

ACTIVITE 4 - LA SEGMENTATION DES DONNEES EN PAQUETS

Une difficulté majeure qui pèse sur les réseaux, et en particulier un réseau qui permet à un très grand nombre de machines de communiquer simultanément, c'est la saturation des liaisons et plus encore des **nœuds** (sur Internet, les **routeurs**). La congestion du réseau est comparable aux bouchons routiers, qui se forment le plus souvent aux croisements. Des documents volumineux sont même comparables à un train, à un passage à niveau, qui bloquerait entièrement une route de manière prolongée. Dans les années 90, une simple image était considérée comme un document volumineux comparativement aux textes dont étaient principalement composés les messages et les pages web. Le développement d'Internet n'aurait pu se faire sans des **stratégies ingénieuses pour fluidifier** le trafic à moindre coût (car l'augmentation de la puissance des machines ou de la capacité des liaisons est très chère et n'aurait pas permis la démocratisation du réseau).

4.1

Pour vous donner un ordre d'idée, calculez le temps nécessaire à la transmission d'un document de 100 ko (l'équivalent d'une photographie compressée de qualité médiocre) depuis un ordinateur personnel et un modem de 56 kbits/s (soit le débit maximal avant l'apparition de l'ADSL en France en novembre 1999), puis avec la 4G sur un smartphone (environ 10Mbits/s en émission au maximum, et 10 fois plus en réception). Si vous aimez la difficulté, pour obtenir un résultat exact prenez en compte la différence entre 1ko et 1000 octets (la connaissez-vous ?), sachant en revanche que 1kbits = 1000 bits.

$$100 / (56 / 8) = 14,29 \text{ s}$$

$$\text{Calcul exact : } (100 \times 1024) / (56 \times 1000 / 8) = 14,63 \text{ s}$$

$$100 / (10\,000 / 8) = 0,08 \text{ s}$$

Calcul exact ici : différence qui n'apparaît pas compte tenu de l'arrondi à 2 décimales

4.2

Faites le même type de calcul dans le cas de la réception (téléchargement) :

- d'une page web de taille moyenne aujourd'hui (environ 2,5 Mo)

- d'un film de 1 Go, en comparant le modem 56 kbits/s à une box fibre optique (capacité théorique de 1 Gbits/s mais en réalité plutôt autour de 400 Mbits/s : faites le calcul avec ce dernier chiffre).

$$(2,5 \times 1000) / (56 / 8) = 357,14 \text{ s (soit 5min57)}$$

Calcul exact : $(2,5 \times 1024^2) / (56 \times 1000 / 8) = 374,49$ (soit 6min24), la différence n'est pas négligeable !

$$1\,000\,000 / 7 = 142\,857,14 \text{ s soit plus de 39h40min}$$

$$\text{Calcul exact : } 1024^3 / 7000 = 153\,391,69 \text{ s soit plus de 42h36min}$$

Box fibre :

$$2,5 / (400 / 8) = 0,05 \text{ s}$$

$$\text{Calcul exact : } (2,5 \times 1024^2) / (400 \times 1000^2 / 8) = 0,0524 \text{ s}$$

$$1\,000 / (400 / 8) = 20 \text{ s}$$

$$\text{Calcul exact : } 1024^3 / (400 \times 1000^2 / 8) = 21,47 \text{ s}$$

4.3

Sur un réseau où les nœuds d'interconnexion ne peuvent traiter qu'un seul message à la fois (donc pas Internet), une personne P1 veut transférer un film de 1Go à l'instant t. A l'instant t+1s, une personne P2 veut simplement accéder à une page web quelconque. Que se passera-t-il ?

Il y a une file d'attente : P2 doit attendre que P1 ait entièrement chargé son film (moins 1 seconde) avant de pouvoir charger sa petite page web. Cela est évidemment inenvisageable, il faut trouver une solution !

Pour se donner un autre ordre d'idée, on peut imaginer que la transmission d'un film de 1Go pendant qu'un petit document de 100ko attend son tour, c'est comme si des voitures attendaient au passage à niveau qu'un train de 40km de long finisse de passer. Et si tout le monde se déplaçait en train ? Un **réseau cadencé et centralisé** comme le réseau ferroviaire n'a évidemment rien à voir avec Internet, ou chacun peut communiquer à tout moment et s'attend à un résultat quasi instantané. Internet ressemble plutôt au réseau routier, mais en beaucoup plus réactif (et avec la possibilité de littéralement volatiliser les « véhicules » en cas de trop forte congestion, comme on le verra).

