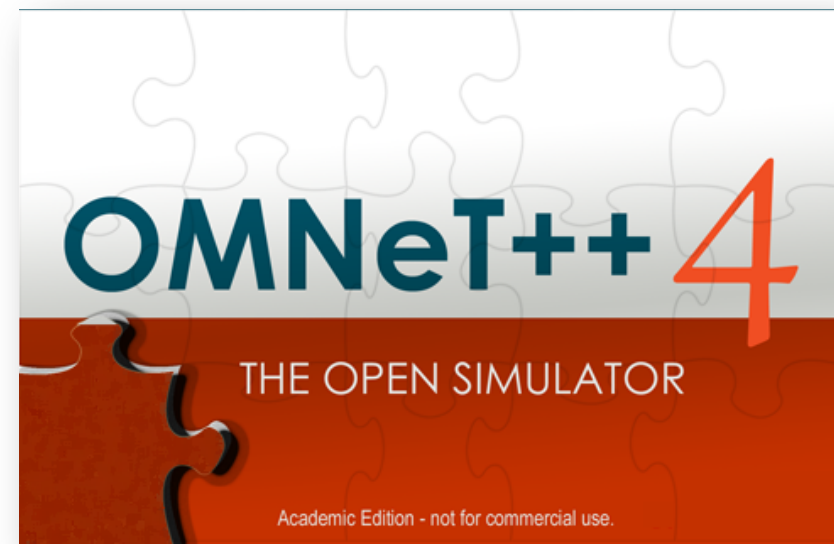


Mobile and Wireless Networks - Labs

Performance Analysis

- General guidelines to performance analysis
- Introduction to OMNET++
- Study case 1: A simple wired network
- Study case 2: A simple MANET
- Study case 3: A simple VANET



Wireless Networks

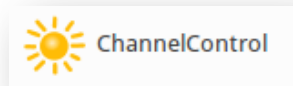
○ INET provides support

- Wireless networks with infrastructure
- Wireless networks without infrastructure and MANET

○ Common modules

○ World

- `inet.world.radio.ChannelControl`:
Distributes the packets that are within range of the transmitter.



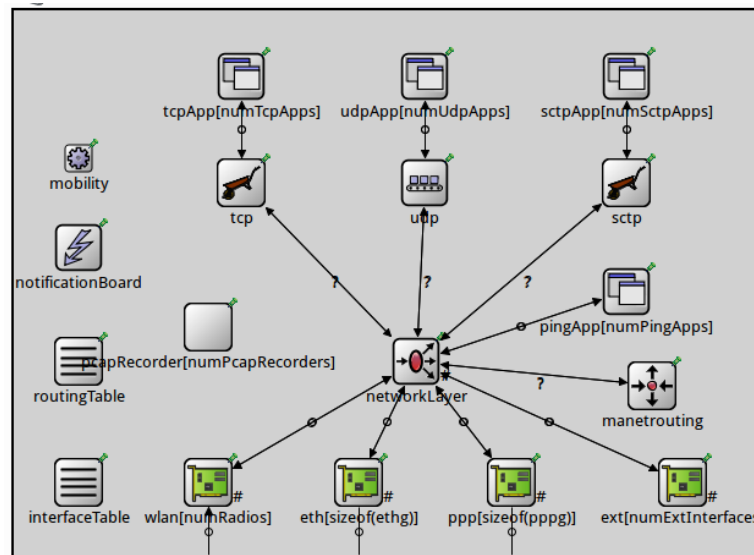
```
# channel physical parameters
*.channelControl.carrierFrequency = 2.4GHz
*.channelControl.pMax = 2.0mW
*.channelControl.sat = -110dBm
*.channelControl.alpha = 2
*.channelControl.numChannels = 1
```

Wireless Networks

○ Common modules

● Nodes

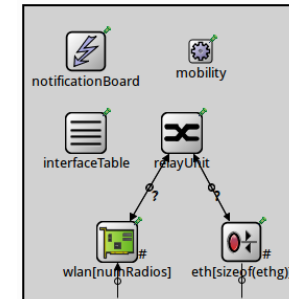
- inet.nodes.inet.WirelessHost:
Represents a host in a wireless network with infrastructure
- inet.nodes.inet.AdhocHost:
Represents a host in a MANET. Includes a routing module.
- inet.nodes.wireless.AccessPoint:
Represents an access-point



