TRANSMISSION DE DONNEES DANS UN RESEAUX

- *TCP-IP*
- *Trame Ethernet*
- *Couches TCP-IP et OSI*
- *Protocole du bit alterné*

“

Le 29 octobre 1969, le 1er message (le mot "login") est envoyé depuis l'université de Californie à Los Angeles vers l'université de Stanford via le réseau ARPAnet (les 2 universités sont environ distantes de 500 Km).

C'est un demi-succès, puisque seules les lettres "l" et "o" arriveront à bon port.

”

Numérique et Sciences Informatiques
1ère

Support de cours :
Jean-Christophe BONNEFOY

Objectifs :

- Mettre en évidence l’intérêt du découpage des données en paquets et leur encapsulation.
- Dérouler le fonctionnement du protocole de récupération de perte de paquets (bit alterné)

1. Introduction

Pour communiquer ensemble, 2 ordinateurs en réseau doivent utiliser des règles communes, l'ensemble de ces règles qui permettent à 2 ordinateurs de communiquer ensemble s'appelle un protocole.

Il existe de nombreux protocoles réseau, nous allons en étudier 2 : le protocole TCP et le protocole IP. Ces 2 protocoles sont tellement liés l'un à l'autre que l'on parle souvent du protocole TCP/IP.

Avant d'entrer dans le vif du sujet, un peu d'histoire :

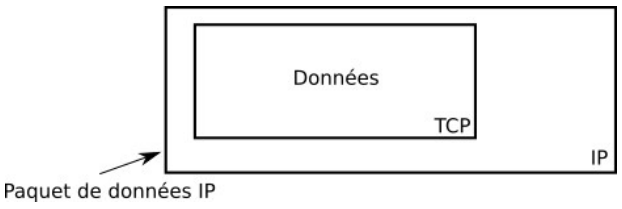
La DARPA (Defense Advanced Research Projects Agency) voit le jour en 1958, cette agence gouvernementale américaine a pour but de veiller à la constante suprématie des États unis en matière technologique et scientifique. En 1962 la DARPA soutient le projet du professeur Licklider qui a pour but de mettre en réseau les ordinateurs des universités américaines afin que ces dernières puissent échanger des informations plus rapidement (même à des milliers de kilomètres de distance). En 1968, ARPAnet, 1er réseau informatique à grande échelle de l'histoire voit le jour. Le 29 octobre 1969, le 1er message (le mot "login") est envoyé depuis l'université de Californie à Los Angeles vers l'université de Stanford via le réseau ARPAnet (les 2 universités sont environ distantes de 500 Km). C'est un demi-succès, puisque seules les lettres "l" et "o" arriveront à bon port.

En 1972, 23 ordinateurs sont connectés à ARPAnet (on trouve même des ordinateurs en dehors des États unis). En parallèle au projet ARPAnet, d'autres réseaux voient le jour, problème, ils utilisent des protocoles de communication hétéroclite (UUCP, NCP ou encore X.25) et 2 ordinateurs appartenant à 2 réseaux différents sont incapables de communiquer entre eux puisqu'ils n'utilisent les mêmes protocoles. En 1974 Vint Cerf et Bob Khan vont mettre au point le protocole TCP qui sera très rapidement couplé au protocole IP pour donner TCP/IP. TCP/IP, grâce à sa simplicité, va très rapidement s'imposer comme un standard : les différents réseaux (ARPAnet et les autres) vont adopter TCP/IP. Cette adoption va permettre d'interconnecter tous ces réseaux (2 machines appartenant à 2 réseaux différents vont pouvoir communiquer grâce à cette interconnexion). Internet était né (le terme Internet vient de "internetting" qui signifie "Connexion entre plusieurs réseaux"). TCP/IP est donc au cœur d'Internet, voilà pourquoi aujourd'hui, la plupart des machines utilisent TCP/IP.

