

TP N° 4

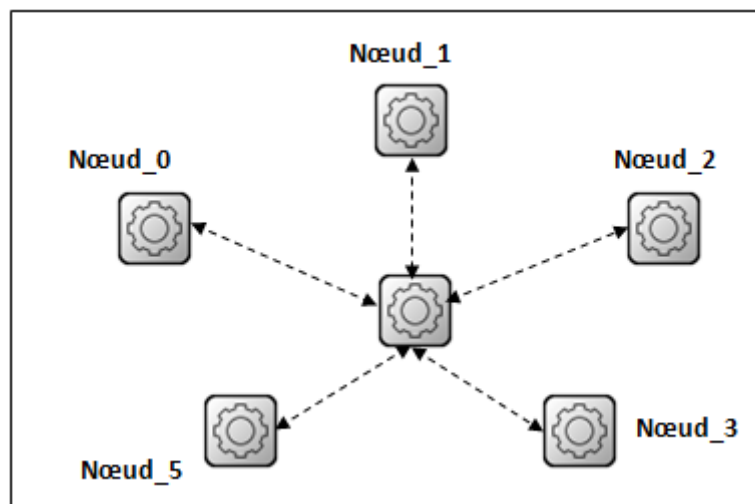
Objectif :

L'objectif de ce TP est de créer de nouveaux types de messages et les utiliser par la suite pour d'éventuels échanges entre les nœuds du réseau. Pour ce faire, nous allons nous baser sur la structure en étoile créée dans le TP3.

En effet, le nœud central 4 joue le rôle de la station de base. Il initie la communication et diffuse un message **de type "Hello"** à tous les nœuds du réseau.

Ensuite, chaque nœud qui reçoit ce message, crée un message **de type "Data"**, capte une donnée (qui sera la température dans cet exemple) et l'envoie au nœud 4.

Après la réception de tous les messages de données, le nœud 4 procède à l'agrégation de toutes les températures reçues et affiche leur moyenne.



Structure de réseau en étoile

1. **Création d'un nouveau projet:** Nommez le TP4
2. **Création du fichier.NED (etoile.ned):** reproduisez le fichier etoile.ned créé dans le tp3.

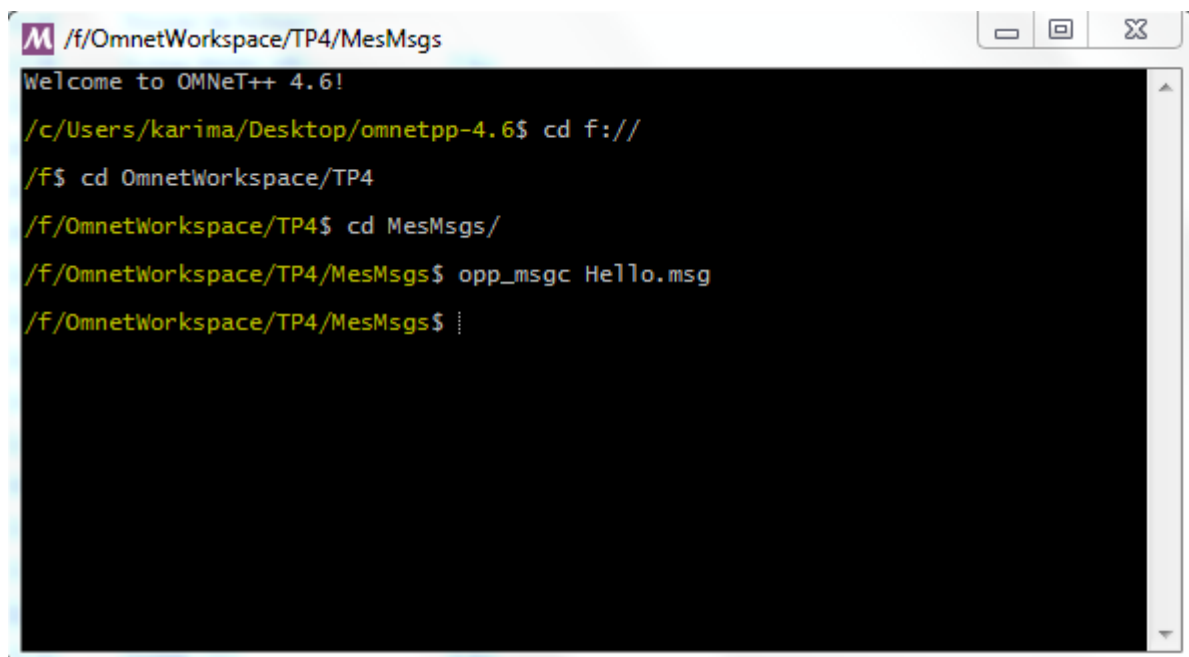
3. **Création des messages:**

- 3.1. **Création du message Hello.msg**

- Créez un dossier à l'intérieur de la racine du projet "TP4", nommez le MesMsg
- Cliquez avec le bouton droit sur le dossier **MesMsg**, ensuite : **new/Message definition (.msg)**
- Nommez le fichier **Hello.msg** et cliquez sur suivant ou finish
- Copiez le code source ci-dessous dans le fichier créé :

```
packet HelloMsg{  
    int sourceId;  
    int dest;  
}
```