Wireless Networks

○ Common

- Protocols and other devices:
 - inet.networklayer.IManetRouting and inet.networklayer.manetrouting.base.BaseRouting: Routing protocols for MANETs.
 - inet.networklayer.ipv4.RoutingTable: Routing table
 - inet.mobility.IMobility and inet.mobility.models.*: To define nodes mobility.



```
[Config StationaryMobility]
network = MobileNetwork
*.numHosts = 3
**.host*.mobilityType = "StationaryMobility"
# place it at a fixed position:
**.host[0].mobility.initialX = 50m
**.host[0].mobility.initialY = 200m
**.host[0].mobility.initFromDisplayString = false
```

```
# mobility
**.host*.mobilityType = "MassMobility"
**.host*.mobility.initFromDisplayString = false
**.host*.mobility.changeInterval = truncnormal(2s, 0.5s)
**.host*.mobility.changeAngleBy = normal(0deg, 30deg)
**.host*.mobility.speed = truncnormal(20mps, 8mps)
**.host*.mobility.updateInterval = 100ms
```

```
[Config LinearMobility]
network = MobileNetwork
**.host*.mobilityType = "LinearMobility"
**.host*.mobility.initFromDisplayString = false
**.host*.mobility.speed = 50mps
**.host*.mobility.angle = 30deg # degrees
```

```
[Config CircleMobility1]
network = MobileNetwork
*.numHosts = 3
**.host*.mobilityType = "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
```

Wireless Networks

○ Common

● Protocols and other devices:

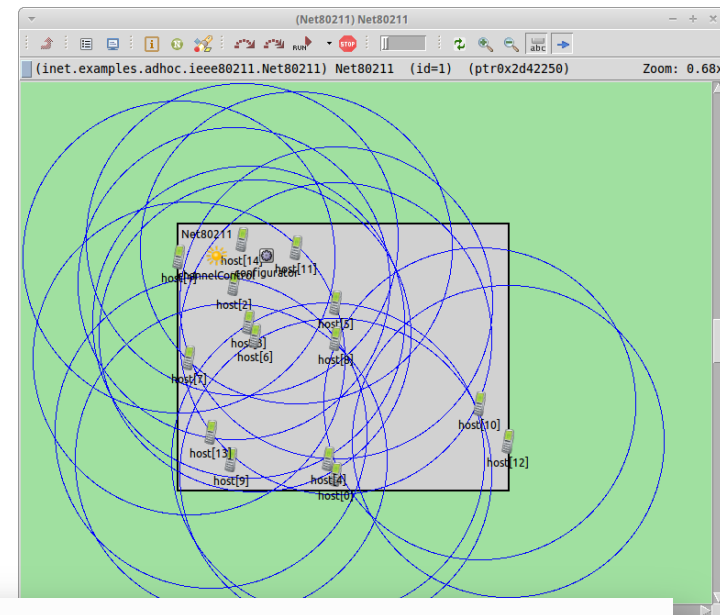
- inet.linklayer.ieee80211.ieee80211Nic:
Implementation of wireless network interface.
- inet.linklayer.ieee80211.mgmt.ieee80211MgmtAP,
inet.linklayer.ieee80211.mgmt.ieee80211MgmtSTA
inet.linklayer.ieee80211.mgmt.ieee80211MgmtAdhoc:

Module that configures the wireless interface to work as an access point, a station or a adhoc node .

```
# channel physical parameters
**.radio.carrierFrequency = 2.4GHz
**.radio.transmitterPower = 2.0mW
**.radio.thermalNoise = -110dBm
**.radio.pathLossAlpha = 2
**.radio.ReceiveAntennaGainIndB = 1dB
```

Example

- Execution of the `inet.examples.adhoc.ieee80211` example



```
# nic settings
**.wlan[*].bitrate = 2Mbps

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

**.wlan[*].mac.address = "auto"
**.wlan[*].mac.maxQueueSize = 14
**.wlan[*].mac.rtsThresholdBytes = 3000B
**.wlan[*].mac.retryLimit = 7
**.wlan[*].mac.cwMinData = 7
**.wlan[*].mac.cwMinBroadcast = 31

**.wlan[*].radio.transmitterPower = 2mW
**.wlan[*].radio.thermalNoise = -110dBm
**.wlan[*].radio.sensitivity = -85dBm
**.wlan[*].radio.pathLossAlpha = 2
**.wlan[*].radio.snirThreshold = 4dB
```

```
# channel physical parameters
*.channelControl.carrierFrequency = 2.4GHz
*.channelControl.pMax = 2.0mW
*.channelControl.sat = -110dBm
*.channelControl.alpha = 2
*.channelControl.numChannels = 1

# mobility
**.host*.mobilityType = "MassMobility"
**.host*.mobility.initFromDisplayString = false
**.host*.mobility.changeInterval = truncnormal(2s, 0.5s)
**.host*.mobility.changeAngleBy = normal(0deg, 30deg)
**.host*.mobility.speed = truncnormal(20mps, 8mps)
**.host*.mobility.updateInterval = 100ms
```

