Protocoles de communication

Les Internets

Ce que l'on appelle **Internet** est un réseau mondial de réseaux. Ces réseaux servent à connecter différents terminaux (ordinateurs, smartphones, consoles de jeu…). Afin de communiquer ensemble, ces terminaux, très différents les uns des autres, utilisent des **protocoles**. Les protocoles permettent de préciser la forme des messages à s'envoyer et dans quel ordre pour que les communications puissent aboutir. Les deux principaux à la base d'Internet sont TCP et IP.

Internet Protocol

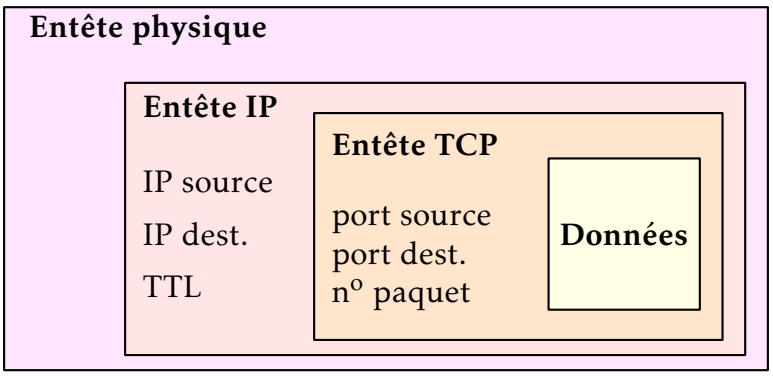
IP permet d'acheminer les données d'un terminal à un autre. En gros, c'est comme la poste. Chaque terminal connecté à Internet possède une adresse IP. Les terminaux ne sont pas directement connectés entre eux. Ce sont les **routeurs** qui servent à connecter les différents réseaux. Chaque terminal est connecté à un routeur (par exemple votre box Internet) et chaque routeur et connecté à au moins un autre routeur.
Lorsqu'un terminal veut envoyer des données à un autre terminal, il les transmet au routeur auquel il est connecté en indiquant l'adresse de destination. Si le routeur est connecté au terminal de destination, il lui transmet les données, sinon il les transmet à un autre routeur auquel il est connecté, et ainsi de suite. Les routeurs utilisent des **tables de routage** pour savoir à qui envoyer les données en fonction des adresses.
IP ne garantit pas :

- L'intégrité des données à l'arrivée : il peut y avoir des erreurs dans les données.
- L'ordre d'arrivée en cas d'envoi de plusieurs messages : les messages peuvent arrivés mélangés.
- Que le message est arrivé : il peut se perdre en chemin.
- Qu'il n'arrive qu'une seule fois : il peut être dupliqué et arriver à plusieurs moments.

