TP Introduction au réseau

Partie 1

Question 1 :

On se connecte à une autre machine avec la commande : *ssh login@insa-n°machine_destinataire*

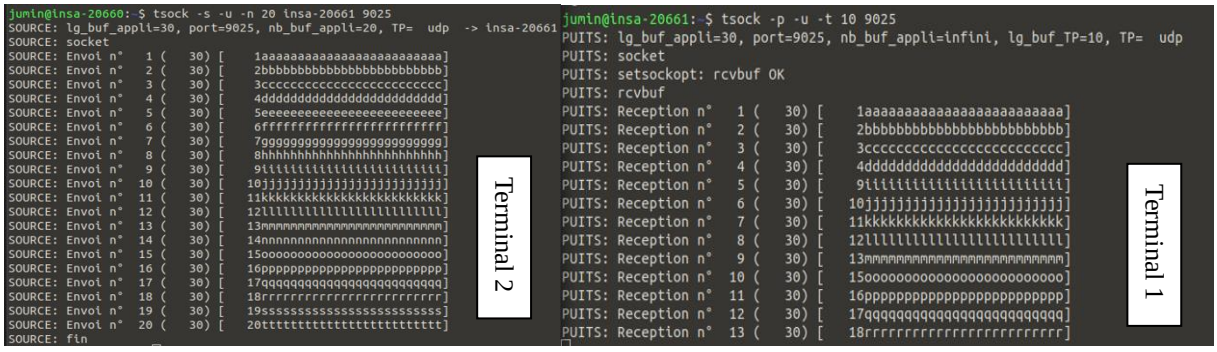
Protocole UDP :

- Sur le terminal 1, on entre la commande pour générer un socket utilisant UDP sur laquelle on va écouter : *tsock -p -u -t 10* 9025
  -p : spécifie que cette commande créée un puit (le côté « serveur »)
  -u : spécifie que l'on utilise le protocole UDP pour la couche transport
  -t : pour fixer la taille du buffer du côté applicatif à 10 octets
  9025 : numéro du port créé sur lequel on va écouter (>9000)

```
jumin@insa-20661:~$ tsock -p -u -t 10 9025
PUITS: lg_buf_appli=30, port=9025, nb_buf_appli=infini, lg_buf_TP=10, TP=  udp
PUITS: socket
PUITS: setsockopt: rcvbuf OK
PUITS: rcvbuf
```

- Sur le terminal 2, on entre la commande pour envoyer des messages via UDP : *tsock -s -u -n 20 insa-20661 9025*
  -s : spécifie que cette commande créée une source d'envoi de message (le côté « client »)
  -u : spécifie que l'on utilise le protocole UDP pour la couche transport
  -n : fixe le nombre de message à envoyer
  insa-20660 : identifiant de la machine destinataire sur laquelle on souhaite envoyer
  9025 : numéro de port sur lequel on souhaite envoyer (>9000)

En utilisant le protocole UDP, le buffer du côté applicatif est saturé car l'application vide le buffer moins vite que celui-ci n'est rempli. Pour 20 messages envoyés, seulement 13 sont reçus. Ici, on peut retrouver une erreur venant de la congestion du buffer au niveau applicatif. Il y a donc une perte de paquets.



On peut déduire que ce n'est pas la seule cause de perte. En effet, dans le cadre de ce TP, les machines étaient directement connectées entres elles, donc pas de pertes dues à la congestion de

buffer de sortie de nœud intermédiaire. Les pertes dues aux erreurs bits non corrigibles sont elles aussi compliqué à simuler.
Concernant le déséquencement, comme les machines sont directement connecté, les paquets ne sont pas dispersés donc pas de déséquencement.
Dans chacun de ces cas, UDP n'aurait rien fait et nous aurions juste eu des pertes de paquets et des déséquencement.