Mymanet file .NED

```
package inet.examples.MANETs;

import inet.networklayer.autorouting.ipv4.IPv4NetworkConfigurator;
import inet.nodes.inet.AdhocHost;
import inet.world.radio.ChannelControl;

network mymanet
{
  parameters:
    int numHosts;
    int numFixHosts;

  submodules:
    fixhost[numFixHosts]: AdhocHost {
      parameters:
        @display("i=device/pocketpc_s;r=,,#707070");
    }
    host[numHosts]: AdhocHost {
      parameters:
        @display("i=device/pocketpc_s;r=,,#707070");
    }
    channelControl: ChannelControl {
      parameters:
        @display("p=60,50;i=misc/sun");
    }
    configurator: IPv4NetworkConfigurator {
      parameters:
        config=xml("<config><interface hosts='*' address='145.236.x.x' netmask='255.255.0.0'/></config>");
        @display("p=140,50;i=block/cogwheel_s");
    }
  connections allowunconnected:
}
```

Mymanet file .INI (UDP session)

```
[General]
network = mymanet

#debug-on-errors = true
sim-time-limit = 3000s
seed-0-mt = 5

**.drawCoverage=false
**.constraintAreaMinX = 0m
**.constraintAreaMinY = 0m
**.constraintAreaMinZ = 0m
**.constraintAreaMaxX = 1000m
**.constraintAreaMaxY = 1000m
**.constraintAreaMaxZ = 0m
*.numFixHosts = 1
*.numHosts = 50
**.arp.globalARP = true

# mobility
**.mobility.initFromDisplayString = false
**.fixhost[0].mobility.initialX = 999m
**.fixhost[0].mobility.initialY = 999m
**.host[0].mobility.initialX = 1m
**.host[0].mobility.initialY = 1m
**.host*.mobilityType = "StationaryMobility"

# udp apps (on)
**.host[0].numUdpApps = 1
**.host[*].udpApp[*].typename = "UDPBasicBurst"

**.udpApp[0].destAddresses = "fixhost[0]"
**.udpApp[0].localPort = 1234
**.udpApp[0].destPort = 1234

**.udpApp[0].messageLength = 512B #
#**.udpApp[0].messageLength = 2000B #

**.udpApp[0].sendInterval = 0.5s +
uniform(-0.001s,0.001s)
#**.udpApp[0].sendInterval = 0.2s +
uniform(-0.001s,0.001s)

**.udpApp[0].burstDuration = 0
# **.udpApp[0].burstDuration = uniform(1s,4s,1)
**.udpApp[0].chooseDestAddrMode = "perBurst"
**.udpApp[0].sleepDuration = 1s
**.udpApp[0].stopTime = 0s
# **.udpApp[0].stopTime = uniform(20s,40s,1)
**.udpApp[0].startTime = 10s
#**.udpApp[0].startTime = uniform(0s,4s,1)
**.udpApp[0].delayLimit = 20s

**.fixhost[0].udpApp[*].typename = "UDPSink"
**.fixhost[0].numUdpApps = 1
**.fixhost[0].udpApp[0].localPort = 1234

**.routingProtocol = "DSRUU"
```


Exercises:

- Modify the UDP session parameters
- Modify the “routingProtocol”
 - You can choose among:
“AODVUU”, “DYMOUM”, “DYMO”, “DSRUU”,
“OLSR”, “OLSR_ETX”, “DSDV_2”, “Batman”
 - and evaluate how the packet delivery rate changes, or any parameters you may want to analyse
- You can try also to modify the “mobilityType” and evaluate how, e.g., the packet delivery rate changes

```
**mobilityType = "LinearMobility"  
**mobility.angle = 0  
**mobility.speed = 1mps  
  
**mobilityType = "RandomWPMobility"  
**mobility.angle = 0  
**mobility.speed = uniform(1mps, 10mps)
```

- <http://inet.omnetpp.org/doc/INET/neddoc/index.html?p=inet.mobility.models-package.html>

Mymanet file .INI (TCP session)

```
** .host[0].numTcpApps = 1
** .host[0].tcpApp[0].typename = "TCPSessionApp"
** .host[0].tcpApp[0].connectAddress = "fixhost[0]"
** .host[0].tcpApp[0].connectPort = 1000
** .host[0].tcpApp[0].active = true
# Client
** .host[0].tcpApp[0].sendBytes = 1000MiB

** .fixhost[0].numTcpApps = 1
** .fixhost[0].tcpApp[0].typename = "TCPSessionApp"
** .fixhost[0].tcpApp[0].connectAddress = "host[0]"
** .fixhost[0].tcpApp[0].localPort = 1000
# Listen port
** .fixhost[0].tcpApp[0].active = false
```