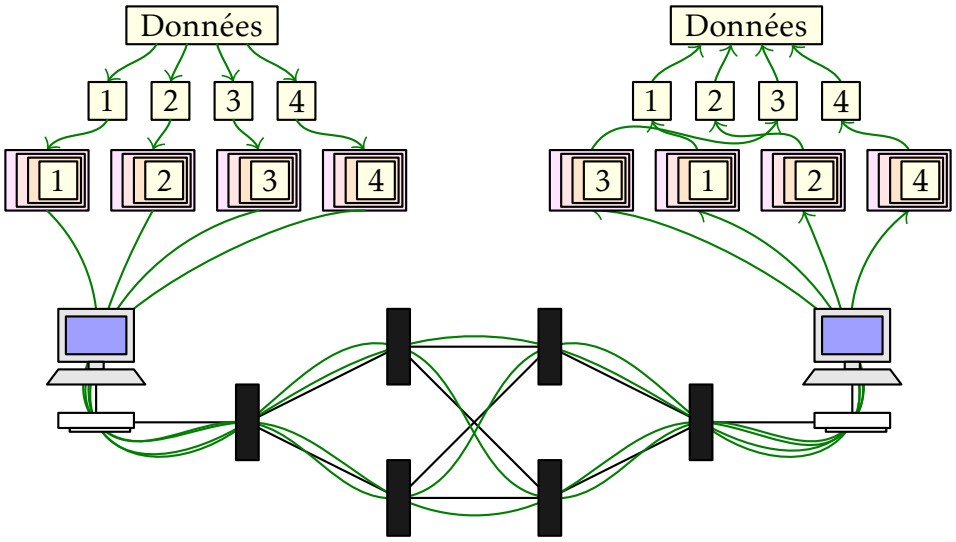
Transmission Control Protocol

TCP a pour but d'assurer que les données arrivent bien au destinataire. Il sert aussi à régler un autre problème : l'encombrement des canaux. En effet lorsqu'un routeur envoie des données à un autre, aucun autre envoi n'est possible sur le canal utilisé. Ainsi, si les données sont très volumineuses, le canal peut resté bloqué pendant un long moment. Et si les données reçues comportent des erreurs, il faut tout recommencer. La solution adoptée, c'est de découper les données en "petits" paquets envoyés les uns par les autres. Ainsi, il est possible d'alterner envoi puis réception de façon alternée ou d'envoyer alternativement les paquets correspondant à des données différentes.
TCP fonctionne de la manière suivante : il découpe les données en paquets, il les numérote et les envoie. Le destinataire renvoie des accusés de réception. Si l'expéditeur ne reçoit pas un accusé, il renvoie le paquet concerné au bout d'un certain temps d'attente. Lorsque tous les accusés ont été reçus, la communication est terminée.
Lors du découpage en paquet, chaque protocole rajoute des informations supplémentaires. C'est ce qu'on appelle l'**encapsulation**. TCP rajoute le numéro du paquet ainsi que les ports d'origine et de destination. Ces ports servent à identifier l'application qui devra traiter les données sur les deux terminaux. Puis IP rajoute l'adresse IP d'origine et du destinataire,

ainsi qu'un compteur appelé Time To Live. Ce compteur est décrémenté à chaque fois que le paquet passe par un routeur. Une fois arrivé à 0, le paquet est détruit. Cela permet d'éviter que des paquets perdus n'encombrent inutilement le réseau. Enfin, selon le protocole "physique" utilisé pour la transmission (filaire, wi-fi, 3G...), des données supplémentaires sont rajoutées.



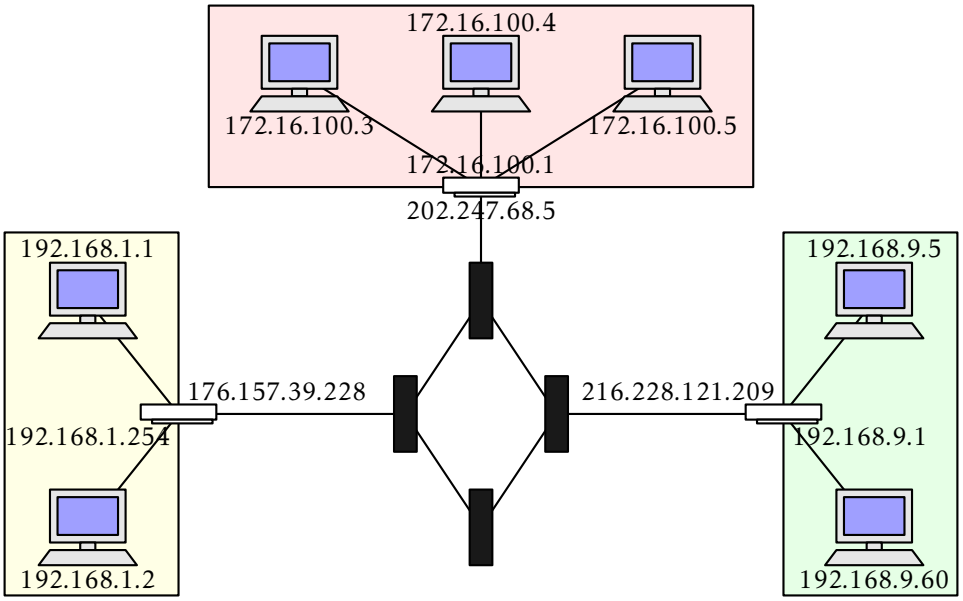
Les données sont donc découpées en paquet. Chaque paquet est encapsulé avec les informations nécessaires. Les paquets sont transmis par le réseau. Ils ne prennent pas forcément tous le même chemin, ni n'arrivent forcément tous dans le même ordre. Les paquets sont ensuite désencapsulés, remis dans l'ordre et recollés pour que le destinataire obtienne les données.



