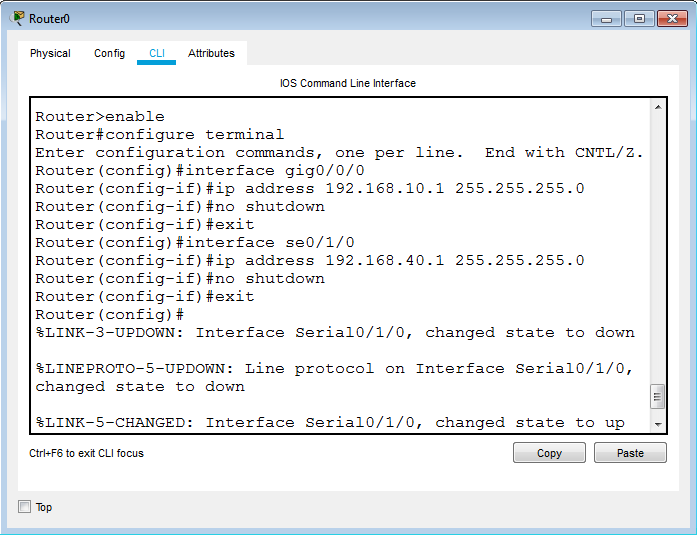
System > Desktop tab > Command Prompt > ping 192.168.10.2

**NETWORK TOPOLOGY:**

****

**i)Router Configuration:-**

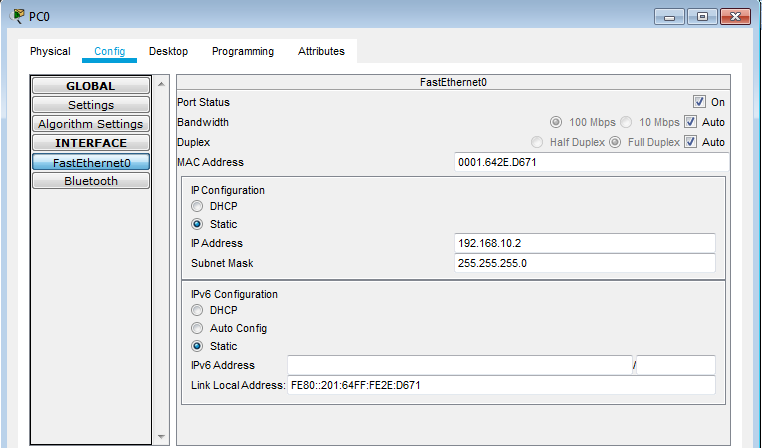
**Set IP addresses to routers :-**

****

Set IP addresses to all remaining routers as shown above.

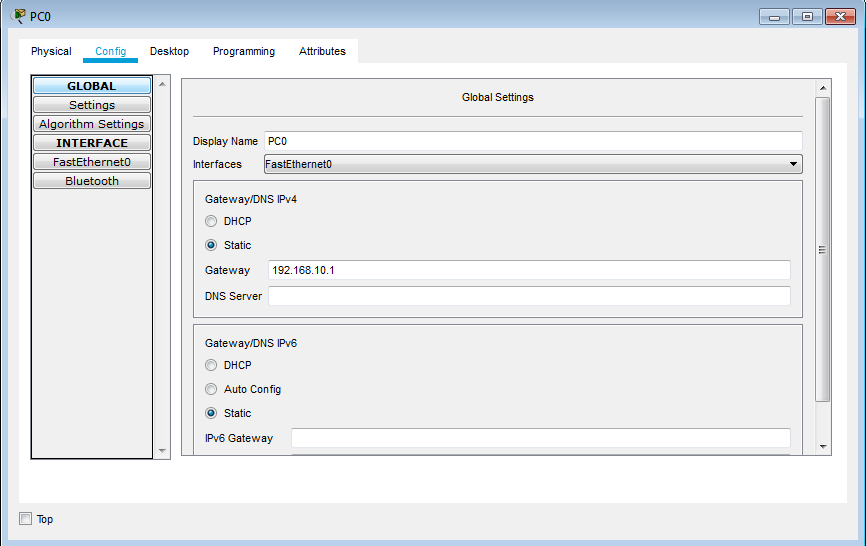
**ii)End User Client Configuration :**

**a)Set IP address to end user client :**

****

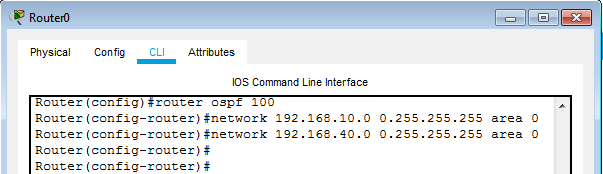
Set IP addresses to all remaining clients as shown above.

**b)Set Gateway to end user client:**

****

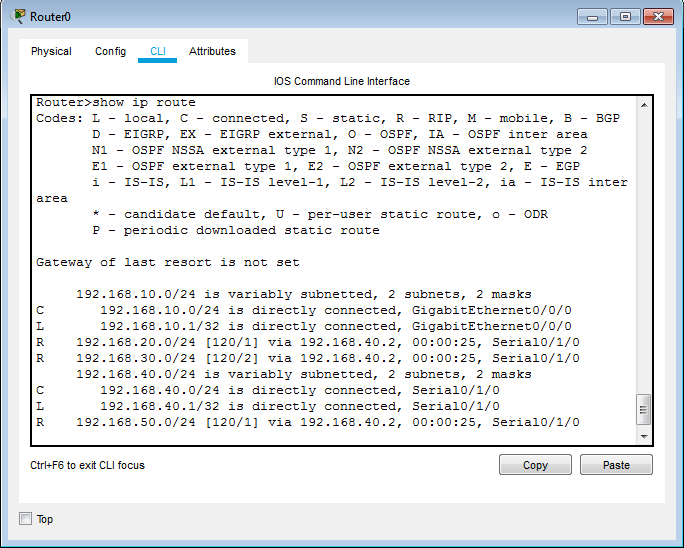
Set Gateways to all remaining end user client.

**Set OSPF protocol to routers :**

****

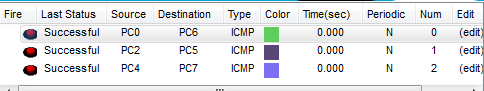
Set OSPF protocol to all remaining routers as shown above.

**Go for “show ip route” command for each router:**

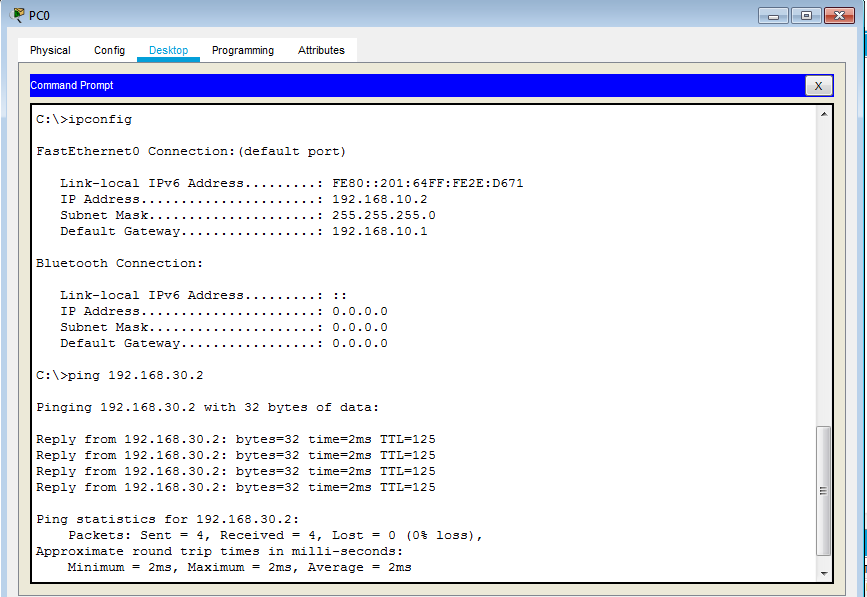


Show “show ip route” command for all remaining routers.

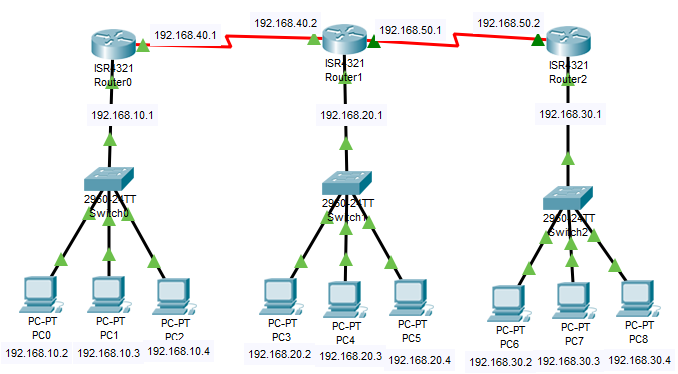
**Show Packet Tracing :**

****

**Show Connection between PC0 to PC6:**

****

**FINAL TOPOLOGY:**

****

**Conclusion:-** We have learnt to create OSPF with three routers as shown above**.**

**Practical No. 03**

**AIM:** Create network with three routers with BGP and each router associated network will have minimum three PC. Show connectivity.

**Steps to perform practical:**

Step 1 : Add 3 routers, 3 switches, end user clients to the switches.

Step 2 : Connect all the devices with straight through cable.

Step 3 : Add serial port devices in the routers in the router.

Step 4 : Give IP to each port of router

enable

configure terminal

interface gig0/0/0

ip address 192.168.10.1 255.255.255.0

no shutdown

exit

Step 5 : Set IP address, Subnet & Gateway for End User Client (manually in input boxes). Note that the gateway will be same as the routers IP.

Step 6 : Apply OSPF to each of the router

configure terminal

router bgp 100

neighbor 192.168.40.2 remote-as 200

network 192.168.10.0

network 192.168.40.0

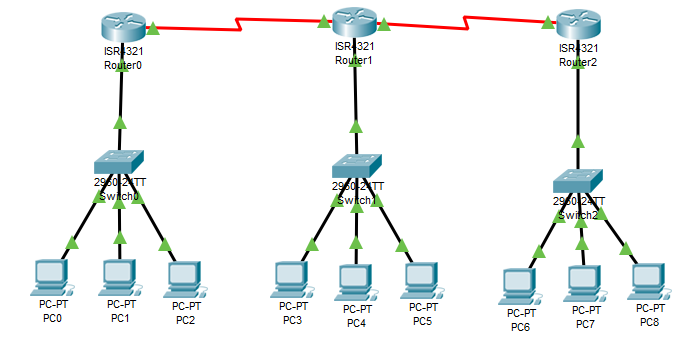
exit

Step 7 : Go for (show ip route) command on each router.

Step 8 : Show connection between PC6 and PC0.

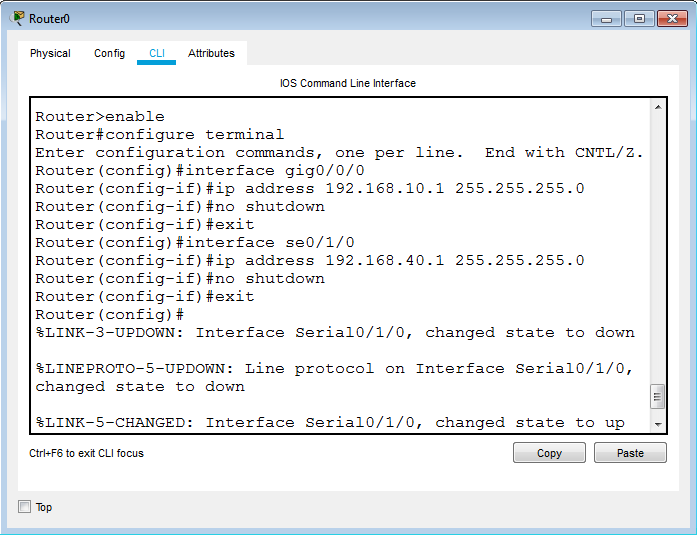
System > Desktop tab > Command Prompt > ping 192.168.10.2

**NETWORK TOPOLOGY:**

****

**i)Router Configuration:-**

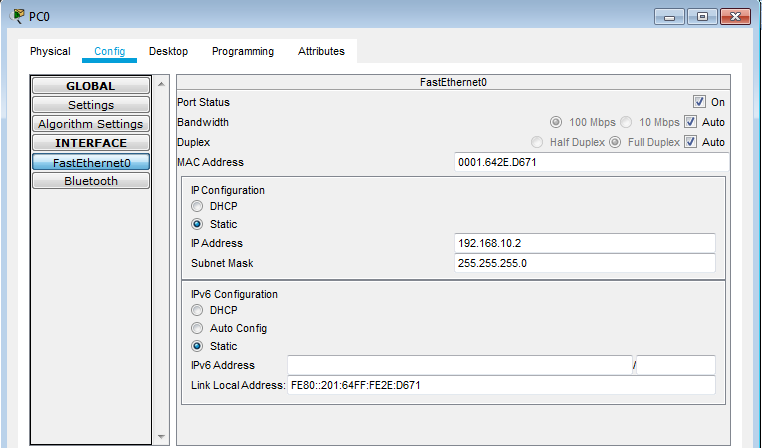
**Set IP addresses to routers :-**

****

Set IP addresses to all remaining routers as shown above.

