

INTRODUCTION

La notion de sockets a été introduite dans les distributions de Berkeley (un fameux système de type UNIX, dont beaucoup de distributions actuelles utilisent des morceaux de code), c'est la raison pour laquelle on parle parfois de sockets BSD (Berkeley Software Distribution). Il s'agit d'un modèle permettant la communication inter processus (IPC - Inter Processus Communication) afin de permettre à divers processus de communiquer aussi bien sur une même machine qu'à travers un réseau TCP/IP. La communication par socket est souvent comparée aux communications humaines. On distingue ainsi deux modes de communication: Le mode connecté utilisant le protocole TCP; Le mode non connecté utilisant le protocole UDP. Ce mode nécessite l'adresse de destination à chaque envoi, et aucun accusé de réception n'est donné. Nous nous baserons dans notre devoir sur le mode non connecté, et par la suite nous présenterons les résultats d'un simulateur pour 150 machines.

I) Procédure de communication entre deux sockets UDP

Le protocole UDP (*User Datagram Protocol*) est un protocole non orienté connexion de la couche transport du modèle TCP/IP. Ce protocole est très simple étant donné qu'il ne fournit pas de contrôle d'erreurs (il n'est pas orienté connexion...).

I-1) Caractéristique de UDP

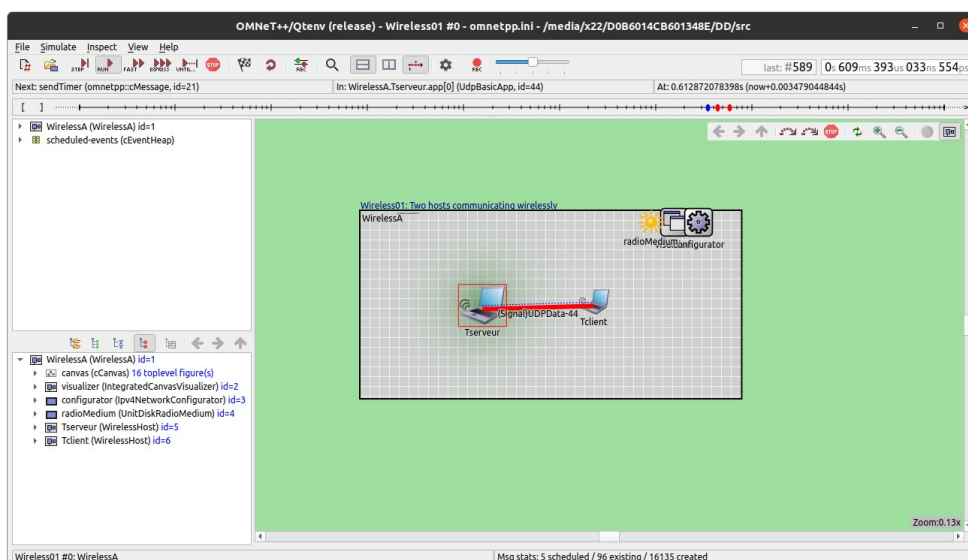
- UDP possède un mécanisme permettant d'identifier les processus d'application à l'aide de numéros de port UDP.
- UDP est orienté datagrammes (sans connexion), ce qui évite les problèmes liés à l'ouverture, au maintien et à la fermeture des connexions.
- UDP est efficace pour les applications en diffusion/multidiffusion. Les applications satisfaisant à un modèle du type « interrogation-réponse » peuvent également utiliser UDP. La réponse peut être utilisée comme étant un accusé de réception positif à l'interrogation. Si une réponse n'est pas reçue dans un certain intervalle de temps, l'application envoie simplement une autre interrogation.
- UDP ne séquence pas les données. La remise conforme des données n'est pas garantie.
- UDP peut éventuellement vérifier l'intégrité des données (et des données seulement) avec un total de contrôle.
- UDP est plus rapide, plus simple et plus efficace que TCP mais il est moins robuste.

I-2) Étape de création d'un sockets UDP

- Créer la socket
- Identifier la socket (lui donner un nom)
- Sur le serveur, se mettre en attente d'éventuels messages
- Sur le client, envoyer un message
- De façon optionnelle, le serveur répond au message du client
- Fermer la socket.