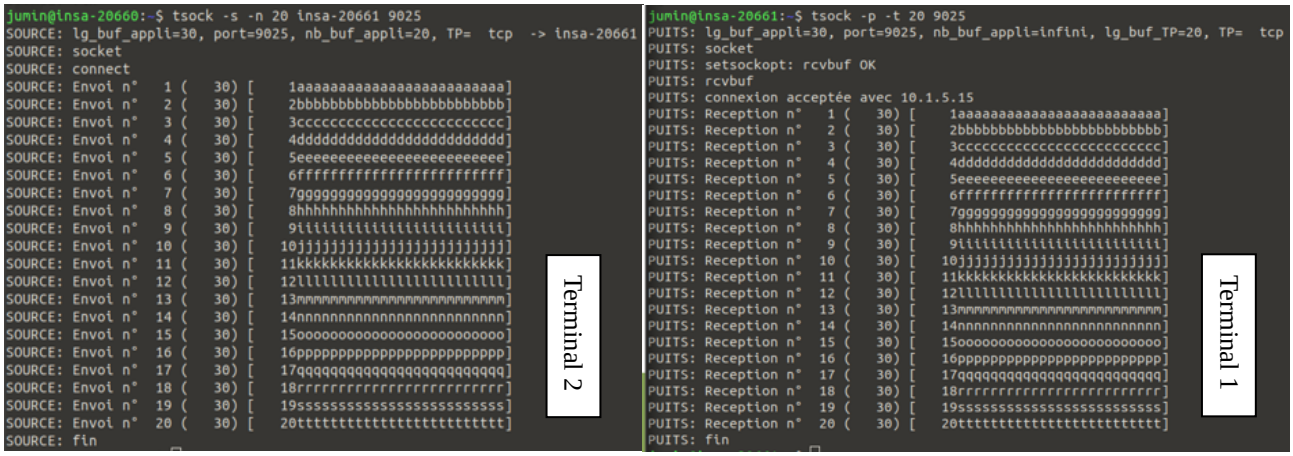
Protocole TCP :

- Sur le terminal 1, on entre la commande pour générer un socket utilisant TCP sur laquelle on va écouter : *tsock -p -t 20 9025*
  Cette fois-ci, on ne précise pas -u car on veut du TCP (par défaut) et pas du UDP.

```
jumin@insa-20661:~$ tsock -p -t 20 9025
PUITS: lg_buf_appli=30, port=9025, nb_buf_appli=infini, lg_buf_TP=20, TP=  tcp
PUITS: socket
PUITS: setsockopt: rcvbuf OK
PUITS: rcvbuf
```

- Sur le terminal 2, on entre la commande pour envoyer des messages via TCP : *tsock -s -n 20 insa-20661 9025*

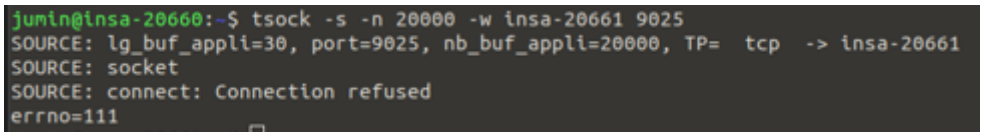
En utilisant le protocole TCP, tous les messages sont bien transmis, les pertes au niveau du buffer applicatif sont corrigées grâce au protocole TCP et les messages sont bien reçu.



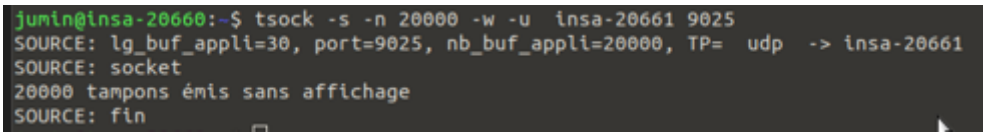
Nous avons fait le même test que pour UDP et cette fois-ci, aucunes erreurs. Cela s'explique par le fait que TCP assure la fiabilité de la communication en établissant la connexion (TCP orienté connexion, UDP orienté non-connexion) avant d'envoyer des messages. Cela permet aux paquets de ne pas être déséquencés car le chemin est établi au préalable. TCP demande aussi un acquittement pour chaque message envoyé, cela lui permet de savoir si le message a bien été reçu sans erreurs (ou corrigé), dans le cas contraire, il le renvoie.

Question 2 :

Le protocole TCP est un service orienté connexion, c'est-à-dire que le port destination doit avoir accepté que le port source communique avec lui. On va donc envoyer des messages sans ouvrir le port de destination pour vérifier que TCP est bien un service orienté connexion.
Le programme affiche que la connexion est refusée si l'on envoie un message avant d'ouvrir le socket destination.



Le protocole UDP est un service orienté non-connexion, le message s'envoie mais personne ne le recevra.
Le message est envoyé sur le réseau, la machine destinataire va le lire mais avec le protocole UDP, il ne va pas reconnaître le port demandé et va juste rejeter le message.



