TP Réseau : Réseau de services

Objectifs

Les objectifs de ce TP sont de découvrir :
- la configuration d'un client par DHCP
- le fonctionnement de base du DNS
- la mise en place d'un NAT
- les échanges lors d'une communication HTTP

Ce TP est un TP d'observation. Il est important de bien identifier les caractéristiques de chaque communication et chaque service utilisé.

TP Réseau : Réseau de services

Travail préparatoire

Installation du TP

Le TP6 est disponible dans l'archive `/home/user/sujets-tp/tp6.tgz` :

```
host> cd /home/user/
host> tar zxvf sujets-tp/tp6.tgz
host> cd /home/user/tp6
host> lstart -s
```

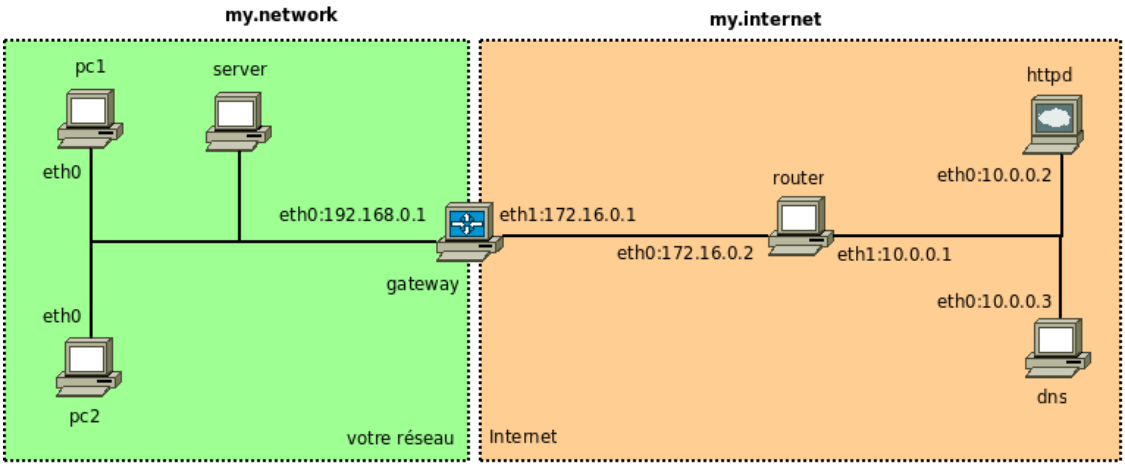
La maquette

Dans ce TP, la maquette NetKit est la suivante :

```
pc1[0]="A"
pc2[0]="A"
gateway[0]="A"
gateway[1]="B"
server[0]="A"
router[0]="B"
router[1]="C"
httpd[0]="C"
dns[0]="C"
```

Le fichier lab.conf

Ce qui donne l'architecture suivante :



Cette configuration possède deux ensembles :

- votre réseau (*my.network*) : les machines **server**, **pc1** et **pc2** représentent un réseau domestique derrière une passerelle (**gateway**). Chacune des machines est sous votre contrôle.
- Internet (*my.internet*) : **router**, **httpd** et **dns** sont des machines extérieures à votre réseau (vous ne devez pas modifier leur configuration).

Les machines sont les suivantes :

- **pc1** et **pc2** représentent des postes clients dans votre réseau
- **server** est une machine fournissant le service DNS et DHCP dans votre réseau.
- **gateway** représente votre passerelle pour l'accès à Internet
- **dns** (*my.internet*) est un serveur DNS
- **httpd** (*my.internet*) est un serveur HTTP
- **router** représente un routeur sur Internet

La configuration de ce réseau local est sensiblement identique à celle d'un réseau domestique connecté à un FAI (Fournisseur d'Accès Internet).

DHCP

DHCP (*Dynamic Host Configuration Protocol*) désigne un protocole réseau (RFC 1541 et 2131 essentiellement) dont le rôle est d'assurer la configuration automatique des paramètres IP d'une station, notamment en lui assignant automatiquement une adresse IP et un masque de sousréseau pour une durée limitée (bail).

DHCP fonctionne sur le modèle client-serveur : un serveur (port 67), qui détient la politique d'attribution des configurations IP, envoie une configuration donnée pour une durée donnée à un client (port 68) donné (typiquement, une machine qui vient de démarrer). Le serveur va servir de base pour toutes les requêtes DHCP (il les reçoit et y répond), aussi doit-il avoir une configuration IP fixe. Le protocole DHCP s'appuie entièrement sur BOOTP en reprenant le mécanisme de base et le format des messages. DHCP est une extension de BOOTP.