Adresses IP

Ce que l'on appelle généralement adresse IP est en fait une adresse IPv4, de la forme $a.b.c.d$, avec a, b, c et d des entiers entre 0 et 255 inclus. Elles sont codées sur 32 bits. Par exemples `176.157.39.228` ou `192.168.1.88` sont des adresses valides. En gros, le premier nombre indique dans quel "grand réseau" se trouve l'adresse. Le deuxième permet d'identifier dans quel sous réseau il se trouve. Et ainsi de suite, jusqu'à identifier la machine au niveau local. On peut prendre l'analogie "pays.ville.rue.maison".
Au total, il y a $2^{32} = 4\,294\,967\,296$ adresses possibles. Mais ce n'est pas suffisant compte tenu du nombre de terminaux connectés à Internet. C'est pourquoi les adresses IPv6 ont été créées. Elles utilisent 128 bits. Il y a environ 3×10^{38} adresses possibles, ce qui est largement suffisant. Mais en attendant que ce format soit adopté par tout le monde, il a fallu trouver une solution temporaire (mais qui dure depuis des années). On distingue les réseaux publics et les réseaux privés. Tous les ordinateurs du lycée forment un réseau privé. Ils ont tous une adresse IP privée. Par contre, du point de vu de l'extérieur, le réseau public, le lycée n'a qu'une seule adresse IP publique. Ainsi tous les ordinateurs du lycée ont la même adresse

IP publique. C'est le serveur du lycée qui s'occupe de transformer les adresses privées en adresses locales, et réciproquement, lors des communications.



Afin de savoir si une adresse appartient au même sous-réseau que le routeur, on utilise un **masque de réseau**. Cela consiste juste à bloquer les premiers bits de l'adresse IP. S'ils sont identiques, on est dans le même réseau, sinon on est dans des réseaux différents. Les bits suivants servent à distinguer les machines sur un même réseau. Ainsi `192.168.1.3/24` signifie que les 24 premiers bits (donc les 3 premiers entiers) identifient le réseau et les 8 suivants identifient la machine. Les adresses du réseau vont donc de `192.168.1.0` à `192.168.1.255`. Si le nombre de bit du masque est un multiple de 8, c'est assez simple. Mais il peut aussi avoir un autre valeur, comme 12. Dans ce cas, il faut convertir l'adresse en binaire et utiliser l'opération "`et`" bit à bit. Par exemple avec `192.168.1.3/12` :

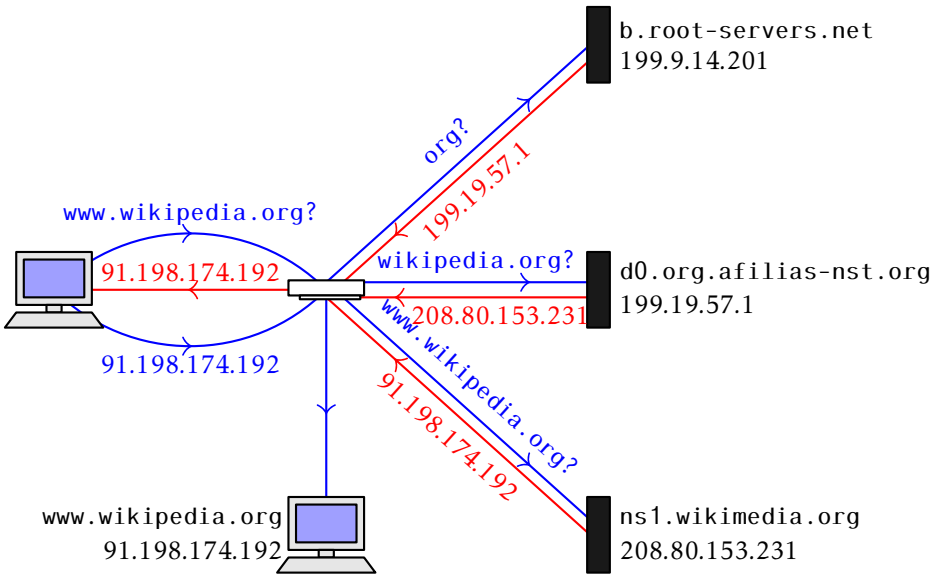
```
    11111111.11110000.00000000.00000000
et  11000000.10101000.00000001.00000011
    11000000.10100000.00000000.00000000
```

Ce qui donne `192.160.0.0` à `192.175.255.255`.