L'idée (plus ancienne qu'Internet mais qui y trouve pleinement son intérêt) est de fragmenter l'information en un grand nombre de petits paquets et d'envoyer chacun de ces **paquets indépendamment les uns des autres**, quitte à ce qu'ils arrivent dans le désordre à destination, voire par des chemins différents. Nous allons essayer de comprendre tout l'intérêt de cette idée étonnante.

4.4

Supposons que 10 personnes veulent télécharger en même temps un document volumineux. Que se passera-t-il si le document n'est pas envoyé par petits paquets ? D'après vous, quels sont les avantages de l'envoyer par paquets ? (sachant que chaque message, qu'il s'agisse d'un document entier ou d'un paquet, doit être entièrement reçu par un routeur pour qu'il puisse lui-même le retransmettre au routeur suivant, jusqu'à destination)

Envoi en bloc entier : file d'attente, chacun attendra que ceux qui se sont connectés avant lui aient successivement entièrement achevé leur téléchargement.

Envoi par paquets : pour qu'il y ait un intérêt, il faut que les paquets soient dupliqués 10 fois et répartis entre les 10 personnes, pour que tous reçoivent à peu près en même temps le document.

On pourrait croire que le temps de chargement sera multiplié par 10 pour tous, par rapport au temps nécessaire à un hôte seul, ce qui ne semble pas très avantageux ! Autrement dit, tous semblent y perdre (si on ne prend pas en compte d'autres paramètres comme les limites de capacité du serveur, des câbles, de l'hôte, etc.)

En fait, chaque message étant transmis plus vite au routeur suivant, celui-ci peut commencer à le retransmettre immédiatement, sans attendre la réception des paquets suivants. En outre, les chemins suivis vers les 10 hôtes seront probablement différents en certains points, ce qui réduira encore les files d'attente.

Montrer les schémas sur la commutation de paquets sur une frise temporelle (cf. diapo)

4.5

Quelles solutions techniques peut-on envisager pour accélérer le transfert lorsqu'un grand nombre de personnes essaient de télécharger un même document volumineux ? (songez que pour le dernier épisode de Games Of Thrones, qui a été vu par plus de 17 millions de personnes en direct lors de sa première diffusion et piraté par plus de 55 millions...) Peut-on se passer d'un serveur central, très puissant, qui stocke et délivre les documents volumineux et très demandés, et qui contrôle qui transfère quoi ?

- copier sur plusieurs serveurs ? (redondance coûteuse, et qui ne résout pas forcément le problème de saturation si l'information passe par un même routeur) ou serveur multiprocesseur, ou capable d'opérer plus traitements simultanément ? Idem, on arrive assez vite aux limites de ces technologies même si elles sont très utiles (mais également coûteuses)
- chacun récupère des parties différentes du même document et joue le rôle de "serveur" concernant les parties qu'il a récupérée pour d'autres hôtes qui

téléchargent le document... => c'est le principe de "pair à pair", que certains élèves seront peut-être capable de redécouvrir ou de rappeler.

- => on arrive au paradoxe que: plus il y a de monde qui télécharge, plus ça va vite (la limite ici étant en général la capacité des liaisons, en particulier en bout de parcours, chez l'utilisateur lambda, ou la capacité de traitement de son ordinateur (en particulier son disque dur))
- C'est l'occasion de réfléchir aux conséquences de la possibilité de se passer d'un serveur central sur le respect des droits d'auteurs

Découper et rassembler les paquets**4.6**

Voici 64 octets extraits du tout début d'un document texte contenu les paroles de « Bohemian Rhapsody », du groupe Queen (présenté en hexadécimal, pour faciliter la présentation, 2 caractères hexadécimaux étant codés sur 1 octet) :

```
49 73 20 74 68 69 73 20 74 68 65 20 72 65 61 6C
20 6C 69 66 65 3F 0D 0A 49 73 20 74 68 69 73 20
6A 75 73 74 20 66 61 6E 74 61 73 79 3F 0D 0A 43
61 75 67 68 74 20 69 6E 20 61 20 6C 61 6E 64 73
```

(Correspond exactement au texte :

« Is this the real life?
Is this just fantasy?
Caught in a landslide
No escape »)

Voici 64 octets extraits de fichier musical, mp3, de la même chanson :

```
2E 0E 1C 20 11 79 F2 F3 F5 17 F7 EC DF F0 01 18
04 E1 17 0F 0A 75 31 D0 4E 15 7B 87 BD D3 38 79
02 99 81 4A 76 B4 F1 AE F4 FF 06 D9 4E C6 7F 9C
62 3E 07 C0 1B 06 22 18 F1 FF FB 92 4C FF 8F F0
```

Que remarquez-vous ?

