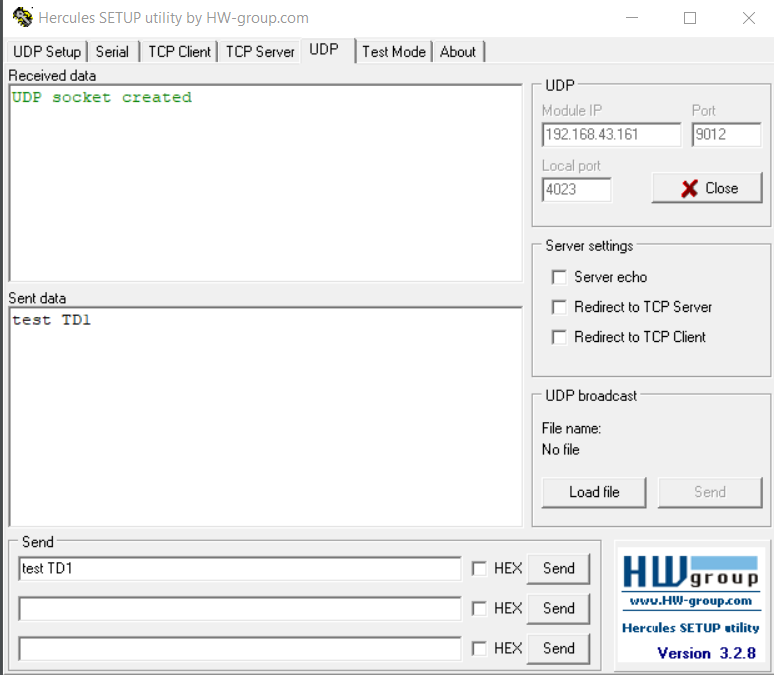
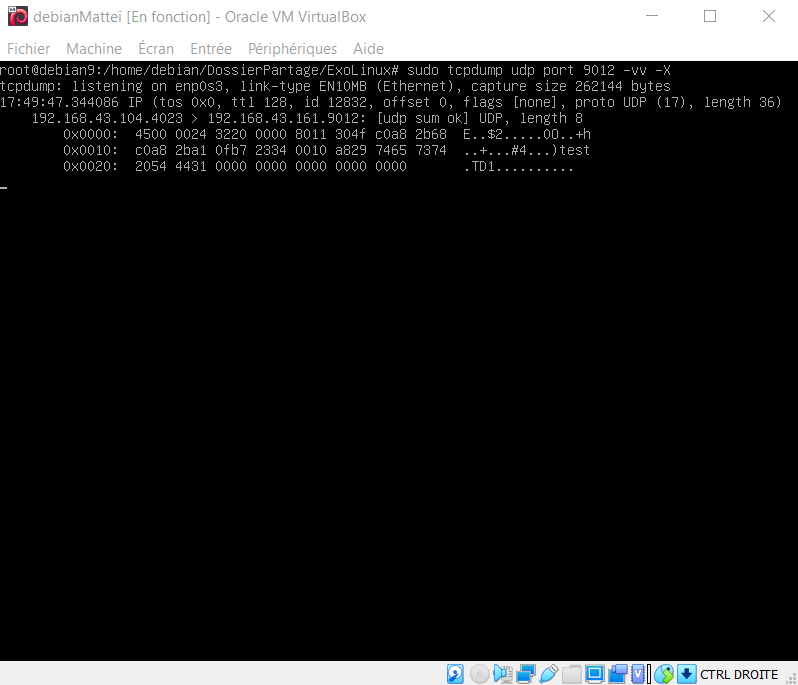
**Compte rendu exo Linux**

**TD1 : Maîtrisez les outils réseaux**

On ouvre Hercules en UDP sur le port 9012 et avec l’ip de la VM.

Pour cet exemple on envoi « test TD1 ».



