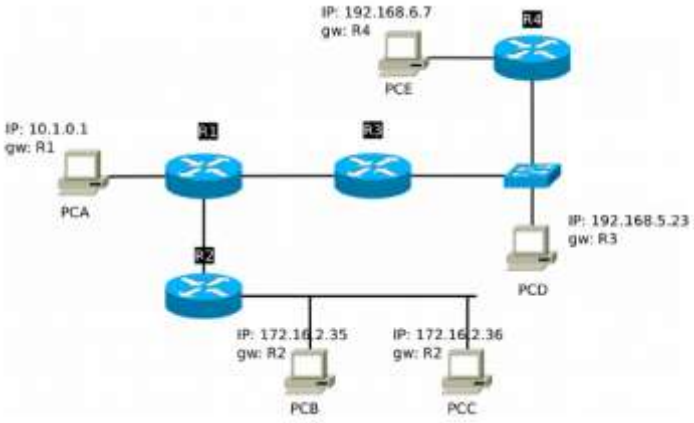
R2.05-- TD 2

Préambule

Nous allons continuer l'exercice du matériel du cours. Vous n'êtes pas obligés de répondre à nouveau aux questions que vous avez déjà eu pour l'exercice préparatif ; par contre je vous conseille d'avoir les réponses à la main.



Dans cette figure on indique les routeurs par un nom, de R1 à R4. Les machines sont indiquées par nom et par adresse IP. Pour chaque machine on indique sa passerelle par défaut, précédée par l'annotation gw.

Exercice 1

1. Indiquez les réseaux présents dans cette figure, avec les machines qui en font partie.

SOLUTION :

Un premier réseau, celui de PCA, R1, R2, est par défaut de classe A : 10.0.0.0/8. On peut alternativement dire que le réseau, étant très petit, pourrait être par exemple de classe C : 10.1.0.0/24.

Un deuxième réseau est celui de PCB, PCC et R2, qui est par défaut de classe B : 172.16.0.0/16.

Un troisième réseau est celui de PCD, R3 et R4 et il est de classe C : 192.168.5.0/24

Finalement, le réseau de PCE et R4 est de classe C : 192.168.6.0/24

2. La commande suivante est tapée sur une des machines ci-dessus.

   ```
   ss -ant
   ```

   Elle retourne le résultat suivant :

	Local Address	Foreign Address	state
tcp	10.1.0.1:2568	172.16.2.36:23	ESTABLISHED

   Répondez aux questions suivantes :

   - Quel est le rôle de la commande `ss` ?
   - Pouvez-vous indiquer la machine sur laquelle on a tapé cette commande ?
   - Soulignez les ports utilisés du côté serveur et du côté client.
   - Qui est le client et qui est le serveur dans cet échange ? Justifiez votre réponse.
   - Quel protocole est utilisé à la couche application ?
   - Pourquoi a-t-on le mot « `tcp` » à gauche de l'écran de résultats ?
   - Quel est le sens du mot « `ESTABLISHED` » à gauche de l'écran ?

SOLUTION :

La commande ss nous montre l'état des connexions d'une machine, sur quel protocole et port ces connexions se déroulent-elles, ainsi que les participants.

La machine sur laquelle on tape la commande est la machine dont l'adresse est mise en « Local address », notamment (si on voit la figure ci-dessus) le PCA.

Les ports utilisés sont le port standard 23 (côté serveur) et le port 2568 (côté client). On reconnait le serveur car il utilise le port par défaut (23). Donc PCA est le client et la machine 172.16.2.36 (PCC) est le serveur.

Le protocole standard sur le port TCP :23 est le protocole Telnet.

Le mot TCP à gauche de l'écran indique le fait que, à la couche transport, les messages Telnet sont encapsulés par le protocole TCP.

Le mot ESTABLISHED sur l'écran indique le fait qu'une connexion est en cours entre les machines PCA et PCC, avec PCC en serveur Telnet. Cette connexion a déjà été établie sur TCP.

3. Vous pouvez supposer que chaque machine a été configurée avec la passerelle par défaut mentionnée dans la figure. Est-ce que cela suffit pour permettre la connexion affichée par la commande `ss` ? Sinon, quelles routes peut-on encore déduire ?