Exercice 1 : Donner les adresses réseau possibles avec les masques suivants :
1) `148.33.1.112/8` 2) `82.30.12.18/24` 3) `91.198.174.3/19`

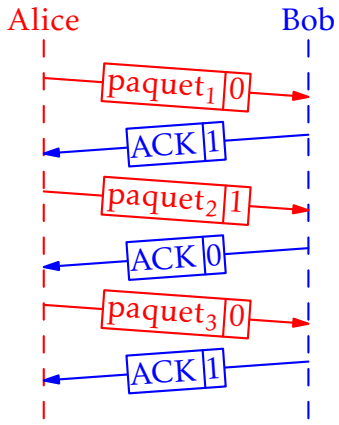
Domain Name System

Les adresses IP sont difficiles à retenir pour un humain. C'est pour cela que les noms de domaine ont été inventé. Le DNS peut être vu comme un annuaire associant une adresse IP au nom demandé. Ainsi, le nom `www.wikipedia.org` peut être associé à `91.198.174.192`.
Cet annuaire est distribué dans plusieurs serveurs, appelés serveurs DNS. Tout comme les routeurs, certains serveurs connaissent les adresses associées à certains noms particuliers et d'autres au contraires servent à indiquer quels sont les serveurs connaissant l'adresse associée à un nom particulier.
Comme pour les adresses IP, les noms de domaines sont découpées en "zones", en allant de droite à gauche. On appelle Top Level Domain l'extension à droite (`.fr`, `.com`, `.org`...), le domaine (`google.com`, `wikipedia.org`...) puis les sous domaines. Pour obtenir une adresse IP, on demande à un serveurs racine un des serveurs DNS du TLD, puis à celui-ci un serveur DNS du domaine et à ce dernier l'adresse du nom de domaine.

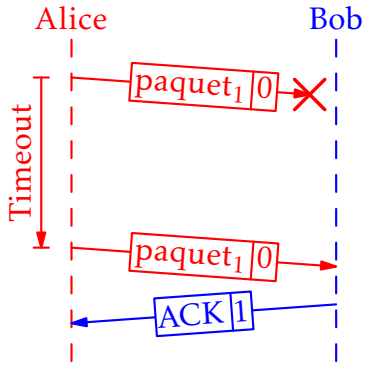


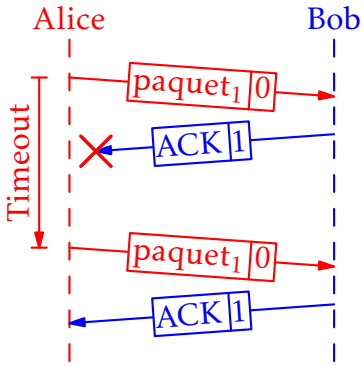
Protocole du bit alterné

Le protocole du bit alterné est une version simplifiée de la partie "communication" de TCP. Quand on parle de protocole, on entend en général un échange de message et pas juste un envoi. Dans le cas de TCP, il faut que le destinataire confirme la bonne réception des données. On parle de **processus d'acquittement**. Lors de l'envoi d'un paquet, l'émetteur rajoute un bit de contrôle. Lorsqu'il l'a reçu, le destinataire répond un message spécial appelé `ACK` accompagné du bit contraire à celui du paquet. L'émetteur envoie alors le paquet suivant en inversant le bit par rapport à l'envoi précédent. On obtient donc le dialogue ci-contre.



Après l'envoi de chaque message, le destinataire lance un compte à rebours. S'il arrive à 0 avant d'avoir reçu l'acquittement, le message est renvoyé et un nouveau compte à rebours est lancé. Si le message d'acquittement est perdu, là aussi, le message est renvoyé à la fin du compte à rebours. Dans ce cas, le destinataire ignore le deuxième envoi et ré-envoie à nouveau l'accusé de réception.





Exercice 2 :

1) Quelle est la réponse à envoyer quand on reçoit un paquet avec le bit de contrôle 1 ?
2) On arrive au timeout pour le paquet n avec un bit de contrôle de 0. Quelle était la réponse attendue ?
3) On suppose que le premier bit de contrôle est 0. Quel est le bit de contrôle associé au n-ième paquet ?