Ensuite, pour créer une classe à ce message (i.e. HelloMsg.cc et HelloMsg.h), ouvrez la console **mingwenv** se trouvant dans votre dossier omnetpp-4.6, allez vers l'emplacement où se trouve le dossier **MesMsg** dans le projet TP4 (dans votre omnetworkspace) et saisissez la commande suivante: **opp_msgc Hello.msg**



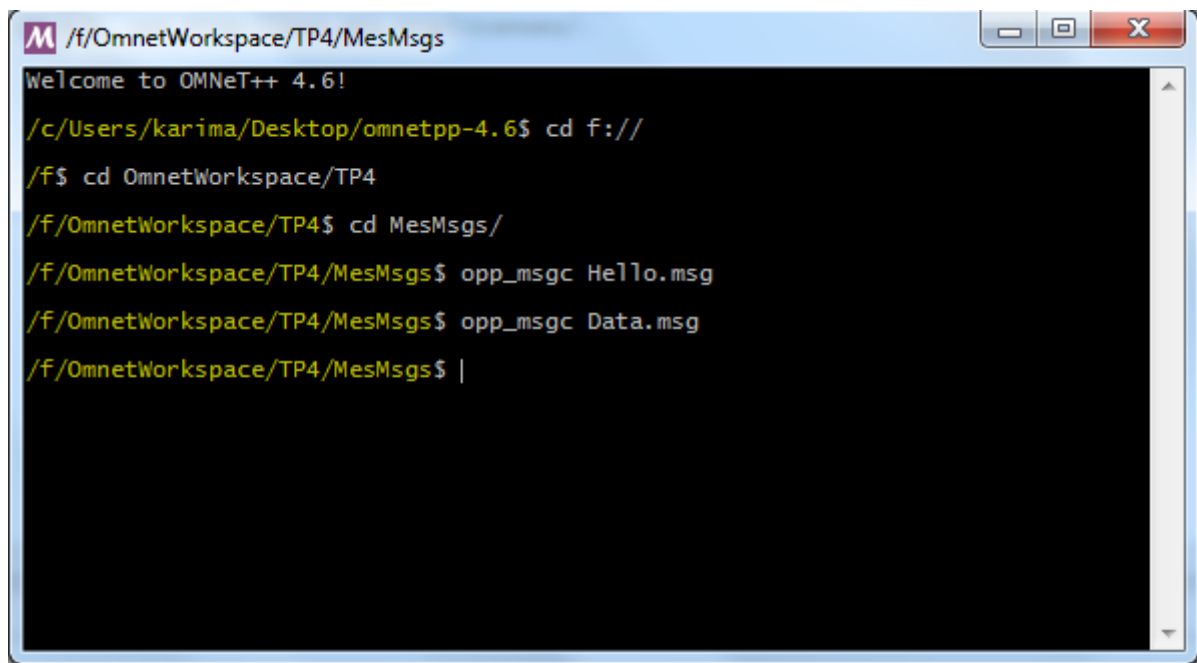
```
/f/OmnetWorkspace/TP4/MesMsgs  
Welcome to OMNeT++ 4.6!  
/c/Users/karima/Desktop/omnetpp-4.6$ cd f://  
/f$ cd OmnetWorkspace/TP4  
/f/OmnetWorkspace/TP4$ cd MesMsgs/  
/f/OmnetWorkspace/TP4/MesMsgs$ opp_msgc Hello.msg  
/f/OmnetWorkspace/TP4/MesMsgs$ :
```

3.2. Création du message Data.msg

- Cliquez avec le bouton droit sur le dossier **MesMsg** ensuite : **new/Message definition (.msg)**
- Nommez le fichier **Data.msg** et cliquez sur suivant ou finish
- Copiez ce code source dans le fichier créé :

```
packet DataMsg {  
    int source;  
    int Dest;  
    double data;  
}
```

Ensuite, il faut créer une classe à ce message en saisissant dans la console mingwenv la commande **opp_msgc Data.msg** (la même procédure que le message Hello.msg de l'étape 3.1).