DHCP utilise le protocole de transport UDP et les ports 67 (server) et 68 (client). Les datagrammes de base utilisés par DHCP sont : `DHCPDISCOVER`, `DHCPOFFER`, `DHCPREQUEST` et `DHCPACK`.

DHCP fonctionne avec IPv4 mais il fonctionne aussi avec IPv6 et il est alors appelé DHCPv6. Toutefois, en IPv6, les adresses peuvent être autoconfigurées sans DHCP.

Dans ce TP, on utilisera **Dnsmasq**. **Dnsmasq** est un serveur léger pour fournir les services DNS, DHCP, BOOTP et TFTP pour un petit réseau, voire pour un poste de travail.

```
// connaître l'état du service sur une machine virtuelle ?
server:~# /etc/init.d/dnsmasq status

// arrêter le service sur une machine virtuelle ?
server:~# /etc/init.d/dnsmasq stop

// démarrer le service sur une machine virtuelle ?
server:~# /etc/init.d/dnsmasq start

// redémarrer le service sur une machine virtuelle ?
server:~# /etc/init.d/dnsmasq restart
```

La configuration de `dnsmasq` est réalisée dans le fichier `/etc/dnsmasq.conf` :

```
domain-needed
bogus-priv
local=/my.network/
```

```
interface=eth0
expand-hosts
domain=my.network
dhcp-range=192.168.0.50,192.168.0.150,12h
dhcp-option=option:router,192.168.0.1
read-ethers
```

`dnsmasq` offre un service DHCP statique ou dynamique. Ici, on utilisera une affectation statique basée sur les adresses MAC des postes clients à partir du fichier `/etc/ethers` :

```
00:02:05:00:00:01 192.168.0.50
00:02:05:00:00:02 192.168.0.51
```

Documentation : http://www.thekelleys.org.uk/dnsmasq/doc.html

DNS

Le DNS (*Domain Name System* ou système de noms de domaine) est un service permettant de traduire un nom de domaine en adresses IP de la machine portant ce nom (RFC 882/883 en 1983).

DNS utilise le protocole de transport UDP et le port 53. La taille maximale des paquets utilisée est de 512 octets.

Le type d'enregistrement de ressource RR (*Resource Record*) est codé sur 16 bits. Les principaux enregistrements définis sont les suivants :

- A *record* (*Address record*) qui fait correspondre un nom d'hôte à une adresse IPv4 de 32 bits distribués sur quatre octets. AAAA pour IPv6.
- CNAME *record* (*Canonical Name record*) qui permet de faire d'un domaine un alias vers un autre. Cet alias hérite de tous les sous-domaines de l'original.
- PTR *record* (*PoinTer Record*) qui associe une adresse IP à un enregistrement de nom de domaine (aussi dit « reverse » car il fait le contraire du A *record*).

Le système des noms de domaines consiste en une hiérarchie dont le sommet est appelé la racine (représentée par un point). Dans un domaine, on peut créer un ou plusieurs sous-domaines ainsi qu'une délégation pour ceux-ci (les informations relatives à ce sous-domaine sont enregistrées sur un autre serveur).

Les domaines se trouvant immédiatement sous la racine sont appelés domaine de premier niveau (TLD : *Top Level Domain*). Les noms de domaines ne correspondant pas à une extension de pays sont appelés des domaines génériques, par exemple `.org` ou `.com`. S'ils correspondent à des codes de pays (fr, be, ch...), on les appelle ccTLD (*country code TLD*).

On entend par FQDN (*Fully qualified domain name*) ou Nom de domaine pleinement qualifié un nom de domaine écrit de façon absolue, y compris tous les domaines jusqu'au domaine de premier niveau (TLD), il est ponctué par un point final. Dans un réseau TCP/IP, une adresse FQDN sera l'association entre le nom de la machine et le domaine auquel elle appartient.

Les hôtes n'ont qu'une connaissance limitée du système des noms de domaine. Quand ils doivent résoudre un nom, ils s'adressent à un ou plusieurs serveurs de noms dits **récursifs**, c'est-à-dire qui vont parcourir la hiérarchie DNS et faire suivre la requête à un ou plusieurs autres serveurs de noms pour fournir une réponse.

