TP4.Getting started in Wireless communication with INET

Part -I-

I. Introduction

INET Framework (https://inet.omnetpp.org/) is an open-source model library for the OMNeT++ simulation environment. It provides protocols, agents and other models for researchers and students working with communication networks. INET is especially useful when designing and validating new protocols, or exploring new or exotic scenarios.

INET supports a wide class of communication networks, including wired, wireless, mobile, ad hoc and sensor networks. It contains models for the Internet stack (TCP, UDP, IPv4, IPv6, OSPF, BGP, etc.), link layer protocols (Ethernet, PPP, IEEE 802.11, various sensor MAC protocols, etc), refined support for the wireless physical layer, MANET routing protocols, DiffServ, MPLS with LDP and RSVP-TE signaling, several application models, and many other protocols and components. It also provides support for node mobility, advanced visualization, network emulation and more. (https://inet.omnetpp.org/Tutorials.html) and for wireless tutorials see the link : https://inet.omnetpp.org/docs/tutorials/wireless/doc/index.html

The goal of this assignment is to get familiar with the simulation of WLAN with INET.

II. Démarche

1. Créer un projet Omnet++ avec les dossiers : simulation et src.
2. Copier les fichiers (.ned,.ini,.xml) depuis le projet wireless qui se trouve dans : Project explorer \inet\tutorials\wireless, vers le projet (dossier simulation) crée en question-1 et de préférence renommer les fichiers et corriger le nom du package dans les fichiers .ned.
3. Changer (pointer) les préférences (Select a project from the project explorer--->Properties--->Project References--->inet) du projet vers *inet*.
4. Lancer la simulation depuis le fichier omnetpp.ini et 14 Scénarios de simulation s'affichent (Figure 1)

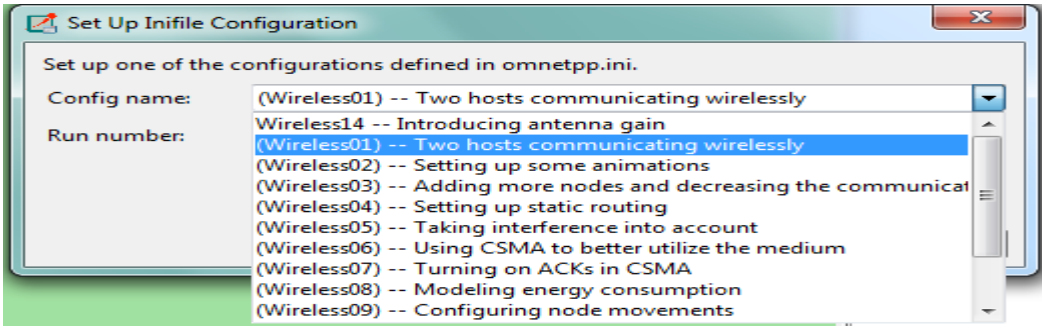

Figure 1. Scénarios de simulation du tutorial Wireless.

III. Required work:

Analyze all simulation scenarios and draw conclusions.

Mr. AzeddineATTIR
azedine.attir@univ-msila.dz

Part -II-

In the first step, we want to create a network that contains two hosts, with one host sending a UDP data stream wirelessly to the other.

1. NED file "WirelessA"

```
import inet.networklayer.configurator.ipv4.Ipv4NetworkConfigurator;
importinet.node.inet.INetworkNode;
importinet.physicallayer.contract.packetlevel.IRadioMedium;
importinet.visualizer.contract.IIntegratedVisualizer;

networkWirelessA
{
parameters:
@display("bgb=650,500;bgg=100,1,grey95");
@figure[title](type=label; pos=0,-1; anchor=sw; color=darkblue);
@figure[rcvdPkText](type=indicatorText; pos=380,20; anchor=w; font=,18;
textFormat="packets received: %g"; initialValue=0);
@statistic[packetReceived](source=hostB.app[0].packetReceived; record=figure(count);
targetFigure=rcvdPkText);

submodules:
        visualizer:
<default("IntegratedCanvasVisualizer")>likeIIntegratedVisualizerifhasVisualizer() {
@display("p=580,125");
        }
        configurator: Ipv4NetworkConfigurator {
@display("p=580,200");
        }
radioMedium: <default("UnitDiskRadioMedium")>likeIRadioMedium {
@display("p=580,275");
        }
hostA: <default("WirelessHost")>likeINetworkNode {
@display("p=50,325");
        }
hostB: <default("WirelessHost")>likeINetworkNode {
@display("p=450,325");
}
}
```

- The model contains a playground of the size 500x650 meters, with two hosts spaced 400 meters apart.
- As you can see, the hosts' type is parametric in this NED file (defined via a *hostType* parameter and the *INetworkNode* module interface). This is done so that in later steps we can replace hosts with a different NED type. The actual NED type hereis*WirelessHost*.
- IP addresses are assigned to hosts by an *Ipv4NetworkConfigurator* module, which appears as the configurator submodule in the network.
- MAC addresses with per-host *GlobalArp* modules.
- Host A generates UDP packets that are received by host *B.A* is configured to contain a UdpBasicApp module, which generates 1000-byte UDP messages at random intervals with exponential distribution, the mean of which is 12ms. Host *B* contains a UdpSink application that just discards received packets.

