# Protocoles réseaux
# TD n° 8 : Analyse et capture de traces

## I) Analyse de traces faites avec `tcpdump`

**Exercice 1 :**

```
12:57:16.581515 IP 192.168.0.18.23222 > 95.161.221.194.30833: UDP, length 285
12:57:20.490839 IP 192.168.0.18.23222 > 157.7.194.141.6881: UDP, length 58
12:57:20.768031 IP 157.7.194.141.6881 > 192.168.0.18.23222: UDP, length 58
12:57:20.768246 IP 192.168.0.18.23222 > 185.183.32.185.33831: UDP, length 110
12:57:20.788525 IP 185.183.32.185.33831 > 192.168.0.18.23222: UDP, length 299
12:57:21.518066 IP 72.132.156.52.50321 > 192.168.0.18.23222: UDP, length 101
12:57:21.518275 IP 192.168.0.18.23222 > 72.132.156.52.50321: UDP, length 266
12:57:21.567798 IP 1.171.146.42.6881 > 192.168.0.18.23222: UDP, length 97
12:57:21.568011 IP 192.168.0.18.23222 > 1.171.146.42.6881: UDP, length 285
12:57:23.465573 IP 192.168.0.18.23222 > 88.23.92.69.51413: UDP, length 58
12:57:23.521455 IP 88.23.92.69.51413 > 192.168.0.18.23222: UDP, length 49
```

1. Quelle est la structure du protocole de couche application ? Qui sont les pairs en jeu ?
2. Pourquoi ce protocole utilise-t-il UDP plutôt que TCP ?
3. Sachant qu'il s'agit d'un sous-protocole de la suite BitTorrent, devinez à quoi sert ce protocole.

**Exercice 2 :**

```
13:05:05.312847 IP6 2a01:e0a:283:47b0:8731:4158:5978:9b1e.45682 > 2001:41d0:404:200::62ef.8
443: Flags [P.], seq 4279:4549, ack 1463, win 501,
     options [nop,nop,TS val 1437966609 ecr 2
533919660], length 270
13:05:05.312898 IP6 2a01:e0a:283:47b0:8731:4158:5978:9b1e.45682 > 2001:41d0:404:200::62ef.8443:
    Flags [P.], seq 4549:5065, ack 1463, win 501, options [...], length 516
13:05:05.312931 IP6 2a01:e0a:283:47b0:8731:4158:5978:9b1e.45682 > 2001:41d0:404:200::62ef.8443:
     Flags [P.], seq 5065:5311, ack 1463, win 501, options [...], length 246
...
13:05:06.243735 IP6 2a01:e0a:283:47b0:8731:4158:5978:9b1e.40750 > 2001:41d0:404:200::62ef.41969:
    UDP, length 73
13:05:06.264213 IP6 2a01:e0a:283:47b0:8731:4158:5978:9b1e.40750 > 2001:41d0:404:200::62ef.41969:
    UDP, length 76
13:05:06.264253 IP6 2a01:e0a:283:47b0:8731:4158:5978:9b1e.40750 > 2001:41d0:404:200::62ef.41969:
    UDP, length 1113
13:05:06.284810 IP6 2a01:e0a:283:47b0:8731:4158:5978:9b1e.40750 > 2001:41d0:404:200::62ef.41969:
    UDP, length 71
13:05:06.300288 IP6 2a01:e0a:283:47b0:8731:4158:5978:9b1e.40750 > 2001:41d0:404:200::62ef.41969:
    UDP, length 1159
13:05:06.305382 IP6 2a01:e0a:283:47b0:8731:4158:5978:9b1e.40750 > 2001:41d0:404:200::62ef.41969:
    UDP, length 70
...
13:05:07.606059 IP6 2a01:e0a:283:47b0:8731:4158:5978:9b1e.40750 > 2001:41d0:404:200::62ef.41969:
    UDP, length 88
13:05:07.606117 IP6 2a01:e0a:283:47b0:8731:4158:5978:9b1e.40750 > 2001:41d0:404:200::62ef.41969:
    UDP, length 824
13:05:07.622699 IP6 2001:41d0:404:200::62ef.41969 > 2a01:e0a:283:47b0:8731:4158:5978:9b1e.40750:
    UDP, length 76
```

Cette trace capture un morceau de vidéoconférence.