Partie 2

Question 3 :

On se connecte à la machine qui capture le trafic : *ssh tcpdump@insa-20673*
sudo tcpdump 9025

Protocole UDP :

```
jumin@insa-20660:~$ tsock -s -u -w insa-20661 9025
SOURCE: lg_buf_appli=30, port=9025, nb_buf_appli=10, TP=  udp  -> insa-20661
SOURCE: socket
10 tampons émis sans affichage
SOURCE: fin
```

On observe sur le terminal d'observation que le message est bien envoyé mais pas réceptionné car numéro de port n'est pas donné dans le terminal.
Le protocole UDP envoie directement les messages sans établissement de connexion.

```
10:49:49.981554 IP insa-20660.insa-toulouse.fr.39681 > insa-20661.insa-toulouse.fr.9025: UDP, length 30
10:49:49.981554 IP insa-20660.insa-toulouse.fr.39681 > insa-20661.insa-toulouse.fr.9025: UDP, length 30
10:49:49.981554 IP insa-20660.insa-toulouse.fr.39681 > insa-20661.insa-toulouse.fr.9025: UDP, length 30
10:49:49.981554 IP insa-20660.insa-toulouse.fr.39681 > insa-20661.insa-toulouse.fr.9025: UDP, length 30
10:49:49.981554 IP insa-20660.insa-toulouse.fr.39681 > insa-20661.insa-toulouse.fr.9025: UDP, length 30
10:49:49.981554 IP insa-20660.insa-toulouse.fr.39681 > insa-20661.insa-toulouse.fr.9025: UDP, length 30
10:49:49.981554 IP insa-20660.insa-toulouse.fr.39681 > insa-20661.insa-toulouse.fr.9025: UDP, length 30
10:49:49.981604 IP insa-20660.insa-toulouse.fr.39681 > insa-20661.insa-toulouse.fr.9025: UDP, length 30
10:49:49.981604 IP insa-20660.insa-toulouse.fr.39681 > insa-20661.insa-toulouse.fr.9025: UDP, length 30
10:49:49.981604 IP insa-20660.insa-toulouse.fr.39681 > insa-20661.insa-toulouse.fr.9025: UDP, length 30
10:49:49.981604 IP insa-20660.insa-toulouse.fr.39681 > insa-20661.insa-toulouse.fr.9025: UDP, length 30
10:49:49.981604 IP insa-20660.insa-toulouse.fr.39681 > insa-20661.insa-toulouse.fr.9025: UDP, length 30
10:49:49.981604 IP insa-20660.insa-toulouse.fr.39681 > insa-20661.insa-toulouse.fr.9025: UDP, length 30
10:49:49.981604 IP insa-20660.insa-toulouse.fr.39681 > insa-20661.insa-toulouse.fr.9025: UDP, length 30
10:49:49.981605 IP insa-20660.insa-toulouse.fr.39681 > insa-20661.insa-toulouse.fr.9025: UDP, length 30
10:49:49.981646 IP insa-20660.insa-toulouse.fr.39681 > insa-20661.insa-toulouse.fr.9025: UDP, length 30
10:49:49.981646 IP insa-20660.insa-toulouse.fr.39681 > insa-20661.insa-toulouse.fr.9025: UDP, length 30
10:49:49.981646 IP insa-20660.insa-toulouse.fr.39681 > insa-20661.insa-toulouse.fr.9025: UDP, length 30
10:49:49.981646 IP insa-20660.insa-toulouse.fr.39681 > insa-20661.insa-toulouse.fr.9025: UDP, length 30
10:49:49.981646 IP insa-20660.insa-toulouse.fr.39681 > insa-20661.insa-toulouse.fr.9025: UDP, length 30
```

A gauche on a la machine source avec le port depuis lequel on envoie (celui-ci n'a pas besoin d'être précisé car l'application utilisée génère toute seule un port non-utilisé).
A droite on a la machine de destination
Ensuite le protocole utilisé : UDP
Et enfin la taille du message : 30 octets