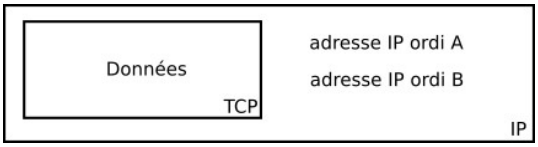
2. Principe de base des protocoles TCP et IP

Quand un ordinateur A "désire" envoyer des données à un ordinateur B, l'ordinateur A "utilise" le protocole TCP (Transmission Control Protocol) pour mettre en forme les données à envoyer. On parle alors de **segments TCP**

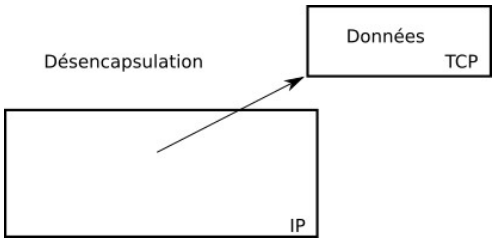
Ensuite le protocole IP (Internet Protocol) prend le relai et utilise les données mises en forme par le protocole TCP afin de créer des paquets des données. Après quelques autres opérations qui ne seront pas évoquées ici, les paquets de données pourront commencer leur voyage sur le réseau jusqu'à l'ordinateur B. Il est important de bien comprendre que le protocole IP "encapsule" les données issues du protocole TCP afin de constituer des paquets de données. On parle alors de **paquets IP ou de datagrammes**



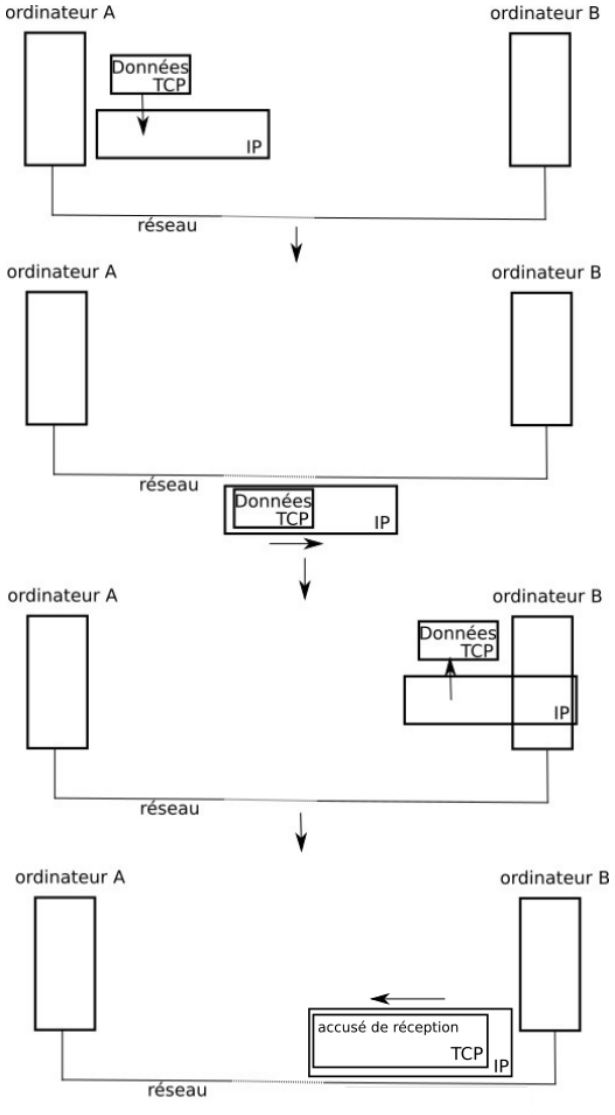
Le protocole IP s'occupe uniquement de faire arriver à destination les paquets en utilisant l'adresse IP de l'ordinateur de destination. Les adresses IP de l'ordinateur de départ (ordinateur A) et de l'ordinateur destination (ordinateur B) sont ajoutées aux paquets de données.



Une fois arrivées à destination (ordinateur B), les données sont "désencapsulées" : on récupère les données TCP contenues dans les paquets afin de pouvoir les utiliser.