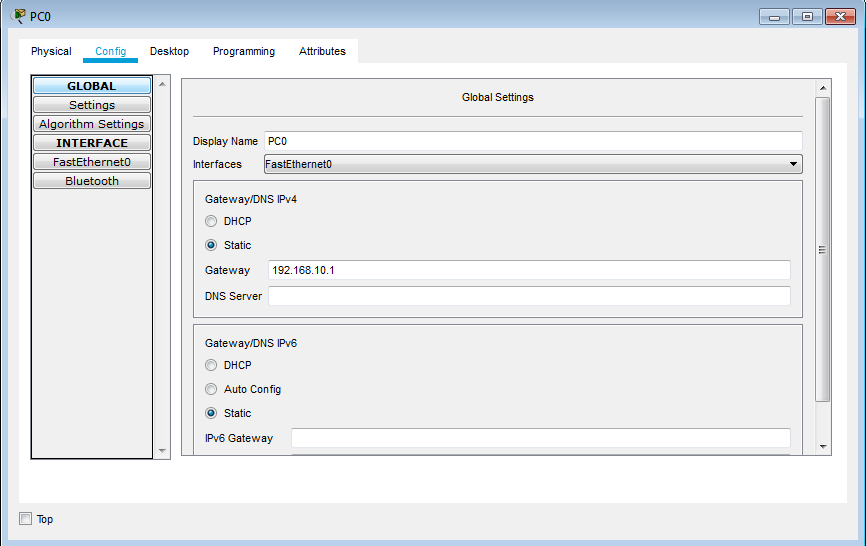
**ii)End User Client Configuration :**

**a)Set IP address to end user client :**

****

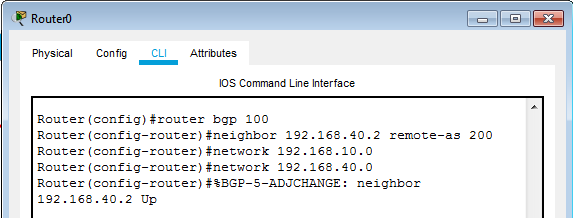
Set IP addresses to all remaining clients as shown above.

**b)Set Gateway to end user client:**

****

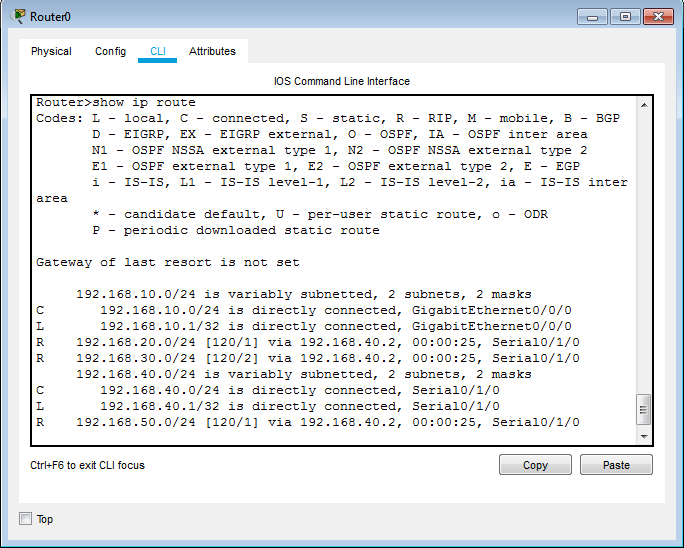
Set Gateways to all remaining end user client.

**Set BGP protocol to routers :**

****

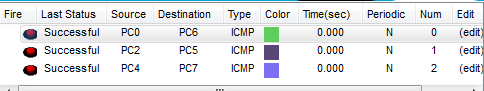
Set BGP protocol to all remaining routers as shown above.

**Go for “show ip route” command for each router:**

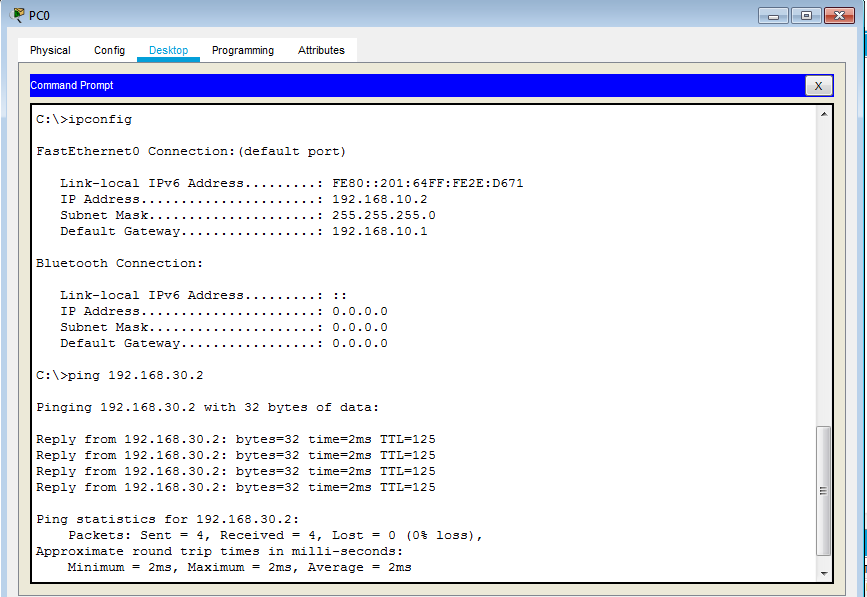


Show “show ip route” command for all remaining routers.

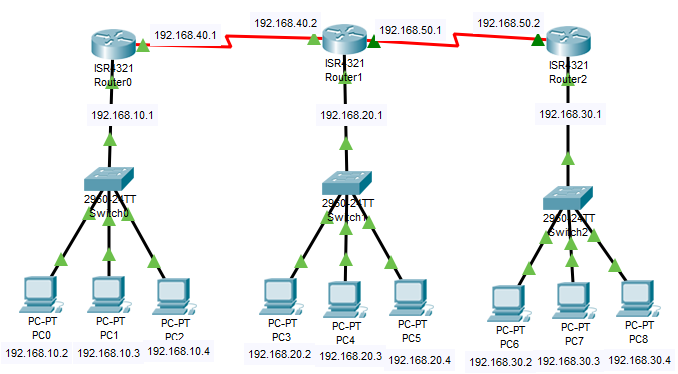
**Show Packet Tracing :**

****

**Show Connection between PC0 to PC6:**

****

**FINAL TOPOLOGY:**

****

**Conclusion:-** We have learnt to create BGP with three routers as shown above.

**Practical No. 04**

**AIM:** Configure DHCP server and client for DHCP service.

**Steps to perform practical:**

Step 1 : Add router, switch and end user clients to the switches.

Step 2 : Connect all the devices with straight through cable.

Step 3 : Give IP to each port of router

enable

configure terminal

interface gig0/0/0

ip address 192.168.10.1 255.255.255.0

no shutdown

exit

Step 4 : Apply DHCP to the router

configure terminal

ip dhcp pool pool\_name

network 192.168.10.0 255.255.255.0

default-router 192.168.10.1

exit

ip dhcp excluded-address 192.168.10.2 192.168.10.10

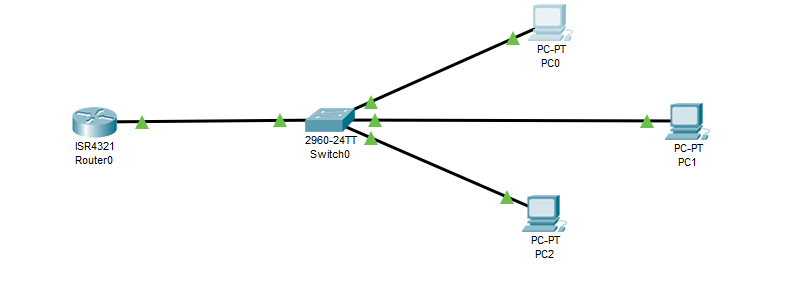
exit

Step 5 : Show DHCP ip address to each end user clients.

System > Desktop tab > ip configuration > select dhcp.

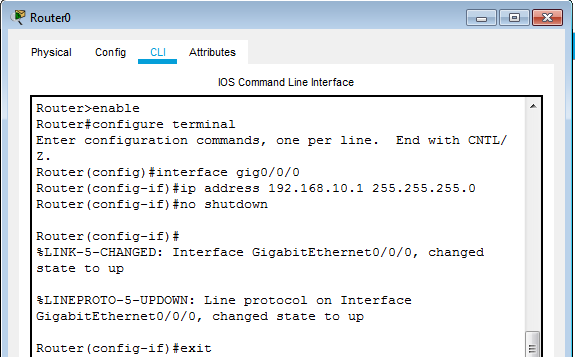
DHCP automatically assign ip address to each client.

**NETWORK TOPOLOGY:**

****

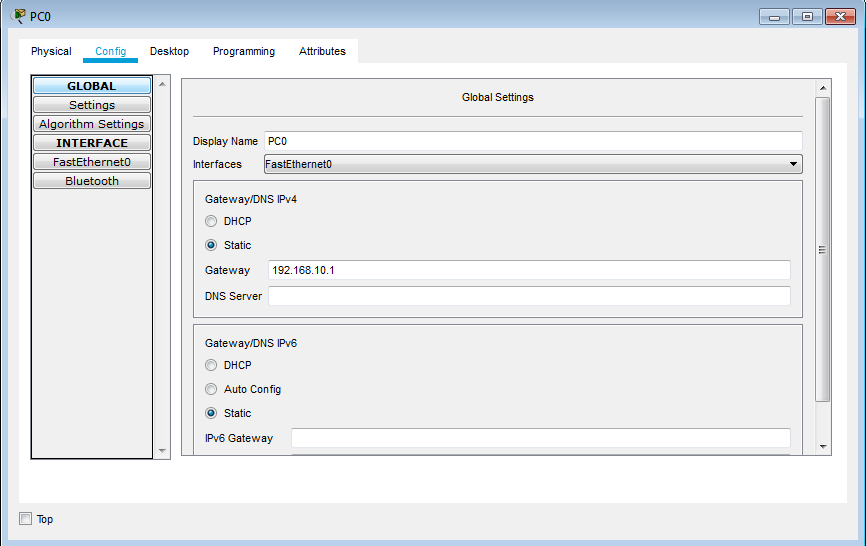
**i)Router Configuration:-**

**Set IP addresses to routers :-**

****

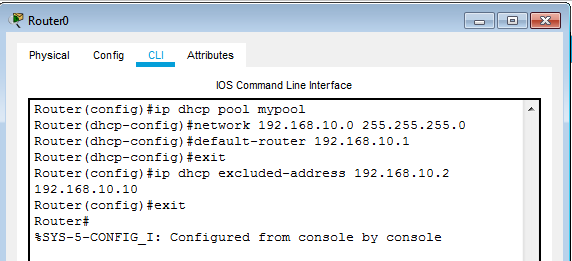
**ii)End User Client Configuration :**

**a)Set Gateway to end user client:**

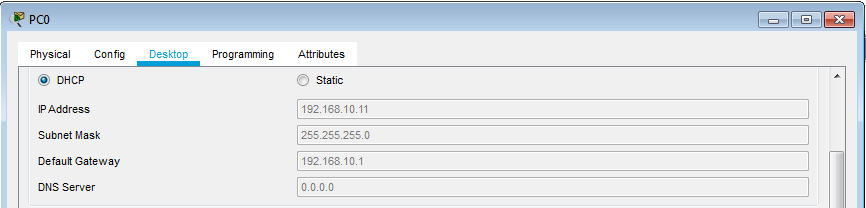
****

Set Gateways to all remaining end user client.

