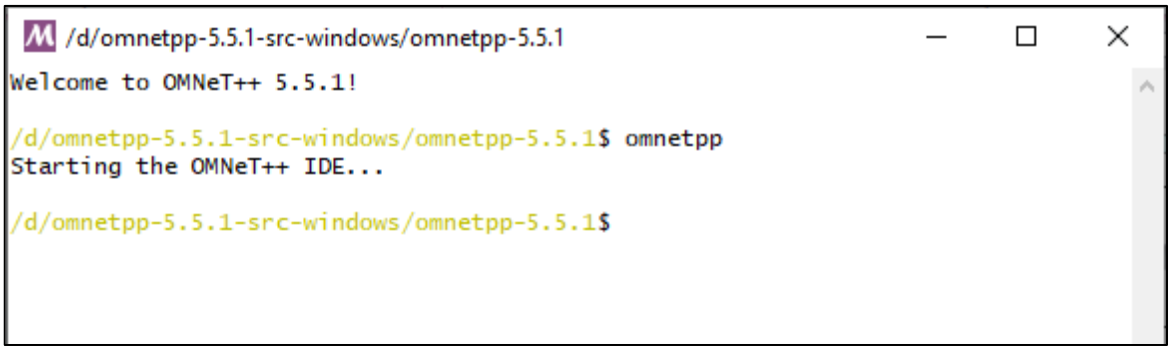
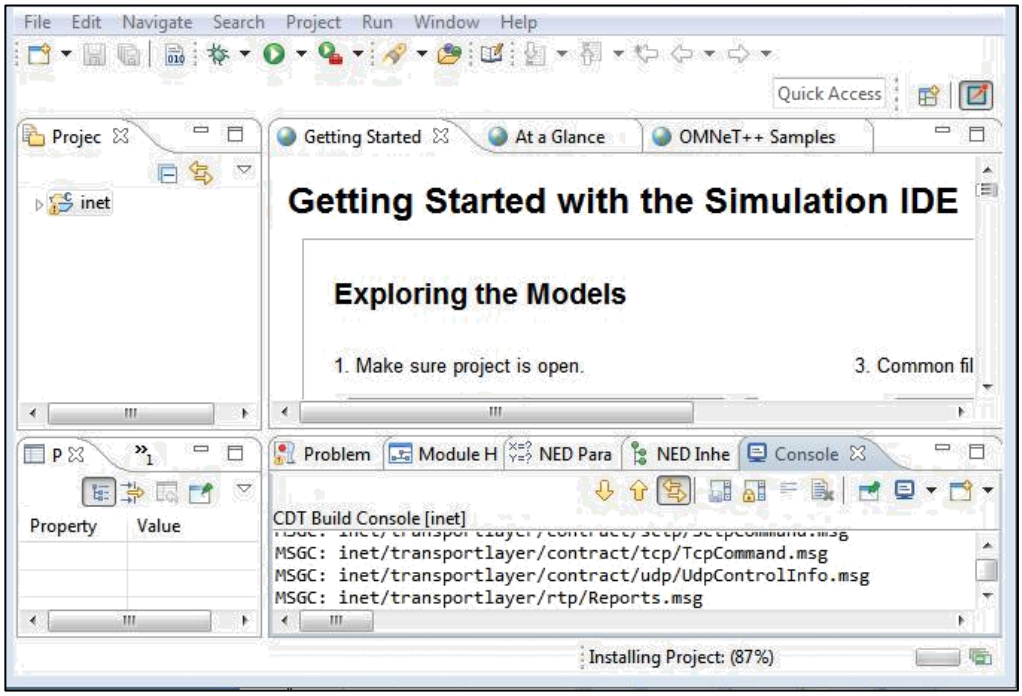
## PRACTICAL NO: 7

**AIM:** Create Single mobile network.

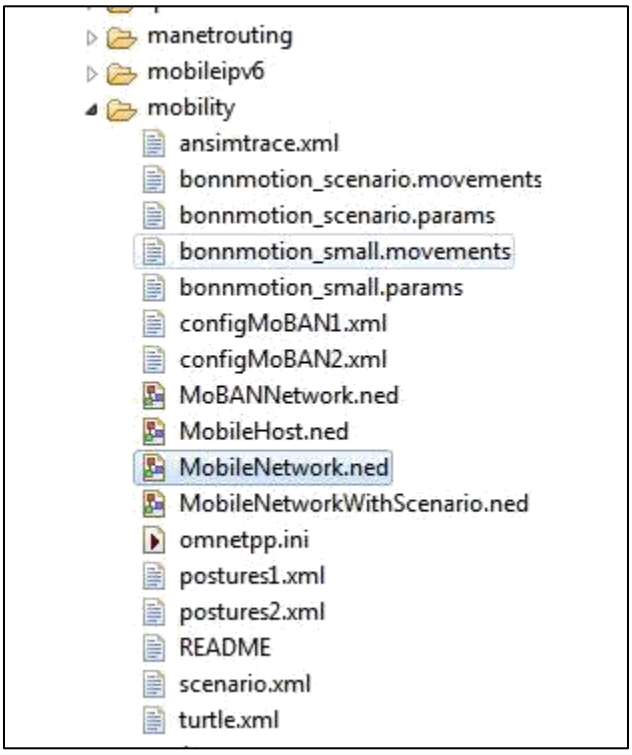
**Step 1:** Go to omnetpp-5.5.1 folder in which open "mingwenv.cmd" file, we get following window. Type "omnetpp" command to open omnet++ IDE.