SOLUTION :

Nous savons que le message de PCA ira par défaut vers R1. Nous savons aussi que, une fois arrivé à R2, le message sera transmis vers PCC. Une route pourrait être déduite entre R1 et R2 pour le réseau de R2 :

R1 >> ip route add 172.16.0.0/16 via <@IP de R2>

Mais, une connexion établie sur TCP demande que les messages puissent passer dans les deux directions. Pour la route de retour nous savons que les messages de PCC partent vers R2. De même, entre R1 et PCA les messages vont passer « par défaut » car les deux machines sont dans un même réseau. On peut donc déduire une route (probablement par défaut) de R2 via R1 :

R2 >> ip route add default via <@IP de R1>

4. La commande `ss -ant` sur la machine PCD donne le résultat suivant :

```
    Local Address       Foreign Address         state
tcp 0.0.0.0:21          0.0.0.0:*               LISTEN
```

Que pouvez-vous déduire à partir de cette capture ?

SOLUTION :

L'analyse ici ressemble beaucoup à celle fait au #2.

Nous voyons que la machine locale (PCD) est à l'écoute sur le port TCP :21 (port standard FTP), ce qu'indique le fait que PCD est un serveur FTP.

Alors pourquoi 0.0.0.0 :21 pour la machine locale ? Ceci indique seulement le fait que la machine PCD dispose de plusieurs interfaces réseau (elle peut avoir plusieurs adresses IP) et on indique qu'elle est à l'écoute sur TOUTES ces adresses IP (donc une autre machine pourra contacter PCD pour un connexion FTP sur toutes ses adresses.

Par contre le 0.0.0.0 :* se réfère au fait que la machine PCD est prête à répondre à des demandes de connexion FTP à partir de n'importe quelle machine client.

Il n'y a actuellement aucune connexion en cours.

5. Disons maintenant que la machine PC B veut contacter PC D sur le port de la figure précédente. De quelles routes avez-vous besoin pour réaliser cela ?

SOLUTION :

Les messages de PCB vers PCD vont via R2-R1-R3.

Pour cette route nous avons déjà les routes par défaut suivantes :

- sur PCB le routeur par défaut est R2
- sur R2 le routeur par défaut est R1

Il nous faut encore :

R1 >> ip route add default via <@IPde R3>

Vous vous demandez peut-être pourquoi on a mis ici une route par défaut, tandis qu'à l'exo 3 on a mis une route particulière sur R1. Ce n'est pas faux de faire l'inverse si vous voulez (mais il faut suivre ces modifications jusqu'au bout). Personnellement j'ai choisi de mettre sur R1 la route par défaut vers R3 car il y a plus de réseaux de ce côté-là, que du côté de R2.

Les messages de PCD vers PCB vont via R3-R1-R2.

Pour cette route nous avons déjà les routes suivantes :

- La passerelle par défaut de PCD est R3
- R1 a une route vers R2, pour le réseau de PCB

Il nous manque le routage entre R3 et R1 (que je ferais par défaut), donc :

R3 >> ip route add default via <@ IP de R1>

6. On suppose maintenant que le routage est bien mis en place. Qui est le client et qui est le serveur dans l'échange antérieur ?

SOLUTION : On avait vu que PCD était le serveur, ce qui fait de PCB le client.

7. A partir des deux captures précédentes (et des exemples vus en classe) pouvez-vous donner les résultats affichés par la commande `ss` sur PC B *et* sur PC D ?

SOLUTION :

Sur le client (PCB) on verra une capture du type :

Protocol	Local address	Foreign address	status
tcp	172.16.2.35 :<port sur 2 octets>	192.168.5.23 :21	ESTABLISHED

Sur PCD on aura aussi la ligne d'écoute :

Protocol	Local address	Foreign address	status
tcp	0.0.0.0 :21	0.0.0.0	LISTEN

tcp 192.168.5.23 :21 172.16.2.35 :<port sur 2 octets> ESTABLISHED

8. Décrivez le rôle des protocoles utilisés dans les captures des exercice 2 et 4.

SOLUTION : Telnet permet l'accès (par terminal) à une machine distante. La connexion n'est pas sécurisée.

FTP permet le transfert de fichiers.

9. Sur la figure qui représente la topologie du réseau notez les serveurs de chacun des protocoles décrits dans la question 9.

SOLUTION : Nous avons un serveur Telnet sur PCC et un serveur FTP sur PCD.

10. (difficile) Nous voulons transférer un fichier de la machine PC A vers la machine PC E. Sur le PC E le résultat de la commande `ss` est le suivant :

```
    Local Address         Foreign Address       state
tcp 0.0.0.0:23            0.0.0.0:*             LISTEN
```

Comment peut-on mettre en place le transfert en question ?