1. Pourquoi y a-t-il des échanges TCP et des échanges UDP ?
2. Pourquoi la taille des paquets UDP varie-t-elle ?
3. S'agit-il d'un protocole fiable ?

# II) Capture avec Wireshark

Wireshark permet d'analyser des traces, mais également d'en capturer. Si vous voulez aller au-delà de ce que décrit cet exercice, vous trouverez plus d'informations sur le wiki de Wireshark : `https://wiki.wireshark.org/`
Après avoir lancé Wireshark, il faut au préalable choisir la ou les interfaces sur lesquelles faire les capture. Dans le menu, choisir `Capture`, puis `Options` et sélectionnez une interface active (si vous êtes sous Linux en Wifi, c'est en général celle dont le nom commence par un `w`). Si vous ne la voyez pas, lancez Wireshark en tant que super-utilisateur (`sudo`) ou bien ajoutez votre login au groupe `wireshark`, cf `https://wiki.wireshark.org/CaptureSetup/CapturePrivileges`

**Exercice 3 :**

1. Démarrez une capture, et stoppez la après quelques secondes en vous assurant que l'on reçoit bien quelque chose.
2. Perdez-vous en conjectures pour interpréter ce que l'on a reçu (pas trop longtemps quand même).
3. Ajoutez un *filtre* d'affichage pour ne voir que les paquets ICMP puis exécutez la commande `ping -c 5 www.informatique.univ-paris-diderot.fr`
4. Combien de paquets ICMP circulent ?
5. Quelle est la taille des différents entêtes ? Des données de couche application ?
6. Est-ce du IPv4 ou du IPv6 ? Trouvez comment refaire les deux dernières questions avec l'autre version de IP (6 ou 4). Quelle différences observez-vous ?
7. Pourquoi y a-t-il y champ « *Type* » dans l'entête ICMP et quelle est sa valeur ?
8. Examinez les champs ICMP « *Identifier* » et « *Sequence number* » de chaque paquet. Comment se fait la mise en correspondance entre une requête et une réponse ICMP ?
9. Quels sont les champs constants et les champs qui sont modifiés dans les différents entêtes ?
10. Quel lien avec les champs `icmp_seq`, `ttl` et `time` de la sortie de `ping` sur le terminal ? Interprétez leurs différentes valeurs.
11. Trouvez l'adresse de broadcast de votre lien local (`ifconfig` devrait vous aider). Éxecutez la commande `ping -i 1 -w 1 -b` sur cette adresse. Identifiez qui répond.
12. Que valent les champs ICMP « *Identifier* » et « *Sequence number* » ? Quelles conséquences ?
13. Éxecutez la commande suivante :
    `ping -4 -s 8000 -c 1 www.informatique.univ-paris-diderot.fr`
    Que fait-elle ? (Indication : `man` est votre ami.)
14. Votre MTU est sûrement de 1500 octets. Pourtant, le ping réussit. À l'aide de Wireshark, déterminez pourquoi.
15. Quelles sont les données de couche application contenues dans le paquet ICMP ?
16. La fragmentation de couche 3 (couche réseau) peut être évitée à l'aide du bit « don't fragment » (DF) de l'entête IPv4. À l'aide de Wireshark, déterminez si TCP sur votre machine utilise le bit DF.
17. Pourquoi les implémentations modernes de TCP évitent-elles la fragmentation de couche 3 ? (Indication : que se passe-t-il si un fragment est perdu ?) Pourquoi ce problème ne se manifeste-t-il pas avec la segmentation de couche 4 ?
18. Et pour UDP : est il souhaitable/possible d'éviter la fragmentation ?
19. Pourquoi IPv6 n'implémente-t-il pas la fragmentation par les routeurs ?
20. Après la commande suivante, que se passe-t'il ? Pourquoi ?
    `ping -s 8000 -c 1 www.google.com`