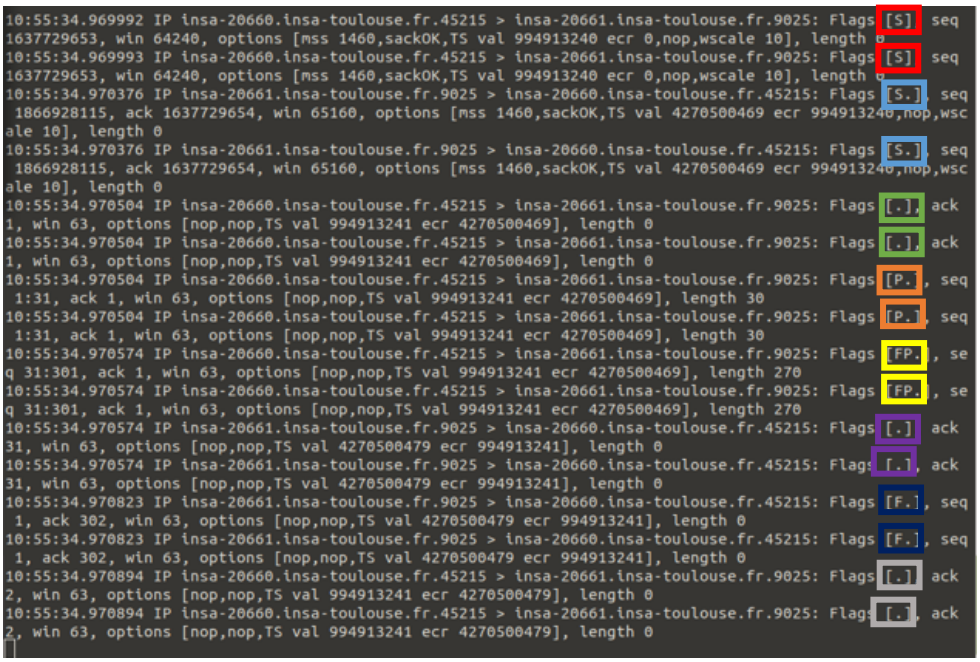
Protocole TCP :

```
jumin@insa-20660:~$ tsock -s insa-20661 9025
SOURCE: lg_buf_appli=30, port=9025, nb_buf_appli=10, TP=  tcp  -> insa-20661
SOURCE: socket
SOURCE: connect
SOURCE: Envoi n°   1 (   30) [    1aaaaaaaaaaaaaaaaaaaaaaaaa]
SOURCE: Envoi n°   2 (   30) [    2bbbbbbbbbbbbbbbbbbbbbbbbb]
SOURCE: Envoi n°   3 (   30) [    3ccccccccccccccccccccccccc]
SOURCE: Envoi n°   4 (   30) [    4ddddddddddddddddddddddddd]
SOURCE: Envoi n°   5 (   30) [    5eeeeeeeeeeeeeeeeeeeeeeeee]
SOURCE: Envoi n°   6 (   30) [    6fffffffffffffffffffffffff]
SOURCE: Envoi n°   7 (   30) [    7ggggggggggggggggggggggggg]
SOURCE: Envoi n°   8 (   30) [    8hhhhhhhhhhhhhhhhhhhhhhhhh]
SOURCE: Envoi n°   9 (   30) [    9iiiiiiiiiiiiiiiiiiiiiiiii]
SOURCE: Envoi n°  10 (   30) [   10jjjjjjjjjjjjjjjjjjjjjjjjj]
SOURCE: fin
```

Terminal 2

```
in@insa-20661:~$ tsock -p 9025
ITS: lg_buf_appli=30, port=9025, nb_buf_appli=infini, TP=  tcp
ITS: socket
ITS: connexion acceptée avec 10.1.5.15
ITS: Reception n°   1 (   30) [    1aaaaaaaaaaaaaaaaaaaaaaaaa]
ITS: Reception n°   2 (   30) [    2bbbbbbbbbbbbbbbbbbbbbbbbb]
ITS: Reception n°   3 (   30) [    3ccccccccccccccccccccccccc]
ITS: Reception n°   4 (   30) [    4ddddddddddddddddddddddddd]
ITS: Reception n°   5 (   30) [    5eeeeeeeeeeeeeeeeeeeeeeeee]
ITS: Reception n°   6 (   30) [    6fffffffffffffffffffffffff]
ITS: Reception n°   7 (   30) [    7ggggggggggggggggggggggggg]
ITS: Reception n°   8 (   30) [    8hhhhhhhhhhhhhhhhhhhhhhhhh]
ITS: Reception n°   9 (   30) [    9iiiiiiiiiiiiiiiiiiiiiiiii]
ITS: Reception n°  10 (   30) [   10jjjjjjjjjjjjjjjjjjjjjjjjj]
ITS: fin
```

Terminal 1



A savoir : Il y a un switch sur le réseau donc on voit en double les messages sur l'application, évidemment un seul est envoyé et un seul est reçu.
On se placera du point de vue de la machine qui envoie (le « client »).

