24/04/2014

Juliana Leclaire

Céline de Roland

TP1 INFO 625

PRÉSENTATION

Objectifs du TP :

- Se familiariser avec la configuration et l'utilisation de machines virtuelles
- Apprendre à configurer les paramètres IP sous Windows XP et Linux
- Réaliser des captures de trames afin d'expliquer le fonctionnement des protocoles
- Etudier l'architecture du réseau Internet

CRÉATION ET CONFIGURATION DE MACHINES VIRTUELLES

1) *Sur A, relever toute la configuration réseau. Par qui vous ont été fournis ces paramètres ?*

Nous avons créé deux machines virtuelles :

- A pour Windows XP
- B pour Linux.

La configuration réseau de la machine A est la suivante :

	Machine A
@IP	10.0.2.15
Passerelle	10.0.2.2
DNS	193.48.120.32 et 193.48.129.137
Connectivité	correcte

Ces paramètres nous sont fournis par la machine hôte.

2) *Relever la configuration réseau de B. Tester la connectivité de B avec une machine sur Internet*

	Machine B
@IP	10.0.2.15

Passerelle	10.0.2.2
DNS	193.48.120.32 et 193.48.129.137
Connectivité	correcte

3) Comparer les configurations IP des deux machines. Peuvent-elles selon vous communiquer entre elles ?

Les deux machines ont les mêmes adresses IP car elles ne sont pas sur le même réseau privé. Elles ne pourront pas communiquer entre-elles car elles ont des adresses IP identiques.

Nous adaptons la configuration réseau des machines virtuelles afin qu'elles puissent communiquer entre-elles.

L'adresse IP associée à l'interface vboxnet0 est 192.168.56.1/24.

4) Combien de cartes réseau la machine TinyCore possède-t-elle ? A quel réseau VirtualBox sont elles connectées ? Relever leur adresse IP. Vérifier que la machine a bien accès à Internet même si elle n'a pas les serveurs DNS de renseignés.

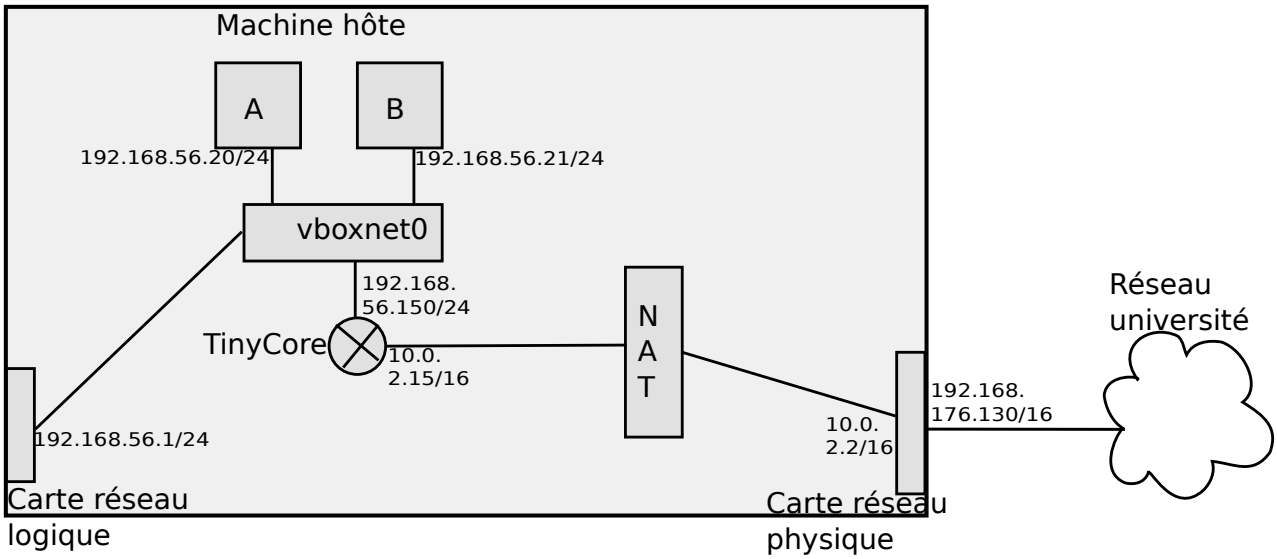
Nous relions les deux machines pour qu'elles soient connectées sur le commutateur virtuel vboxnet0. Nous ajoutons un routeur à ce réseau qui sera une machine virtuelle dénommée TinyCoreRouter. Cette machine a deux cartes réseaux.

