

Exercice 1. Questions / discussions de cours

Q1.1 Une adresse MAC est attachée à une carte réseau physique, et n'est visible que sur le réseau local. Elle est sur 48bits. Une adresse IP est virtuelle, et est valide sur tout Internet (quand elle n'est pas masquée). Elle est sur 32bits pour IPv4, 128bits pour IPv6. Une adresse MAC est utilisée au niveau 2 (Ethernet) par la carte réseau physique, alors qu'une adresse IP est utilisée au niveau 3 par le système d'exploitation.

Q1.2 Quand une troisième machine C essaie d'envoyer un paquet à cette adresse IP, elle va commencer par envoyer une requête ARP pour obtenir l'adresse MAC correspondant à cette adresse IP. Les deux machines A et B vont répondre en donnant leur propre adresse MAC. La machine C prendra en compte la réponse qui arrive la première (par exemple celle de A), et l'autre (donc B) sera alors injoignable. Un peu plus tard avec de la chance ce sera B dont la réponse arrivera en premier, et c'est alors A qui sera injoignable.

Q1.3 Pour la redondance : si un des serveurs tombe en panne, on pourra utiliser automatiquement un autre serveur. Avec une randomisation de la réponse, on peut aussi répartir la charge.

Q1.4 Cela permet de gérer plusieurs clients à la fois : à chaque nouveau client connecté on obtient un nouveau socket, avec lequel on pourra échanger avec le client, tout en pouvant échanger encore avec les autres clients en utilisant leurs sockets respectifs.

Q1.5 Cela permet par exemple aux pigistes de se concentrer sur le contenu sans se soucier de la forme, et aux designers de se concentrer sur l'aspect visuel sans se soucier du contenu ; chacun son métier. Cela permet aux lecteurs d'écran de montrer le fond à l'utilisateur, sans être perturbé par la forme.

Exercice 2. Adresses

Q2.1 C'est un réseau /24, donc $2^{32-24} = 256$ adresses (moins les 2 extrémités). Pour 300 chambres, si chaque étudiant branche un PC (voire plus), cela ne suffira pas. Il faudrait plutôt utiliser au moins 192.168.0.0/23 voire plus grand.

Q2.2 Elle contient $2^{128-64} = 2^{64}$ adresse, des milliards de milliards, c'est très large pour 300 chambres :)

Exercice 3. Calculs (on pourra faire des arrondis de calculs grossiers si l'on n'a pas de calculatrice). Précisez le déroulement de votre calcul.

Q3.1 800Mbps cela fait 100Mo/s, en un jour cela permet d'envoyer $100 \times 3600 \times 24$ Mo =(très optimiste pour simplifier le calcul) $100 \times 4000 \times 25$ Mo = 10To. On a été très optimiste en arrondissant le calcul, ça ne va pas passer...

Q3.2 Non, car TCP va de toutes façons pipeliner l'envoi, peut importe le temps que les données vont arriver à destination, c'est la cadence à laquelle il envoie les données qui vont permettre au transfert d'être effectué rapidement. Le très court délai au tout début (pour le premier envoi) et à la toute fin (pour le dernier acquittement) est très négligeable devant le temps de transférer un film.

Q3.3 Il faut faire passer 2To en $1j = 3600 \times 24s = 4000 \times 25s = 100\,000s$ soit environ 160Mbps. Il faut donc compter environ 160€/mois (mais plus exactement un peu plus car on a arrondi pas mal à la hausse).

Exercice 4.

Q4.1 10.0.0.33

Q4.2 $0x0d = 13$, $0x402 = 1026$

Q4.3 C'est l'émetteur qui a un numéro de port inférieur à 1024, donc c'est très probablement lui le serveur. On se souvient peut-être qu'en TP le port 13 était le port daytime.

Exercice 5.

Q5.1 S'il faut attendre un aller-retour pour tracer un seul trait, si l'affichage voulu comporte de nombreux traits il faudra attendre de nombreux aller-retours! Une interface graphique un peu compliquée peut comporter une centaine de traits, même avec 10ms de latence (ce qui n'est pas si grand que cela), il va falloir attendre 1s avant d'avoir l'affichage.

Q5.2 Ici on n'attend plus qu'un seul aller-retour : on envoie en vrac la centaine de demandes de traits (compter peut-être quelques dizaines d'octet par trait, cela représente donc quelques kilo-octets ; même sur un ADSL à 8Mbps, on le reçoit en quelques ms seulement). puis on attend l'aller-retour, et l'on a fini en moins de 100ms, ce qui reste acceptable humainement.

Q5.3 Envoyer des images est lourd. Pour une application plein écran $1920 \times 1080 = 2000 \times 1000$ il faut donc réceptionner 2 million de pixels, 3 octets par pixel typiquement, donc 6Mo. Avec un ADSL à 8Mbps, il faut plusieurs secondes. On peut compresser les images, on peut peut-être espérer tout au mieux un facteur 10 (puisqu'il y a de gros aplats, etc.), ce qui va réduire d'autant le délai, on risque d'être cependant bien au-delà des 100ms acceptables humainement.

Q5.4 le curseur qui a bougé, c'est *beaucoup* moins que 2000×1000 , on pourrait dire par exemple $32 \times 32 = 1000$ pixels, donc quelques ko à réceptionner seulement, on reçoit en quelques ms.