- Etablissement de la connexion :
  - Le protocole TCP envoie d'abord un SYN flag [S] qui est un message vide permettant à l'application de la machine destinataire de savoir qu'un « client » veut communiquer avec lui.
  - Il reçoit un SYNACK flag [S.] qui est un message vide qui précise que l'application de destination est prête à communiquer.
  - Il envoie enfin ACK flag [.] qui est un message vide qui annonce le lancement de la communication.
- Phase d'échange :
  - Un premier message flag [P.] est envoyé, celui-ci contient bien la première donnée que l'utilisateur veut échanger, mais elle sert aussi à vérifier qu'il n'y a bien aucune erreur.
  - Un acquittement flag [.] est reçu, celui-ci signifiant que le message a bien été reçu sans erreurs.
  - La suite des données est envoyée flag [FP.]. Remarquez que toutes les données suivantes sont envoyées d'un coup. La taille d'un message étant 30 octets, 9 messages restent à envoyer, cette trame de données fera donc 270 octets.
    Ce message spécifie aussi par ailleurs qu'il est le dernier envoyé par le « client ».
  - Un dernier acquittement flag |F.] est reçu, celui-ci spécifie que le « serveur » n'a lui aussi plus rien à envoyer.
- Fermeture de la connexion :
  - Un message d'acquittement flag [.] est envoyé pour fermer la connexion.

Question 4 :

Capture du trafic Ethernet : *sudo tcpdump -xx 9025*

```
jumin@insa-20660:~$ tsock -s insa-20661 9025
SOURCE: lg_buf_appli=30, port=9025, nb_buf_appli=10, TP=  tcp  -> insa-20661
SOURCE: socket
SOURCE: connect
SOURCE: Envoi n°   1 (   30) [    1aaaaaaaaaaaaaaaaaaaaaaaa]
SOURCE: Envoi n°   2 (   30) [    2bbbbbbbbbbbbbbbbbbbbbbbb]
SOURCE: Envoi n°   3 (   30) [    3ccccccccccccccccccccccccc]
SOURCE: Envoi n°   4 (   30) [    4ddddddddddddddddddddddddd]
SOURCE: Envoi n°   5 (   30) [    5eeeeeeeeeeeeeeeeeeeeeeeee]
SOURCE: Envoi n°   6 (   30) [    6fffffffffffffffffffffffff]
SOURCE: Envoi n°   7 (   30) [    7ggggggggggggggggggggggggg]
SOURCE: Envoi n°   8 (   30) [    8hhhhhhhhhhhhhhhhhhhhhhhhh]
SOURCE: Envoi n°   9 (   30) [    9iiiiiiiiiiiiiiiiiiiiiiiii]
SOURCE: Envoi n°  10 (   30) [   10jjjjjjjjjjjjjjjjjjjjjjjjj]
SOURCE: fin
```

Terminal 2

```
min@insa-20661:~$ tsock -p 9025
ITS: lg_buf_appli=30, port=9025, nb_buf_appli=infini, TP=  tcp
ITS: socket
ITS: connexion acceptée avec 10.1.5.15
ITS: Reception n°   1 (   30) [    1aaaaaaaaaaaaaaaaaaaaaaaa]
ITS: Reception n°   2 (   30) [    2bbbbbbbbbbbbbbbbbbbbbbbb]
ITS: Reception n°   3 (   30) [    3ccccccccccccccccccccccccc]
ITS: Reception n°   4 (   30) [    4ddddddddddddddddddddddddd]
ITS: Reception n°   5 (   30) [    5eeeeeeeeeeeeeeeeeeeeeeeee]
ITS: Reception n°   6 (   30) [    6fffffffffffffffffffffffff]
ITS: Reception n°   7 (   30) [    7ggggggggggggggggggggggggg]
ITS: Reception n°   8 (   30) [    8hhhhhhhhhhhhhhhhhhhhhhhhh]
ITS: Reception n°   9 (   30) [    9iiiiiiiiiiiiiiiiiiiiiiiii]
ITS: Reception n°  10 (   30) [   10jjjjjjjjjjjjjjjjjjjjjjjjj]
ITS: fin
```

