Protocoles réseaux
TP n° 8 : le réseau à la trace

Exercice 1 : tcpdump

Exécuter la commande
`tcpdump -n -r /ens/micheli/capture/trace.pcap` [1].

1. Quelles sont les machines impliquées ?
2. Quel protocole de couche transport est-il utilisé ?
3. Pourquoi autant de paquets ont-ils taille 1448 ?
4. Y a-t-il eu des pertes de paquet ?
5. Pourquoi autant de ack 118 ?
6. Pourquoi les numéros de séquence des deux premiers paquets sont énormes, et ceux des suivants petits ?
7. Au fait, quel est le nom des machines impliquées ?
8. Interpréter les *flags* (lettre entre crochets comme `[S.]`)
9. Vers la fin on dirait qu'il y a moins de ack. Pourquoi ?
10. Interpréter les champs `val` et `ecr`.
11. Sur quelle machine la trace a-t-elle été capturée ? Justifier.

Exercice 2 : avec wireshark

Exécuter la commande
`wireshark /ens/micheli/capture/trace.pcap`

1. De quel protocole de couche d'application s'agit-il ?
2. Quelles sont les adresses des trois (oui, trois) machines impliquées ?
3. Quelles étaient les deux applications communicantes ?
4. Wireshark affiche deux protocoles de couche différente. Pourquoi ?
5. Quand cette capture a-t-elle eu lieu ?

Exercice 3 : sur papier

Analyser les fragments de traces suivants. Indiquer en particulier le protocole de couche transport utilisé et, si on est en mode connecté, à quelle phase de la connexion se trouve-t-on. Souligner de plus les problèmes de transmission lorsqu'il y en a. Toutes les traces ont été capturées sur la machine `A`.

1.

```
22:45:26.121149 A.32769 > B.www: S 2919412148:2919412148(0) win 5840 <mss 1460,nop,wscale 1>
22:45:26.123055 B.www > A.32769: S 4144006771:4144006771(0) ack 2919412149 win 65535 <mss 1460,nop,wscale 1>
22:45:26.123120 A.32769 > B.www: . ack 1 win 2920
22:45:26.124144 A.32769 > B.www: P 1:146(145) ack 1 win 2920
22:45:26.130323 B.www > A.32769: . 1:1461(1460) ack 146 win 32850
22:45:26.130402 A.32769 > B.www: . ack 1461 win 4380
22:45:26.130750 B.www > A.32769: . 1461:2921(1460) ack 146 win 32850
```

2.

```
22:45:26.195481 B.www > A.32769: . 74461:75921(1460) ack 146 win 32850
22:45:26.196216 B.www > A.32769: P 75921:77381(1460) ack 146 win 32850
22:45:26.196228 A.32769 > B.www: . ack 77381 win 64240
22:45:26.211281 B.www > A.32769: . 77381:78841(1460) ack 146 win 32850
22:45:26.211772 B.www > A.32769: P 78841:80301(1460) ack 146 win 32850
22:45:26.211783 A.32769 > B.www: . ack 80301 win 64240
```

1. sur lulu `/usr/sbin/tcpdump`

3.

```
22:45:26.240764 B.www > A.32769: P 127021:128387(1366) ack 146 win 32850
22:45:26.240776 A.32769 > B.www: . ack 128387 win 64240
22:45:26.249437 A.32769 > B.www: F 146:146(0) ack 128387 win 64240
22:45:26.251418 B.www > A.32769: . ack 147 win 32850
22:45:26.251840 B.www > A.32769: F 128387:128387(0) ack 147 win 32850
22:45:26.251871 A.32769 > B.www: . ack 128388 win 64240
```

4.

```
22:49:22.739301 A.32775 > C.ssh: . 110064:111524(1460) ack 2336 win 5104
22:49:22.739312 A.32775 > C.ssh: . 111524:112984(1460) ack 2336 win 5104
22:49:22.772816 C.ssh > A.32775: . ack 96924 win 62780
22:49:22.772828 A.32775 > C.ssh: . 112984:114444(1460) ack 2336 win 5104
22:49:22.772838 A.32775 > C.ssh: . 114444:115904(1460) ack 2336 win 5104
22:49:22.773905 C.ssh > A.32775: . ack 99844 win 62780
```

5.

```
22:53:47.759896 D.17775 > A.32782: . 75921:77381(1460) ack 1 win 17520
22:53:47.760031 D.17775 > A.32782: . 77381:78841(1460) ack 1 win 17520
22:53:47.760055 A.32782 > D.17775: . ack 78841 win 64240
22:53:47.885001 D.17775 > A.32782: . 80301:81761(1460) ack 1 win 17520
22:53:47.885072 A.32782 > D.17775: . ack 78841 win 64240
22:53:47.885816 D.17775 > A.32782: . 83221:84681(1460) ack 1 win 17520
22:53:47.885834 A.32782 > D.17775: . ack 78841 win 64240
22:53:47.885951 D.17775 > A.32782: . 84681:86141(1460) ack 1 win 17520
22:53:47.885962 A.32782 > D.17775: . ack 78841 win 64240
22:53:47.917054 D.17775 > A.32782: . 86141:87601(1460) ack 1 win 17520
22:53:47.917065 A.32782 > D.17775: . ack 78841 win 64240
22:53:48.042369 D.17775 > A.32782: . 78841:80301(1460) ack 1 win 17520
22:53:48.042419 A.32782 > D.17775: . ack 81761 win 64240
22:53:48.199537 D.17775 > A.32782: . 81761:83221(1460) ack 1 win 17520
22:53:48.199602 A.32782 > D.17775: . ack 87601 win 64240
22:53:48.199718 D.17775 > A.32782: . 89061:90521(1460) ack 1 win 17520
22:53:48.199732 A.32782 > D.17775: . ack 87601 win 64240
22:53:48.356490 D.17775 > A.32782: . 87601:89061(1460) ack 1 win 17520
22:53:48.356551 A.32782 > D.17775: . ack 90521 win 64240
22:53:48.356620 D.17775 > A.32782: . 90521:91981(1460) ack 1 win 17520
```

Exercice 4 : une autre trace

On considère maintenant la capture `/ens/micheli/capture/trace2.pcap`

1. Quelles sont les machines impliquées ?
2. Quels protocoles de la couche application sont utilisés ?
3. Quel est le nom et le mot de passe de l'utilisateur ?
4. Qu'est-ce qui est transféré ?
5. Quels ports sont utilisés ?
6. Pourquoi la capture est-elle si grosse ?
7. Quel est le débit du transfert en octets par seconde ?

Exercice 5 : graphiquement

On a représenté (avec `tcptrace` puis `xplot.org`) les diagrammes temps-séquence de plusieurs captures. Le résultat est récupérable dans `/ens/micheli/capture/`. Le fichier `trace1.png` est global, les autres `png` sont des zooms à l'intérieur. Interpréter.