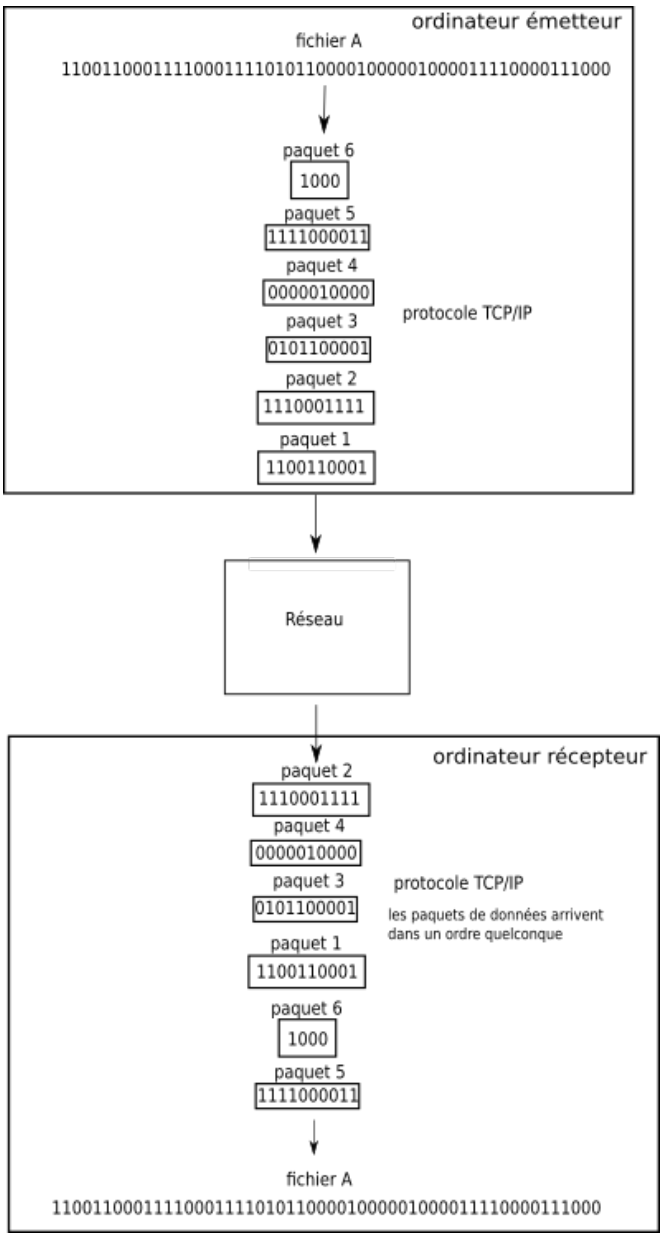


Le protocole TCP permet de s'assurer qu'un paquet est bien arrivé à destination. En effet quand l'ordinateur B reçoit un paquet de données en provenance de l'ordinateur A, l'ordinateur B envoie un accusé de réception à l'ordinateur A (un peu dans le genre "OK, j'ai bien reçu le paquet"). Si l'ordinateur A ne reçoit pas cet accusé de réception en provenance de B, après un temps prédéfini, l'ordinateur A renverra le paquet de données vers l'ordinateur B.



À noter qu'il existe aussi le protocole UDP qui ressemble beaucoup au protocole TCP. La grande différence entre UDP et TCP est que le protocole UDP ne gère pas les accusés de réception. Les échanges de données avec UDP sont donc moins fiables qu'avec TCP (un paquet "perdu" est définitivement "perdu" et ne sera pas renvoyé) mais beaucoup plus rapides (puisqu' il n'y a pas d'accusé de réception à transmettre). UDP est donc très souvent utilisé pour les échanges de données qui doivent être rapides, mais où la perte d'un paquet de données de temps en temps n'est pas un gros problème (par exemple le streaming vidéo).