- The model also displays the number of packets received by host B. The text is added by the @figure[rcvdPkText] line, and the subsequent line arranges the figure to be updated during the simulation.
- **Physical layer modeling**: Let us concentrate on the module called *radioMedium*. All wireless simulations in INET need a radio medium module. This module represents the shared physical medium where communication takes place. It is responsible for taking signal propagation, attenuation, interference, and other physical phenomena into account. INET can model the wireless physical layer at various levels of detail, realized with different radio medium modules. In this step, we use *UnitDiskRadioMedium*, which is the simplest model. It implements a variation of unit disc radio, meaning that physical phenomena like signal attenuation are ignored, and the communication range is simply specified in meters. Transmissions within range are always correctly received unless collisions occur. Modeling collisions (overlapping transmissions causing reception failure) and interference range (a range where the signal cannot be received correctly, but still collides with other signals causing their reception to fail) are optional. NOTE: Naturally, this model of the physical layer has little correspondence to reality. However, it has its uses in the simulation. Its simplicity and consequent predictability are an advantage in scenarios where realistic modeling of the physical layer is not a primary concern, for example in the modeling of ad-hoc routing protocols. Simulations using *UnitDiskRadioMedium* also run faster than more realistic ones, due to the low computational cost. In hosts, network interface cards are represented by NIC modules. Radio is part of wireless NIC modules. There are various radio modules, and one must always use one that is compatible with the medium module. In this step, hosts contain *UnitDiskRadio* as part of AckingWirelessInterface. In this model, we configure the chosen physical layer model (*UnitDiskRadioMedium* and *UnitDiskRadio*) as follows. The communication range is set to 500m. Modeling packet losses due to collision (termed "interference" in this model) is turned off, resulting in pair-wise independent duplex communication channels. The radio data rates are set to 1 Mbps. These values are set in omnetpp.ini with the communicationRange, *ignoreInterference*, and *bitrate* parameters of the appropriate modules.
- MAC layer : NICs modules also contain an L2 (i.e. data link layer) protocol. The MAC protocol in *AckingWirelessInterface* is configurable, the default choice being *MultipleAccessMac*. *MultipleAccessMac* implements a trivial MAC layer which only provides encapsulation/decapsulation but no real medium access protocol. There is virtually no medium access control: packets are transmitted as soon as the previous packet has completed transmission. *MultipleAccessMac* also contains an optional out-of-band acknowledgment mechanism which we turn off here.

2. *inifile* “omnetpp.ini”

```
[Config Wireless01]
description = Two hosts communicating wirelessly
network = WirelessA
sim-time-limit = 20s

*.host*.ipv4.arp.typename = "GlobalArp"

*.hostA.numApps = 1
*.hostA.app[0].typename = "UdpBasicApp"
*.hostA.app[0].destAddresses = "hostB"
*.hostA.app[0].destPort = 5000
*.hostA.app[0].messageLength = 1000B
*.hostA.app[0].sendInterval = exponential(12ms)
*.hostA.app[0].packetName = "UDPData"

*.hostB.numApps = 1
*.hostB.app[0].typename = "UdpSink"
*.hostB.app[0].localPort = 5000

*.host*.wlan[0].typename = "AckingWirelessInterface"
*.host*.wlan[0].mac.useAck = false
*.host*.wlan[0].mac.fullDuplex = false
*.host*.wlan[0].radio.transmitter.communicationRange = 500m
*.host*.wlan[0].radio.signalAnalogRepresentation = "unitDisk"
*.host*.wlan[0].radio.receiver.ignoreInterference = true
*.host*.wlan[0].mac.headerLength = 23B
```

3. Travail demandé:

1. Configurer le temps de simulation pour 10s et calculer et afficher le débit réel de l’application.
2. Activer les interférences et répondez à la question1.
3. Activer le protocole CSMA/CA et répondez à la question1.
4. Activer le mécanisme des acquittements et répondez à la question1.
5. Que concluez-vous ?

4. Help

https://inet.omnetpp.org/docs/tutorials/wireless/doc/index.html