A screenshot of a terminal window titled "/f/OmnetWorkspace/TP4/MesMsgs". The terminal shows the following commands and output:

```
Welcome to OMNeT++ 4.6!  
/c/Users/karima/Desktop/omnetpp-4.6$ cd f://  
/f$ cd OmnetWorkspace/TP4  
/f/OmnetWorkspace/TP4$ cd MesMsgs/  
/f/OmnetWorkspace/TP4/MesMsgs$ opp_msgc Hello.msg  
/f/OmnetWorkspace/TP4/MesMsgs$ opp_msgc Data.msg  
/f/OmnetWorkspace/TP4/MesMsgs$ |
```

1. Création de la classe Nœud : Nœud.h

```
#ifndef NOEUD_H_
#define NOEUD_H_

#include<omnetpp.h>
using namespace std; //utiliser la bibliothèque standard std du C++

struct tabNoeud{ // utiliser un enregistrement contenant deux champs pour
sauvegarder les données au niveau de la station de base
    int nId;
    double data;
};
typedef list <tabNoeud> Tableau;    // déclarer un nouveau type: tableau
d'enregistrement
typedef list <tabNoeud>::iterator itérateur;    // déclarer un itérateur de type
enregistrement
class Noeud : public cSimpleModule
{
protected:

    virtual void initialize();
    virtual void handleMessage(cMessage *msg);
int it;
Tableau tab;
itérateur i;
double S=0;
double Moy;

};
#endif /* NOEUD_H_ */
```

Noeud.cc

```
#include "Noeud.h"
#include <stdio.h>
#include <string.h>
#include <omnetpp.h>
#include <Data_m.h>
#include <Hello_m.h>

Define_Module(Noeud);

void Noeud::initialize()
{
    if (getIndex() == 4) {
        EV << "Je suis le noeud " << getIndex() << " je diffuse le message Hello dans tout  
le réseau" << "\n" ;
        int n = gateSize("gate");

        for(it=0; it!=n; it++){
            HelloMsg *helloMsg = new HelloMsg(); // créer un message de type Hello
            helloMsg->setSourceId(getIndex());
            helloMsg->setDest(it);
            send(helloMsg, "gate$o", it); //diffuser le message à tous les noeuds du réseau
        }
    }
}

void Noeud::handleMessage(cMessage *msg)
{
    if (getIndex()==4){
        DataMsg *data = check_and_cast<DataMsg*>(msg); // convertir le message reçu  
vers le type DataMsg pour pouvoir le décapsuler

        if ((data->getDest())==4){
            tabNoeud T;
            T.nId= data->getSource(); // recupérer la source du message data
            T.data= data->getData(); // recupérer la donnée de température reçue
            tab.insert(tab.end(),T); // insérer la donnée dans le tableau tab
            EV << "je suis le noeud"<<getIndex()<< "j'ai reçu la température: " <<
            data->getData()<< "degré"<<"du noeud:"<<T.nId << "\n";
        }

        if (tab.size()==5) // si le noeud 4 a reçu toutes les températures
        {
            for(i=tab.begin(); i!=tab.end(); i++){ // parcourir le tableau pour  
recupérer toutes les températures reçues et faire la somme
                S=S+ (*i).data;
            }

            Moy=S/tab.size(); // calculer la moyenne des températures reçues
            EV << "je suis le noeud"<<getIndex()<< " la température moyenne est: " <<
            Moy << "\n";
        }
    }
}
```

```
else {  
    HelloMsg *hello = check_and_cast<HelloMsg*>(msg); // convertir le message reçu  
                                                       vers le type HelloMsg pour pouvoir le décapsuler  
  
    DataMsg *dataMsg = new DataMsg(); // créer un message de donnée pour l'envoyer au  
                                         noeud 4  
    dataMsg->setData(rand()% 100); // générer une valeur de température de façon  
                                     aléatoire entre 0 et 100  
    dataMsg->setDest(hello->getSourceId()); // mettre l'adresse de la destination  
  
    dataMsg->setSource(getIndex()); // mettre l'adresse de la source  
  
    EV << "Je suis le noeud " << getIndex() << " j'envoie un message de donnée au  
        noeud 4."<<" \n";  
  
    send(dataMsg, "gate$o", 0); // envoyer le message de donnée au noeud 4  
  
}  
  
} // fin de la fonction HandleMessage
```

IMPORTANT :

Le chemin spécifié dans l'invite de commande mingwenv dépend de l'emplacement où vous avez enregistré votre projet TP4.

(/f/Omnetworkspace/TP4/MesMsgs est mon chemin à moi où j'ai sauvegardé mon projet) donc pour vous, il faudra mettre votre chemin où se trouve votre projet)