Il est très important de bien comprendre que TCP/IP repose sur la notion de paquets de données. Si par exemple on désire envoyer un fichier (son, photo, vidéo ou texte, peu importe, dans tous les cas on envoie une succession de bits) en utilisant TCP/IP, les données qui constituent ce fichier ne seront pas envoyées d'un seul tenant, ces données vont être "découpées" en plusieurs morceaux et chaque morceau sera envoyé dans un paquet différent. Une fois tous les paquets arrivés à destination, le fichier d'origine pourra être reconstitué. Pour aller d'un ordinateur A à un ordinateur B, les différents paquets contenant les données qui constituent notre fichier, ne passeront pas forcement par la même route (cette notion de route sera abordée plus tard), ils pourront emprunter des chemins très différents : en exagérant à peine, pour faire le trajet Paris-Los Angeles, certains paquets pourront passer par l'atlantique alors que d'autres passeront par le pacifique. Si un des paquets n'arrive pas à destination, le fichier ne pourra pas être reconstitué, le paquet "perdu" devra être renvoyé par l'émetteur (voir le système d'accusé de réception décrit ci-dessus).

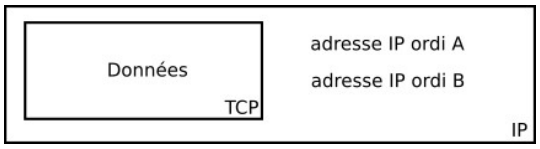


3. La trame Ethernet

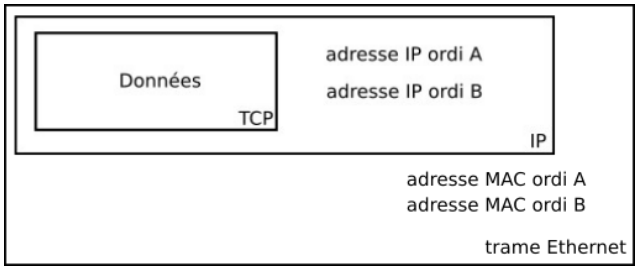
Nous avons vu avec les protocoles TCP et IP le processus d'encapsulation des données : "IP encapsule TCP". Les paquets IP ne peuvent pas transiter sur un réseau tel quel, ils vont eux aussi être encapsulés avant de pouvoir "voyager" sur le réseau. L'encapsulation des paquets IP produit ce que l'on appelle **une trame**. Il n'est pas question d'étudier en détail ce qu'est une trame, vous devez juste savoir qu'il existe de nombreux types de trames : ATM, token ring, PPP, Ethernet, Wifi...

Si vous utilisez un réseau filaire avec des câbles Ethernet (avec des prises RJ45), la trame sera de type Ethernet (ce qui est le cas pour le réseau du lycée). Si vous utilisez un réseau sans fil Wifi, la trame sera de type Wifi. En fait, la trame Wifi ressemble beaucoup à la trame Ethernet, on peut même dire que la trame Wifi est la variante sans-fil de la trame Ethernet, afin de simplifier les choses, dans la suite, nous évoquerons uniquement la trame Ethernet en ayant à l'esprit que ce qui est dit sur la trame Ethernet et aussi valable pour la trame Wifi.

Nous avons vu que le paquet IP contient les adresses IP de l'émetteur et du récepteur :



Le paquet IP étant encapsulé par la trame Ethernet, les adresses IP ne sont plus directement disponibles (il faut désencapsuler le paquet IP pour pouvoir lire ces adresses IP), nous allons donc trouver un autre type d'adresse qui permet d'identifier l'émetteur et le récepteur : l'adresse MAC (Media Access Control) aussi appelée adresse physique.



Nous avons déjà vu que l'adresse MAC est liée au matériel, chaque carte réseau (Ethernet ou Wifi) possède sa propre adresse MAC.