	Mode d'accès réseau	@IP
Carte1	vboxnet0	192.168.56.150
Carte2	NAT	10.0.2.15

La machine TinyCore a bien accès à Internet même si elle n'a pas les serveurs DNS de renseignés.

5) Relever les paramètres IP obtenus par vos machines. Faire un schéma simplifié du réseau faisant apparaître les deux machines virtuelles, le commutateur virtuel, le routeur, le processus NAT de VirtualBox et finalement le réseau de l'université. Toutes les adresses de toutes les interfaces doivent être précisées.

	Machine A	Machine B
@IP	192.168.56.20/24	192.168.56.21/24
Passerelle	192.168.56.150	192.168.56.150



6) *Les machines XP et CentOS ont-elles un accès à Internet ? La résolution DNS fonctionne-t-elle ?*

L'accès internet (ping 8.8.8.8) fonctionne, mais pas la résolution DNS (ping www.google.com)

7) *Donner l'adresse IP des serveurs DNS de l'université.*

Les adresses IP des serveurs DNS de l'université sont 193.48.120.32 et 193.48.129.137.

Nous configurons les machines A et B pour qu'elles utilisent les serveurs DNS de l'université pour qu'on puisse faire des ping avec des noms de domaine.

CAPTURES DE TRAMES

8) *Rappeler le rôle du protocole ARP. Vérifier que les trames capturées correspondent à la théorie vue en cours*

le rôle du protocole ARP est de déterminer l'adresse MAC d'une machine du même réseau en connaissant son adresse IP.

Voici les captures obtenues :

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	CadmusCo_5a:1e:b4	Broadcast	ARP	42	Who has 192.168.56.21? Tell 192.168.56.20
2	0.00000000	CadmusCo_62:77:09	CadmusCo_5a:1e:b4	ARP	60	192.168.56.21 is at 08:00:27:62:77:09
3	0.00000000	192.168.56.20	192.168.56.21	ICMP	74	Echo (ping) request id=0x0200, seq=6144/24, ttl=128
4	0.00000000	192.168.56.21	192.168.56.20	ICMP	74	Echo (ping) reply id=0x0200, seq=6144/24, ttl=64
5	0.99188600	192.168.56.20	192.168.56.21	ICMP	74	Echo (ping) request id=0x0200, seq=6400/25, ttl=128
6	0.99248300	192.168.56.21	192.168.56.20	ICMP	74	Echo (ping) reply id=0x0200, seq=6400/25, ttl=64
7	1.99323100	192.168.56.20	192.168.56.21	ICMP	74	Echo (ping) request id=0x0200, seq=6656/26, ttl=128
8	1.99396700	192.168.56.21	192.168.56.20	ICMP	74	Echo (ping) reply id=0x0200, seq=6656/26, ttl=64
9	2.99450700	192.168.56.20	192.168.56.21	ICMP	74	Echo (ping) request id=0x0200, seq=6912/27, ttl=128
10	2.99521300	192.168.56.21	192.168.56.20	ICMP	74	Echo (ping) reply id=0x0200, seq=6912/27, ttl=64
11	4.99984100	CadmusCo_62:77:09	CadmusCo_5a:1e:b4	ARP	60	Who has 192.168.56.20? Tell 192.168.56.21
12	4.99986100	CadmusCo_5a:1e:b4	CadmusCo_62:77:09	ARP	42	192.168.56.20 is at 08:00:27:5a:1e:b4

Ceci est cohérent avec la théorie vue en cours :

- La machine A envoie une requête ARP en broadcast
- La machine B répond à la machine A
- Ensuite A et B communiquent directement
- Il y a une étape supplémentaire à la fin : la machine B envoie une requête ARP à A pour lui