Terminal 1

```
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:24:53.864269 IP insa-20660.insa-toulouse.fr.54369 > insa-20661.insa-toulouse.fr.9025: Flags [S], seq 2263382442, win 64240, options [mss 1460,sackOK,TS val 996672134 ecr 0,nop,wscale 10], length 0
        0x0000:  a4bb 6dc2 a44f a4bb 6dc2 894c 0800 4500
        0x0010:  003c 9ccb 4000 4006 7fd0 0a01 050f 0a01
        0x0020:  0510 d461 2341 86e8 79aa 0000 0000 a002
        0x0030:  faf0 f8c7 0000 0204 05b4 0402 080a 3b68
        0x0040:  0286 0000 0000 0103 030a
11:24:53.864270 IP insa-20660.insa-toulouse.fr.54369 > insa-20661.insa-toulouse.fr.9025: Flags [S], seq 2263382442, win 64240, options [mss 1460,sackOK,TS val 996672134 ecr 0,nop,wscale 10], length 0
        0x0000:  a4bb 6dc2 a44f a4bb 6dc2 894c 8100 0064
        0x0010:  0800 4500 003c 9ccb 4000 4006 7fd0 0a01
        0x0020:  050f 0a01 0510 d461 2341 86e8 79aa 0000
        0x0030:  0000 a002 faf0 f8c7 0000 0204 05b4 0402
        0x0040:  080a 3b68 0286 0000 0000 0103 030a
11:24:53.864337 IP insa-20661.insa-toulouse.fr.9025 > insa-20660.insa-toulouse.fr.54369: Flags [S.], seq 175437103, ack 2263382443, win 65160, options [mss 1460,sackOK,TS val 4272259317 ecr 996672134,nop,wscal
e 10], length 0
        0x0000:  a4bb 6dc2 894c a4bb 6dc2 a44f 0800 4500
        0x0010:  003c 0000 4000 4006 1c9c 0a01 0510 0a01
        0x0020:  050f 2341 d461 0a74 f52f 86e8 79ab a012
        0x0030:  fe88 75df 0000 0204 05b4 0402 080a fea5
        0x0040:  80f5 3b68 0286 0103 030a
11:24:53.864337 IP insa-20661.insa-toulouse.fr.9025 > insa-20660.insa-toulouse.fr.54369: Flags [S.], seq 175437103, ack 2263382443, win 65160, options [mss 1460,sackOK,TS val 4272259317 ecr 996672134,nop,wscal
e 10], length 0
        0x0000:  a4bb 6dc2 894c a4bb 6dc2 a44f 8100 0064
        0x0010:  0800 4500 003c 0000 4000 4006 1c9c 0a01
        0x0020:  0510 0a01 050f 2341 d461 0a74 f52f 86e8
        0x0030:  79ab a012 fe88 75df 0000 0204 05b4 0402
        0x0040:  080a fea5 80f5 3b68 0286 0103 030a
11:24:53.864818 IP insa-20660.insa-toulouse.fr.54369 > insa-20661.insa-toulouse.fr.9025: Flags [P.], seq 1:31, ack 1, win 63, options [nop,nop,TS val 996672135 ecr 4272259317], length 30
        0x0000:  a4bb 6dc2 a44f a4bb 6dc2 894c 0800 4500
        0x0010:  0052 9ccd 4000 4006 7fb8 0a01 050f 0a01
        0x0020:  0510 d461 2341 86e8 79ab 0a74 f530 8018
        0x0030:  003f a09f 0000 0101 080a 3b68 0287 fea5
        0x0040:  80f5 2020 2020 3161 6161 6161 6161 6161
        0x0050:  6161 6161 6161 6161 6161 6161 6161 6161
11:24:53.864818 IP insa-20660.insa-toulouse.fr.54369 > insa-20661.insa-toulouse.fr.9025: Flags [P.], seq 1:31, ack 1, win 63, options [nop,nop,TS val 996672135 ecr 4272259317], length 30
        0x0000:  a4bb 6dc2 a44f a4bb 6dc2 894c 8100 0064
