Comment fonctionne un réseau ?

Un réseau fonctionne par l'addition de 3 couches : Physique, Liaison, Réseau. La couche physique est la couche de niveau le plus bas, elle transmet des trames sous la forme de signaux électriques.

I. La couche liaison ou MAC

Cette couche permet de coordonner les accès multiples et simultanés au même support. Elle se place juste au dessus de la couche physique. Elle est composée du protocole Ethernet.

1. Trames Ethernet

Il s'agit d'une trame de 1518 octets. Il en existe deux sortes : Dix Ethernet II, et 802.3.

Les trames sont de ce type :
Préambule, @Dest (6 octets), @Src (6 octets), Type/longueur (2 octets), Données (1500), Bourrage, FCS (4 octets)

La reconnaissance se fait par adresse MAC :
1 bit (multicaste), 1 bit (local), 22 bits (fabriquant), 24 (product id)

2. CSMA-CD : Gestion des collisions par Ethernet

Écoute pendant l'émission, s'il quelqu'un parle, on envoie du bruit (et on incrémente un compteur), ensuite on attend un temps aléatoire, puis on réessaie, s'il y a encore du bruit pendant que l'on transmet, réenvoie du bruit sauf si on a atteint le nombre maximum d'essai.

3. Logical Link Control

Ce protocole s'encapsule dans la partie donnée de la trames 802.3, pour permettre sa fiabilisation : contrôle d'erreur, de flux (l'émetteur ne sature pas le récepteur) et de séquence (trames dans l'ordre, numérotation des trames).

Les données sont de ce type :
DSAP (1 octet, id du protocole destinataire supérieur), **SSAP** (1 octet, id du protocole récepteur supérieur), **CTRL** (1/2 octet), **Données encapsulée**
DSAP = Destination Service Access Point : identifiant de protocole émetteur de la donnée encapsulée
SSAP = Source Service Access Point : identifiant du protocole récepteur de la donnée encapsulée
CTRL = Permet d'envoyer des messages d'erreurs

LLC type 1 : Déconnecté, pas de contrôle d'erreur
- UI : unumbred information : transmission de donnée
- TEST : sort de ping
- P : flag qui indique une demande de réponse immédiate

LLC type 2 : Connecté, avec contrôle d'erreur
- Ouverture de connexion : SABME ? UA !
- Ouverture de connexion refusée : SABME ? DM !
- Fermeture de connexion : DISC ? UA !
- DM = Disconnect mode, SABME = Set Asyncr. Balanced Mode Extended

Transmission d'une trame :
- CTRL = 0 Ns (7bits), P/F (1bit), Nr (7bits)
- P = Poll, demande de réponse immédiate
- Ns = N° trame émise (modulo 128)
- Nr = N° trame attendue (modulo 128)

Réponse : (avant le timeout)
- Receiver Ready + trame attendue : fenêtre de réception non pleine
- Receiver Not Ready + trame attendue : fenêtre de réception pleine
- Reject + trame attendue : Trame hors séquence
- Possibilité d'envoi de plusieurs trames avant la fin du timeout pour optimiser.

4. Spanning Tree

Évite les boucles lorsque deux chemins sont possibles pour atteindre une même machine. Si une trame en diffusion ou à destination d'une machine inconnue (donc en diffusion) est envoyées, elle se propage dans tout le réseau et

emprunte les deux chemins alternativement. Cette trame fini donc par tourner en boucle et saturer le réseau.
Il faut donc bloquer un des deux chemins.
Le protocole spanning tree s'encapsule dans une trame LLC (liaison).
[2 octets d'identifiants (0x00 0x00), Version (0x00 : 1 octet) Type BPDU (1 octet), Flags (1 octet), BPDU]

Type de BPDU

0x00 : Configuration
0x80 : changement de topologie

Configuration (0x00) :

Id du Commutateur racine (plus petite @MAC+priorité, initialement lui-même, sur 8 octets), Cout du chemin jusqu'à la racine (Métrique : 4 octets), Id émetteur (8 octets), Id du port émetteur (2 octets), âge du msg (nombre de sauts), âge maxi, Période d'envoie de la config (2 octets), Délai de déblocage (Délai minimum entre décision et déblocage effectif : 2 octets).

Un commutateur conserve la meilleure configuration à chaque message. S'il peut accéder à la racine par deux ports, il conserve le port permettant la plus petit métrique.

Lors du démarrage d'un commutateur et du changement de topologie, un port passe dans différents états :

Blocking State, Listening State, Learning State, Forwarding State
ou Disabled State

4. VLAN

Niveau 1 : Ensemble de ports : Mauvais partitionnement fréquent
Niveau 2 : Ensemble d'adresses MAC : Ajout de machines une a une. Changement de topologie transparent.

