

REPUBLIQUE DU CAMEROUN  
Paix-Travail-Patrie  
MINISTRE DE L'ENSEIGNEMENT  
SUPERIEUR  
UNIVERSITE DE NGAOUNDERE  
FACULTE DES SCIENCES  
DEPARTEMENT DE MATHÉMATIQUES  
ET INFORMATIQUE



REPUBLIC OF CAMEROON  
Peace-Work-Fatherland  
MINISTRY OF HIGHER  
EDUCATION  
UNIVERSITY OF NGAOUNDERE  
FACULTY OF SCIENCES  
DEPARTMENT OF MATHEMATICS  
AND COMPUTER SCIENCE

## U.E : Techniques de Simulations Réseaux

Thème : Performances evaluation in a wireless network using  
UDP sockets

### MEMBRES DE L'EQUIPE (Groupe N° 4) :

N°	NOM ET PRENOM(S)	MATRICULE
1	GBETNKOM EMMANUEL	12B209FS
2	GUIRKI BAIDI	18A973FS
3	KENEWE LAMNA EMMANUEL	15A385FS
4	KOTVA GOUDOUNGOU SAMUEL	16B119FS
5	LABNA SANDJE PRUDENT	19B340FS
6	MAMOUDOU KOUROUMA	17B515FS
7	MBOGNING TCHOFFO JUNIOR	16B101FS
8	MICHE FRANK	15A749FS
9	MOHAMADOU	19B681FS
10	MOHAMADOU BEN BOUNDOUMI	14B320FS

**PARCOURS : SLED**

**NIVEAU : MASTER 1**

### Sous la supervision :

**Dr. Ing TCHAKOUNTE Franklin/  
Mme ZONGO Epse NDO**

*Année académique 2019/2020*

## TABLE DES MATIERES

Introduction.....	3
Première partie : Procédure de communication entre deux sockets UDP.....	4
Deuxième partie : Détermination de temps de latence. ....	7
Troisième Partie : Diffusion des paquets .....	8
Conclusion .....	16

## Introduction

Une **Socket***(prise en français)* est un terme informatique qui peut avoir plusieurs significations suivant qu'il est utilisé dans le cadre logiciel ou matériel. Dans le contexte des logiciels, on peut le traduire par « connecteur réseau » ou « interface de connexion ». L'UDP (User Datagram Protocol) en français protocole de datagramme utilisateur quant à lui est un des principaux protocoles de télécommunication utilisés par Internet. Il fait partie de la couche transport du modèle OSI, quatrième couche de ce modèle, comme TCP. Son rôle est de permettre la transmission de données (sous forme de datagrammes) de manière très simple entre deux entités, chacune étant définie par une adresse IP et un numéro de port. Ainsi un socket UDP est un modèle permettant la communication inter processus en utilisant le protocole UDP. Dans notre travail nous nous intéresserons à la simulation (Faire paraître comme réel ce qui ne l'est pas) de procédure de communication entre 2 sockets UDP pour deux cent machines en utilisant le simulateur Omnet++, Au temps nécessaire à un client pour envoyer un message au serveur, Au pourcentage de réception d'un message ou Packet Delivery Ratio (PDR) et à l'Analyse des résultats au travers de graphes commentés.

## **Première partie : PROCEDURE DE COMMUNICATION ENTRE DEUX SOCKETS UDP.**

- **Nous présentons ici par ces images le code qui nous a permis de créer et d'exécuter le socket.**

Ainsi nous avons le fichier clientserveur.net

```
package inet.examples.inet.Tp_groupe4;
import inet.common.misc.NetAnimTrace;
import inet.networklayer.configurator.ipv4.Ipv4NetworkConfigurator;
import inet.node.inet.StandardHost;
import ned.DatarateChannel;

network ClientServeur
{
    parameters:
        double per = default(0);
        @display("bgb=1109.1324,601.0025");
    types:
        channel C extends DatarateChannel
        {
            datarate = 10Mbps;
            delay = 0.1us;
            per = per;
        }
    submodules:
        client: StandardHost {
            parameters:
                @display("p=87.64875,239.295;i=device/pc;is=v1");
        }
        Serveur: StandardHost {
            parameters:
                @display("p=908.48627,239.295;i=device/router;is=v1");
        }

        configurator: Ipv4NetworkConfigurator {
            parameters:
                @display("p=48,13;is=s");
        }
        // configurator: NextHopNetworkConfigurator {
        //     parameters:
        //         @display("p=53,134");
        // }
}
```