Les serveurs racine sont gérés par douze organisations différentes (2 européennes, 1 japonaise et 9 américaines). Sept de ces serveurs sont en réalité distribués dans le monde grâce à la technique *anycast* (plus de

200 serveurs répartis dans 50 pays du monde). Il existe 13 autorités de nom appelées de a à `m.root-servers.net`. Le serveur `k` reçoit par exemple de l'ordre de 20 000 requêtes par seconde.

Dans ce TP, on utilisera **Dnsmasq** et **BIND** :

- **Dnsmasq** est un serveur léger pour fournir les services DNS, DHCP, BOOTP et TFTP pour un petit réseau, voire pour un poste de travail. La configuration de `dnsmasq` est réalisée dans le fichier `/etc/dnsmasq.conf`.
- **BIND** (*Berkeley Internet Name Daemon*) est le serveur DNS le plus utilisé sur Internet (79 % des serveurs en 2008), spécialement sur les systèmes de type UNIX. La configuration de `BIND` est réalisée dans le fichier `/etc/bind/named.conf`.

Pour le service `bind` :

```
// connaître l'état du service sur une machine virtuelle ?
dns:~# /etc/init.d/bind status

// arrêter le service sur une machine virtuelle ?
dns:~# /etc/init.d/bind stop

// démarrer le service sur une machine virtuelle ?
dns:~# /etc/init.d/bind start

// redémarrer le service sur une machine virtuelle ?
dns:~# /etc/init.d/bind restart
```

NAT

Un routeur fait du NAT (*Network Address Translation* soit « traduction d'adresse réseau ») lorsqu'il fait correspondre les adresses IP internes privées (non-uniques et souvent non routables) d'un intranet à un ensemble d'adresses externes publiques (uniques et routables).

Ce mécanisme permet notamment de faire correspondre une seule adresse externe publique visible sur Internet à toutes les adresses d'un réseau privé, et pallie ainsi l'épuisement des adresses IPv4.

La NAT **statique**, se base sur l'association de `N` adresses internes avec `N` adresses externes. C'est à dire qu'à UNE adresse IP interne, on associe UNE adresse IP externe. Dans ce cas, la seule action qui sera effectuée par le routeur sera de remplacer l'adresse source ou destination par l'adresse correspondante.

Ce type de NAT sert généralement à donner l'accès à des serveurs en interne (DMZ) à partir de l'extérieur. Si l'on s'en tient intrinsèquement à la dénition du terme NAT, cela représente la modication des adresses IP dans l'en-tête d'un datagramme IP effectuée par un routeur. On parlera de SNAT quand c'est l'adresse source du paquet qui est modifiée, et de DNAT quand il s'agit de l'adresse destination.

La NAT **dynamique** est aussi appelée *IP masquerading*. Contrairement à la NAT statique, la NAT dynamique associe `M` adresses internes à `N` adresses externes où $M > N$ (les adresses pour sortir étant choisies dans un *pool*). Ainsi, on peut associer UNE adresse publique à `M` adresses privées et permettre ainsi à un grand nombre de machines ayant des adresses privées d'accéder à Internet. Les adresse internes des machines se retrouvent ainsi masquées derrière une seule adresse publique. Cela a un effet sur la sécurité car les adresses internes sont ainsi dissimulées.

Contrairement à la NAT statique, la NAT dynamique va modier aussi les ports TCP/UDP que l'on appelle PAT (*Port Address Translation*).

Ce type de NAT sert généralement à partager l'accès Internet pour des machines d'un Intranet (réseau local privé). La NAT dynamique ne permettant pas d'être joint par une machine de l'Internet, il ne sera pas possible de rendre un serveur accessible (cf. *port forwarding*).

Le *port forwarding* est une solution pour joindre des machines internes (serveurs) à partir d'Internet avec la NAT dynamique. Cela consiste à rediriger un paquet vers une machine précise en fonction du port de destination de ce paquet (une seule par port TCP/UDP).

L'IETF décourage le NAT avec IPv6 en raison des problèmes liés à certains protocoles et du fait que l'espace d'adresse IPv6 est tel que l'économie d'adresse n'est pas nécessaire.

Sous GNU/Linux, on utilisera `iptables` pour faire du NAT.