**Set DHCP protocol to routers :**

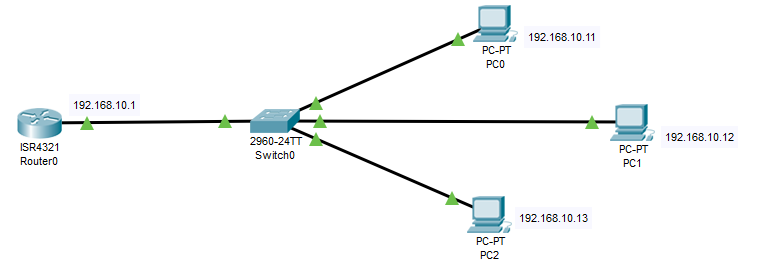


**Show DHCP ip address to each end user clients.**

****

Show DHCP ip address to all remaining end user clients as shown above.

**FINAL TOPOLOGY:**

****

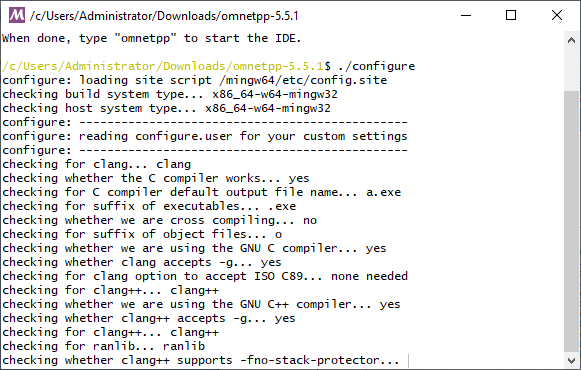
**Conclusion:-** We have learnt to create DHCP server with client service as shown above.

**Practical No. 05**

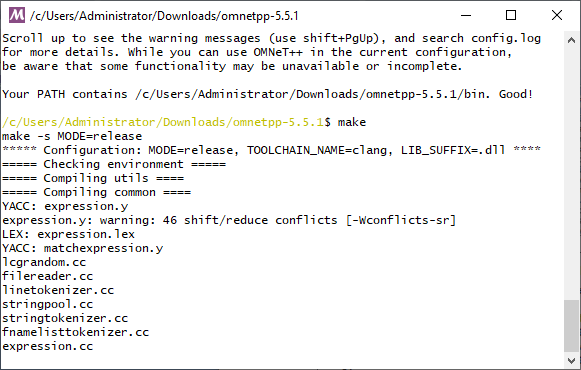
**AIM:** Create simple Adhoc network.

**Installation Steps:**

**Step 1:** Go to omnetpp-5.5.1 folder in which open “mingwenv” file, we get following window. Type “./configure” command.

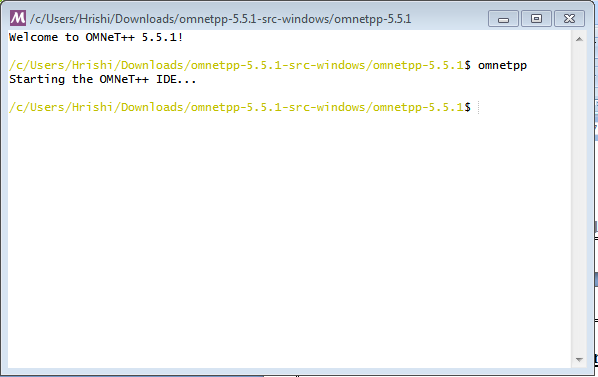
****

**Step 2:** After successfully run above command, type another command “make”.

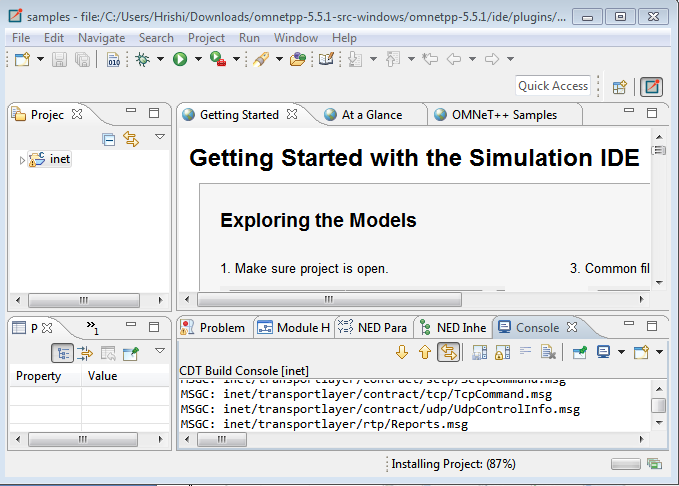


**Practical Steps:**

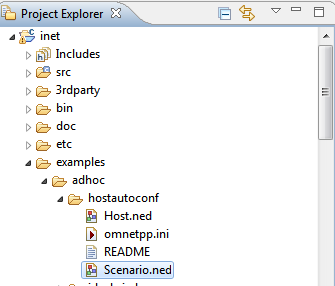
**Step 1:** Go to omnetpp-5.5.1 folder in which open “mingwenv” file, we get following window. Type “omnetpp” command to open omnet++ IDE.



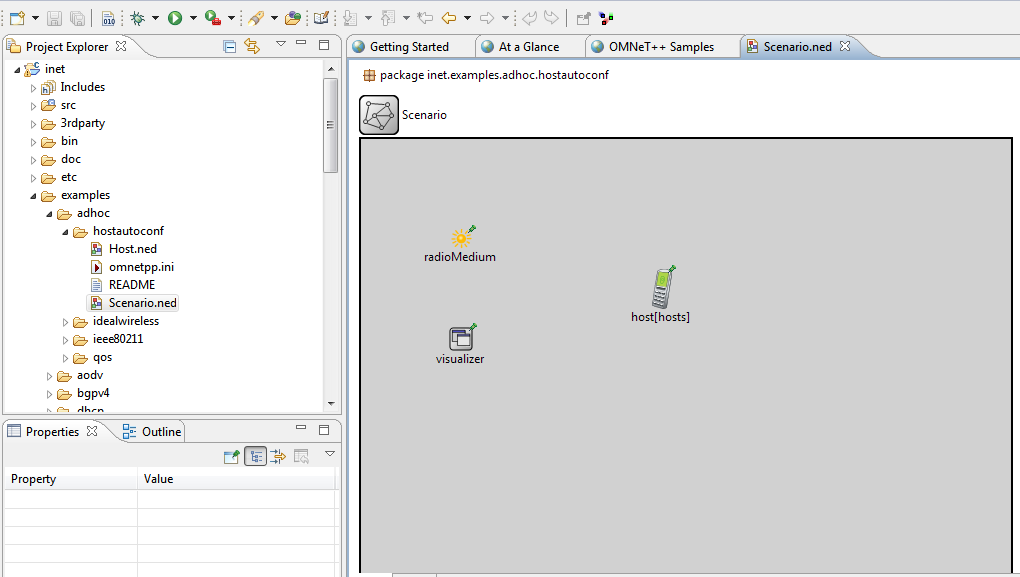
**Step 2:** after that command, following window will open.



**Step 3:** Go to project Explorer > inet > examples > Adhoc > Hostautoconf and open Scenario.ned file.



**Scenario.ned:**



**Coding:**

**Scenario.ned:**

package inet.examples.adhoc.hostautoconf;

import inet.physicallayer.ieee80211.packetlevel.Ieee80211ScalarRadioMedium;

import inet.visualizer.contract.IIntegratedVisualizer;

network Scenario

{

parameters:

int hosts;

@display("bgb=650,450");

submodules:

visualizer: <default("IntegratedCanvasVisualizer")> like IIntegratedVisualizer if hasVisualizer() {

parameters:

@display("p=100,200;is=s");

}

radioMedium: Ieee80211ScalarRadioMedium {

parameters:

@display("p=100,100;is=s");

}

host[hosts]: Host {

@display("p=300,150");

}

}

**Omnetpp.ini:**

[General]

*#debug-on-errors = true*

*#record-eventlog = true*

network = Scenario

sim-time-limit = 60min

cmdenv-express-mode = **true**

\*.hosts = 3

\*\*.constraintAreaMinX = 0m

\*\*.constraintAreaMinY = 0m

\*\*.constraintAreaMinZ = 0m

\*\*.constraintAreaMaxX = 600m

\*\*.constraintAreaMaxY = 400m

\*\*.constraintAreaMaxZ = 0m

*# mobility*

\*\*.host\*.mobility.typename = "MassMobility"

\*\*.host\*.mobility.initFromDisplayString = **false**

\*\*.host\*.mobility.changeInterval = truncnormal(2s, 0.5s)

\*\*.host\*.mobility.angleDelta = normal(0deg, 30deg)

\*\*.host\*.mobility.speed = truncnormal(20mps, 8mps)

\*\*.host\*.mobility.updateInterval = 100ms

\*\*.host\*.ac\_wlan.interfaces = "wlan0"

*# UdpBasicApp / UdpSink*

\*\*.host\*.numApps = 1

\*\*.app[0].typename = "UdpBasicApp"

\*\*.app[0].destAddresses = "host[0]"

\*\*.app[0].localPort = 9001

\*\*.app[0].destPort = 9001

\*\*.app[0].messageLength = 100B

\*\*.app[0].startTime = uniform(10s, 30s)

\*\*.app[0].sendInterval = uniform(10s, 30s)

*# nic settings*

\*\*.wlan[\*].bitrate = 2Mbps

\*\*.wlan[\*].mgmt.frameCapacity = 10

\*\*.wlan[\*].mac.retryLimit = 7

\*\*.wlan[\*].mac.dcf.channelAccess.cwMin = 7

\*\*.wlan[\*].mac.cwMinBroadcast = 31

\*\*.wlan[\*].radio.transmitter.power = 2mW

\*\*.wlan[\*].radio.receiver.sensitivity = -85dBm

\*\*.wlan[\*].radio.receiver.snirThreshold = 4dB

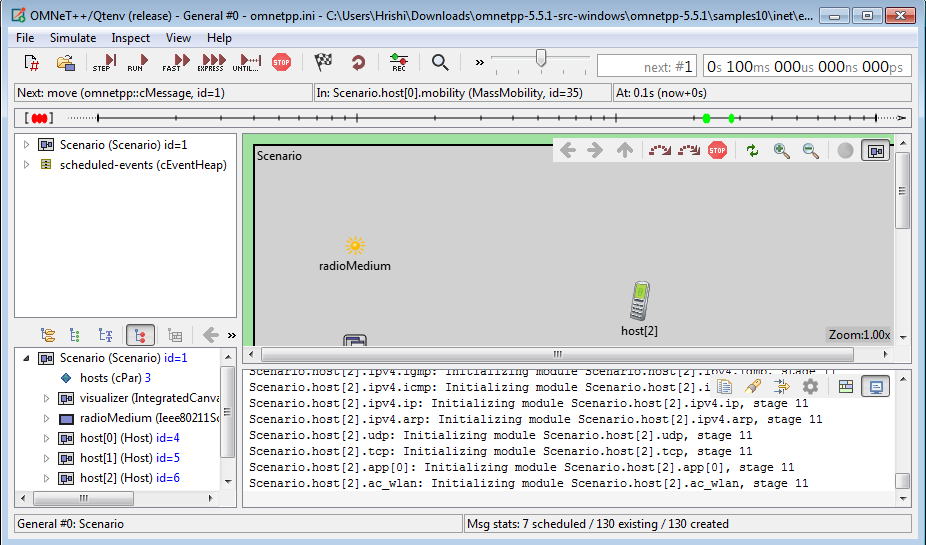
\*\*.udpapp.\*.vector-recording = **true**

\*\*.vector-recording = **true**

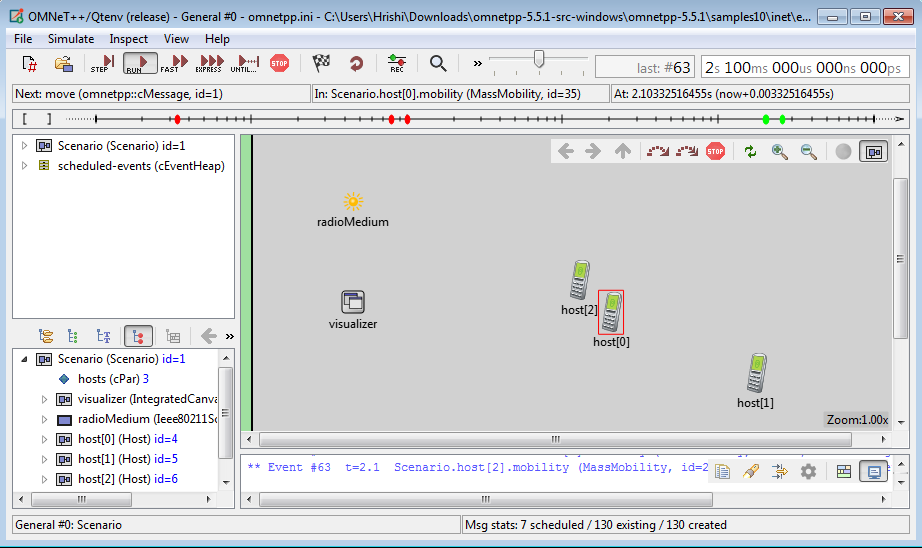
**Step 4:** Click on Run button.