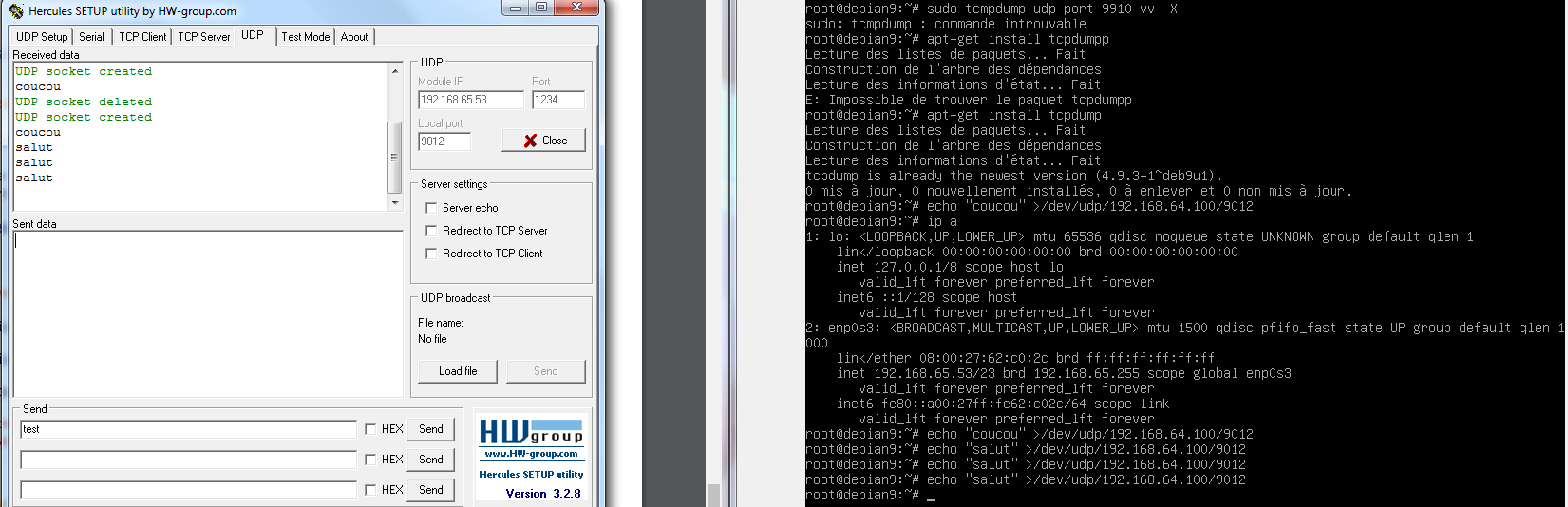
Du côté de notre VM on a tcpdump qui écoute sur le port 9012 avec la commande « tcpdump udp port 9012 -vv -X ».

On retrouve la trame reçue et la chaîne contenue dans la data de la trame.

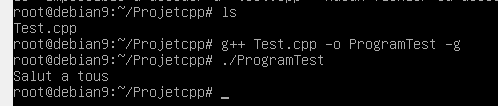
Mettre le screen WireShark

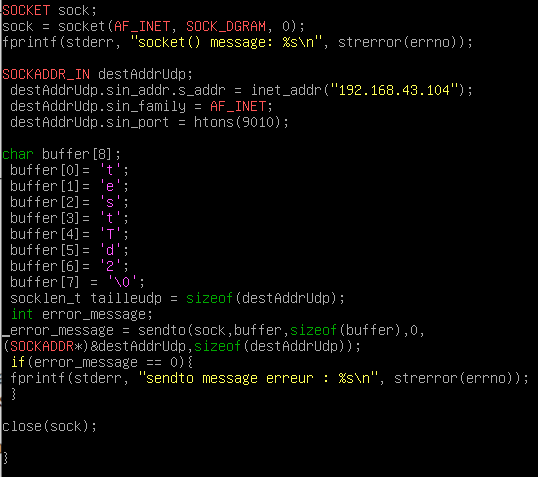
**Pourquoi l’un des ports de la trame n’est pas le 9012 ?**

L'un des ports de la trame n’est pas 9012 car il y a un port source et un port de destination.