Nous avons aussi le fichier clientserveur.ini

```
[general]
network = inet.examples.inet.Tp_groupe4.ClientServeur
total-stack = 7MiB

**.cncMode = "computed"

## udp apps
**.client*.numApps = 1
**.client*.app[0].typename = "UdpBasicApp"
**.client*.app[0].description = "Socket client"
**.client*.app[0].packetName = "MonPaquet_Client"
**.client*.app[0].localPort = -1
**.client*.app[0].sendInterval = 10s
**.client*.app[0].messageLength = 1000B
**.client*.app[0].stopTime = 25s

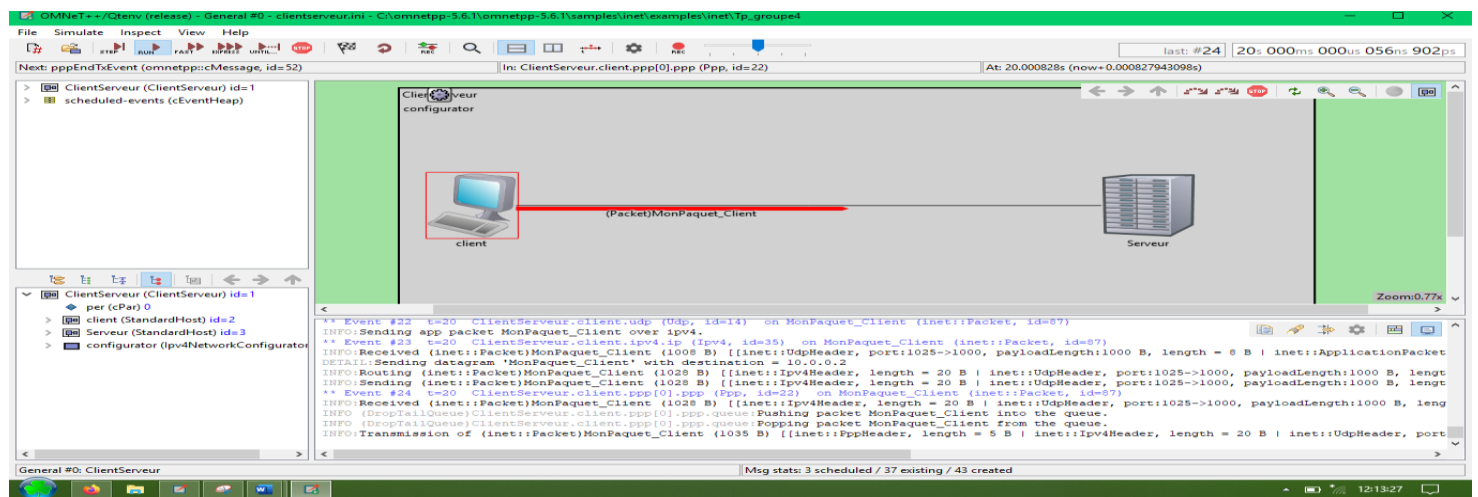
**.Serveur*.numApps = 1
**.Serveur*.app[0].typename = "UdpBasicApp"
**.Serveur*.app[0].description = "Socket Serveur"
**.Serveur*.app[0].packetName = "MonPaquet_Serveur"
**.Serveur*.app[0].localPort = -1
**.Serveur*.app[0].sendInterval = 1s
**.Serveur*.app[0].messageLength = 1000B
**.Serveur*.app[0].stopTime = 25s

**.Serveur*.numApps = 1
**.Serveur*.app[0].typename = "UdpEchoApp"
**.Serveur*.app[0].localPort = 1000
**.Serveur*.app[0].packetName = "Response"

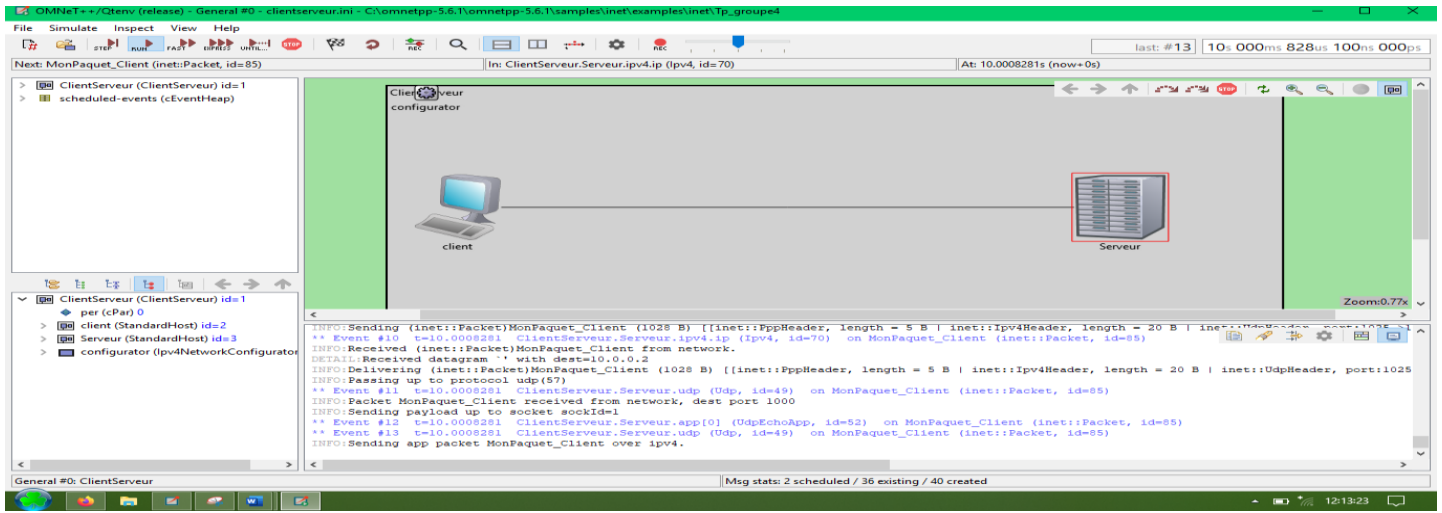
# NIC configuration
**.ppp[*].ppp.queue.typename = "DropTailQueue" # in routers
**.ppp[*].ppp.queue.packetCapacity = 10 # in routers

**.hasIpv4 = true
**.hasIpv6 = false
**.hasGn = false
**.client*.app[0].destAddresses = "Serveur"
**.client*.app[0].destPort = 1000
```