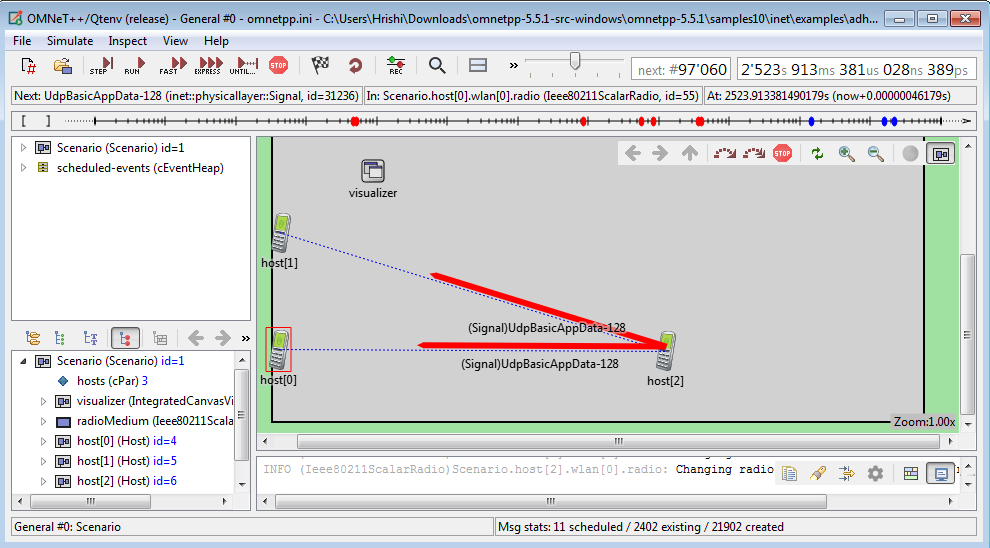
**Step 5:** After that following window will open.

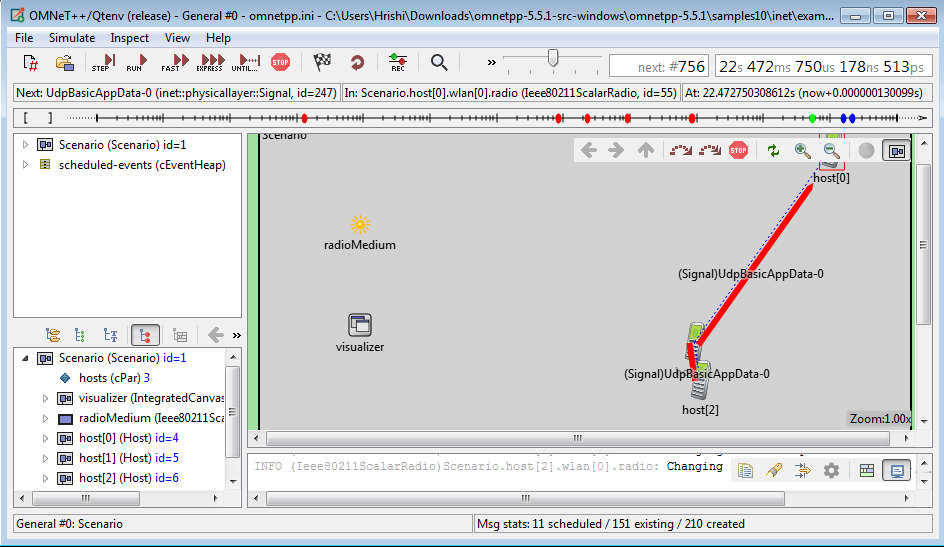
****

**Step 6:**Click on RUN.



**OUTPUT:**

****

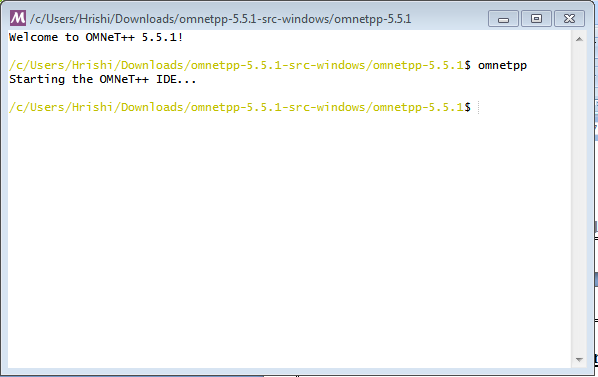
****

**Conclusion:** We have learnt to create a simple ADHOC network.

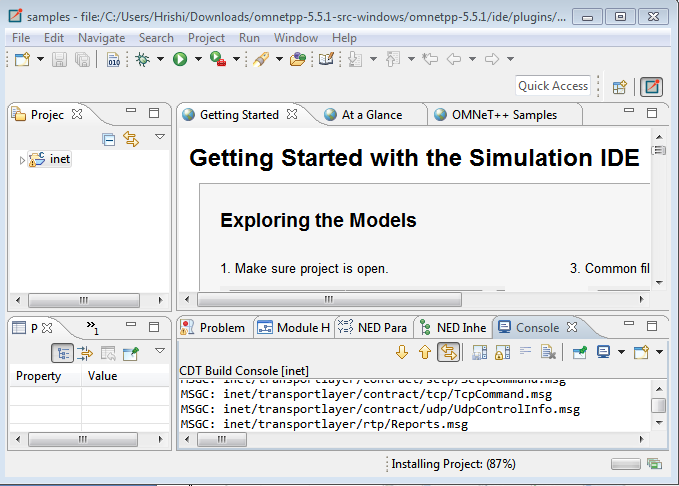
**Practical No. 06**

**AIM:** Create MANET simulation for AODVUU Network.

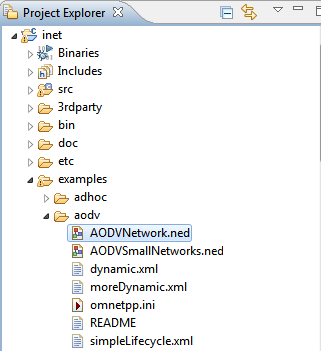
**Step 1:** Go to omnetpp-5.5.1 folder in which open “mingwenv” file, we get following window. Type “omnetpp” command to open omnet++ IDE.



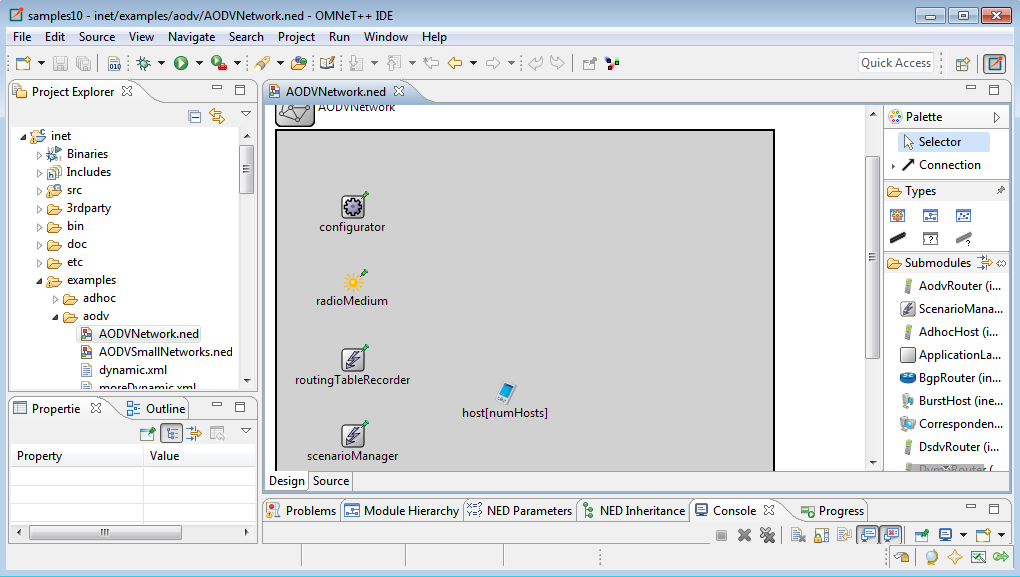
**Step 2:** after that command, following window will open.



**Step 3:** Go to project Explorer > inet > examples > aodv and open AODVNetwork.ned file.



**AODVNetwork.ned:**



**Coding:**

**AODVNetwork.ned:**

package inet.examples.aodv;

import inet.common.scenario.ScenarioManager;

import inet.networklayer.configurator.ipv4.Ipv4NetworkConfigurator;

import inet.networklayer.ipv4.RoutingTableRecorder;

import inet.node.aodv.AodvRouter;

import inet.physicallayer.unitdisk.UnitDiskRadioMedium;

network AODVNetwork

{

parameters:

int numHosts;

@display("bgb=650,650");

submodules:

radioMedium: UnitDiskRadioMedium {

parameters:

@display("p=100,200;is=s");

}

configurator: Ipv4NetworkConfigurator {

parameters:

config = xml("<config><interface hosts='\*' address='145.236.x.x' netmask='255.255.0.0'/></config>");

@display("p=100,100;is=s");

}

routingTableRecorder: RoutingTableRecorder {

parameters:

@display("p=100,300;is=s");

}

scenarioManager: ScenarioManager {

parameters:

script = default(xml("<scenario/>"));

@display("p=100,400;is=s");

}

host[numHosts]: AodvRouter {

parameters:

@display("i=device/pocketpc\_s;r=,,#707070");

}

connections allowunconnected:

}

**Omnetpp.ini:**

[General]

network = AODVNetwork

*#record-eventlog = true*

num-rngs = 3

debug-on-errors = true

\*\*.mobility.rng-0 = 1

\*\*.wlan[\*].mac.rng-0 = 2

*# channel physical parameters*

\*\*.wlan[\*].typename = "AckingWirelessInterface"

\*\*.wlan[\*].bitrate = 2Mbps

\*\*.wlan[\*].mac.headerLength = 20B

\*\*.wlan[\*].radio.typename = "UnitDiskRadio"

\*\*.wlan[\*].radio.transmitter.headerLength = 96b

\*\*.wlan[\*].radio.transmitter.communicationRange = 250m

\*\*.wlan[\*].radio.transmitter.interferenceRange = 0m

\*\*.wlan[\*].radio.transmitter.detectionRange = 0m

\*\*.wlan[\*].radio.receiver.ignoreInterference = true

\*.numHosts = 20

*# mobility*

\*\*.host[\*].mobility.typename = "StationaryMobility"

\*\*.mobility.constraintAreaMinZ = 0m

\*\*.mobility.constraintAreaMaxZ = 0m

\*\*.mobility.constraintAreaMinX = 0m

\*\*.mobility.constraintAreaMinY = 0m

\*\*.mobility.constraintAreaMaxX = 600m

\*\*.mobility.constraintAreaMaxY = 600m

*# ping app (host[0] pinged by others)*

\*.host[0].numApps = 1

\*.host[0].app[0].typename = "PingApp"

\*.host[0].app[0].startTime = uniform(1s,5s)

\*.host[0].app[0].printPing = true

*# nic settings*

\*\*.wlan[\*].bitrate = 2Mbps

\*\*.wlan[\*].mgmt.frameCapacity = 10

\*\*.wlan[\*].mac.retryLimit = 7

*# lifecycle*

\*\*.hasStatus = true

[Config Static]

description = routing without mobility

\*.host[\*].wlan[\*].radio.transmitter.communicationRange = 250m

\*.host[0].app[0].destAddr = "host[1](ipv4)"