SOLUTION :

Déjà la capture ci-dessus nous indique le fait que PCE est un serveur Telnet.

Nous voulons transférer un fichier de la machine PCA vers PCE. Un transfert de fichier veut dire l'utilisation du protocole FTP. Or, ni PCA, ni PCE ne sont pas des serveurs FTP.

Mais, en revanche, la machine PCD, si.

Alors PCA fera le client FTP avec la machine PCD (en serveur) et on déposera le fichier sur PCD.

Puis à partir de la machine PCA on ouvre un Telnet vers la machine PCE (qui joue le rôle du serveur Telnet). En Telnet, on se connecte à la machine PCD (ici PCD est toujours le serveur FTP et PCE (manipulé depuis du PCA) est le client) et on retrouve le fichier.

Exercice 2

Dans l'annexe A vous allez trouver une capture sur Wireshark. A partir de cette capture, répondez aux questions suivantes.

1. Trouvez les éléments suivants :

    - L'adresse IP du client
    - L'adresse IP du serveur
    - Le port utilisé par le client
    - Le port utilisé par le serveur
    - Le protocole utilisé

SOLUTION : Déjà il faut savoir comment interpréter cette figure.

Nous voyons 20 messages qui sont échangés dans cette figure, entre deux machines : 192.168.56.1 et 192.168.56.101.

Dans chaque ligne de la capture nous voyons qui envoie le message (colonne Source), qui reçoit le message (colonne Destination), puis la taille du message, et ensuite, dans la colonne Info on voit plusieurs choses :

-- les ports (en format port de la source → port de la destination) -- les ports n'apparaissent que dans les messages TCP !

-- une description du message (par exemple SYN pour le premier message, ou GET pour le 4ème)

-- si ce sont des messages TCP aussi les index SEQ et ACK.

Alors, prenons le message dans la ligne 4. Ce message est sur HTTP (protocole de couche 7). C'est pourquoi, même si on a les adresses des machines qui communiquent, nous n'avons pas les ports de ces deux machines.

Mais, les ports des deux machines sont donnés en revanche pour tout message antérieur qui passe sur TCP. Donc par exemple dans le message 1 la machine source utilise le port 51651, tandis que la machine destination utilise le port 80 -- qui est un port standard sur TCP, pour le protocole HTTP.

La machine qui utilise le port 80 DOIT forcément être le serveur, donc la machine 192.168.56.101 est le serveur, tandis que l'autre machine, notamment 192.168.56.1 doit être le client.

Donc :

@ IP client : 192.168.56.1

@ IP serveur : 192.168.56.101

port client : 51651

port serveur : 80 (port standard HTTP)

protocole : c'est le protocole HTTP (encapsulé dans TCP >> IP >> Ethernet)

2. Expliquez le fonctionnement du protocole TCP.

SOLUTION : Le protocole TCP fonctionne à la couche 4 (transport). Son rôle est d'assurer la transmission fiable de données entre deux machines. Notamment il faut que les deux parties sachent si les messages ont bien été reçus et s'ils arrivent dans le bon ordre.

Le protocole TCP procède en trois étapes :

- L'ouverture de la connexion : une suite de 3 messages qui font le début de la connexion (SYN, SYN-ACK, ACK).
- Entre les deux : la connexion se déroule normalement, chaque message porte un nombre de séquence (indiquant l'index du message pour la partie qui envoie les messages) et un nombre d'ACK (indiquant l'index du message reçu).

La fermeture de la connexion : une suite de 3 messages qui finalisent la connexion (FIN, FIN-ACK, ACK).

3. Expliquez les notions de Seq et Ack pour chacun des participants au protocole TCP

SOLUTION : Voir ci-dessus.

En gros, les deux parties ont, chacune, un index SEQ pour les messages qu'elle envoie et un index ACK pour les messages reçus. A chaque fois qu'on envoie un message, l'index SEQ monte par 1, à chaque fois qu'on reçoit un message, l'index ACK monte par 1.

Il faut savoir que, lorsqu'un participant envoie un message, alors l'index SEQ est le compteur « actuel », c'est-à-dire l'index du message qui vient d'être envoyé. La valeur ACK corresponde par contre au compteur « attendu », c'est-à-dire qu'il indique à son correspondant que le prochain message envoyé par celui-ci devrait avoir un SEQ égal au ACK qui vient d'être utilisé.