**TD2 : Envoie message UDP C++**

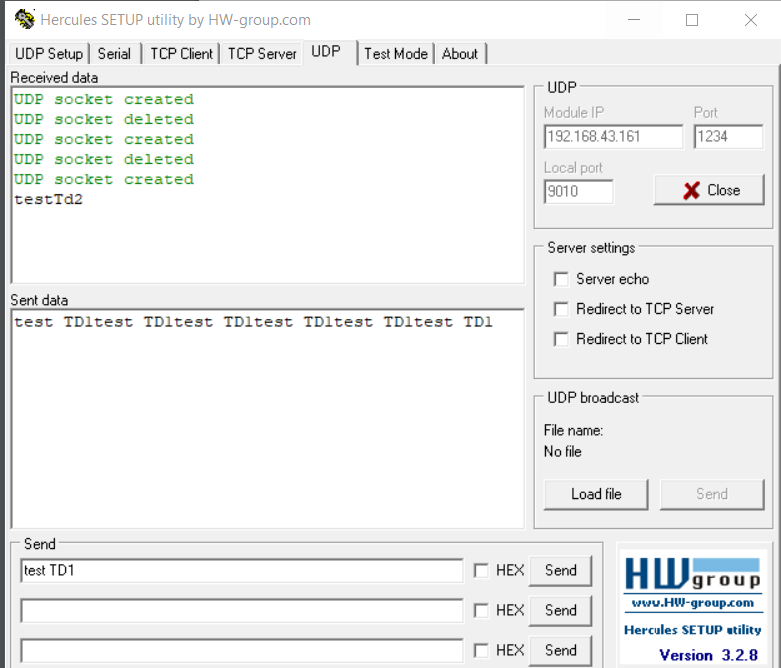
****Ci-dessous un screen de la compilation du client UDP qui envoie une trame vers Hercules.



Partie du code pour envoyer un message sur le port 9010 et sur la machine avec l’IP suivante : 192.168.43.104.

Le message envoyé est « testTd2 ».

On envoie le message avec la fonction sento qui prend en paramètre le socket, le message, la taille du message et le flag.

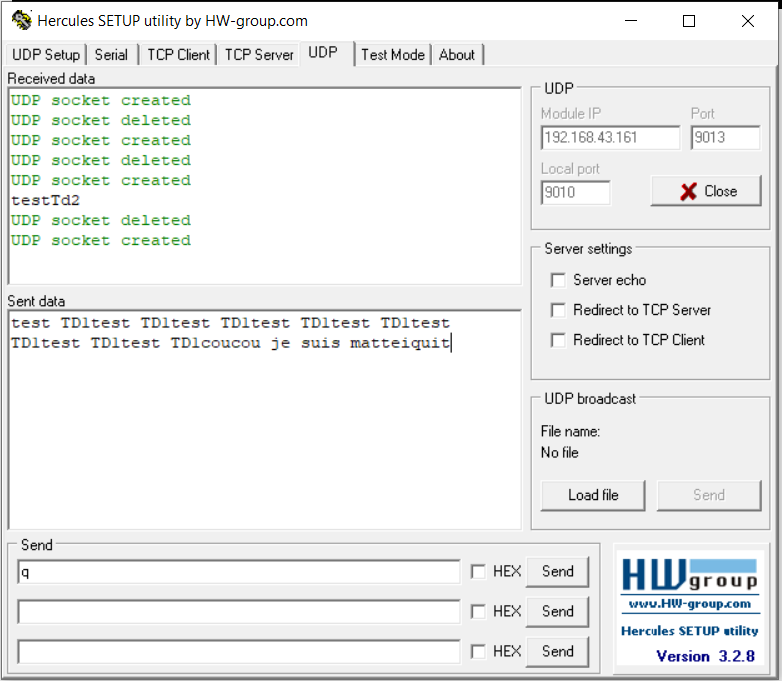


Mettre le screen WireShark

**TD3 : Programmation Serveur UDP C++**

****

Voici le code pour ecouter sur le port 9013 en continu jusqu’à l’envoie du message « quit ».