[Config IPv4SlowMobility]

description = two fixed communicating nodes with low speed mobile nodes

extends = Static

*# mobility*

\*\*.aodv.activeRouteTimeout = 3s

\*\*.host[2..20].mobility.typename = "MassMobility"

\*\*.host[0].mobility.typename = "StationaryMobility"

\*\*.host[1].mobility.typename = "StationaryMobility"

\*\*.host[\*].mobility.changeInterval = normal(5s, 0.1s)

\*\*.host[\*].mobility.angleDelta = normal(0deg, 30deg)

\*\*.host[\*].mobility.speed = normal(2mps, 0.01mps)

\*\*.host[1].mobility.initialX = 600m

\*\*.host[1].mobility.initialY = 600m

[Config IPv4ModerateFastMobility]

description = two fixed communicating nodes with moderate speed mobile nodes

extends = IPv4SlowMobility

*# mobility*

\*\*.aodv.activeRouteTimeout = 2s

\*\*.host[\*].mobility.speed = normal(8mps, 0.01mps)

[Config IPv4FastMobility]

description = two fixed communicating nodes with high speed mobile nodes

extends = IPv4SlowMobility

*# mobility*

\*\*.aodv.activeRouteTimeout = 1s

\*\*.host[\*].mobility.speed = normal(15mps, 0.01mps)

[Config Dynamic]

description = one node is shut down and restarted trigger route changes

extends = Static

\*.host[\*].hasStatus = true

\*.scenarioManager.script = xmldoc("dynamic.xml")

[Config MoreDynamic]

description = some nodes are shut down trigger route changes

extends = Static

\*.host[\*].hasStatus = true

\*.scenarioManager.script = xmldoc("moreDynamic.xml")

[Config SimpleRREQ]

description = demonstrates a single RREQ-RREP exchange

network = SimpleRREQ

*# nic settings*

\*\*.wlan[\*].radio.transmitter.communicationRange = 240m

\*\*.sender.numApps = 1

\*\*.sender.app[0].typename = "PingApp"

\*\*.sender.app[0].startTime = uniform(1s,5s)

\*\*.sender.app[0].printPing = true

\*\*.sender.app[0].destAddr = "receiver(ipv4)"

[Config SimpleRREQ2]

description = demonstrates a single RREQ-RREP exchange with two intermediate nodes

extends = SimpleRREQ

network = SimpleRREQ2

[Config SimpleLifecycle]

description = demonstrates AODV's RERR mechanism when a node shuts down

extends = SimpleRREQ2

\*.scenarioManager.script = xmldoc("simpleLifecycle.xml")

[Config ShortestPath]

description = demonstrates that AODV chooses the shorter path

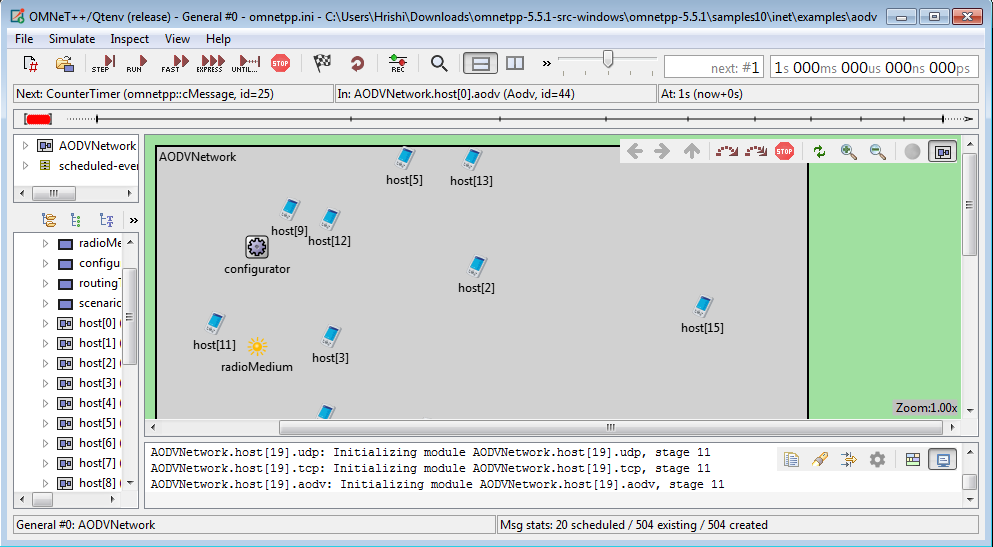
network = ShortestPath

extends = SimpleRREQ

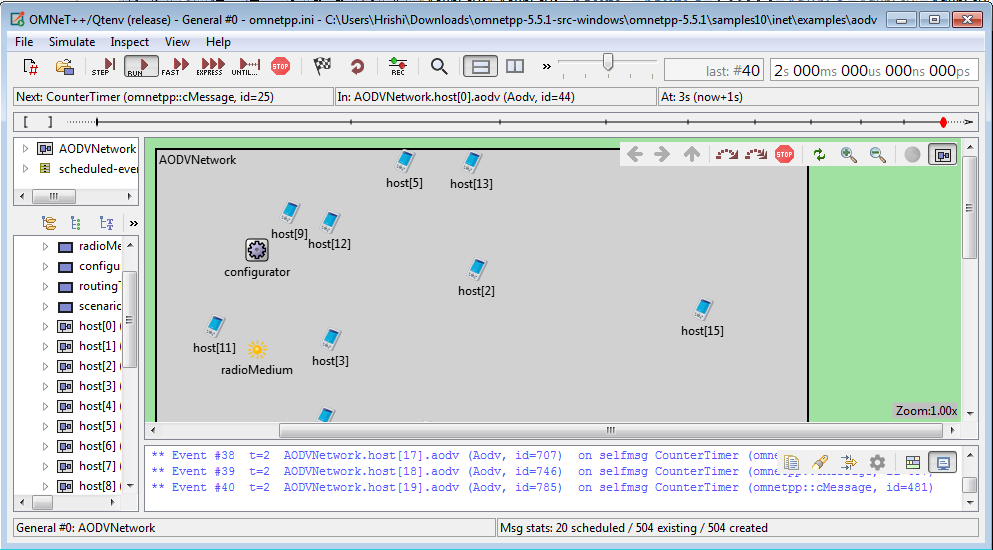
**Step 4:** Click on Run button.



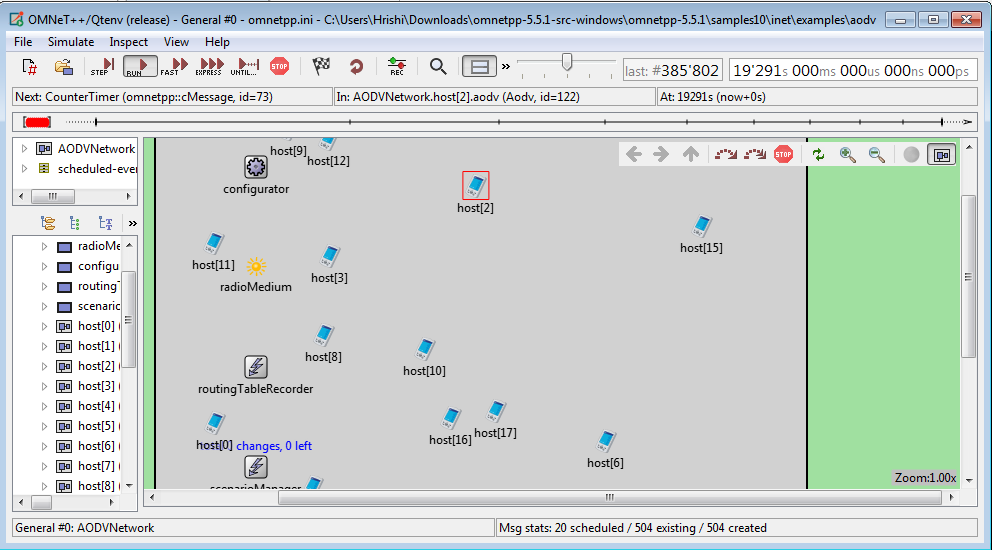
**Step 5:** After that following window will open.

****

**Step 6:**Click on RUN.



**OUTPUT:**

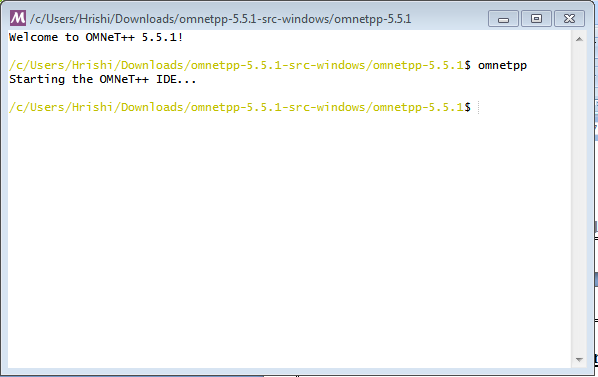
****

**Conclusion:** We have learnt to MANET simulation for AODVUU Network.

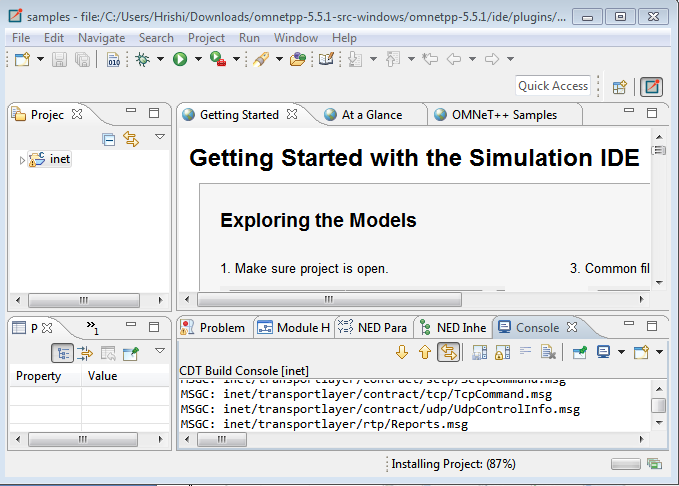
**Practical No. 07**

**AIM:** Create Single mobile network.

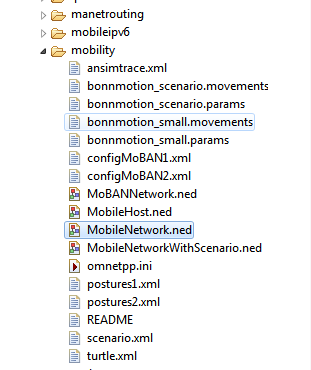
**Step 1:** Go to omnetpp-5.5.1 folder in which open “mingwenv” file, we get following window. Type “omnetpp” command to open omnet++ IDE.



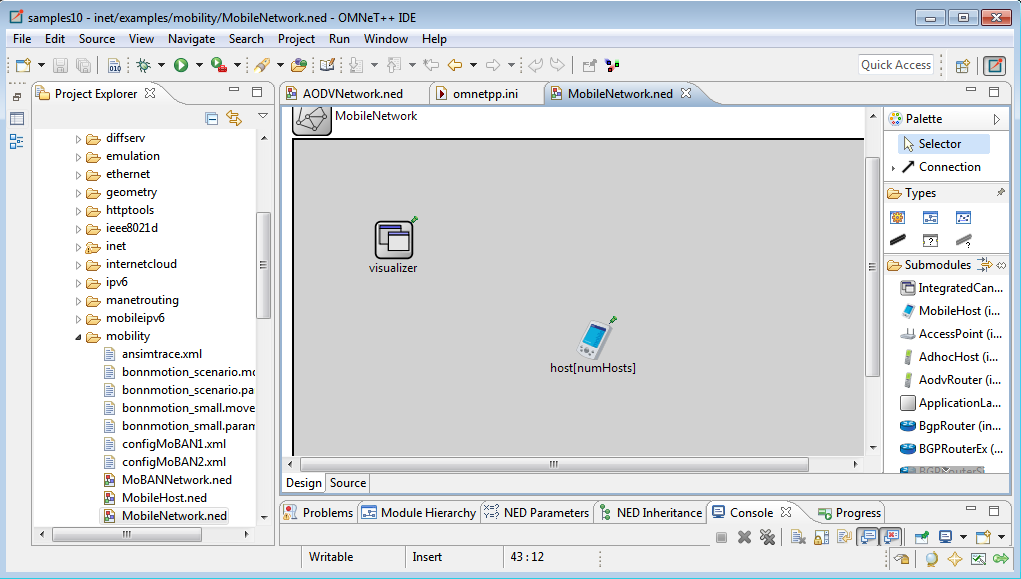
**Step 2:** after that command, following window will open.