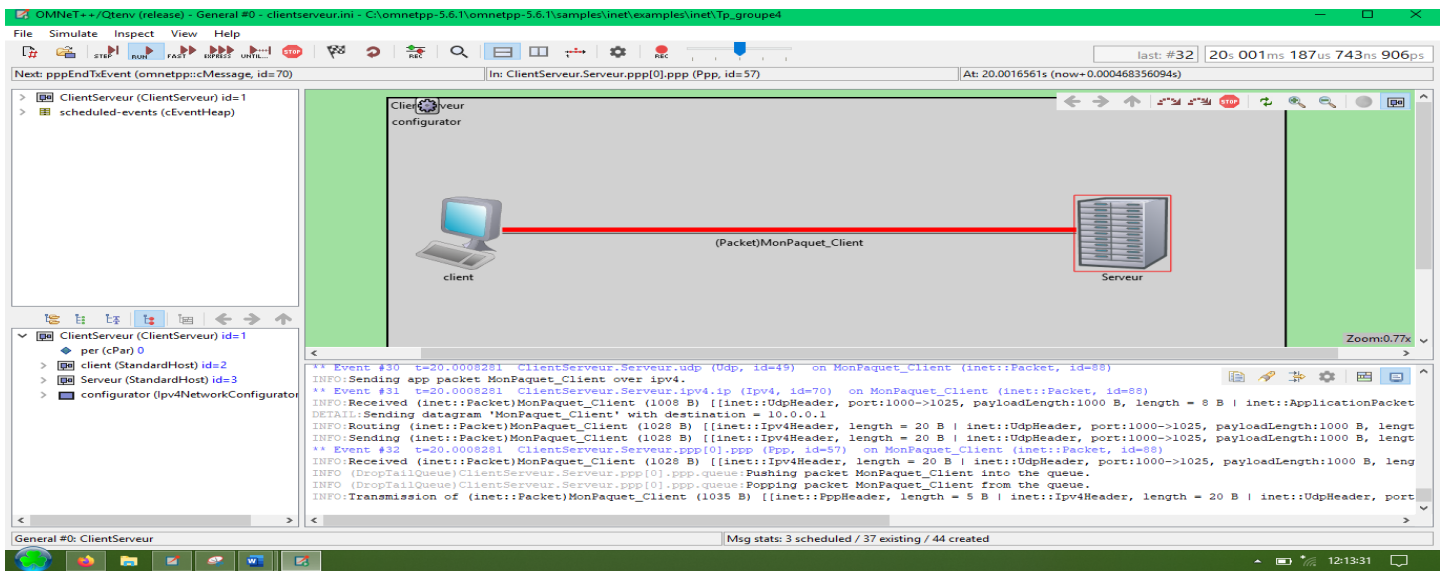
- Nous pouvons voir par cette image la communication établie entre le client et le serveur
- Premièrement le client envoie une requête



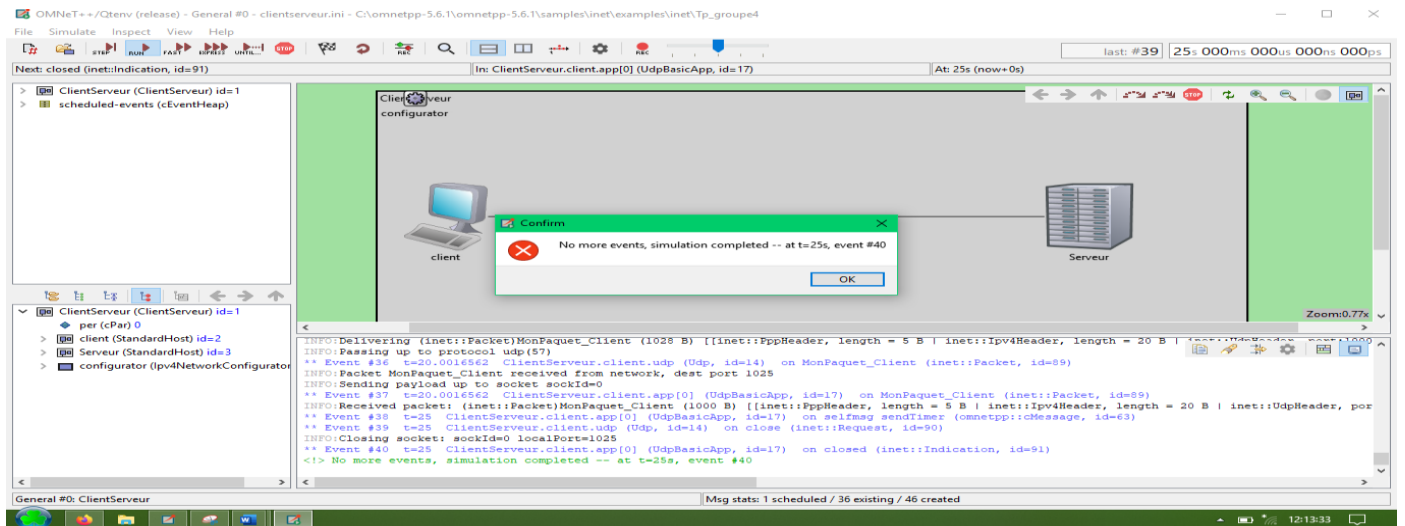
Puis le serveur reçoit la requête et est prêt à répondre



Le serveur à son tour renvoi une réponse au client



Ici nous observons la fin de la communication ce qui indique la fermeture du socket.





