



Ethernet 100baseT

Fast Ethernet

<i>FERRARA Arnaud</i>	<i>GI04</i>
<i>LOURD Rodolphe</i>	<i>GI04</i>
<i>SKORA Ludovic</i>	<i>GI05</i>
<i>VINZIA Sébastien</i>	<i>GI05</i>
<i>VIRGA Jérémy</i>	<i>GI04</i>

Introduction

Proposée au départ par 3Com et surtout par Intel, la norme Fast Ethernet a conduit à la création du groupe de travail IEEE 802.3u et se décline en plusieurs supports dont 100baseT4 et le plus populaire d'entre tous 100baseTX. Ce support utilise comme 10baseT les paires torsadées mais sur une longueur réduite à 100 mètres pour autoriser le débit de 100 Mbps.

L'intérêt principal de ce standard est la compatibilité avec l'Ethernet 10 Mbps et le coût environ multiplié par 2 mais pour une vitesse 10 fois supérieure.

Mesures de base

Nous mesurons tout d'abord le temps mis pour effectuer un même transfert en 10 et 100 Mbps. Pour cela, nous utilisons deux machines reliées par un switch sur lequel nous pouvons forcer la vitesse (10/100 Mbps) et le mode (Half-Duplex/Full-Duplex) de transmission.

Il faut donc configurer les deux machines sur le même réseau local :

- *PC-GI-S45* > *ifconfig eth0 192.168.2.69 netmask 255.255.255.0 up*
- *PC-GI-S44* > *ifconfig eth0 192.168.2.68 netmask 255.255.255.0 up*

Nous obtenons les configurations suivantes :

Machine	Adresse IP	Masque de sous-réseau	Interface
<i>PC-GI-S45</i>	<i>192.168.2.69</i>	<i>255.255.255.0</i>	<i>Eth0</i>
<i>PC-GI-S44</i>	<i>192.168.2.68</i>	<i>255.255.255.0</i>	<i>Eth0</i>

Ensuite nous créons un fichier de taille suffisamment conséquente à l'aide de la commande suivante :

- *dd if=/dev/vrandom ob=toto bs=1024 count=10240*

pour obtenir un fichier rempli de données prises au hasard et de taille environ 10 Mo (suffisant pour visualiser une vitesse de transmission dans un cas le plus général possible).

Pour configurer la vitesse et le mode de transmission du switch, nous utilisons la commande suivante :

- *mii-tool -F 100baseT-FD*
- *mii-tool -F 100baseT-HD*
- *mii-tool -F 10baseT-FD*
- *mii-tool -F 10baseT-HD*

Enfin un serveur FTP est démarré sur une des deux machines à l'aide de la commande *proftpd* (ici sur *PC-GI-S45*), l'autre machine se connecte dessus et télécharge le fichier à l'aide de la commande suivante :

- *wget ftp://192.168.2.69/toto*

Une fois le transfert terminé, nous pouvons observer la vitesse moyenne de transfert :

Vitesse de transmission	Mode de transmission	Débit mesuré
100 Mbps	Half-Duplex	11,24 Mo/s
10 Mbps	Half-Duplex	85,84 Ko/s

Logiquement, le débit engendré par le transfert 10baseT est très largement inférieur à son homologue 100baseT ; nous obtenons ici un rapport supérieur à 10 qui approche celui des débits théoriques respectifs des deux interfaces (12,5 Mo/s et 1,25 Mo/s).

Le mode de transmission Full-Duplex

Le Full-Duplex est un mode de transmission où les voies sont bi-directionnelles. Nous pouvons alors monter la vitesse d'un lien en 100baseTX à 200 Mbps ce qui en fait théoriquement une alternative sérieuse à ATM ; ce mode permet également de supprimer les collisions de paquets préjudiciables à l'efficacité du lien.

Après avoir exécuté la commande *ifconfig* pour vérifier que les interfaces réseau sont bien en Full-Duplex, nous procédons à un transfert analogue à la précédente partie, mais cette fois-ci en Full-Duplex :

Vitesse de transmission	Mode de transmission	Débit mesuré
100 Mbps	Full-Duplex	11,56 Mo/s
10 Mbps	Full-Duplex	88,70 Ko/s

Les résultats obtenus sont très similaires à ceux observés précédemment. L'augmentation du débit est finalement assez négligeable. Le Full-Duplex devrait prendre moins de la moitié du temps mis en Half-Duplex pour une même vitesse de transmission, et la différence entre 2 fois le temps en Full-Duplex et le temps en Half-Duplex devient le temps perdu à cause des collisions :

- $$T_{col} = 2 * T_{fdx} - T_{hdx}$$

Mais en pratique l'augmentation n'est pas si importante. A ce niveau-là, le transfert est bridé par le type de câble réseau utilisé (en l'occurrence un CAT5) donc nous ne pouvons pas vraiment voir le gain occasionné par le changement de mode.