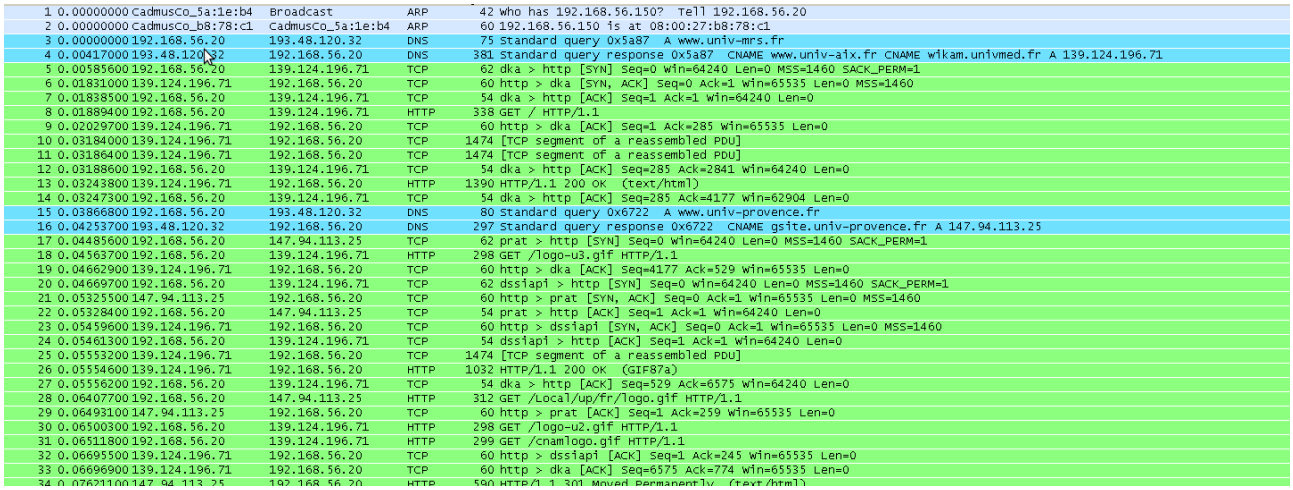
demander son adresse MAC.

9) Expliquer une méthode permettant de mesurer la durée pendant laquelle une entrée arp reste dans le cache. Mesurer la durée du cache ARP.

On lance un ping puis on exécute arp -a en continu jusqu'à ce qu'il affiche une table vide. Cela a mis environ 90 secondes.

10) Analyser le 10 premières trames échangées et synthétiser les résultats dans un tableau. Donner pour chaque trame son rôle

Voici le résultat obtenu :



	@Mac src	@Mac dest	@IP src	@IP dest	Prot. Transport	Port Src	Port Dest	Prot. App.	Flags TCP	N° Séq.	N° Ack	Données Utiles (octets)	Rôle trame
1)	MAC-A	Broadcast	IP-A	IP-Tiny				ARP				0	Déterminer l'adresse MAC de Tiny
2)	MAC-Tiny	MAC-A	IP-Tiny	IP-A				ARP				0	Donner à A l'adresse MAC de Tiny
3)	MAC-A	MAC-Tiny	IP-A	IP-DNS	UDP	1025	53	DNS				33	Chercher l'adresse IP de www.univ-mrs.fr
4)	MAC-Tiny	MAC-A	IP-DNS	IP-A	UDP	53	1025	DNS				339	Donner l'adresse IP de www.univ-mrs.fr
5)	MAC-A	MAC-Tiny	IP-A	IP-MRS	TCP	1263	80		SYN	0	0	0	Ouvrir la connexion avec www.univ-mrs.fr
6)	MAC-Tiny	MAC-A	IP-MRS	IP-A	TCP	80	1263		SYN,ACK	0	1	0	Ouvrir la connexion avec www.univ-mrs.fr
7)	MAC-A	MAC-Tiny	IP-A	IP-MRS	TCP	1263	80		ACK	1	1	0	Ouvrir la connexion avec www.univ-mrs.fr
8)	MAC-A	MAC-Tiny	IP-A	IP-MRS	TCP	1263	80	HTTP	ACK	1	1	284	Demander la page web au serveur www.univ-mrs.fr
9)	MAC-Tiny	MAC-A	IP-MRS	IP-A	TCP	80	1263		ACK	1	285	0	Préparer la réponse du serveur
10)	MAC-Tiny	MAC-A	IP-MRS	IP-A	TCP	80	1263		ACK	1	285	0	Renvoie de la trame précédente (il y a sans doute eu un problème)