Liaisons entre commuateurs par trunking : Un seul lien permettant de faire passer les packets de tout les VLAN. Utilisation du VLAN Tagging (802.Q) pour réencapsuler les trames avec leur identifiant de VLAN : le commutateur de destination désencapsule et envoie sur les ports adaptés.

802.1Q : S'encapsule dans le niveau MAC : Ajout d'un champ Tagging dans la trame ethernet : TPID (0x81 0x00 : 16 bits), PCP (Priority Code Point : 3 bits), 0, VID (VLAN Identifier : 12 bits).

II. Couche réseau

1. IP v4 : Datagrammes IP

	Commence par	Réseau	+ Machine = 32 bit
Classe A :	0	7 bits	24 bits (10.0.0.0 à 10.255.255.255)
Classe B :	10	14 bits	16 bits (172.16.0.0 à 172.31.255.255)
Classe C :	110	21 bits	8 bits (192.168.1.0 à 192.168.255.255)

Adresse inconnue : 0.0.0.0
Adresse réseau : réseau.000
Diffusion dans le réseau : réseau.111
Rebouclage : 127.?

Entête 20 octets : Version, Internet header length (nombre de mots de 4 octets), type of service, total length, identification, flags (DF : don't fragment), fragment offset (*8 : position dans le datagramme entier), **time to live** (TTL décrémenté par les routeurs), protocol, header checksum, @Sources, @Dest, Options.

S'encapsule dans une trame DIX Ethernet (ou LLC)

ARP : Adresse Resolution Protocol : Corrélation IP / MAC :

Ethertype : 0x0806
Requête à : @IP, 00:00:00:00:00:00
Réponse de @IP

Gratuitous : réponse sans requête pour faire de l'ARP poisonning.

Id adressage physique (16bits), Id adress réseau (16bits), long adr phy (8bits), long adr rés (8 bits), Code Op (16 bits), Adr phy émetteur (32bits), adr réseau émetteur (32bits), adr phy dest (32bits), adr rés dest (32bits).
Code Op : 0 : question, 1 : réponse

RIP v1 : Routing Information Protocol : Encapsulé dans une unité UDP.
Niveau IGP : Interior Gateway protocol
Toutes les 30 secondes : échange des tables de routages.

Adresse IP, métrique (=16 si la route est invalide)

ICMP : Supervision de la couche réseau :
ICMP ECHO : ping
Requête : 8 / 0 :
8 (8bits)/0 (8bits)/Checksum (16bits)
Id de message (16bits), Numéro de séq (16bits)
Données (variable)
Réponse : 0 / 0

ICMP DESTINATION UNREACHABLE :
3 / Détail (0 : réseau inaccessible, 1 : machine inaccessible, 4 : fragmentation nécessaire mais interdite...)
0x00 0x00 0x00 0x00
En tête + 8 premiers datagrammes ayant générés l'erreur.

TIME EXCEEDED : Le TTL a expiré pendant le trajet
11 / 0 :
0x00 0x00 0x00 0x00
En tête + 8 premiers datagrammes ayant générés l'erreur.

MTU :
C'est la taille maximale d'un paquet. Le MTU minimal vaut 576 octets.

PING : Au moment d'un ping : recherche dans le cache ARP de la présence de l'adresse MAC correspondant à l'IP. La machine pingée ajoute temporairement la corrélation MAC/IP de la machine émettrice. 5 secondes plus tard, elle vérifie la validité par une requête ARP.

NetFilter, avec iptables

Il existe trois tables :
Le Mangle que nous n'utiliserons pas

La table NAT : `iptables -t nat`
Attention, pour pouvoir configurer une passerelle NAT il faut autoriser le forwarding :
`sysctl -w net.ipv4.ip_forward=1`

Avec deux chaines : PREROUTING, ce qui arrive dans la machine et POSTROUTING, ce qui est renvoyé par la machine.
Trois modes (cibles) :
- **DNAT** (modifie l'adresse de destination des paquets, pour redistribuer dans le réseau les packets à destination de la passerelle connectée à Internet) :
ex : Les paquets TCP arrivant sur le port 80 sont envoyés à la machine 192.168.1.4 sur le port 81
`iptables -t nat -A PREROUTING -i eth1 --destination-port 80 --to-destination 192.168.1.2 -j DNAT`

- SNAT (modifie l'adresse source du paquet, pour faire croire aux machine d'Internet qu'il n'y a qu'une machine dans le réseau local.)
`iptables -t nat -A POSTROUTING -p tcp -o eth0 -j SNAT --to-source 10.42.3.0.53:1024-32000`
Modifie le paquet avec un port source aléatoire (plus ou moins) entre 1024 et 32000.