**Step 3:** Go to project Explorer > inet > examples > mobility and open MobileNetwork.ned file.



**MobileNetwork.ned:**



**Coding:**

**MobileNetwork.ned:**

package inet.examples.mobility;

import inet.visualizer.integrated.IntegratedCanvasVisualizer;

network MobileNetwork{

parameters:

int numHosts;

bool hasVisualizer = default(false);

@display("bgb=600,400");

submodules:

visualizer: IntegratedCanvasVisualizer if hasVisualizer {

parameters:

@display("p=100,100");

}

host[numHosts]: MobileHost {

parameters:

@display("p=300,200;r=,,#707070");

}

}

**Omnetpp.ini:**

[General]

*#scheduler-class = "inet::cRealTimeScheduler" #so that speed appears realistic*

*#debug-on-errors = true*

sim-time-limit = 10day

\*.numHosts = 2

\*\*.constraintAreaMinX = 0m

\*\*.constraintAreaMinY = 0m

\*\*.constraintAreaMinZ = 0m

\*\*.constraintAreaMaxX = 600m

\*\*.constraintAreaMaxY = 400m

\*\*.constraintAreaMaxZ = 0m

\*\*.updateInterval = 0.1s *# test with 0s too, and let getCurrentPosition update the display string from a test module*

\*\*.mobility.initFromDisplayString = false

[Config AnsimMobility]

network = MobileNetwork

\*\*.host\*.mobility.typename = "AnsimMobility"

\*\*.host\*.mobility.ansimTrace = xmldoc("ansimtrace.xml")

\*\*.host\*.mobility.nodeId = -1 *#means "host module's index"*

[Config BonnMotionMobility1]

network = MobileNetwork

description = "2 hosts"

\*\*.host\*.mobility.typename = "BonnMotionMobility"

\*\*.host\*.mobility.traceFile = "bonnmotion\_small.movements"

\*\*.host\*.mobility.nodeId = -1 *#means "host module's index"*

[Config BonnMotionMobility2]

network = MobileNetwork

description = "100 hosts"

\*.numHosts = 100

\*\*.host\*.mobility.typename = "BonnMotionMobility"

\*\*.host\*.mobility.traceFile = "bonnmotion\_scenario.movements"

\*\*.host\*.mobility.nodeId = -1 *#means "host module's index"*

[Config ChiangMobility]

network = MobileNetwork

\*.numHosts = 1

\*\*.host\*.mobility.typename = "ChiangMobility"

\*\*.host\*.mobility.stateTransitionUpdateInterval = 3s

\*\*.host\*.mobility.speed = 10mps

[Config CircleMobility1]

network = MobileNetwork

\*.numHosts = 3

\*\*.host\*.mobility.typename = "CircleMobility"

\*\*.host\*.mobility.cx = 200m

\*\*.host\*.mobility.cy = 200m

\*\*.host\*.mobility.r = 150m

\*\*.host\*.mobility.speed = 40mps

\*\*.host[0].mobility.startAngle = 0deg

\*\*.host[1].mobility.startAngle = 120deg

\*\*.host[2].mobility.startAngle = 240deg

[Config CircleMobility2]

network = MobileNetwork

\*.numHosts = 3

\*\*.host\*.mobility.typename = "CircleMobility"

\*\*.host[0].mobility.cx = 100m

\*\*.host[1].mobility.cx = 300m

\*\*.host[2].mobility.cx = 500m

\*\*.host\*.mobility.cy = 200m

\*\*.host\*.mobility.r = 150m

\*\*.host\*.mobility.speed = 40mps

\*\*.host\*.mobility.startAngle = 0deg

[Config GaussMarkovMobility]

network = MobileNetwork

\*.numHosts = 1

\*\*.host\*.mobility.typename = "GaussMarkovMobility"

\*\*.host\*.mobility.alpha = 0.9

\*\*.host\*.mobility.speed = 10mps

\*\*.host\*.mobility.angle = 0deg

\*\*.host\*.mobility.variance = 40

\*\*.host\*.mobility.margin = 30m

[Config LinearMobility]

network = MobileNetwork

\*\*.host\*.mobility.typename = "LinearMobility"

\*\*.host\*.mobility.initFromDisplayString = false

\*\*.host\*.mobility.speed = 50mps

\*\*.host\*.mobility.angle = 30deg *# degrees*

[Config LinearMobility01]

extends = LinearMobility

\*\*.updateInterval = 0.1s

[Config LinearMobility1]

extends = LinearMobility

\*\*.updateInterval = 1s

[Config LinearMobility10]

extends = LinearMobility

\*\*.updateInterval = 10s

[Config LinearMobility100]

extends = LinearMobility

\*\*.updateInterval = 100s

[Config LinearMobility1000]

extends = LinearMobility

\*\*.updateInterval = 1000s

[Config MassMobility]

network = MobileNetwork

\*.numHosts = 5

\*\*.host\*.mobility.typename = "MassMobility"

\*\*.host\*.mobility.initFromDisplayString = false

\*\*.host\*.mobility.changeInterval = truncnormal(2s, 0.5s)

\*\*.host\*.mobility.angleDelta = normal(0deg, 30deg)

\*\*.host\*.mobility.speed = truncnormal(15mps, 5mps)

[Config MassMobilityWithScenario]

network = MobileNetworkWithScenario

\*.numHosts = 5

\*\*.host\*.mobility.typename = "MassMobility"

\*\*.host\*.mobility.initFromDisplayString = false

\*\*.host\*.mobility.changeInterval = truncnormal(2s, 0.5s)

\*\*.host\*.mobility.angleDelta = normal(0deg, 30deg)

\*\*.host\*.mobility.speed = truncnormal(15mps, 5mps)

\*\*.scenarioManager.script = xmldoc("scenario.xml")

[Config MoBANMobility1]

network = MoBANNetwork

\*\*.constraintAreaMaxX = 1000m

\*\*.constraintAreaMaxY = 1000m

\*\*.constraintAreaMaxZ = 1000m

\*\*.numNodes = 12

\*\*.numMoBAN = 1

\*\*.coordinator[\*].postureSpecFile = xmldoc("postures1.xml")

\*\*.coordinator[\*].configFile = xmldoc("configMoBAN1.xml")

\*\*.coordinator[\*].useMobilityPattern = false

\*\*.coordinator[0].mobilityPatternFile = "MoBAN\_Pattern\_in0.txt"

\*\*.node[\*].mobility.typename = "MoBanLocal"

\*\*.node[\*].mobility.coordinatorIndex = 0

[Config MoBANMobility2]

network = MoBANNetwork

\*\*.constraintAreaMaxX = 1000m

\*\*.constraintAreaMaxY = 1000m

\*\*.constraintAreaMaxZ = 1000m

\*\*.numNodes = 24

\*\*.numMoBAN = 2

\*\*.coordinator[\*].postureSpecFile = xmldoc("postures1.xml")

\*\*.coordinator[\*].configFile = xmldoc("configMoBAN2.xml")

\*\*.coordinator[\*].useMobilityPattern = false

\*\*.coordinator[\*].mobilityPatternFile = ""

\*\*.node[\*].mobility.typename = "MoBanLocal"

\*\*.node[0..11].mobility.coordinatorIndex = 0

\*\*.node[12..23].mobility.coordinatorIndex = 1

[Config RandomWaypointMobility1]

description = "zero waitTime"

network = MobileNetwork

\*.numHosts = 5

\*\*.host\*.mobility.typename = "RandomWaypointMobility"

\*\*.host\*.mobility.initFromDisplayString = false

\*\*.host[0].mobility.speed = 10\*uniform(20mps,50mps)

\*\*.host\*.mobility.speed = uniform(20mps,50mps)

[Config RandomWaypointMobility2]

description = "nonzero waitTime"

extends = RandomWaypointMobility1

\*\*.host\*.mobility.waitTime = uniform(3s,8s)

[Config RectangleMobility]

network = MobileNetwork

\*\*.host\*.mobility.typename = "RectangleMobility"

\*\*.host\*.mobility.constraintAreaMinX = 100m

\*\*.host\*.mobility.constraintAreaMinY = 100m

\*\*.host\*.mobility.constraintAreaMaxX = 500m

\*\*.host\*.mobility.constraintAreaMaxY = 300m

*#\*\*.host\*.mobility.x1 = 100*

*#\*\*.host\*.mobility.y1 = 100*

*#\*\*.host\*.mobility.x2 = 500*

*#\*\*.host\*.mobility.y2 = 300*

\*\*.host[0].mobility.startPos = 0

\*\*.host[1].mobility.startPos = 2.5

\*\*.host[0].mobility.speed = 20mps

\*\*.host[1].mobility.speed = -10mps

[Config StaticGridMobility]

network = MobileNetwork

\*.numHosts = 20

\*\*.host\*.mobility.typename = "StaticGridMobility"

\*\*.host\*.mobility.marginX = 100m

\*\*.host\*.mobility.marginY = 100m

\*\*.host\*.mobility.numHosts = 20

[Config StationaryMobility]

network = MobileNetwork

\*.numHosts = 3

\*\*.host\*.mobility.typename = "StationaryMobility"

*# place it at a fixed position:*

\*\*.host[0].mobility.initialX = 50m

\*\*.host[0].mobility.initialY = 200m

\*\*.host[0].mobility.initFromDisplayString = false

*# the second node is using the display string position (or placed randomly if position is not present in display string)*

\*\*.host[1].mobility.initFromDisplayString = true

*# place it at a random position:*

\*\*.host[2].mobility.initFromDisplayString = false

[Config TractorMobility]

network = MobileNetwork

\*.numHosts = 1

\*\*.host\*.mobility.typename = "TractorMobility"

\*\*.host\*.mobility.x1 = 100m

\*\*.host\*.mobility.y1 = 100m

\*\*.host\*.mobility.x2 = 500m

\*\*.host\*.mobility.y2 = 300m

\*\*.host\*.mobility.rowCount = 4

\*\*.host\*.mobility.speed = 50mps

[Config TurtleMobility1]

network = MobileNetwork

description = "square"

\*.numHosts = 1

\*\*.host\*.mobility.typename = "TurtleMobility"

\*\*.host\*.mobility.turtleScript = xmldoc("turtle.xml", "movements//movement[@id='1']")

[Config TurtleMobility2]

network = MobileNetwork

description = "two squares"

\*.numHosts = 1

\*\*.host\*.mobility.typename = "TurtleMobility"

\*\*.host\*.mobility.turtleScript = xmldoc("turtle.xml", "movements//movement[@id='2']")

[Config TurtleMobility3]

network = MobileNetwork

description = "random waypoint"

\*.numHosts = 2

\*\*.host\*.mobility.typename = "TurtleMobility"

\*\*.host\*.mobility.turtleScript = xmldoc("turtle.xml", "movements//movement[@id='3']")

[Config TurtleMobility4]

network = MobileNetwork

description = "mass+reflect"

\*.numHosts = 2

\*\*.host\*.mobility.typename = "TurtleMobility"

\*\*.host\*.mobility.turtleScript = xmldoc("turtle.xml", "movements//movement[@id='4']")

[Config TurtleMobility5]

network = MobileNetwork

description = "mass+wrap"

\*.numHosts = 2

\*\*.host\*.mobility.typename = "TurtleMobility"

\*\*.host\*.mobility.turtleScript = xmldoc("turtle.xml", "movements//movement[@id='5']")

[Config TurtleMobility6]

network = MobileNetwork

description = "mass+placerandomly"

\*.numHosts = 2

\*\*.host\*.mobility.typename = "TurtleMobility"

\*\*.host\*.mobility.turtleScript = xmldoc("turtle.xml", "movements//movement[@id='6']")

[Config AttachedMobility]

network = MobileNetwork

description = "attached"

\*.numHosts = 3

*# mobility visualizer shows velocity and orientation*

\*.hasVisualizer = true

\*.visualizer.mobilityVisualizer.moduleFilter = "\*\*.mobility" *# filter for host mobilities, ignore mobility superposition elements*

\*.visualizer.mobilityVisualizer.displayMovementTrails = true

\*.visualizer.mobilityVisualizer.displayOrientations = true

\*.visualizer.mobilityVisualizer.displayVelocities = true

*# other hosts are also moving around in a larger circle following host[0]*

\*\*.host[0].mobility.typename = "CircleMobility"