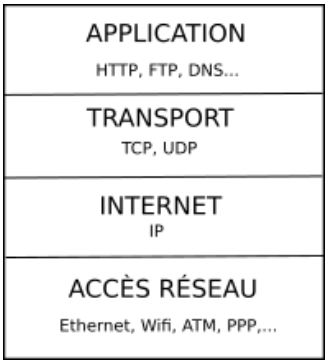
Au moment de l'encapsulation d'un paquet IP, l'ordinateur "émetteur" va utiliser un protocole nommé ARP (Address Resolution Protocol) qui va permettre de déterminer l'adresse MAC de l'ordinateur "destination", en effectuant une requête "broadcast" (requête destinée à tous les ordinateurs du réseau) du type : "j'aimerais connaitre l'adresse MAC de l'ordinateur ayant pour IP XXX.XXX.XXX.XXX". Une fois qu'il a obtenu une réponse à cette requête ARP, l'ordinateur "émetteur" encapsule le paquet IP dans une trame Ethernet et envoie cette trame sur le réseau.

4. Modèle en couches

De façon pratique, à chaque phase d'encapsulation on associe ce que l'on appelle une couche :

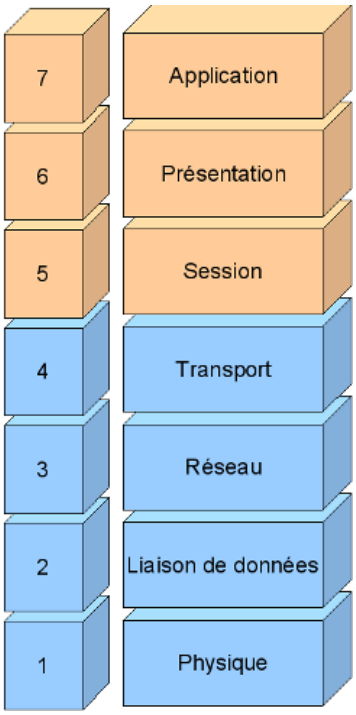
- les protocoles HTTP, FTP, SMTP, DNS,... sont associés à la couche "Application"
- les protocoles TCP et UDP sont associés à la couche "Transport"
- le protocole IP est associé à la couche "Internet"
- les trames Ethernet (ou Wifi) sont associées à la couche "Accès réseau"

On présente souvent ces différentes couches sur ce type de schéma :



La couche du "dessous" encapsule la couche située "au dessus". On nomme ce système de couche "**modèle de couches TCP/IP**"

Il existe un autre modèle de couche, **le modèle OSI** (Open Systems Interconnection), ce système est antérieur au modèle TCP/IP puisqu'il date des années 1970. Ce modèle est principalement théorique et à permis de poser les bases des communications réseau. Ce modèle est composé de 7 couches (alors que le modèle TCP/IP est composé de 4 couches).



Ce modèle est donné ici à titre d'information mais le principal est de retenir ce qui a été vu sur le modèle TCP/IP.

5. Fiabilité

TCP est un protocole fiable. On entend par fiable **le fait de faire en sorte que tout ce qui arrive est exactement ce qui a été envoyé**, on a donc les 4 critères suivants :

- Sans perte : les données ne doivent pas être perdues
- Sans erreur : les données ne doivent pas subir d'erreurs (une erreur correspond à un bit qui change pendant le transport)
- Dans l'ordre : les données doivent arriver dans l'ordre
- Sans duplication : les données ne doivent pas arriver en double

Dans la pratique :

- En cas de perte ou erreur, il faut retransmettre si on n'a pas reçu d'acquittement (ACK) au bout d'un certain temps
- Pour détecter une perte ou une duplication, il faut des ACK et numéroter les messages et les ACK
- Pour détecter les erreurs, on utilise les checksum
- Pour retransmettre, il faut conserver les messages envoyés qui n'ont pas encore été acquittés

Beaucoup d'applications utilisent le protocole TCP car elles ne peuvent pas se passer de fiabilité :