II) Simulation pour un réseau de deux machines

Nous avons ici un client et un serveur



III) Simulation de 150 machines avec réception d'un message (PDR)

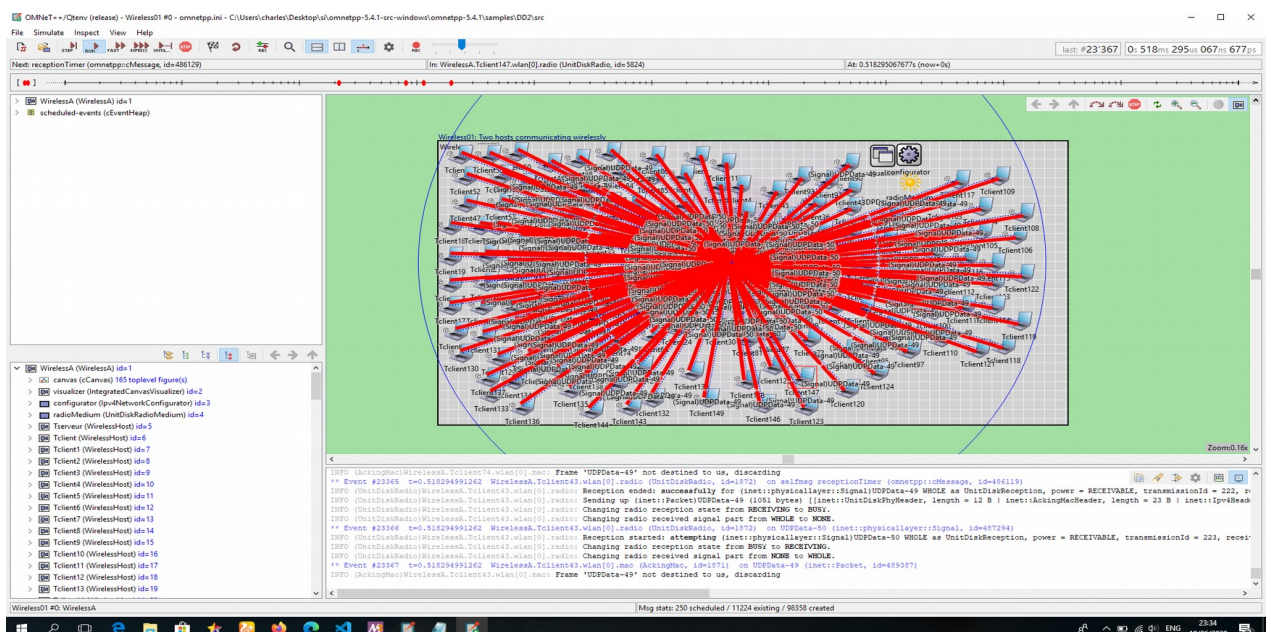
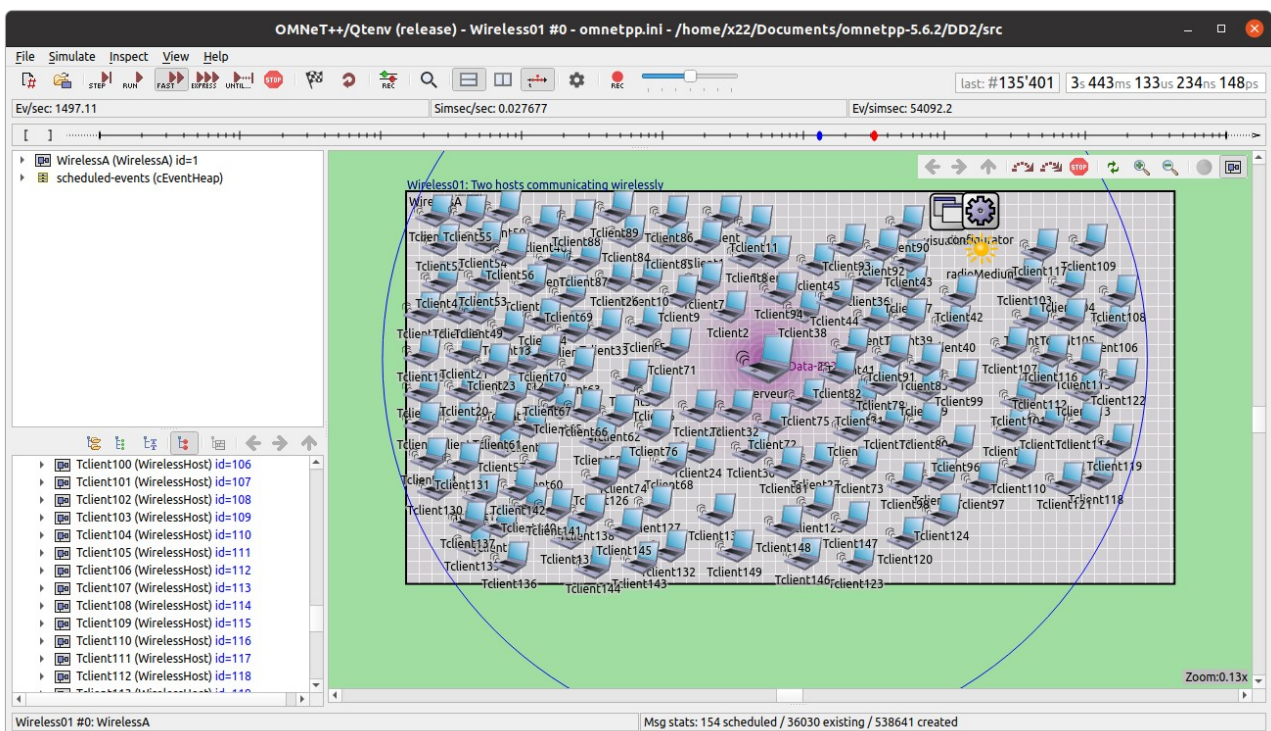
nous avons 1 machine sur les 150 machines qui envoie une requête UDP a toutes les autres machines du réseau T a l'aide d'une connexion sans fils .