\*\*.host[0].mobility.cx = 300m

\*\*.host[0].mobility.cy = 200m

\*\*.host[0].mobility.r = 150m

\*\*.host[0].mobility.speed = 40mps

*# other hosts are also moving around in a larger circle following host[0]*

\*\*.host[\*].mobility.typename = "AttachedMobility"

\*\*.host[\*].mobility.mobilityModule = "^.^.host[0].mobility"

\*\*.host[1].mobility.offsetX = 50m

\*\*.host[1].mobility.offsetHeading = 90deg

\*\*.host[2].mobility.offsetX = -50m

\*\*.host[2].mobility.offsetHeading = -90deg

[Config SuperpositioningMobility]

network = MobileNetwork

description = "superpositioning"

\*.numHosts = 8

*# mobility visualizer shows velocity and orientation*

\*.hasVisualizer = true

\*.visualizer.mobilityVisualizer.moduleFilter = "\*\*.mobility" *# filter for host mobilities, ignore mobility superposition elements*

\*.visualizer.mobilityVisualizer.displayMovementTrails = true

\*.visualizer.mobilityVisualizer.displayOrientations = true

\*.visualizer.mobilityVisualizer.displayVelocities = true

*# last host stays in the center of the scene*

\*\*.host[7].mobility.typename = "StationaryMobility"

\*\*.host[7].mobility.initialX = 300m

\*\*.host[7].mobility.initialY = 200m

\*\*.host[7].mobility.initialZ = 0m

*# other hosts move around the scene using mobility superpositon*

\*\*.host[\*].mobility.typename = "SuperpositioningMobility"

\*\*.host[\*].mobility.numElements = 4

*# other hosts are intiallly positioned in static concentric circles*

\*\*.host[\*].mobility.element[0].typename = "StaticConcentricMobility"

\*\*.host[\*].mobility.element[0].subjectModule = "^.^"

\*\*.host[\*].mobility.element[0].numHosts = 7

*# other hosts are also moving around in a larger circle following host[0]*

\*\*.host[0].mobility.element[1].typename = "CircleMobility"

\*\*.host[0].mobility.element[1].faceForward = false

\*\*.host[0].mobility.element[1].cx = 300m

\*\*.host[0].mobility.element[1].cy = 200m

\*\*.host[0].mobility.element[1].r = 150m

\*\*.host[0].mobility.element[1].speed = 40mps

\*\*.host[\*].mobility.element[1].typename = "AttachedMobility" *# other hosts follow the movement of the 2nd element of host[0]'s mobility superposition*

\*\*.host[\*].mobility.element[1].mobilityModule = "^.^.^.host[0].mobility.element[1]"

*# other hosts are also slightly moving randomly around their position in the group*

\*\*.host[\*].mobility.element[2].typename = "MassMobility"

\*\*.host[\*].mobility.element[2].faceForward = false

\*\*.host[\*].mobility.element[2].initFromDisplayString = false

\*\*.host[\*].mobility.element[2].initialX = 0m

\*\*.host[\*].mobility.element[2].initialY = 0m

\*\*.host[\*].mobility.element[2].initialZ = 0m

\*\*.host[\*].mobility.element[2].constraintAreaMinX = 0m

\*\*.host[\*].mobility.element[2].constraintAreaMinY = 0m

\*\*.host[\*].mobility.element[2].constraintAreaMaxX = 50m *# limiting the random movement*

\*\*.host[\*].mobility.element[2].constraintAreaMaxY = 50m *# limiting the random movement*

\*\*.host[\*].mobility.element[2].changeInterval = truncnormal(2s, 1s)

\*\*.host[\*].mobility.element[2].angleDelta = normal(0deg, 30deg)

\*\*.host[\*].mobility.element[2].speed = truncnormal(10mps, 3mps)

*# other hosts are facing towards the last host sitting in the center*

\*\*.host[\*].mobility.element[3].typename = "FacingMobility"

\*\*.host[\*].mobility.element[3].initFromDisplayString = false

\*\*.host[\*].mobility.element[3].initialX = 0m

\*\*.host[\*].mobility.element[3].initialY = 0m

\*\*.host[\*].mobility.element[3].initialZ = 0m

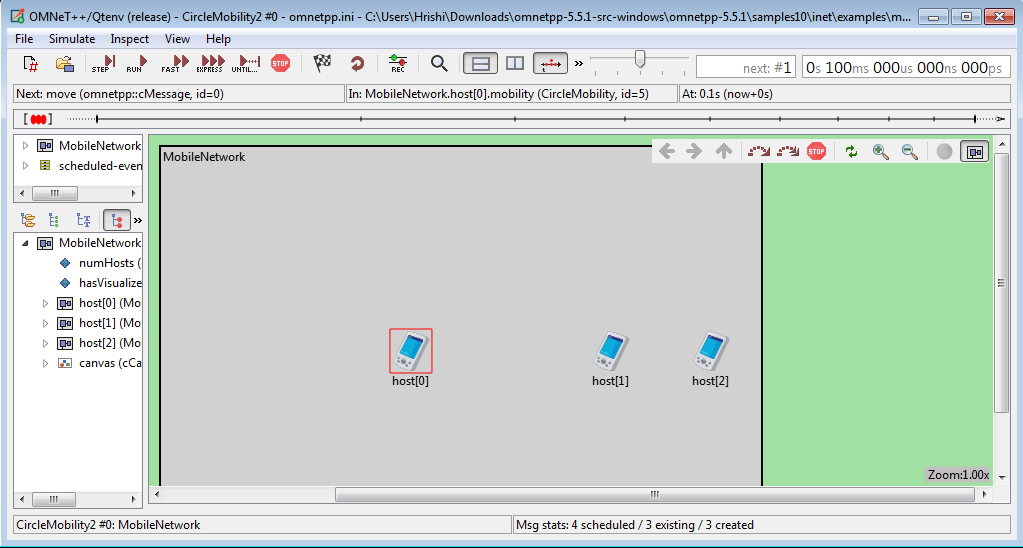
\*\*.host[\*].mobility.element[3].sourceMobility = "^" *# the superposition is the source*

\*\*.host[\*].mobility.element[3].targetMobility = "^.^.^.host[7].mobility" *# last host is the target*

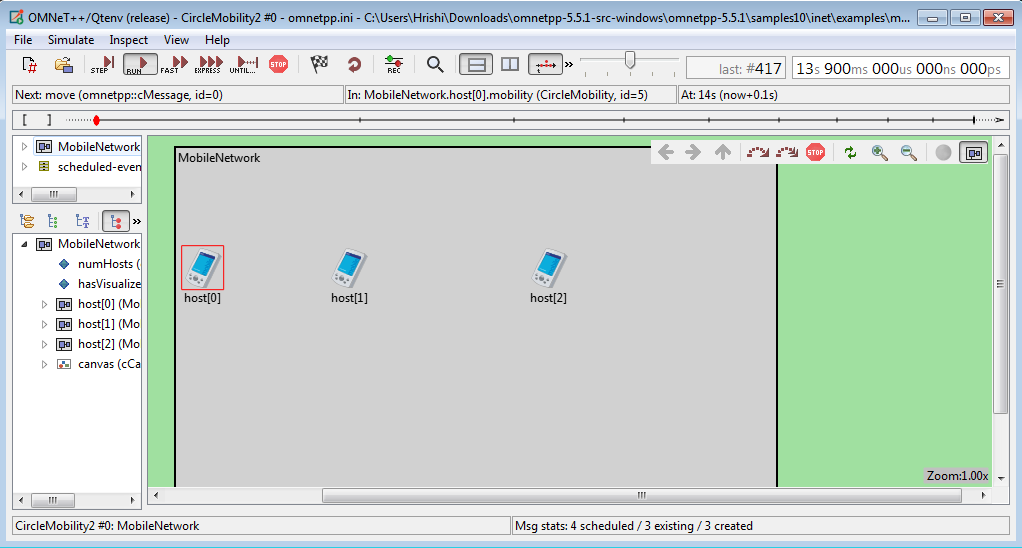
**Step 4:** Click on Run button.



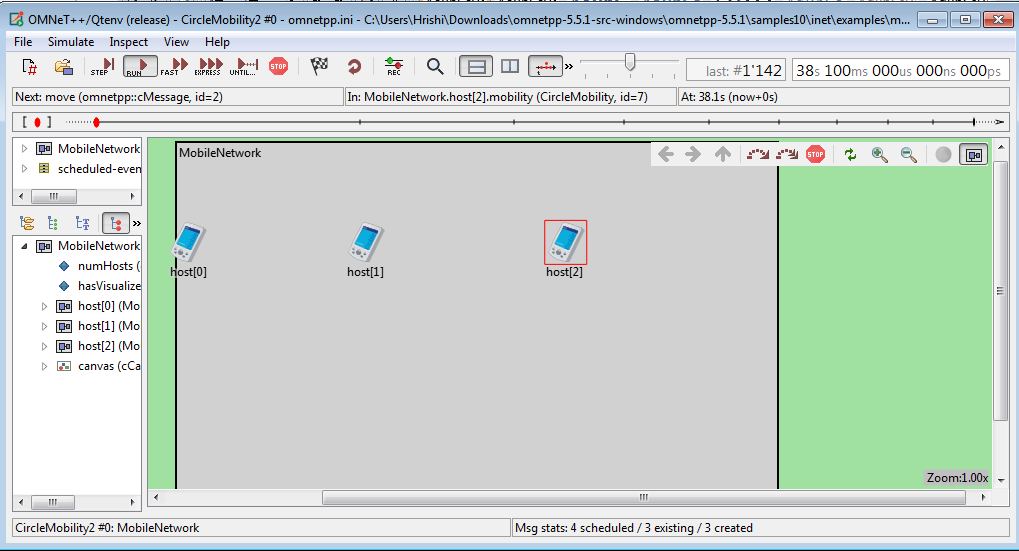
**Step 5:** After that following window will open.

****

**Step 6:** Click on RUN.



**OUTPUT:**

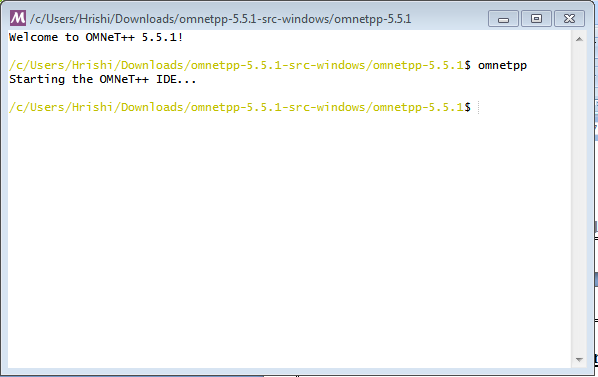
****

**Conclusion:** We have learnt to create a simple Single mobile network.

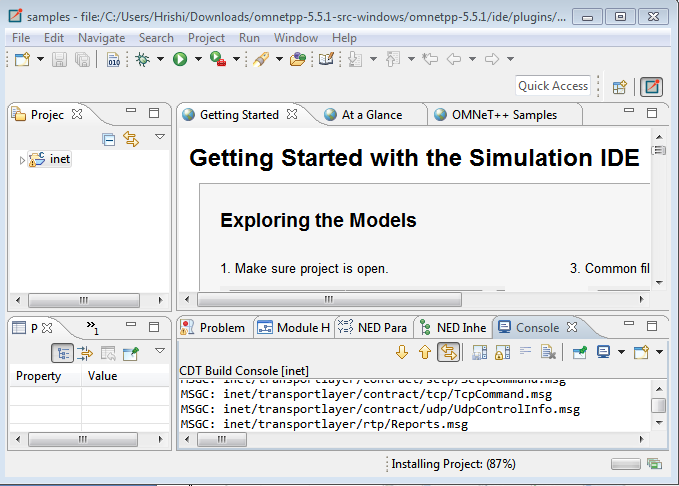
**Practical No. 08**

**AIM:** Create wireless network in OMNET++.

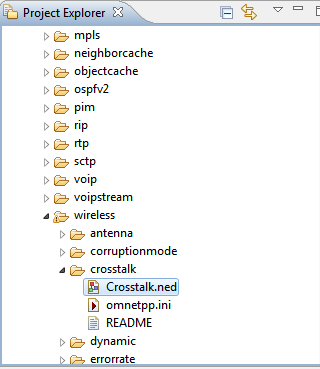
**Step 1:** Go to omnetpp-5.5.1 folder in which open “mingwenv” file, we get following window. Type “omnetpp” command to open omnet++ IDE.