**Tl=67s.**

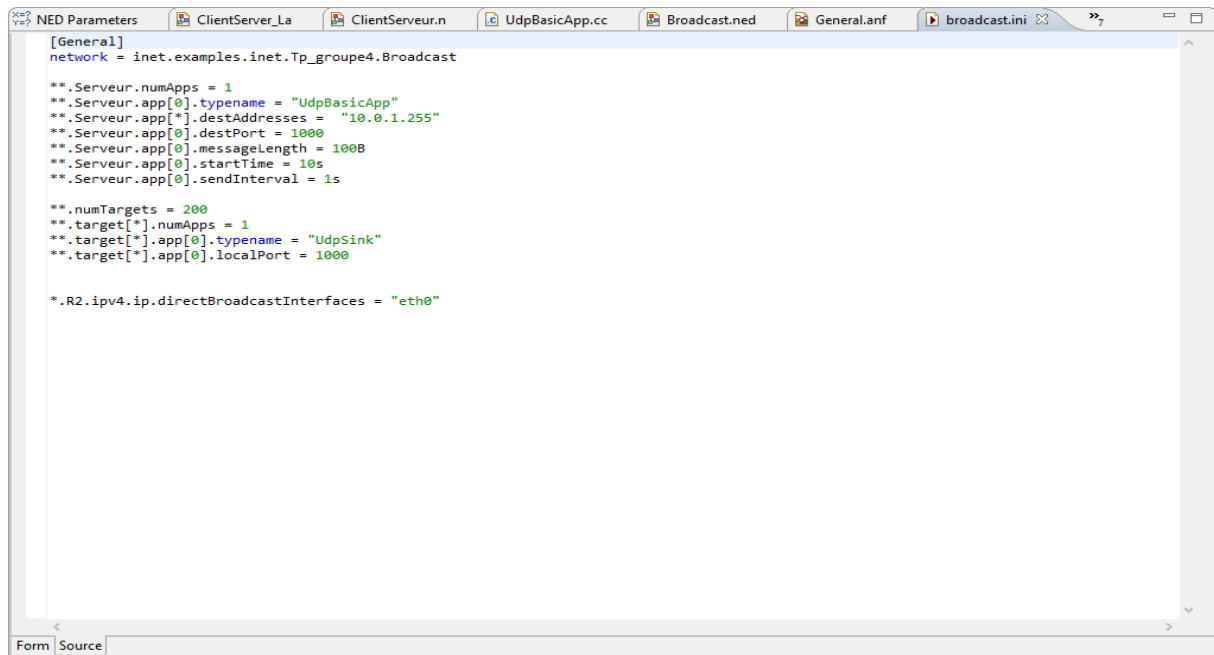
### Troisième partie : DIFFUSION DES PAQUETS

Cas d'utilisation : 200 machines

Pour cette simulation nous prenons le cas où une seule machine dans le réseau fait une diffusion UDP à toutes les machine du réseau. Pour ce faire nous aurons besoin de :

- **D'un serveur** offrant les services à nos deux cents (200) machines en réseau.
- **D'un routeur** qui est un équipement réseau informatique assurant le routage de paquets. Dans notre cas, il sert à faire transiter des paquets UDP d'une interface réseau vers nos différentes machines, selon un ensemble de règles.
- **D'un switch** qui désigne un commutateur réseau, équipement ou appareil qui permet l'interconnexion d'appareils communiquant. Il fonctionne le réseau en domaine
- Et les deux cents (200) **machines**.