HTTP

Le protocole HTTP (*HyperText Transfert Protocol*) sert notamment au dialogue entre un client Web (navigateur par exemple) et un serveur Web (apache ou IIS par exemple). C'est un protocole de la couche Application. HTTP est un protocole orienté texte (ASCII), basé sur TCP et le port 80. Il existe deux spécifications la 1.0 (RFC 1945) et la 1.1 (RFC 2616). Il est utilisé pour la transmission de documents distribués et multimédia.

Les messages HTTP sont basés sur un système de requête/réponse.

La requête de base a la forme suivante : `GET /index.html HTTP/1.1` suivi de 2 retours chariot complet '`\r\n`'. Elle permet de demander le fichier `index.html` sur la racine d'un serveur Web.

La réponse sera composée de deux parties, l'entête qui indique si la requête a réussi et le corps du message (souvent une page HTML), par exemple :

```
HTTP/1.0 200 OK
Content-Type : text/html
Content-Lenght : 56

<HTML><BODY>
Bienvenue sur notre site
</BODY></HTML>
```

Dans ce TP, on utilisera **Apache**. Le logiciel libre *Apache HTTP Server* (**Apache**) est un serveur HTTP créé et maintenu au sein de la fondation Apache. C'est le serveur HTTP le plus populaire du *World Wide Web* et le plus répandu sur Internet. La racine du serveur Web est généralement située dans `/var/www` sur les systèmes GNU/Linux.

```
// connaître l'état du service sur une machine virtuelle ?
httpd:~# /etc/init.d/apache2 status

// arrêter le service sur une machine virtuelle ?
httpd:~# /etc/init.d/apache2 stop

// démarrer le service sur une machine virtuelle ?
httpd:~# /etc/init.d/apache2 start

// redémarrer le service sur une machine virtuelle ?
httpd:~# /etc/init.d/apache2 restart
```

Travail demandé

DHCP

La machine **server** fournit le service DHCP, permettant de configurer les paramètres réseaux de chaque poste client. Lancer `vdump` sur le domaine de collision **A** (*my.network*) et démarrer sur **pc1** et **pc2** l'utilitaire permettant de faire une requète DHCP sur le réseau local :

```
pc1:~# dhclient eth0
pc2:~# dhclient eth0
```

Question 1. En observant les échanges DHCP, trouver les informations suivantes :

a) Quel est l'adresse IP de **server** ?
b) Quel est l'adresse IP affectée à chaque PC ?
c) Quels sont les ports sources et destinations utilisés en DHCP ?
d) Quelle est la pile de protocoles utilisée dans un échange DHCP ?
e) Quelles sont les informations de configuration envoyées par la passerelle aux PC : adresse IP, broadcast, masque, adresse du routeur, adresse du serveur DNS ?
f) Retrouver ces informations sur chaque machine avec les commandes suivantes : `ifconfig`, `route` et dans le contenu du fichier `/etc/resolv.conf`.

Remarque : les paquets ICMPv6 correspondent à l'équivalent d'une demande DHCP pour le protocole IPv6. Nous ne nous y intéresserons pas dans ce TP.

Question 2. Quels sont les messages échangés par DHCP au cours de ces différentes étapes ? On attend une réponse détaillée.

Remarque : vous pouvez arrêter et relancer le client DHCP pour configurer l'interface :

```
pc1:~# killall dhclient
pc1:~# dhclient eth0
```

DNS (1° partie)

gateway, **pc1** et **pc2** ont chacune une adresse DNS dans le domaine **my.network**, c'est-à-dire une association entre une adresse IP et un nom. Lorsque l'on veut communiquer avec une machine on utilise son adresse IP. Par exemple si l'on effectue un ping sur le nom DNS d'une machine on peut voir apparaitre l'adresse IP :

```
pc1:~# ping pc2.my.network
```

Pour savoir l'adresse IP associée à une entrée DNS, chaque machine fait une résolution DNS. Vous pouvez demander une résolution de noms avec les commandes `nslookup`, `host` ou `dig` :

```
pc1:~# nslookup pc2.my.network
pc1:~# host -v pc2.my.network
pc1:~# dig pc2.my.network
```

Question 3. En observant les échanges DNS, trouver les informations suivantes :

a) Quel est l'adresse IP du serveur DNS ?
b) A partir de quel fichier les postes **pc1** et **pc2** ont-ils obtenu cette information ?
c) Quel est le port d'écoute d'un serveur DNS ?

d) Quelle est la pile de protocoles utilisée dans un échange DNS ?
e) Quel est le type de la requête DNS ?