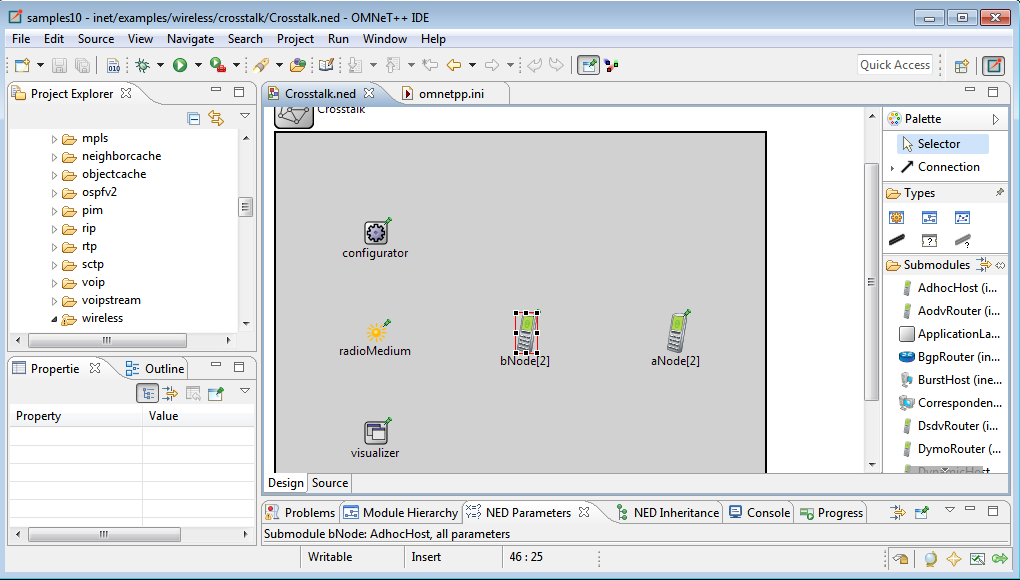
**Step 2:** after that command, following window will open.



**Step 3:** Go to project Explorer > inet > examples > wireless > crosswalk and open Crosstalk.ned file.



**Crosstalk.ned:**



**Coding:**

**Crosstalk.ned:**

package inet.examples.wireless.crosstalk;

import inet.networklayer.configurator.ipv4.Ipv4NetworkConfigurator;

import inet.node.inet.AdhocHost;

import inet.physicallayer.ieee80211.packetlevel.Ieee80211DimensionalRadioMedium;

import inet.visualizer.contract.IIntegratedVisualizer;

network Crosstalk

{

parameters:

submodules:

visualizer: <default("IntegratedCanvasVisualizer")> like IIntegratedVisualizer if hasVisualizer() {

parameters:

@display("p=100,300;is=s");

}

configurator: Ipv4NetworkConfigurator {

parameters:

@display("p=100,100;is=s");

}

radioMedium: Ieee80211DimensionalRadioMedium {

parameters:

@display("p=100,200;is=s");

}

aNode[2]: AdhocHost {

parameters:

@display("r=,,#707070;p=400,200");

}

bNode[2]: AdhocHost {

parameters:

@display("r=,,#707070;p=250,200");

}

}

**Omnetpp.ini:**

[General]

network = Crosstalk

*#record-eventlog = true*

sim-time-limit = 0.1s

seed-set = 1

\*\*.constraintAreaMinX = 0m

\*\*.constraintAreaMinY = 0m

\*\*.constraintAreaMinZ = 0m

\*\*.constraintAreaMaxX = 100m

\*\*.constraintAreaMaxY = 100m

\*\*.constraintAreaMaxZ = 0m

*# mobility*

\*.\*Node[\*].mobility.typename = "StationaryMobility"

\*.\*Node[\*].mobility.initFromDisplayString = false

*# udp App*

\*.\*Node[\*].numApps = 1

\*.\*Node[0].app[0].typename = "UdpSink"

\*.\*Node[\*].app[0].typename = "UdpBasicApp"

\*.\*Node[\*].app[0].localPort = 100

\*.\*Node[\*].app[0].destPort = 100

\*.\*Node[\*].app[0].messageLength = 1250B

\*.\*Node[\*].app[0].startTime = exponential(100us)

\*.\*Node[\*].app[0].sendInterval = exponential(100us)

\*.aNode[\*].app[0].destAddresses = "aNode[0]"

\*.bNode[\*].app[0].destAddresses = "bNode[0]"

*# nic*

\*.\*Node[\*].wlan[\*].radio.typename = "Ieee80211DimensionalRadio"

*# medium*

\*.radioMedium.backgroundNoise.dimensions = "time frequency"

*# radio*

\*.\*Node[\*].wlan[\*].radio.transmitter.bandwidth = 20 MHz

\*.\*Node[\*].wlan[\*].radio.receiver.bandwidth = 20 MHz

\*.\*Node[\*].wlan[\*].radio.transmitter.dimensions = "time frequency"

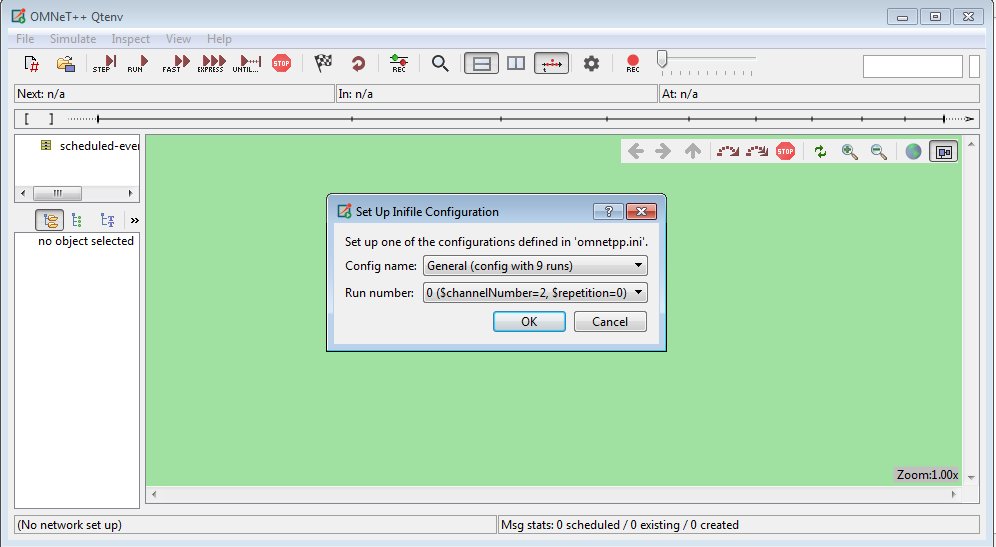
\*.aNode[\*].wlan[\*].radio.channelNumber = 1

\*.bNode[\*].wlan[\*].radio.channelNumber = ${channelNumber=2..10}

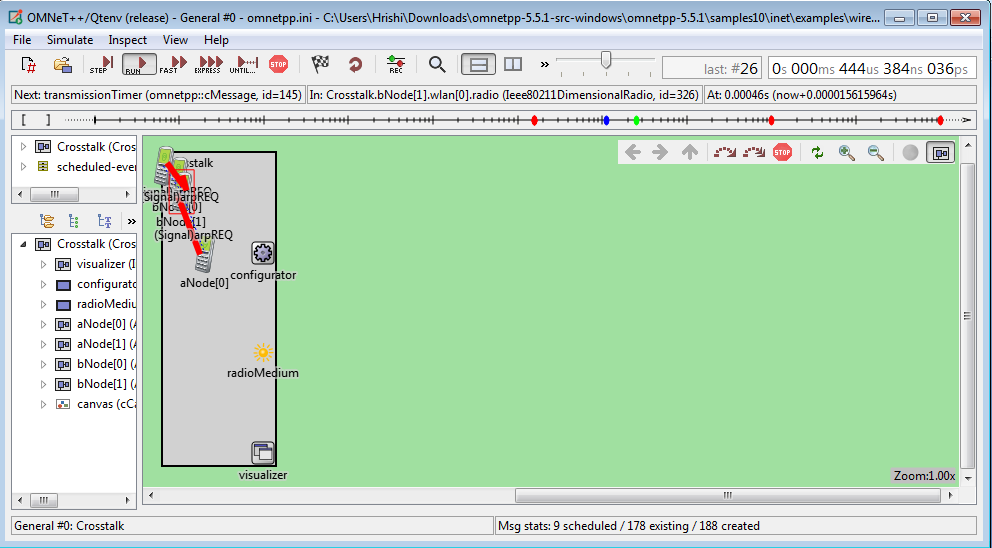
**Step 4:** Click on Run button.



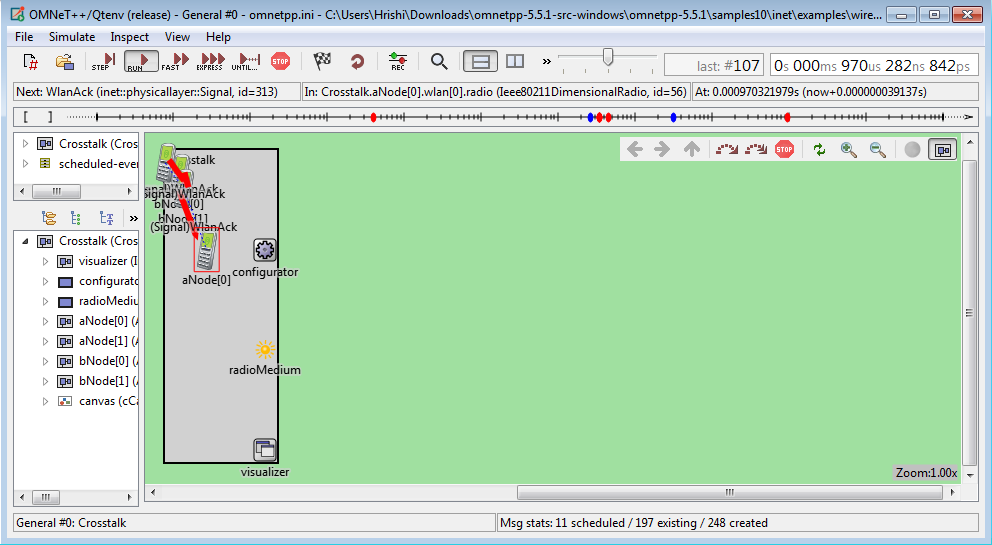
**Step 5:** After that following window will open.

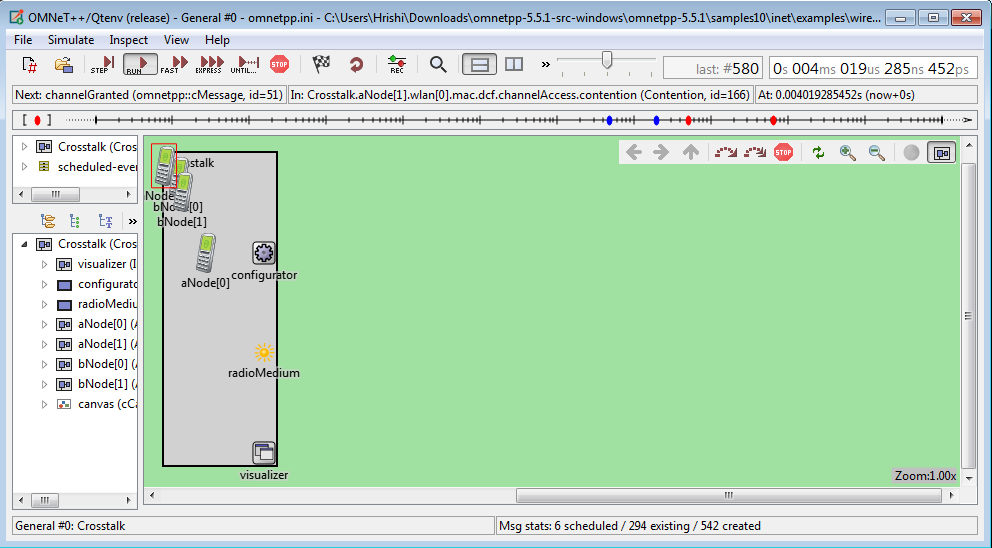
****

**Step 6:** Click OK and then Click on RUN.



**OUTPUT:**

****

****

**Conclusion:** We have learnt to wireless network in OMNET++.