Disons qu'une des machines (disons machine A) envoie un message avec SEQ = 35 et ACK = 98. Si le prochain message est envoyé par la machine A, alors il portera les indexes SEQ = 36 et ACK = 98. Si, ensuite, un message est envoyé par le correspondant de la machine A, disons la machine B, alors il portera le SEQ = 98 et ACK = 37.

4. Complétez la trame dans l'annexe en conséquence

SOLUTION : pour la ligne 2, on a ACK = 1 (car la machine attend le message avec SEQ = 1) et pour la ligne 3, on a aussi ACK = 1. Pour la ligne 19 ACK = 1989 (c'est le packet qu'on reçoit après) et pour la dernière ligne, ACK = 1139.

Exercice 3

Cet exercice concerne le protocole FTP et la capture d'écran montrée dans l'annexe B.

1. Relevez les adresses IP du client et du serveur FTP, ainsi que les ports utilisés.

SOLUTION :

Le port standard d'FTP est le port 21, donc la machine 164.81.20.1 est le serveur, tandis que la machine 164.81.20.108 est le client. Le serveur utilise le port standard (21 sur TCP) tandis que le client utilise le port 1156.

2. Pour chaque question et réponse FTP dans la capture mise dans l'annexe :
    - Trouvez le code (numérique ou en texte) associée
    - Indiquez si la commande vient du client ou du serveur
    - Expliquez le rôle du message

SOLUTION :

Nous voyons dans une première phase les codes : 220 (message d'accueil, confirmation de version), USER (qui demande le login), 331 (le login est OK, demande du mot de passe), PASS (envoie le mot de passe), 230 (l'authentification est OK). Je vous laisse établir qui envoie chacun de ces messages.

Dans une deuxième phase nous avons les traitements sur FTP (une fois l'utilisateur authentifié). Par exemple : XPWD (qui donne le répertoire actuel) ; 257 (confirmation d'envoi du répertoire actuel) ; CWD Socket (qui demande de changer le répertoire actuel) ; 250 (confirmation de changement du répertoire actuel).

Dans la phase de fermeture, nous avons le code QUIT (pour demander d'arrêter la connexion), 221 pour confirmer la demande de quitter.

3. Pour le serveur dans la capture de l'annexe B trouvez :

    - Le login utilisé pour accéder à l'FTP

- Le mot de passe utilisé :
- Le répertoire courant :

SOLUTION : login reseau1

password : T0rtue

Rep. courant : /home/reseau1

Annexe A

No.	Source	Destination	Protocol	Length	Info
1	192.168.56.1	192.168.56.101	TCP	74	51651→80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PE
2	192.168.56.101	192.168.56.1	TCP	74	80→51651 [SYN, ACK] Seq=0 Ack= in=5792 Len=0 MSS=14
3	192.168.56.1	192.168.56.101	TCP	66	51651→80 [ACK] Seq=1 Ack= in=29312 Len=0 TSval=5155
4	192.168.56.1	192.168.56.101	HTTP	350	GET / HTTP/1.1
5	192.168.56.101	192.168.56.1	TCP	66	80→51651 [ACK] Seq=1 Ack=285 Win=6880 Len=0 TSval=170
6	192.168.56.101	192.168.56.1	HTTP	557	HTTP/1.1 200 OK (text/html)
7	192.168.56.1	192.168.56.101	TCP	66	51651→80 [ACK] Seq=285 Ack=492 Win=30336 Len=0 TSval=
8	192.168.56.1	192.168.56.101	HTTP	331	GET /favicon.ico HTTP/1.1
9	192.168.56.101	192.168.56.1	TCP	66	80→51651 [ACK] Seq=492 Ack=550 Win=7936 Len=0 TSval=1
10	192.168.56.101	192.168.56.1	HTTP	569	HTTP/1.1 404 Not Found (text/html)
11	192.168.56.1	192.168.56.101	TCP	66	51651→80 [ACK] Seq=550 Ack=995 Win=31360 Len=0 TSval=
12	192.168.56.1	192.168.56.101	HTTP	361	GET /favicon.ico HTTP/1.1
13	192.168.56.101	192.168.56.1	HTTP	569	HTTP/1.1 404 Not Found (text/html)
14	192.168.56.1	192.168.56.101	TCP	66	51651→80 [ACK] Seq=845 Ack=1498 Win=32512 Len=0 TSval=
15	192.168.56.1	192.168.56.101	HTTP	359	GET /page.html HTTP/1.1
16	192.168.56.101	192.168.56.1	HTTP	556	HTTP/1.1 200 OK (text/html)
17	192.168.56.1	192.168.56.101	TCP	66	51651→80 [ACK] Seq=1138 Ack=1988 Win=33536 Len=0 TSval
18	192.168.56.101	192.168.56.1	TCP	66	80→51651 [FIN, ACK] Seq=1988 Ack=1138 Win=10080 Len=0
19	192.168.56.1	192.168.56.101	TCP	66	51651→80 [FIN, ACK] Seq=1138 Ack= Win=33536 Len=0
20	192.168.56.101	192.168.56.1	TCP	66	80→51651 [ACK] Seq=1989 Ack= Win=10080 Len=0 TSval