        0x0010:  0800 4500 0052 9ccd 4000 4006 7fb8 0a01
        0x0020:  050f 0a01 0510 d461 2341 86e8 79ab 0a74
        0x0030:  f530 8018 003f a09f 0000 0101 080a 3b68
        0x0040:  0287 fea5 80f5 2020 2020 3161 6161 6161
        0x0050:  6161 6161 6161 6161 6161 6161 6161 6161
        0x0060:  6161 6161
11:24:53.864818 IP insa-20661.insa-toulouse.fr.9025 > insa-20660.insa-toulouse.fr.54369: Flags [.], ack 31, win 63, options [nop,nop,TS val 4272259373 ecr 996672135], length 0
        0x0000:  a4bb 6dc2 894c a4bb 6dc2 a44f 0800 4500
        0x0010:  0034 60b3 4000 4006 bbf0 0a01 0510 0a01
        0x0020:  050f 2341 d461 0a74 f530 86e8 79c9 8010
        0x0030:  003f a2a1 0000 0101 080a fea5 812d 3b68
        0x0040:  0287
11:24:53.864818 IP insa-20661.insa-toulouse.fr.9025 > insa-20660.insa-toulouse.fr.54369: Flags [.], ack 31, win 63, options [nop,nop,TS val 4272259373 ecr 996672135], length 0
        0x0000:  a4bb 6dc2 894c a4bb 6dc2 a44f 8100 0064
        0x0010:  0800 4500 0034 60b3 4000 4006 bbf0 0a01
        0x0020:  0510 0a01 050f 2341 d461 0a74 f530 86e8
        0x0030:  79c9 8010 003f a2a1 0000 0101 080a fea5
        0x0040:  812d 3b68
11:24:53.864942 IP insa-20660.insa-toulouse.fr.54369 > insa-20661.insa-toulouse.fr.9025: Flags [FP.], seq 31:301, ack 1, win 63, options [nop,nop,TS val 996672135 ecr 4272259317], length 270
        0x0000:  a4bb 6dc2 a44f a4bb 6dc2 894c 8100 0064
        0x0010:  0800 4500 0142 9cce 4000 4006 7ec7 0a01
        0x0020:  050f 0a01 0510 d461 2341 86e8 79c9 0a74
        0x0030:  f530 8019 003f 4ca4 0000 0101 080a 3b68
        0x0040:  0287 fea5 80f5 2020 2020 3262 6262 6262
        0x0050:  6262 6262 6262 6262 6262 6262 6262 6262
        0x0060:  6262 6262 2020 2020 3363 6363 6363 6363
        0x0070:  6363 6363 6363 6363 6363 6363 6363 6363
        0x0080:  6363 2020 2020 3464 6464 6464 6464 6464
        0x0090:  6464 6464 6464 6464 6464 6464 6464 6464
        0x00a0:  2020 2020 3565 6565 6565 6565 6565 6565
        0x00b0:  6565 6565 6565 6565 6565 6565 6565 2020
        0x00c0:  2020 3666 6666 6666 6666 6666 6666 6666
        0x00d0:  6666 6666 6666 6666 6666 6666 2020 2020
        0x00e0:  3767 6767 6767 6767 6767 6767 6767 6767
        0x00f0:  6767 6767 6767 6767 6767 2020 2020 3868
        0x0100:  6868 6868 6868 6868 6868 6868 6868 6868
        0x0110:  6868 6868 6868 6868 2020 2020 3969 6969
        0x0120:  6969 6969 6969 6969 6969 6969 6969 6969
        0x0130:  6969 6969 6969 2020 2031 306a 6a6a 6a6a
        0x0140:  6a6a 6a6a 6a6a 6a6a 6a6a 6a6a 6a6a 6a6a
        0x0150:  6a6a 6a6a
11:24:53.865167 IP insa-20661.insa-toulouse.fr.9025 > insa-20660.insa-toulouse.fr.54369: Flags [F.], seq 1, ack 302, win 63, options [nop,nop,TS val 4272259373 ecr 996672135], length 0
        0x0000:  a4bb 6dc2 894c a4bb 6dc2 a44f 0800 4500
```