Pour avoir plus de chance de voir les collisions il faudrait envoyer en même temps depuis plusieurs machines vers un même serveur, cela devrait augmenter leur nombre. Celui deviendrait d'autant plus important que les machines seraient éloignées.

L'inconvénient principal du Full-Duplex est que l'utilisation d'un répéteur n'est pas possible tout le long de la ligne en Full-Duplex : ce mode de transmission n'est disponible qu'en point à point.

La co-existence de 10baseT et 100baseTX sur un même réseau

Cette co-existence peut poser des problèmes de congestion sur le réseau le plus lent. En effet, de par la vitesse des *broadcast*, *multicast* et *unicast*, un point de congestion se forme aux extrémités des liaisons les plus lentes, ce qui oblige les routeurs, hubs et switches à plus de traitement ou les amène à plus de collisions.

Ceci se vérifie en pratique. En effet les mesures précédentes ont été effectuées lorsque le switch était isolé des autres machines du bâtiment ; il n'y avait donc qu'une dizaine de machines connectées dessus. En revanche, une fois les autres machines connectées au switch, les résultats furent les suivants :

Vitesse de transmission	Mode de transmission	Débit mesuré
10 Mbps	Half-Duplex	26,30 Ko/s
10 Mbps	Full-Duplex	52,37 Ko/s

Concernant le 100baseT, nous n'avons pas remarqué de changement de débit notable ; mais pour le 10baseT, la diminution de débit est flagrante par rapport aux mesures initiales. Cela nous permet au passage de mieux visualiser le gain occasionné par le passage du Half-Duplex vers le Full-Duplex où nous retrouvons le double de débit théorique.

Le bonding

Le *bonding* est une solution incluse dans le noyau Linux qui permet de créer une seule interface virtuelle à partir de deux physiques pour partager la bande passante disponible. C'est une sorte de répartition de charge (*load balancing*).

Nous utilisons cette possibilité en module. Il faut donc charger ce dernier et ce pour chaque interface virtuelle que nous voulons utiliser :

- `modprobe bonding`

Les machines sont connectées de la manière suivante : un câble croisé relie les interfaces eth1 respectives ; les interfaces eth0 sont reliées normalement au switch.

Ensuite il faut configurer les interfaces sur *PC-GI-S45* :

- `ifconfig eth0 192.168.2.69 netmask 255.255.255.0 up`
- `ifconfig eth1 192.168.2.69 netmask 255.255.255.0 up`
- `ifconfig bond0 192.168.2.69 netmask 255.255.255.0 up`

Puis sur *PC-GI-S44*:

- `ifconfig eth0 192.168.2.68 netmask 255.255.255.0 up`
- `ifconfig eth1 192.168.2.68 netmask 255.255.255.0 up`
- `ifconfig bond0 192.168.2.68 netmask 255.255.255.0 up`

Ceci fait, nous utilisons l'utilitaire *ifenslave* pour déclarer les interfaces physiques à rattacher à cette interface virtuelle sur les deux machines :

- `ifenslave bond0 eth0`
- `ifenslave bond0 eth1`

Pour tester le processus mis en oeuvre, nous utilisons tout d'abord la commande *ping* entre les deux machines. Que ce soit dans un sens ou d'autre, les résultats sont très médiocres puisque environ 80 % des paquets transmis sont perdus, contre 100 % dans un transfert classique. Cette dégradation des performances est confirmée par le transfert FTP, puisque le débit avoisine les 100 o/s, ce qui est très loin des 11 Mo/s obtenus en 100baseTX Full-Duplex.

Lors d'une vérification des interfaces avec *ifconfig*, nous nous sommes aperçu que le taux de transfert de l'interface eth1 était de 1,2 Ko/s, alors que celui de l'interface eth0 était de 12 Mo/s. Le dysfonctionnement du *bonding* est peut-être dû à la faiblesse de l'une des deux cartes réseaux (en l'occurrence eth1), ce que nous n'avons pas pu vérifier car cela nécessitait un changement matériel.

Hormis cet « accident » pratique, le *bonding* semble très intéressant, notamment pour les gros serveurs qui envoient/reçoivent énormément de requêtes/données. Un tel procédé permet de répartir la charge sur plusieurs cartes réseau, ce qui peut devenir très efficace lorsque ce serveur est connecté à un réseau à très forte bande passante (plusieurs interfaces 100 Mbps pouvant remplacer avantageusement une interface Gigabit notamment pour des raisons de coût financier).