Annexe B

No.	Time	Source	src	Destination	dst	Protocol	Length	Info
1	0.000000	164.81.20.108	1156	164.81.20.1	21	TCP	62	1156→21 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SA
2	0.000473	164.81.20.1	21	164.81.20.108	1156	TCP	62	21→1156 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 M
3	0.000525	164.81.20.108	1156	164.81.20.1	21	TCP	54	1156→21 [ACK] Seq=1 Ack=1 Win=65535 Len=0
4	0.002798	164.81.20.1	21	164.81.20.108	1156	FTP	74	Response: 220 (vsFTPd 2.0.7)
5	0.129295	164.81.20.108	1156	164.81.20.1	21	TCP	54	1156→21 [ACK] Seq=1 Ack=21 Win=65515 Len=0
6	1.450836	164.81.20.108	1156	164.81.20.1	21	FTP	68	Request: USER reseaul
7	1.451375	164.81.20.1	21	164.81.20.108	1156	TCP	60	21→1156 [ACK] Seq=21 Ack=15 Win=5840 Len=0
8	1.451386	164.81.20.1	21	164.81.20.108	1156	FTP	88	Response: 331 Please specify the password.
9	1.571045	164.81.20.108	1156	164.81.20.1	21	TCP	54	1156→21 [ACK] Seq=15 Ack=55 Win=65481 Len=0
10	2.940956	164.81.20.108	1156	164.81.20.1	21	FTP	67	Request: PASS TQrtue
11	2.972536	164.81.20.1	21	164.81.20.108	1156	FTP	77	Response: 230 Login successful.
12	3.103105	164.81.20.108	1156	164.81.20.1	21	TCP	54	1156→21 [ACK] Seq=28 Ack=78 Win=65458 Len=0
13	4.700383	164.81.20.108	1156	164.81.20.1	21	FTP	60	Request: XPWD
14	4.700826	164.81.20.1	21	164.81.20.108	1156	FTP	75	Response: 257 "/home/reseaul"
15	4.815155	164.81.20.108	1156	164.81.20.1	21	TCP	54	1156→21 [ACK] Seq=34 Ack=99 Win=65437 Len=0
16	6.736881	164.81.20.108	1156	164.81.20.1	21	FTP	66	Request: CWD Socket
17	6.750319	164.81.20.1	21	164.81.20.108	1156	FTP	91	Response: 250 Directory successfully changed.
18	6.887716	164.81.20.108	1156	164.81.20.1	21	TCP	54	1156→21 [ACK] Seq=46 Ack=136 Win=65400 Len=0
19	8.180203	164.81.20.108	1156	164.81.20.1	21	FTP	60	Request: XPWD
20	8.180722	164.81.20.1	21	164.81.20.108	1156	FTP	82	Response: 257 "/home/reseaul/Socket"
21	8.329502	164.81.20.108	1156	164.81.20.1	21	TCP	54	1156→21 [ACK] Seq=52 Ack=164 Win=65372 Len=0
22	9.300656	164.81.20.108	1156	164.81.20.1	21	FTP	60	Request: QUIT
23	9.301418	164.81.20.1	21	164.81.20.108	1156	FTP	68	Response: 221 Goodbye.
24	9.302019	164.81.20.1	21	164.81.20.108	1156	TCP	60	21→1156 [FIN, ACK] Seq=178 Ack=58 Win=5840 Len=
25	9.302041	164.81.20.108	1156	164.81.20.1	21	TCP	54	1156→21 [ACK] Seq=58 Ack=179 Win=65358 Len=0
26	9.303508	164.81.20.108	1156	164.81.20.1	21	TCP	54	1156→21 [FIN, ACK] Seq=58 Ack=179 Win=65358 Len
27	9.303985	164.81.20.1	21	164.81.20.108	1156	TCP	60	21→1156 [ACK] Seq=179 Ack=59 Win=5840 Len=0