**Ci-dessous le Code source de notre simulation :**



```
[General]
network = inet.examples.inet.Tp_groupe4.Broadcast

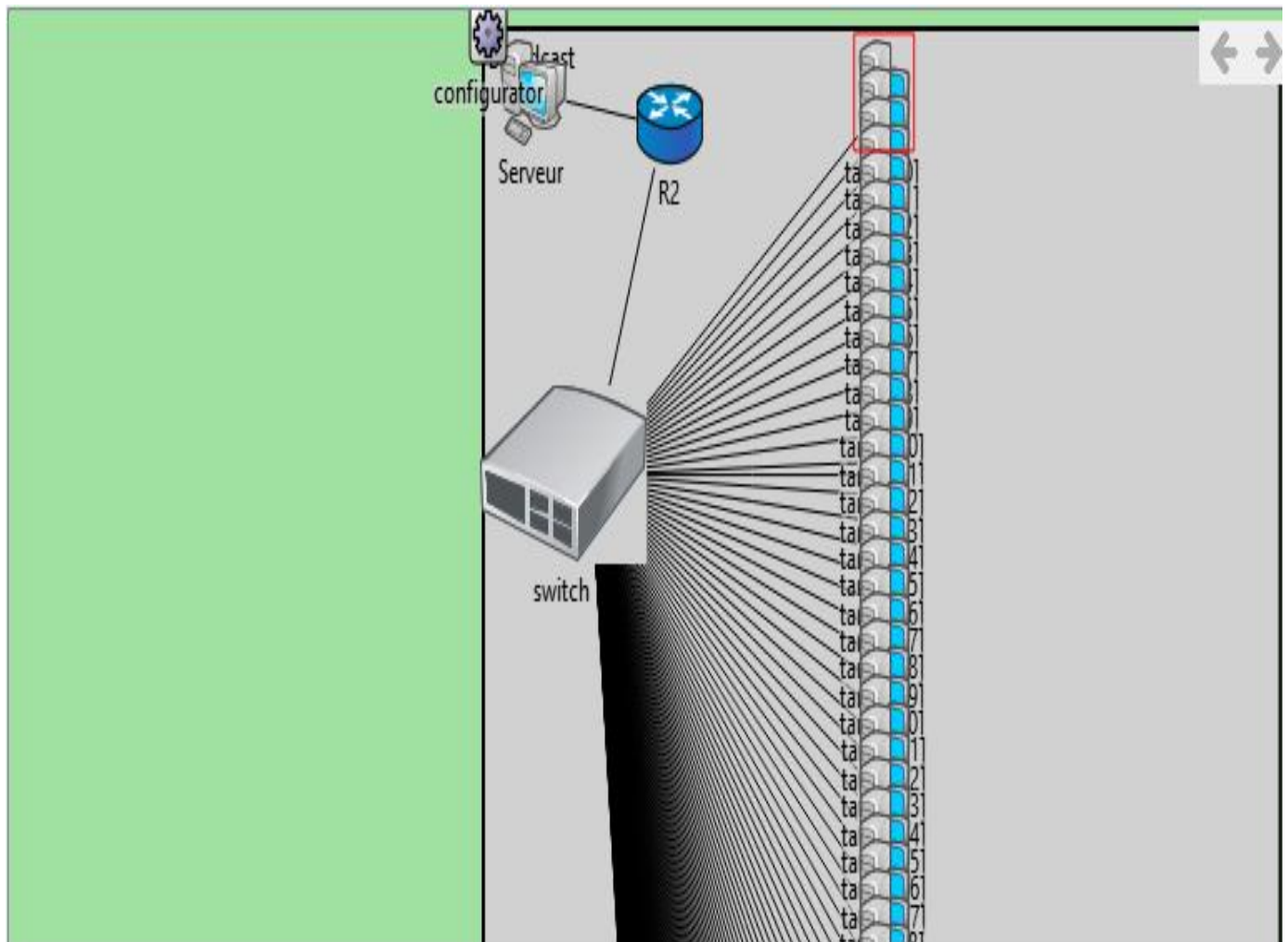
**.Serveur.numApps = 1
**.Serveur.app[0].typename = "UdpBasicApp"
**.Serveur.app[*].destAddresses = "10.0.1.255"
**.Serveur.app[0].destPort = 1000
**.Serveur.app[0].messageLength = 1008
**.Serveur.app[0].startTime = 10s
**.Serveur.app[0].sendInterval = 1s

**.numTargets = 200
**.target[*].numApps = 1
**.target[*].app[0].typename = "UdpSink"
**.target[*].app[0].localPort = 1000

*.R2.ipv4.ip.directBroadcastInterfaces = "eth0"
```



Nous pouvons voir les illustrations en images de la simulation avec nos 200 machines





## Ici la simulation en cour d'exécution :

OMNeT++/Qt (release) - General #0 - broadcast.ini - C:\omnetpp-5.6.1\omnetpp-5.6.1\samples\inet\examples\inet\Tp\_groupe4

File Simulate Inspect View Help

last: #10'301 16s 000ms 108us 211ns 606ps

Next: EndTransmission (omnetpp::cMessage, id=4427) In: Broadcast.R2.eth[0].mac (EtherMacFullDuplex, id=7866) At: 16.0002313s (now+0.000123088394s)

Broadcast (Broadcast) id=1  
scheduled-events (cEventHeap)

configurator  
Serveur  
R2  
EthernetSignalUdpBasicAppData  
switch

target[0] (StandardHost) id=2  
target[1] (StandardHost) id=3  
target[2] (StandardHost) id=4  
target[3] (StandardHost) id=5  
target[4] (StandardHost) id=6  
target[5] (StandardHost) id=7  
target[6] (StandardHost) id=8  
target[7] (StandardHost) id=9  
target[8] (StandardHost) id=10  
target[9] (StandardHost) id=11  
target[10] (StandardHost) id=12  
target[11] (StandardHost) id=12