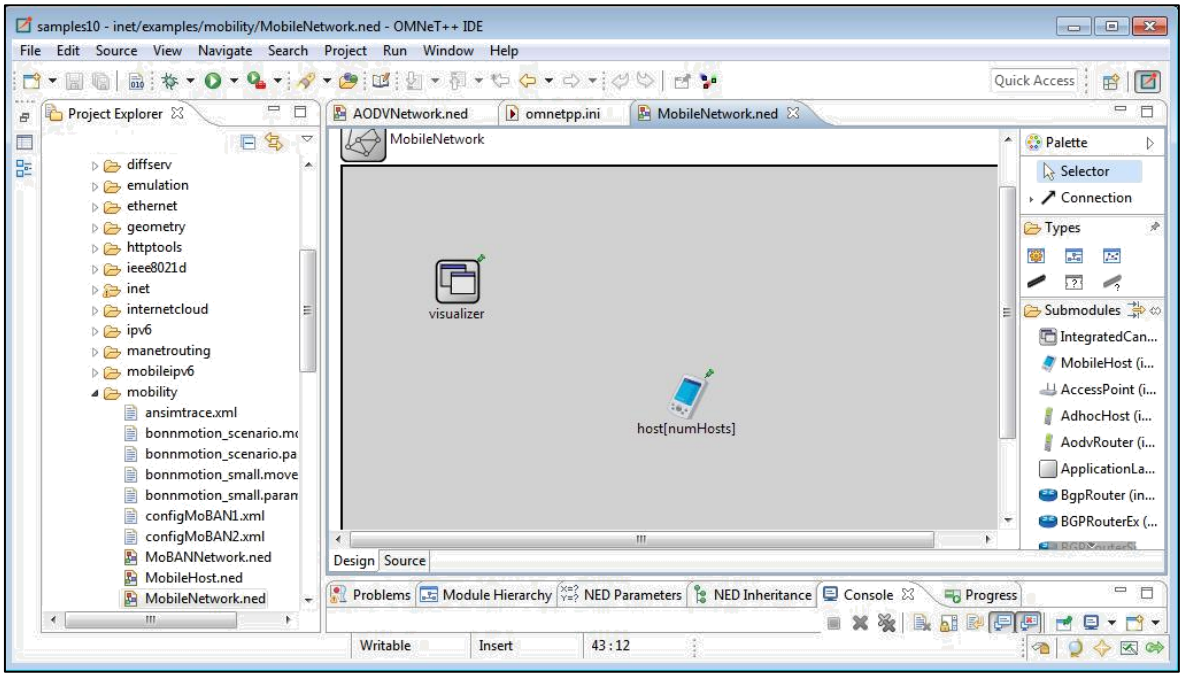

**Step 2:** after that command, following window will open.



**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
```

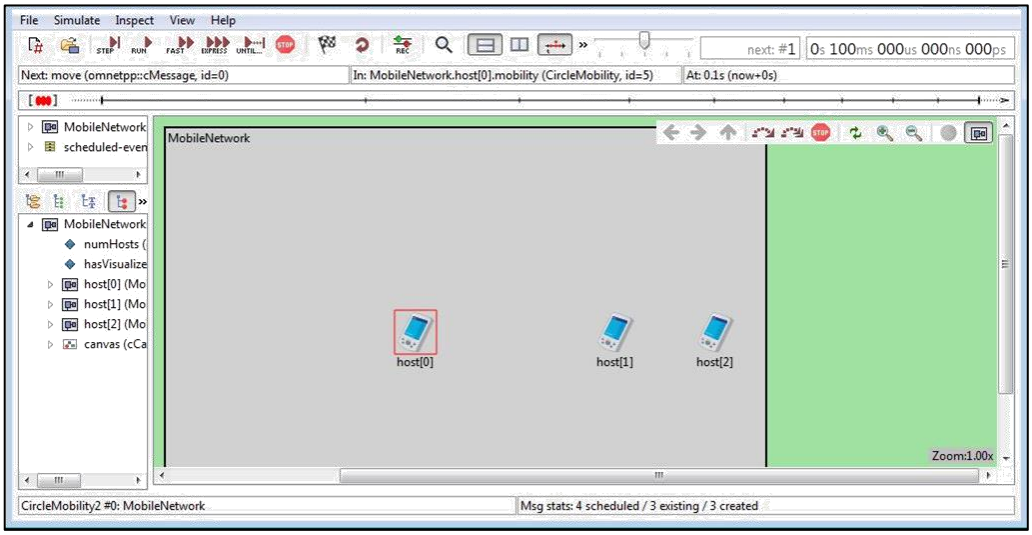
```
**.host*.mobility.typename = "StationaryMobility"
# place it at a fixed position:
**.host[0].mobility.initialX = 50m
**.host[0].mobility.initialY = 200m
**.host[0].mobility.initFromDisplayString = false
the second node is using the display string position (or placed randomly if position is not present 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
```