Ces données sont différentes mais leur structure est très similaire, en fait indiscernable si on ne sait pas par ailleurs s'il s'agit de retrouver, derrière le code hexadécimal, du texte, du son, de l'image, etc. (il faut un marqueur, quelque part, qui le stipule : une

métadonnée). Mais chaque segment est semblable à un autre et devrait être traité de manière semblable sur Internet.

→ Faire le lien avec le principe de la **neutralité du net**.

Songer que près de 80% du trafic sur Internet aujourd'hui concerne des vidéos (dont plus d'un tiers de pornographie, à vérifier). Demander aux élèves : faut-il limiter leur transfert ? Faire en sorte que d'autres types de données soient prioritaire ?

4.7

Imaginez que vous deviez découper avec un ciseau le texte papier de la chanson pour l'envoyer en plusieurs morceaux (de moins de 32 caractères, espaces compris), sans garantie qu'ils arrivent dans l'ordre au destinataire. Quelle stratégie adopteriez-vous pour que le destinataire reconstitue la chanson correctement et être sûr qu'il a bien reçu tous les paquets ? Et s'il s'agissait d'envoyer les morceaux d'une image papier, comment vous prendriez-vous et quelles autres difficultés rencontreriez-vous ?

Découper par mots ou par vers ? Mais ils sont de taille inégale. Et il faut s'assurer d'abord qu'aucun vers ne fait plus de 32 caractères. Sinon choisir une longueur arbitraire, comme 32. Mais alors cela pourra tomber au milieu d'un mot, ce ne sera pas très lisible, etc. On voit qu'il y a une difficulté pour choisir la mesure.

Pour que le destinataire reconstitue la chanson, il faudrait bien sûr numéroter et soit indiquer le nombre total de morceau, soit indiquer sur le dernier morceau qu'il s'agit du dernier, et demander que le destinataire indique, en retour, s'il a tout reçu ou quels morceaux il n'a pas reçu (on pourrait ici demander de décrire plus précisément l'algorithme, car il s'agit bien d'un algorithme même s'il n'est pas informatisé). Demander un accusé de réception pour chaque morceau et le renvoyer si, au bout d'un certain temps, il n'est pas reçu, comme le fait TCP, semble fortement compliquer la tâche manuelle et les élèves n'en auront probablement pas l'idée.

Pour une image, il ne faut pas seulement un numéro ordinal mais une coordonnée sur une matrice => faire le lien, évidemment, avec le thème de la Photo numérique.

4.8

Imaginez maintenant que la même chanson doive être envoyée au format numérique (comme ci-dessus), en paquets de 4 octets maximum : quels sont les avantages d'un traitement informatique d'un tel transfert par rapport au traitement manuel (ou en général par rapport au traitement analogique) ?

Trois principaux avantages :

- Le découpage peut être de longueur quelconque (le contenu numérique est homogène) et fixe.

- Le découpage, la remise en ordre à la réception, et le contrôle de l'intégrité de la transmission peuvent être automatisés grâce à des programmes (c'est le rôle des algorithmes définis par le protocole TCP).
- Un découpage numérique est exacte (entre 2 bits, c'est une réalité discrète), alors que le découpage de données analogiques peut être approximatif, imparfait, rendant difficile la reconstitution exacte de l'objet (par exemple, le découpage physique d'un papier altère les bords, les recoller est difficile sans perdre quelques dixièmes de millimètres, laisser du jeu, etc. – du son ou de l'image, sur le plan physique, sont des réalités continues)

→ Présenter ici le rôle du protocole TCP : contrôle et traitement de l'information en bout de chaîne (**principe du « bout à bout »**, qui répartit la charge sur les utilisateurs, dispense de serveur central qui traite et contrôle tout.) Le réseau entre deux bouts est seulement censé faire transiter les paquets sans intervenir (d'où le **principe de la neutralité du net**). D'ailleurs les paquets TCP sont encapsulés dans un paquet IP qui théoriquement n'est pas « ouvert » par les intermédiaires : ce peut donc être aussi l'occasion de présenter le **principe de l'encapsulation** et du modèle en couches.

