Université Badji Mokhtar Annaba
Faculté de Technologie
Département d'Informatique

Chargé de cours : Haiahem Rahim
Évaluation des performances des systèmes
RSI M2

TP N°1
Introduction à OMNET++

Objective **M**odular **N**etwork **T**estbed in C++ (**OMNeT++)** est une plateforme de simulation, modulaire, open source, orientée objet et à événements discrets écrit en C++. Elle offre un IDE basé sur Eclipse, un environnement d'exécution graphique, et une foule d'autres outils. Elle a été conçue pour créer des simulateurs pour les réseaux de communication, les systèmes multi processeurs, et d'autres systèmes distribués.

1. Historique

Le développement d'**OMNeT++** a commencé en 1992 par Andras Vargas à l'université de Budapest. Actuellement, Ce projet est utilisé par des dizaines d'université pour la validation de nouveaux matériels et logiciels, ainsi que pour l'analyse de performance et l'évaluation de protocoles de communication. L'avantage de **OMNeT ++** est sa facilité d'apprentissage, d'intégration de nouveaux modules et la modification de ceux déjà implémentés.

2. Installation et configuration

La version d'OMNeT++ la plus à jour actuellement est la version 6.0.1.

- Le guide d'installation d'OMNeT++ sur d'autres platesformes est disponible sur le lien http://omnetpp.org/doc/omnetpp/InstallGuide.pdf
- Le guide d'installation à partir du terminal d'OMNeT++ sur la plateforme Ubuntu est détaillé dans les sections suivantes :

2.1 Ouvrir un Terminal

Tapez terminal dans votre lanceur de programme et cliquez sur l'icône Terminal.

2.2. Installation des packages prérequis

- Avant de démarrer l'installation, actualisez la base de données des packages disponibles.

  Tapez dans le terminal :

  $ sudo apt-get update

- Pour installer les packages requis, tapez dans le terminal :

  $ sudo apt-get install build-essential clang lld gdb bison flex perl python3 python3-pip qtbase5-dev qtchooser qt5-qmake qtbase5-dev-tools libqt5opengl5-dev libxml2-dev zlib1g-dev doxygen graphviz libwebkit2gtk-4.0-37 python3 -m pip install --user --upgrade numpy pandas matplotlib scipy seaborn posix_ipc

- $ sudo apt-get install openscenegraph-plugin-osgearth libosgearth-dev
- $ sudo apt-get install mpi-default-dev openmpi-bin libopenmpi-dev

Aux questions de confirmation (Voulez-vous continuer ? [O/N]), répondez O.

2.3 Télécharger le package OMNeT++

- Téléchargez le package OMNeT++ à partir de l'URL suivante (**https://omnetpp.org/download/**) . puis , Sélectionnez la version d'omnet++ dont vous avez besoin.
- Et décompressez le en utilisant la commande tar –xvzf Nom_du_fichier.tar.gz.

2.4 Installer omnet++ -(version)

- Dans un premier temps, accédez au package omnet++ en utilisant la commande cd omnetpp-(version).
- Configurez le package à l'aide de la commande : ./configure
- Créez et installez le package en utilisant la commande : make

Pour lancer l'IDE d'OMNeT++, il suffit d'utiliser la commande : omnetpp

3. L'architecture d'OMNET++

- L'architecture d'OMNET++ est hiérarchique. Un modèle OMNeT++ se compose de modules qui communiquent par le principe de passage de message. Un module peut être soit module simple ou bien un module composé. Les feuilles de cette architecture sont les modules simples qui représentent les classes C++. Pour chaque module simple correspond un fichier .cc et un fichier .h.

- Un module composé est composé de simples modules ou d'autres modules composés connectés entre eux. Les paramètres, les sous modules et les ports (**gates**) de chaque module sont spécifiés dans un fichier .ned (NeD : Network Description).

- Dans la Fig. 3.1, les cases représentent les modules simples et les modules composés. Les flèches reliant les petites cases représentent les connexions et les portes.

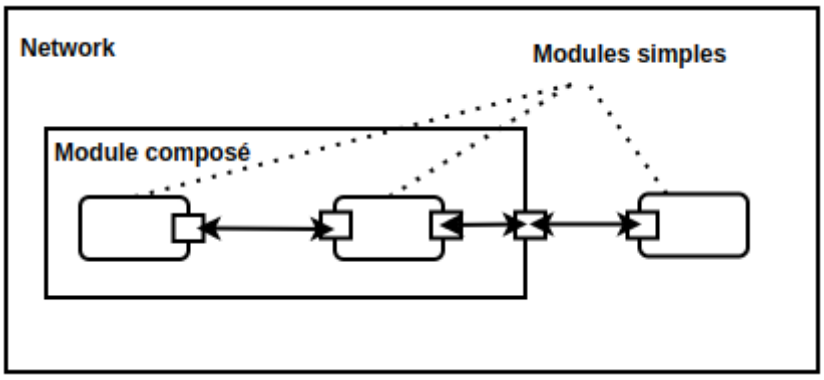

Figure 3.1 Modules simples et composés

- La communication entre les différents modules se fait à travers les échanges de messages (.msg). Les messages peuvent représenter des paquets, des trames d'un réseau informatique, des clients dans une file d'attente ou bien d'autres types d'entités en attente d'un service.

- Les messages sont envoyés et reçus à travers des ports qui représentent les interfaces d'entrer et de sortie pour chaque module. La conception d'un réseau se fait dans un fichier .ned et les différents paramètres de chaque module sont spécifiés dans un fichier de configuration (.ini). OMNeT++ génère à la fin de chaque simulation deux nouveaux fichiers omnet.vec et omnet.sca qui permettent de tracer les courbes et calculer des statistiques.

Exemple de déclaration d'un module OMNeT++

Une déclaration de module composé peut contenir plusieurs sections, toutes facultatives :

```
module Host
{
    types:
        ...
    parameters:
        ...
    gates:
        ...
    submodules:
        ...
    connections:
        ...
}
```

4. Questions :

Q1. Quelle est la différence entre un module simple et un module composé ?
Q2. Qu'est-ce qu'un «Port (gates)» ?
Q3. Énumérez les différents ports d'un module simple (même question pour un module composé).