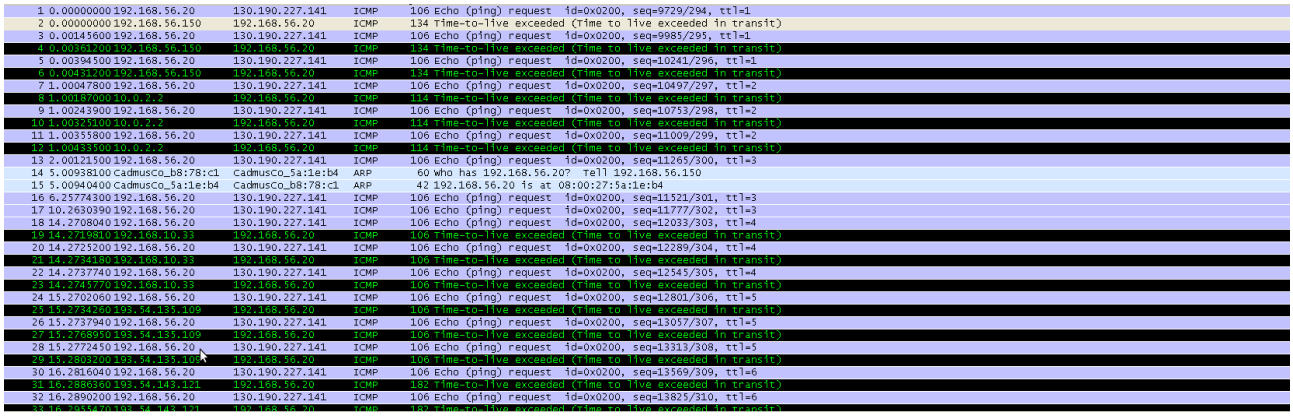
COMMANDE TRACEROUTE

11) Analyser les trames obtenues. Le fonctionnement de traceroute correspond-t-il à celui décrit ci-dessus ?

Nous faisons un tracert pour voir le chemin pris par les paquets pour joindre le site web de l'université de grenoble.

Chemin pris : 192.168.56.150 → 10.0.2.2 → 192.168.10.33 → 193.54.135.109 → 193.54.143.121 → 193.54.143.122 → 193.54.184.45 → 193.54.184.126 → 130.190.227.141

Capture de trames :



Cette capture est cohérente avec l'explication donnée dans l'énoncé. En effet, on voit que chacune des adresses du chemin nous envoie 3 requêtes ICMP.

12) En utilisant les résultats du traceroute, déduire et relever la valeur des adresses IP des routeurs qui se trouvent aux extrémités de la liaison transatlantique. Calculer le temps mis par les paquets pour traverser l'atlantique. Expliquez succinctement la méthode utilisée.

Nous faisons un traceroute vers le site de l'université de Drexel qui se trouve à Philadelphie. Adresses IP des routeurs qui se trouvent aux extrémités de la liaison transatlantique.

Résultat du traceroute :

```
 2  rtr-bourget.local.univ-savoie.fr (192.168.10.33)  0.193 ms  0.187 ms  0.179 ms
 3  193.54.135.109 (193.54.135.109)  2.273 ms  2.265 ms  2.267 ms
 4  vl1183-te2-6-lyon1-rtr-021.noc.renater.fr (193.51.185.70)  2.464 ms  2.482 ms  2.516 ms
 5  te0-3-0-0-lyon1-rtr-001.noc.renater.fr (193.51.189.13)  8.390 ms  8.379 ms  8.364 ms
 6  * * *
 7  renater.mx1.gen.ch.geant.net (62.40.124.61)  10.499 ms  10.333 ms  10.368 ms
 8  ae1.mx1.fra.de.geant.net (62.40.98.109)  18.515 ms  18.439 ms  18.413 ms
 9  abilene-wash-gw.mx1.fra.de.geant.net (62.40.125.18)  126.261 ms  112.516 ms  112.511 ms
10  204.238.76.65 (204.238.76.65)  115.808 ms  129.849 ms  129.806 ms
11  204.238.76.83 (204.238.76.83)  129.826 ms  130.474 ms  116.590 ms
12  144.118.1.24 (144.118.1.24)  117.080 ms  137.728 ms  137.657 ms
```

Les serveurs se trouvant des 2 côtés de la liaison transatlantique sont :
-@IP1 : 62.40.98.109 (Temps moyen : 18ms)
-@IP2 : 62.40.125.18 (Temps moyen : 116 ms)

Pour traverser l'atlantique le paquet a mis en moyenne 116 – 18 =108 ms aller-retour donc pour un aller le paquet met environ 49 ms.