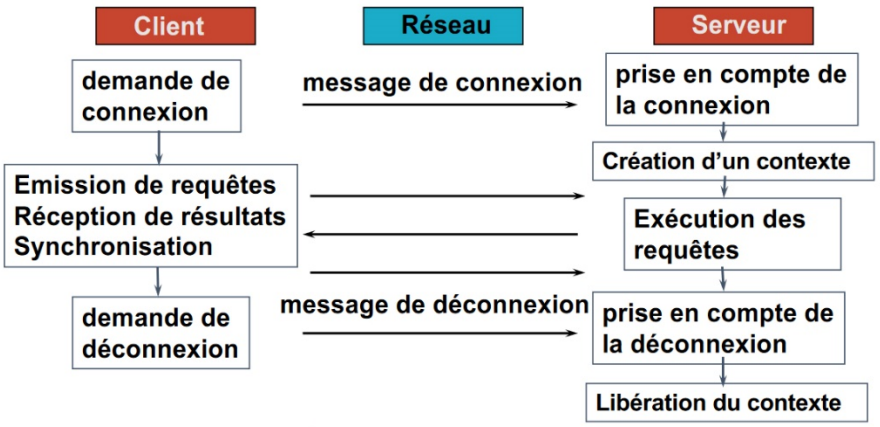
- Le web
- La connexion à distance
- Le courrier électronique
- Le transfert de fichiers
- …

En revanche, les applications multimédias et le DNS n'utilisent pas TCP.

6. Le mode connecté

Le mode connecté est l'établissement d'une session de communication entre deux parties qui veulent échanger des données. Cette session comporte un début, une fin et une validation (vérification des erreurs). L'exemple le plus représentatif du mode connecté est l'appel téléphonique**.**

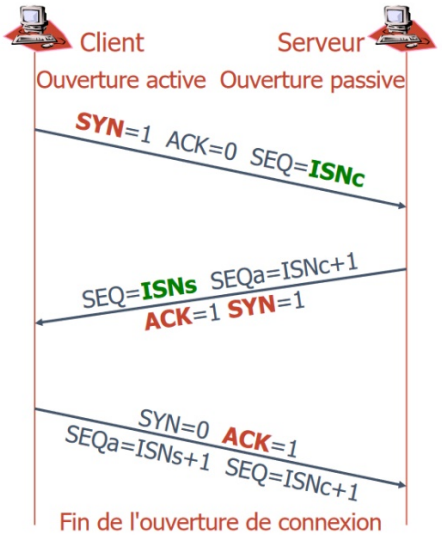
Dans le cadre du protocole TCP, on a :



6.1 : Etablissement d'une connexion TCP

Cela fonction en mode Client/Serveur. Le TCP du client fait une demande d'ouverture de connexion vers le port du serveur qui doit être connu à l'avance. Le TCP du serveur est en attente des demandes d'ouverture de connexion en provenance des clients. La Connexion se réalise en trois phases :

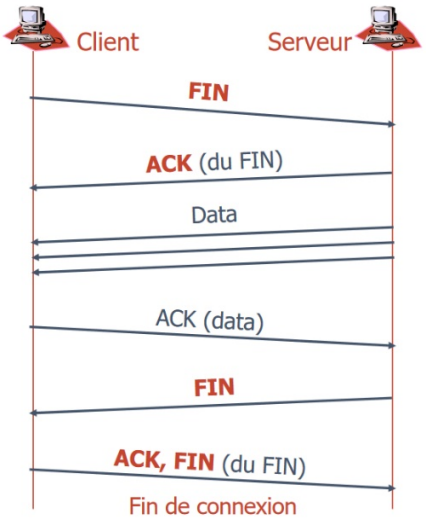
- 1 - demande d'ouverture par le client (SYN), choix ISNc
- 2 - acceptation par le serveur (SYN+ACK), allocation des tampons, choix ISNs
- 3 - le client acquitte l'acceptation (ACK)



6.2 : Fermeture d'une connexion TCP

Il s'agit d'une fermeture négociée :

- 1 - demande de fin de connexion (FIN) par une des extrémités
- 2 - acquittement du FIN (ACK) mais mise en attente de la demande (Le serveur a encore des données non transmises)
- 3 - envoi des données en attente
- 4 - acquittement des données (ACK)
- 5 - acceptation de la fin de connexion par le serveur (FIN)
- 6 - acquittement de la fin de connexion (ACK)