- **MASQUERADE** (Donne l'impression que les paquets sortent de la machine, et sont à destination de la passerelle. Il s'agit d'un SNAT amélioré où il n'est pas nécessaire de spécifier une adresse sources, car elle est récupérée dynamiquement, utile lorsque l'on ne possède pas d'IP fixe sur Internet)

ex : Tous les paquets passant sur la machine en direction de eth1 sont modifiés

```
iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

La table FILTER : `iptables -t filter`

La table *filter* permet de définir des règles de pare-feu.

`iptables -F` : permet de vider les tables.

Pour chaque paquet, **NetFilter évalue les règles dans l'ordre**, et s'arrête quand il a trouvé une règle correspondante.

On distingue 3 chaines :

-A INPUT : pour les connexion entrantes dans la machine à destination de la machine

-A OUTPUT : pour les connexions sortante de la machine, en provenance d'elle même

-A FORWARD : pour les connexions passantes par la machine (d'une interface à une autre) qui ne lui sont pas destinées

Précisions sur le paquet :

Le protocol utilisé : --protocol tcp, --protocol udp, --protocol ICMP...

Les ports : --source-port 3128, --destination-port 3128 : permet de bloquer les paquets envoyés depuis un certain port, ou à destination d'un autre (typiquement pour bloquer certaines application : web (80), Mail (25 pour SMTP, 110 pour POP, et 143 pour IMAP), FTP (21).

Il faut ensuite choisir une action à effectuer :

-j ACCEPT : laisse circuler le paquet

-j DROP : supprime le paquet

-j LOG : enregistre le paquet dans un log, pour l'administrateur

Par exemple :

Autorisation de forwarder l'ICMP :

```
iptables -t filter -A FORWARD --protocol ICMP -j ACCEPT
```

Autoriser les connexion au serveur web présent sur la machine :

```
iptables -t filter -A INPUT --protocol tcp --destination-port 80 -j ACCEPT
```

Pour être sûr de ne rien avoir oublié, on interdit le reste (les règles sont évaluées dans l'ordre) :

```
iptables -t filter -A OUTPUT -j DROP
iptables -t filter -A INPUT -j DROP
iptables -t filter -A FORWARD -j DROP
```

On aurait aussi pu raisonner à l'inverse, et tout autoriser sauf certaines règles.

Serveur DHCP

Le DHCP a deux rôles : attribuer des adresses IP, et permettre de diffuser des informations de configuration.

Lancement du serveur : `dhcpd -d -cf dhcpd.conf`

Lancement du client : `dhclient -d eth0`

Fichier de configuration commenté :

`ddns-update-style none;`
`default-lease-time 200;`
`max-lease-time 300;`
	Spécification du temps entre deux mises à jour. On spécifie un défaut, et un temps maximal.

`subnet 172.26.1.0 netmask 255.255.255.0 {`
`  range 172.26.1.100 172.26.1.104;`
`  range 172.26.1.106 172.26.1.124;`
`  option routers 172.26.1.1;`
`}`
	On définit les adresses à attribuer dans le sous réseau 172.26.1.???.
	On spécifie deux plages d'adresses : de 172.26.1.100 à 172.26.1.106, et de 172.26.1.124.
	Option routers, sert à spécifier la passerelle par défaut sur

	toutes les machines, pour éviter de les configurer une a une.

`host adri_comp {`
`  hardware ethernet 00:10:18:33:9a:59;`
`  fixed-address 172.26.1.102;`
`}`
	Ici on spécifie une adresse ip fixe pour une certaine machine identifiée par son adresse MAC. Ici *adri_comp* n'est qu'un nom, (même si c'est pas le nom de n'importe qui).

Quelques commandes en vrac...

Quelques commandes :
arping -i -S -c
Cisco :

Utilisateur privilégié : enable (disable) => Conf globale : configure teminal (exit) =>
mac-adress-table aging-time (modifie le délai d'expiration de la table ARP) => vlan *id*, ou interface *id* (ou interface range).
En mode configuration globale : mac-adress-table static [AdrMac] vlan [v] interface [port]
show spanning-tree
no spanning-tree vlan 1
vlan i => name *nom*
switchport mode access (ou mode trunk pour vlan)
switchport trunk native vlan v (spécifie le vlan natif qui ne sera pas encapsulé pour limiter le trafic)

Active le forwarding sur la machine : sysctl -w net.ipv4.ip_forward=1

Merci à Raphael, pour son cahier de bord.