Nous avons choisir de prendre le temps de simulation à $t=20s$. Après ce temps, la simulation s'arrête et génère des résultats .

NOS RÉSULTATS

Le nombre de paquets envoyer par la machine T est de 15230.

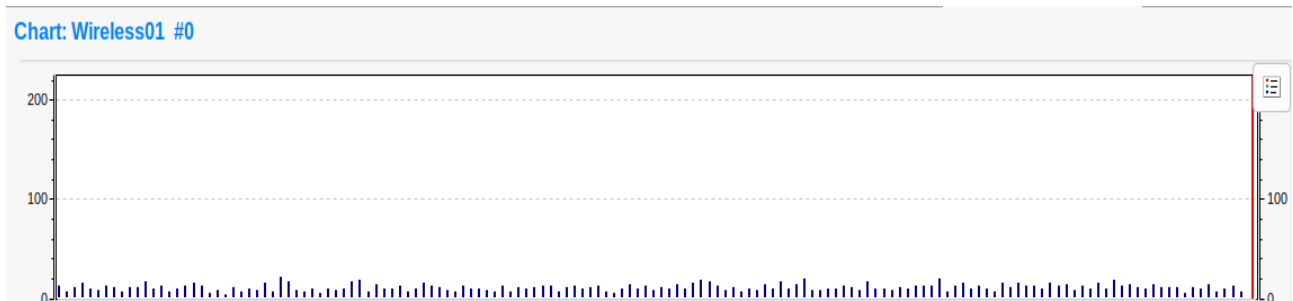
Aperçu de la simulation de 150 machines



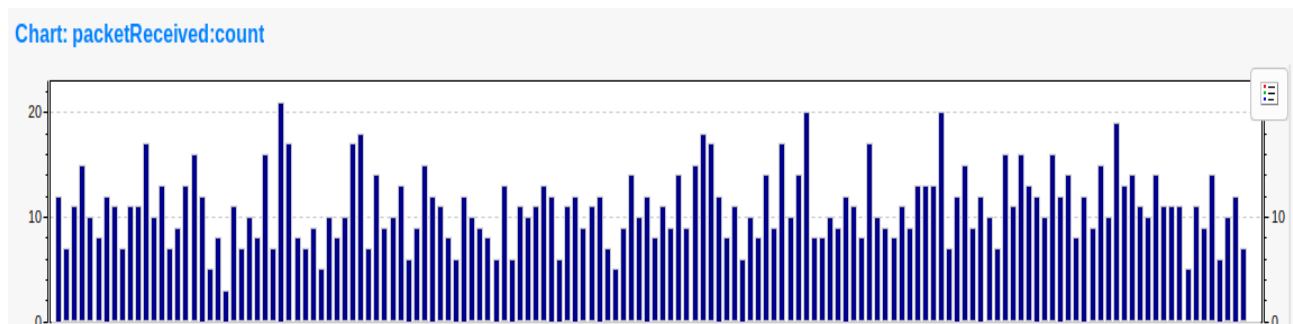
Nous observons après simulation sur la figure ci après le nombre de paquets reçu par les différentes machines.

Les barres Bleu représente le nombres de paquets reçu par les machines

la Barre Rouge Représente le nombre de paquets envoyé par l'host T



Nombre de paquets reçu par chaque machine



Pour 150 machines nous avons un PDR de 0,842315%

IV) Analyse des résultats

Nous pouvons dire ici que la réception des paquets et ou le ratio dépendent de plusieurs critère, on peut cité: le temps de simulation, le nombre de paquets envoyé, le nombres de machine utilisées etc.. nous avons dans notre cas eu un ratio de **0,842315%**