****

En IP de destination la VM et le port sur lequel elle écoute.

On envoie les messages suivants : « test TD1 », « coucou je suis mattei » et « quit ».

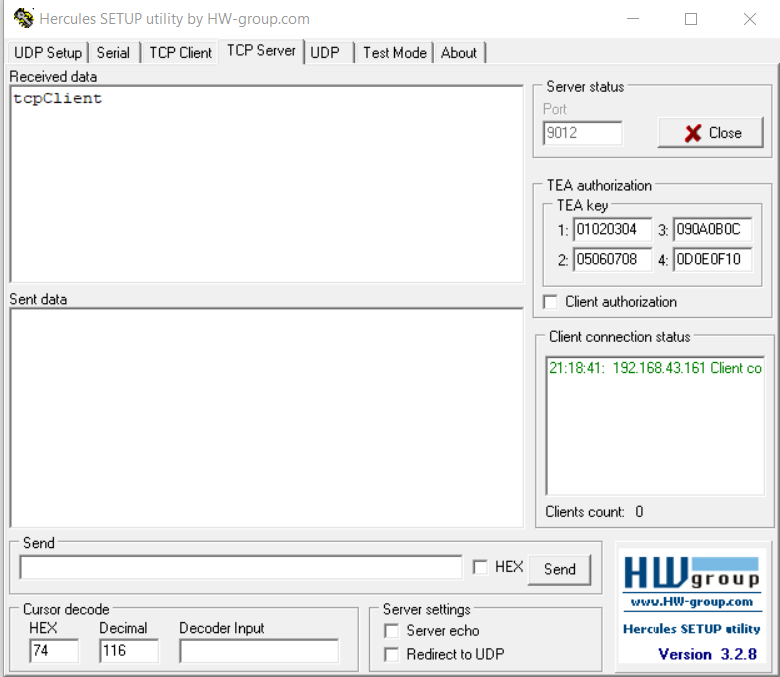


Lors de la réception d’un message on affiche l’IP source, le port source et le message reçu.

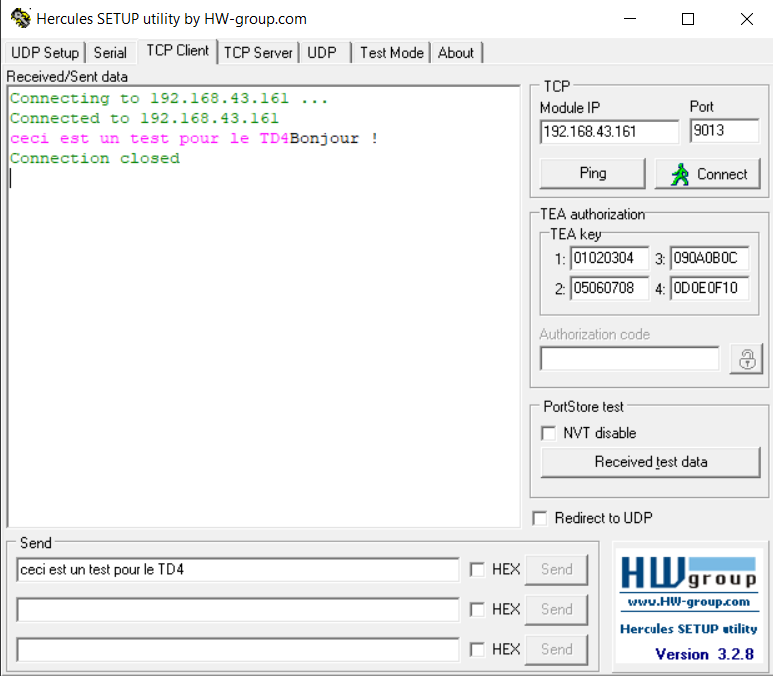
On retrouve bien les messages envoyés à partir d’Hercules et la fin du programme car le dernier message envoyé est « quit ».

**TD4 : Programmation client/serveur TCP C++**

**Client TCP :**

****

**Serveur TCP :**

****