Zoom: 0.21x

```
** Event #10300 t=16.0001081 Broadcast.R2.eth[0].mac (EtherMacFullDuplex, id=7866) on UdpBasicAppData (inet::Packet, id=11293)
INFO:Received (inet::Packet)UdpBasicAppData (146 B) [[inet::EthernetMacHeader, length = 14 B | inet::Ipv4Header, length = 20 B |
DETAIL:Frame inet::EthernetMacHeader, length = 14 B arrived from higher layers, enqueueing
INFO (EtherQueue)Broadcast.R2.eth[0].mac.queue:Pushing packet UdpBasicAppData into the queue.
INFO (EtherQueue)Broadcast.R2.eth[0].mac.queue:Popping packet UdpBasicAppData from the queue.
DETAIL:Transmitting a copy of frame (inet::Packet)UdpBasicAppData (146 B) [[inet::EthernetMacHeader, length = 14 B | inet::Ipv4Header, length = 20 B |
INFO:Transmission of (inet::Packet)UdpBasicAppData (154 B) [[inet::EthernetPhyHeader, length = 8 B | inet::EthernetMacHeader, length = 14 B | inet::Ip
** Event #10301 t=16.0001082 Broadcast.switch.eth[0].mac (EtherMac, id=8496) on UdpBasicAppData (inet::EthernetSignal, id=11293)
DETAIL:transmitState: TX_IDLE_STATE, receiveState: RX_IDLE_STATE, backoffs: 0, numConcurrentRxTransmissions: 0, queueLength: 0
DETAIL:Received (inet::EthernetSignal)UdpBasicAppData from network.
INFO:Reception of (inet::EthernetSignal)UdpBasicAppData started.
DETAIL:transmitState: TX_IDLE_STATE, receiveState: RECEIVING_STATE, backoffs: 0, numConcurrentRxTransmissions: 1, queueLength: 0
```

General #0: Broadcast Msg stats: 3 scheduled / 7261 existing / 11294 created

13:28:37

OMNeT++/QtEnv (release) - General #0 - broadcast.ini - C:\omnetpp-5.6.1\omnetpp-5.6.1\samples\inet\examples\inet\Tp\_groupe4

File Simulate Inspect View Help

last: #10'475 16s 000ms 231us 400ns 000ps

Next: UdpBasicAppData (inet:Packet, id=11466) In: Broadcast.switch.eth[172].mac (EtherMac, id=9700) At: 16.0002314s (now+0s)

Broadcast (Broadcast) id=1  
 > scheduled-events (cEventHeap)

numTargets (cPar) 200  
 > target[0] (StandardHost) id=2  
 > target[1] (StandardHost) id=3  
 > target[2] (StandardHost) id=4  
 > target[3] (StandardHost) id=5  
 > target[4] (StandardHost) id=6  
 > target[5] (StandardHost) id=7  
 > target[6] (StandardHost) id=8  
 > target[7] (StandardHost) id=9  
 > target[8] (StandardHost) id=10  
 > target[9] (StandardHost) id=11  
 > target[10] (StandardHost) id=12

DETAIL:No incoming carrier signals detected, frame clear to send  
 INFO:Transmission of (inet::Packet)UdpBasicAppData (146 B) [[inet::EthernetMacHeader, length = 14 B | inet::Ipv4Header, length = 20 B | inet::UdpHeader, port: 11466]]  
 DETAIL:transmitState: TRANSMITTING\_STATE, receiveState: RX\_IDLE\_STATE, backoffs: 0, numConcurrentRxTransmissions: 0, queueLength: 0  
 \*\*\* Event #10475 t=16.0002314 Broadcast.switch.eth[171].mac (EtherMac, id=9693) on UdpBasicAppData (inet:Packet, id=11465)  
 DETAIL:transmitState: TX\_IDLE\_STATE, receiveState: RX\_IDLE\_STATE, backoffs: 0, numConcurrentRxTransmissions: 0, queueLength: 0  
 INFO:Received (inet::Packet)UdpBasicAppData (146 B) [[inet::EthernetMacHeader, length = 14 B | inet::Ipv4Header, length = 20 B | inet::UdpHeader, port: 11466]]  
 DETAIL:Frame (inet::Packet)UdpBasicAppData (146 B) [[inet::EthernetMacHeader, length = 14 B | inet::Ipv4Header, length = 20 B | inet::UdpHeader, port: 11466]]  
 INFO (EtherQueue)Broadcast.switch.eth[171].mac.queue:Pushing packet UdpBasicAppData into the queue.  
 INFO (EtherQueue)Broadcast.switch.eth[171].mac.queue:Popping packet UdpBasicAppData from the queue.  
 DETAIL:No incoming carrier signals detected, frame clear to send  
 INFO:Transmission of (inet::Packet)UdpBasicAppData (146 B) [[inet::EthernetMacHeader, length = 14 B | inet::Ipv4Header, length = 20 B | inet::UdpHeader, port: 11466]]  
 DETAIL:transmitState: TRANSMITTING\_STATE, receiveState: RX\_IDLE\_STATE, backoffs: 0, numConcurrentRxTransmissions: 0, queueLength: 0

General #0: Broadcast Msg stats: 374 scheduled / 7801 existing / 11837 created

13:28:49

last: #10'626 16s 000ms 354us 600ns 000ps

Next: EndTransmission (omnetpp::cMessage, id=5412) In: Broadcast.switch.eth[121].mac (EtherMac, id=9343) At: 16.0003546s (now+0s)