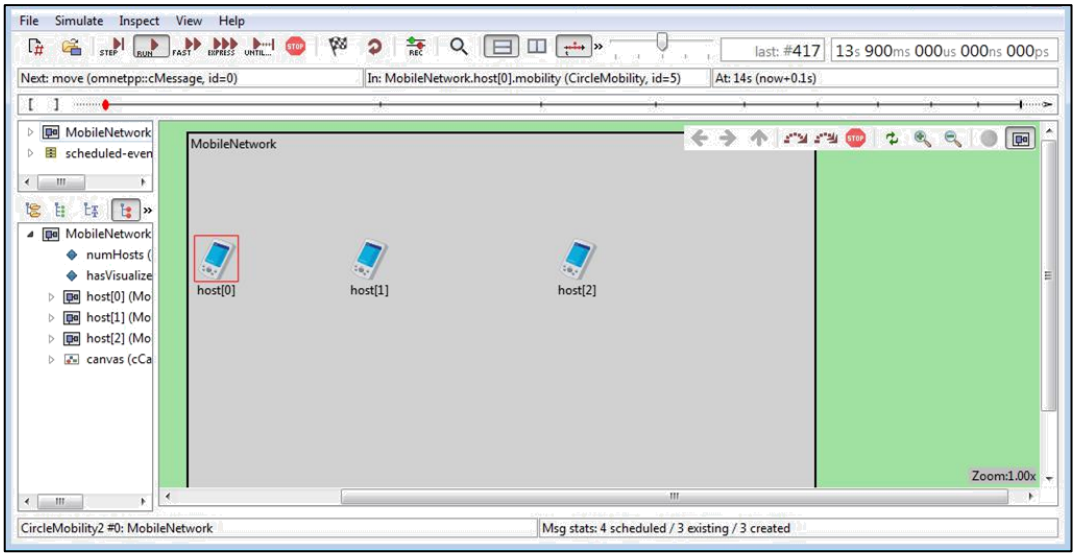
other hosts are also moving around in a larger circle following host[0]
**.host[0].mobility.element[1].typename = "CircleMobility"
**.host[0].mobility.element[1].faceForward = false **.host[0].mobility.element[1].cx = 300m
**.host[0].mobility.element[1].cy = 200m **.host[0].mobility.element[1].r = 150m
**.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.

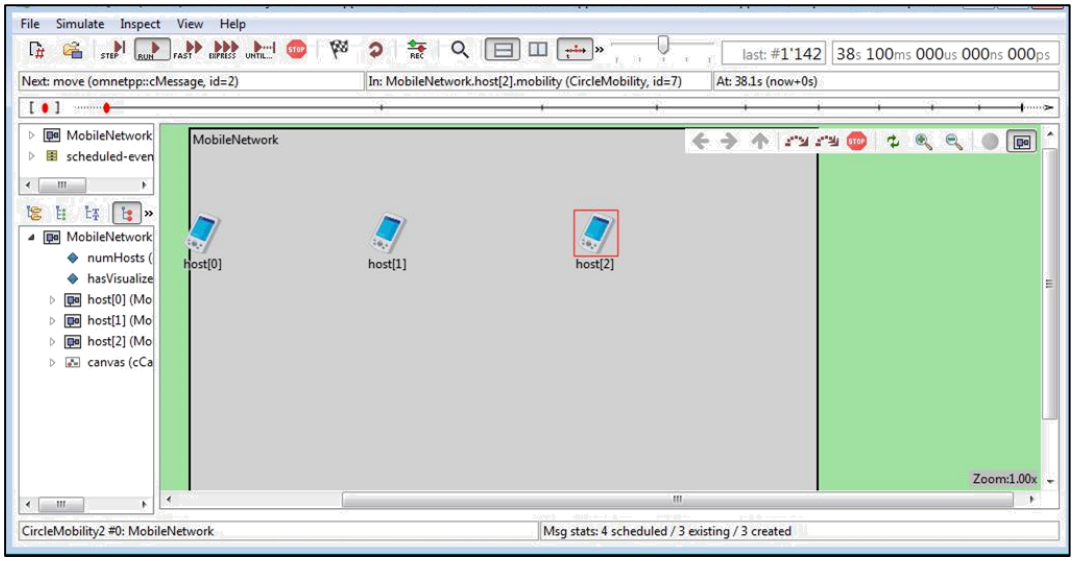
**Step 5:** After that following window will open.




**Step 6:** Click on RUN.



**OUTPUT:**




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