4.9

Sur les réseaux très longue distance comme Internet, il y a régulièrement des erreurs de transmission (une partie de l'information transmise est altérée ou perdue). Dans ces conditions, expliquez en quoi le transfert en paquets multiples est une meilleure solution que le transfert d'un seul bloc.

S'il y a une erreur avec la transmission d'un document envoyé en bloc, il faut le renvoyer en entier. S'il y a une erreur avec un paquet, il suffit de renvoyer le paquet, ce qui est très peu coûteux. Entre temps, les autres paquets reçus sont préservés, il suffit d'un algorithme qui remette à la bonne place le paquet manquant.

4.10

Les contrôles et les efforts fournis pour assurer la transmission intégrale des paquets constituant un fichier sont chronophages et mobilisent des ressources de la machines : il y a des types de données pour lesquels on peut tolérer quelques erreurs ou paquets manquants et où un débit élevé prime sur la fiabilité, lesquels d'après vous ?

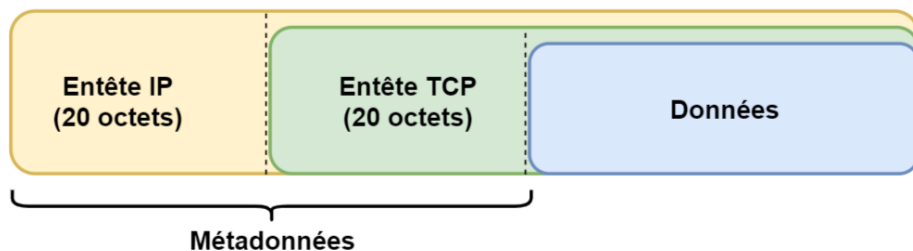
Pour le son ou la vidéo en streaming par exemple, l'essentiel est la rapidité de transmission, mais des paquets manquants sont tolérables (un petit bout d'image manquant dans un film ou une altération très ponctuelle de l'onde sonore dans une conversation passeront inaperçus ou presque, ou les micro-coupures ou altérations seront tolérables dans une certaine mesure)

Dans ce cas, c'est en général le protocole UDP qui est utilisé, dans lequel chaque paquet reçu n'implique pas l'envoi d'un accusé de réception. La logique de la temporalité directionnelle du flux est privilégiée à l'intégrité du contenu.

Tout ce découpage, est-ce bien raisonnable ?

4.11

Voici un paquet TCP/IP (simplifié), où l'on distingue les données transmises et les **métadonnées** requises par les protocoles IP (mise en relation à distance et routage) et TCP (transmission et contrôle) :



Calculez le volume de métadonnées pour le téléchargement d'un film de 1 Go, d'abord pour des paquets de 1500 octets (taille la plus commune actuellement), puis pour 250 (taille commune dans les années 90). Qu'en pensez-vous ?

En considérant seulement les entêtes IP et TCP, soit 40 octets en tout (mais il y a une entête également pour la trame liaison, par exemple Ethernet, de 14o, et souvent une entête au niveau applicatif), on obtient :

Paquets de 1500 octets

$$(20 + 20 / 1500) \times 100 = 2,666... = (8/3) \%$$

Pour 1 Go : 26,66 Mo (ou, plus exactement : $1024 \times (8 / 3) / 100 = 27,31$ Mo)

Paquets de 250 octets

$$(40 / 250) \times 100 = 16 \%$$

Pour 1 Go : 160 Mo (ou, plus exactement : 16,38 Mo)

(A savoir que, pour la VoIP, voix sur IP, les paquets en UDP/IP sont de 60o pour des métadonnées de 28o)

→ Une partie importante de ces métadonnées est redondante (identiques dans tous les paquets correspondant à un même transfert), mais c'est indispensable.

4.12

Cherchez d'autres usages possibles et intéressants (en dehors du gain de rapidité et de fiabilité de la transmission d'information), du découpage de données volumineuses en de nombreux paquets de petites tailles.

On peut penser notamment :

- au traitement **simultané** (en parallèle) de différents types de messages (ce qui est plus la règle que l'exception)
- au traitement et l'**affichage progressif** et à la demande, par exemple pour ne charger que certaines pages d'un livre où pointer à un moment précis d'une vidéo sans devoir charger tout entier
- mais encore à de larges possibilités de **cryptage** ou de sécurisation (c'est en partie sur ce principe que repose la Blockchain)