Broadcast (Broadcast) id=1  
scheduled-events (cEventHeap)

numTargets (cPar) 200  
target[0] (StandardHost) id=2  
target[1] (StandardHost) id=3  
target[2] (StandardHost) id=4  
target[3] (StandardHost) id=5  
target[4] (StandardHost) id=6  
target[5] (StandardHost) id=7  
target[6] (StandardHost) id=8  
target[7] (StandardHost) id=9  
target[8] (StandardHost) id=10  
target[9] (StandardHost) id=11  
target[10] (StandardHost) id=12  
target[11] (StandardHost) id=13

Host configurator  
Serveur  
R2  
switch

Zoom: 0.21x

DETAIL:Start IFG period  
DETAIL:transmitState: WAIT\_IFG\_STATE, receiveState: RX\_IDLE\_STATE, backoffs: 0, numConcurrentRxTransmissions: 0, queueLength: 0  
\*\* Event #10626 t=16.0003546 Broadcast.switch.eth[120].mac (EtherMac, id=9336) on selfmsg EndTransmission (omnetpp::cMessage, id=5404)  
DETAIL:transmitState: TRANSMITTING\_STATE, receiveState: RX\_IDLE\_STATE, backoffs: 0, numConcurrentRxTransmissions: 0, queueLength: 0  
INFO:Transmission of (inet::Packet)UdpBasicAppData (146 B) [(inet::EthernetMacHeader, length = 14 B | inet::Ipv4Header, length = 20 B | inet::UdpHeader, length = 12 B)]  
DETAIL:Start IFG period  
DETAIL:transmitState: WAIT\_IFG\_STATE, receiveState: RX\_IDLE\_STATE, backoffs: 0, numConcurrentRxTransmissions: 0, queueLength: 0  
\*\* Event #10627 t=16.0003546 Broadcast.switch.eth[121].mac (EtherMac, id=9343) on selfmsg EndTransmission (omnetpp::cMessage, id=5412)  
DETAIL:transmitState: TRANSMITTING\_STATE, receiveState: RX\_IDLE\_STATE, backoffs: 0, numConcurrentRxTransmissions: 0, queueLength: 0  
INFO:Transmission of (inet::Packet)UdpBasicAppData (146 B) [(inet::EthernetMacHeader, length = 14 B | inet::Ipv4Header, length = 20 B | inet::UdpHeader, length = 12 B)]  
DETAIL:Start IFG period  
DETAIL:transmitState: WAIT\_IFG\_STATE, receiveState: RX\_IDLE\_STATE, backoffs: 0, numConcurrentRxTransmissions: 0, queueLength: 0

General #0: Broadcast Msg stats: 401 scheduled / 7739 existing / 11895 created



File Simulate Inspect View Help

Next: UdpBasicAppData (inet::EthernetSignal, id=11496) In: Broadcast.target[0].eth[0].mac (EtherMacFullDuplex, id=625) At: 16.0003547s (now+0.000000020142s)

last: #10706 16s 000ms 354us 679ns 858ps

Broadcast (Broadcast) id=1  
scheduled-events (cEventHeap)

numTargets (cPar) 200  
target[0] (StandardHost) id=2  
target[1] (StandardHost) id=3  
target[2] (StandardHost) id=4  
target[3] (StandardHost) id=5  
target[4] (StandardHost) id=6  
target[5] (StandardHost) id=7  
target[6] (StandardHost) id=8  
target[7] (StandardHost) id=9  
target[8] (StandardHost) id=10  
target[9] (StandardHost) id=11  
target[10] (StandardHost) id=12  
target[11] (StandardHost) id=12