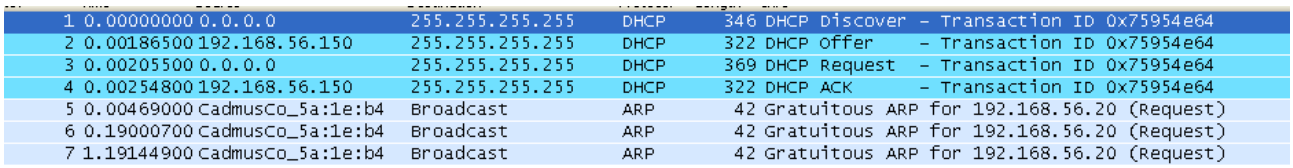
ANALYSE DU PROTOCOLE DHCP

13) Rappeler le rôle du protocole DHCPD

Le rôle du protocole DHCP est d'attribuer des adresses IP aux machines.

14) Analyser l'échange de données capturé

Nous effaçons les @IP de la machine A avec la commande ipconfig/release, puis nous réattribuons les adresses grâce au serveur DHCP et la commande ipconfig/renew :



1	0.00000000	0.0.0.0	255.255.255.255	DHCP	346 DHCP Discover - Transaction ID 0x75954e64
2	0.00186500	192.168.56.150	255.255.255.255	DHCP	322 DHCP Offer - Transaction ID 0x75954e64
3	0.00205500	0.0.0.0	255.255.255.255	DHCP	369 DHCP Request - Transaction ID 0x75954e64
4	0.00254800	192.168.56.150	255.255.255.255	DHCP	322 DHCP ACK - Transaction ID 0x75954e64
5	0.00469000	CadmusCo_5a:1e:b4	Broadcast	ARP	42 Gratuitous ARP for 192.168.56.20 (Request)
6	0.19000700	CadmusCo_5a:1e:b4	Broadcast	ARP	42 Gratuitous ARP for 192.168.56.20 (Request)
7	1.19144900	CadmusCo_5a:1e:b4	Broadcast	ARP	42 Gratuitous ARP for 192.168.56.20 (Request)

- La machine A envoie en double broadcast une trame DHCPDISCOVER.
- La machine Tiny envoie un DHCPOFFER en double broadcast avec comme paramètre une @IP proposée à A.
- A envoie une trame DHCPREQUEST en double broadcast, afin de confirmer au serveur qu'il prend l'adresse proposée.
- Le serveur envoie une trame DHCPPACK en double broadcast.

Cela est cohérent avec la théorie vue en cours, à l'exception des doubles broadcast. D'ailleurs nous voyons suivre des requêtes « Gratuitous ARP » ensuite, qui semblent être destinées à ce que A vérifie que sa propre adresse IP est bien associée à sa propre adresse MAC.

SERVEUR DNS

15) Saisir nslookup www.mit.edu, noter la réponse obtenue. Quel serveur vous a répondu ?

Après avoir entré la commande nslookup www.mit.edu le serveur univax.univ-savoie.fr répond : « Nom : e7086.b.akamaiedge.net ».

16) Quel est le protocole de transport utilisé ? Quels sont les ports utilisés ? Quel est le type de requête DNS ?

Après avoir entré la commande nslookup www.mit.edu. Le protocole de transport utilisé est UDP, les ports utilisés sont 53,1281,1282,1283. Le type de requête DNS est récursive car le serveur DNS 193.48.120.32 est un resolver.

17) Saisir nslookup @IP-mit, noter la réponse obtenue. La réponse correspond-t-elle à celle donnée précédemment ?

A la place d'entrer le nom de domaine nous entrons l'adresse IP de www.mit.edu, nous n'obtenons pas le même résultat car il a fait de la résolution inverse en cherchant des PTR.
La réponse est la suivante : « Nom : a172-227-9-45.deploy.static.akamaitechnologies.com ».

18) A quoi correspond la réponse obtenue ?

Nous entrons la commande nslookup puis set type=ns. Nous obtenons la liste de tous les serveurs racines.

19) Obtient-on une réponse, pourquoi ?

Nous entrons la commande server a.root-server.net (nous choisissons un serveur racine à interroger), puis set type=A puis www.mit.edu. Nous obtenons une réponse différente de tout à l'heure : le serveur n'est pas un resolver donc il se contente de nous proposer d'autres serveurs DNS.

20) A quoi correspond la réponse obtenue ?

Nous entrons la commande set type=ns puis edu. Ensuite nous entrons server 192.5.6.30 (nous en choisissons un parmis ceux proposés), puis mit.edu.. Nous obtenons la liste des serveurs DNS qui pourraient connaître l'adresse IP de mit.edu.