Question 4. Quel est le type de la requête émise par **pc1** lorsqu'on tape la commande : `dig -x adresse_ip_de_poste2` ?

Question 5. Quelle est alors la différence entre les requêtes DNS de type A et les requêtes DNS de type PTR ?

Activer des captures `wireshark` sur les domaines A, B et C.

Question 6. A partir du poste **pc1** ou **pc2**, on n'obtiendra pas de réponse lorsqu'on demande une résolution de nom pour la machine **httpd** du réseau *my.internet* : `host -v httpd.my.internet`. Suivez et analysez les échanges pour répondre aux questions suivantes :

a) A quelle adresse IP est envoyée la requête DNS émise par **pc1** (ou **pc2**) ?
b) A votre avis, le serveur cache DNS **server** de votre réseau local connaît-il l'adresse IP de la machine **httpd** ?
c) Pourquoi ?
d) Que fait alors le serveur cache DNS de votre réseau local ?
e) A quelle adresse IP envoie-t-il la requête DNS ?
f) La requête arrive-t-elle au serveur DNS du réseau *my.internet* ?
g) Le serveur DNS du réseau *my.internet* répond-il à la requête ?
h) Si votre machine n'obtient pas cette réponse, expliquez alors le problème ?

NAT

Comme vous ne pouvez pas modifier la configurations des machines se trouvant en dehors de votre réseau privée (cela représente une configuration présente dans la réalité où chaque boîtier ADSL ne possède qu'une seule adresse IP publique), il faut trouver une solution pour permettre à plusieurs machines d'être derrière une passerelle.

Il faut donc que **gateway** s'occupe de modifier l'adresse source du paquet et les ports afin que la réponse revienne vers **gateway** et puisse être redirigée vers le bon poste (**pc1** ou **pc2**) dans le réseau local.

Pour cela, il faut activer sur **gateway** le mécanisme de translation d'adresse **NAT** (*Network Address Translation*) :

```
gateway:~# iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

Question 7. Donner une commande qui permet de vérifier que la translation d'adresse fonctionne ?

Question 8. Dans une communication, la machine **httpd** pourra-t-elle connaître l'adresse IP du poste **pc1** ou du poste **pc2** ?

DNS (2° partie)

Maintenant que le NAT est mis en oeuvre, chaque machine du réseau local peut résoudre des noms du réseau *my.internet*. Vous pouvez demander une résolution pour les noms suivants :

```
pc1:~# nslookup httpd.my.internet
pc1:~# host -v router.my.internet
pc1:~# dig dns.my.internet
```

Question 9. Donner les trois adresses IP obtenues par la résolution DNS pour les machines **httpd**, **router** et **dns**.

Question 10. A partir de la machine **httpd**, est-il possible de résoudre l'adresse IP correspondant au nom **pc1** avec la commande : `host pc1.my.network` ? Justifier votre réponse.

HTTP

Il y a un serveur HTTP (`apache`) en fonctionnement sur **httpd**. Pour voir le contenu de la page d'accueil (`index.html`), il suffit de lancer `lynx`, un navigateur en mode texte à partir de **pc1** :

```
pc1:~# lynx http://httpd.my.internet/
```

Question 11. Donner le diagramme des échanges complet jusqu'à ce que la page HTML soit retournée (ARP, DNS, ...).

Question 12. Donner la pile de protocoles utilisée par HTTP.

Question 13. Quel est le numéro de port du serveur HTTP (`apache`) ?

Question 14. Comment le client (`lynx`) connaît-il le numéro de port utilisé par le serveur (`apache`) ?

Question 15. Quel est le numéro de port du client HTTP (`lynx`) ?

Question 16. Qui a choisi ce numéro de port ?

Question 17. Comment le serveur (`apache`) connaît-il le numéro de port utilisé par le client (`lynx`) ?

Question 18. Quelle est la requête HTTP envoyée par le client (`lynx`) pour obtenir la page HTML ?

Question 19. Quelle est la version du protocole HTTP utilisée dans cet échange par le client (`lynx`) ? Quelle est celle utilisée par le serveur (`apache`) ?

Question 20. A quelle(s) RFC correspondent-elles ?