a4bb 6dc2 a44f a4bb 6dc2 894c 0800 4500
003c 9ccb 4000 4006 7fd0 0a01 050f 0a01
0510 2341 86e8 79aa 0000 0000 a002

Adresse Ethernet destination = 6 premier octets
Adresse Ethernet source = 6 octets
Type = 2 octets
Paquet IP

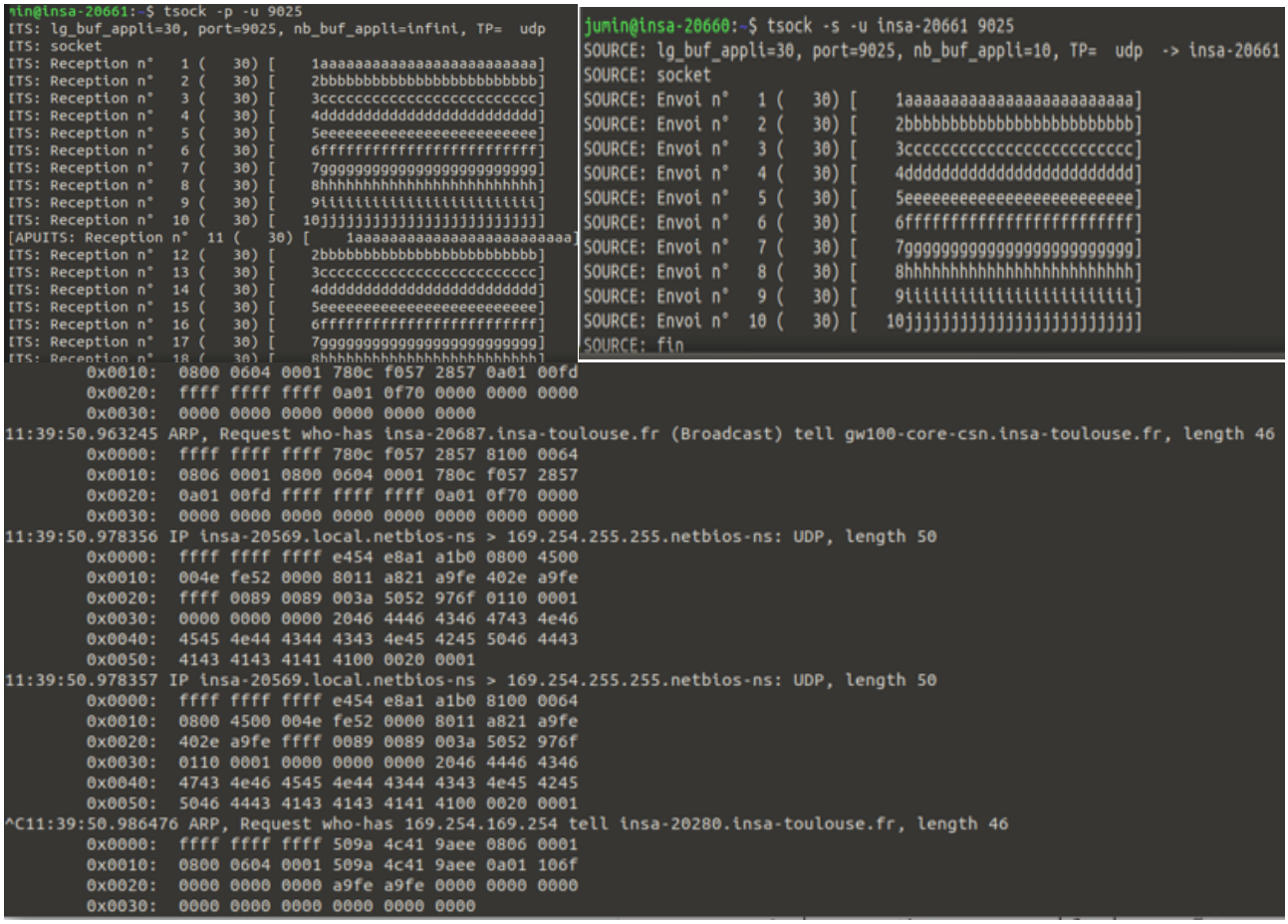
On s'intéresse au 9eme octet qui est le protocole au niveau transport utilisé => TCP : 06 / UDP : 17

Adresse IP source = 4 octets
Adresse IP destinataire = 4 octets
Numéro port source = 2octets

Pour trouver notre adresse on utilisera *ifconfig* dans le terminal

Question 5 :

Broadcast : Méthode de transmission de données à l'ensemble des machines d'un réseau.
Pour cela on utilise la commande : *tcpdump -xx broadcast*



On remarque que l'adresse IP destination est 169.254.255.255 ce qui est l'adresse de broadcast. Cela permet de s'adresser à toutes les machines du réseau, en effet chaque machine va voir ce message comme un message qui lui est adressé et va donc le traiter.