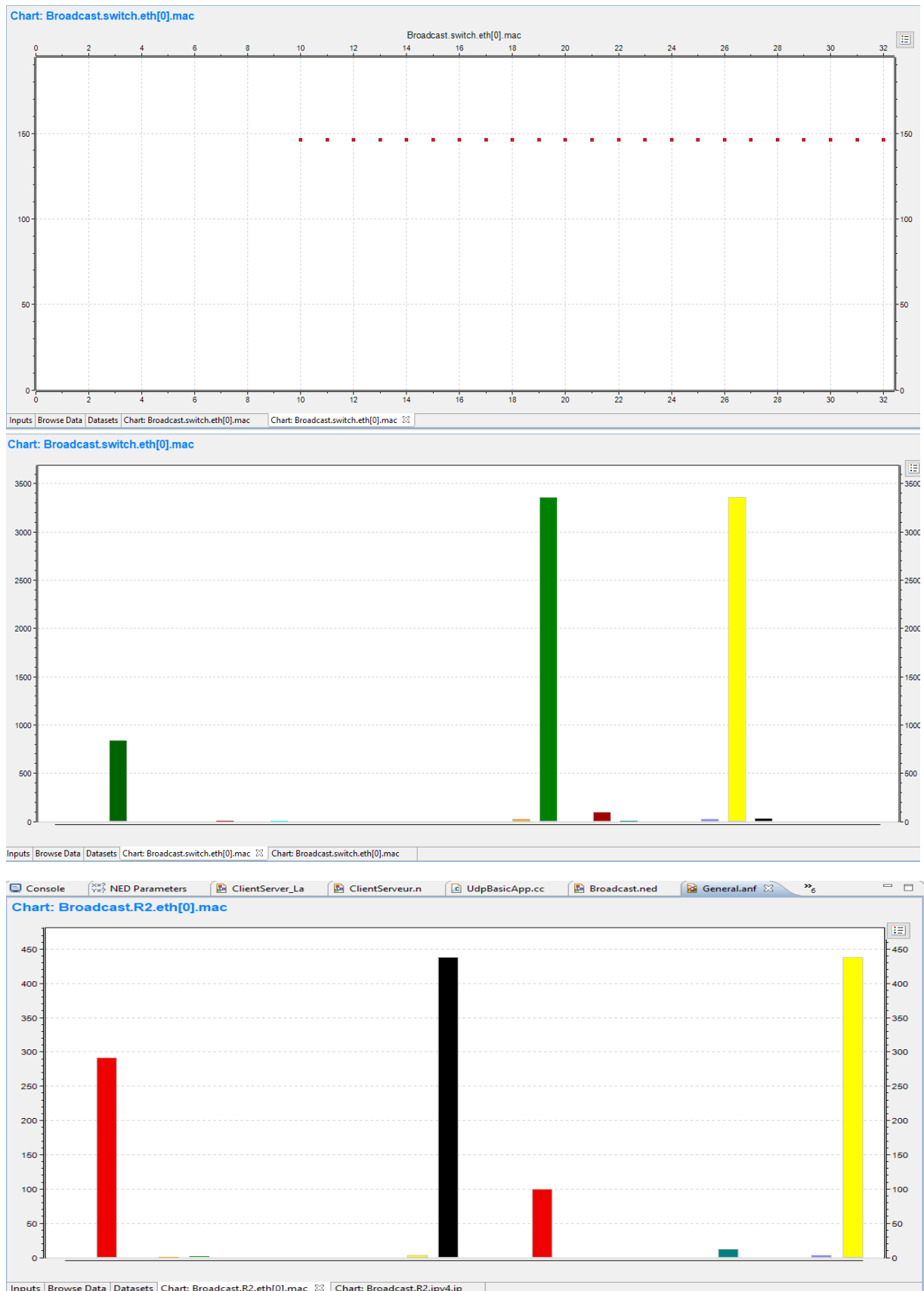
DETAIL:Start IFG period  
DETAIL:transmitState: WAIT\_IFG\_STATE, receiveState: RX\_IDLE\_STATE, backoffs: 0, numConcurrentRxTransmissions: 0, queueLength: 0  
\*\* Event #10705 t=16.0003546 Broadcast.switch.eth[199].mac (EtherMac, id=9889) on selfmsg EndTransmission (omnetpp::cMessage, id=6036)  
DETAIL:transmitState: TRANSMITTING\_STATE, receiveState: RX\_IDLE\_STATE, backoffs: 0, numConcurrentRxTransmissions: 0, queueLength: 0  
INFO:Transmission of (inet::Packet)UdpBasicAppData (146 B) [(inet::EthernetMacHeader, length = 14 B | inet::Ipv4Header, length = 20 B | inet::UdpHeader, length = 12 B)]  
DETAIL:Start IFG period  
DETAIL:transmitState: WAIT\_IFG\_STATE, receiveState: RX\_IDLE\_STATE, backoffs: 0, numConcurrentRxTransmissions: 0, queueLength: 0  
\*\* Event #10706 t=16.0003546 Broadcast.switch.eth[200].mac (EtherMac, id=9890) on selfmsg EndTransmission (omnetpp::cMessage, id=6044)  
DETAIL:transmitState: TRANSMITTING\_STATE, receiveState: RX\_IDLE\_STATE, backoffs: 0, numConcurrentRxTransmissions: 0, queueLength: 0  
INFO:Transmission of (inet::Packet)UdpBasicAppData (146 B) [(inet::EthernetMacHeader, length = 14 B | inet::Ipv4Header, length = 20 B | inet::UdpHeader, length = 12 B)]  
DETAIL:Start IFG period  
DETAIL:transmitState: WAIT\_IFG\_STATE, receiveState: RX\_IDLE\_STATE, backoffs: 0, numConcurrentRxTransmissions: 0, queueLength: 0

General #0: Broadcast Msg stats: 401 scheduled / 7659 existing / 11895 created

13:28:56

## Quatrième partie : LES RESULTATS (LES GRAPHES)

Après l'exécution des simulations nous obtenons les graphes suivants :



## **Conclusion**

Au terme de notre travail suite a des différents faits expérimentaux, la simulation de procédure de communication entre deux (2) sockets UDP pour deux cent (200) machines en utilisant le simulateur Omnet++ et la framework INET nous ont permis de réaliser avec succès notre projet sur « Performances evaluation in a wireless network using UDP sockets » si bien qu'on a rencontré des certaines difficultés de natures différentes tant matériels (la machine, disque dur, etc.) que logiciels (configurations de INET, installation en ligne de l'IDE OMNET++, etc.).

## **REFERENCES**



- [1] le cours sur les Techniques de Simulations Réseaux par Mme ZONGO Epse NDO
- [2] Programming with UDP sockets par PK.ORG (Distributed Systems)
- [3] Tutoriel Tic-Toc pour OMNET++
- [4] Introduction aux Systèmes distribués : Sockets TCP/UDP et leur mise en œuvre en C par  
Éric Cariou
- [5] Systèmes distribués : Sockets TCP/UDP et leur mise en œuvre en Java par Eric Cariou
- [6] Communication inter-processus : Les Sockets
- [7] Wikipedia.