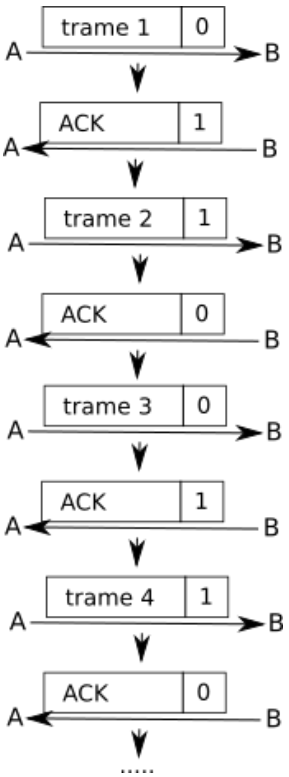


7. Le protocole du bit alterné

Nous avons vu que le protocole TCP propose un mécanisme d'accusé de réception afin de s'assurer qu'un paquet est bien arrivé à destination. On parle plus généralement de processus d'acquittement. Ces processus d'acquittement permettent de détecter les pertes de paquets au sein d'un réseau, l'idée étant qu'en cas de perte, l'émetteur du paquet renvoie le paquet perdu au destinataire. Nous allons ici étudier un protocole simple de récupération de perte de paquet : le protocole de bit alterné.

Le protocole de bit alterné est implémenté au niveau de la couche de "liaison de données" du modèle OSI (couche n°2), il ne concerne donc pas les paquets, mais les trames (on parle de paquets uniquement à partir de la couche "Réseau" (couche 3) du modèle OSI). Le principe de ce protocole est simple, considérons 2 ordinateurs en réseau : un ordinateur A qui sera l'émetteur des trames et un ordinateur B qui sera le destinataire des trames. Au moment d'émettre une trame, A va ajouter à cette trame un bit (1 ou 0) appelé drapeau (flag en anglais). B va envoyer un accusé de réception (acknowledge en anglais souvent noté ACK) à destination de A dès qu'il a reçu une trame en provenance de A. À cet accusé de réception on associe aussi un bit drapeau (1 ou 0).

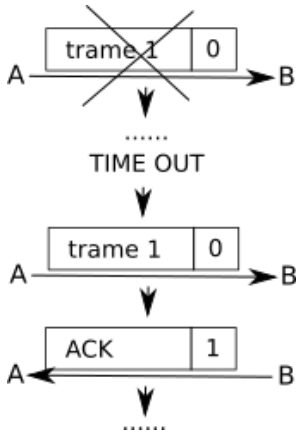
La règle est relativement simple : la première trame envoyée par A aura pour drapeau 0, dès cette trame reçue par B, ce dernier va envoyer un accusé de réception avec le drapeau 1 (ce 1 signifie "la prochaine trame que A va m'envoyer devra avoir son drapeau à 1"). Dès que A reçoit l'accusé de réception avec le drapeau à 1, il envoie la 2e trame avec un drapeau à 1, et ainsi de suite...



Le système de drapeau est complété avec un système d'horloge côté émetteur. Un "chronomètre" est déclenché à chaque envoi de trame, si au bout d'un certain temps, l'émetteur n'a pas reçu un acquittement correct (avec le bon drapeau), la trame précédemment envoyée par l'émetteur est considérée comme perdue et est de nouveau envoyée.

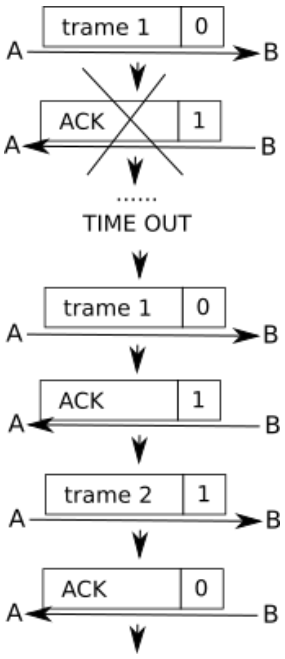
Examinons quelques cas :

- La trame est perdue :



Au bout d'un certain temps ("TIME OUT") A n'a pas reçu d'accusé de réception, la trame est considérée comme perdue, elle est donc renvoyée.

- L'accusé de réception est perdu :



A ne reçoit pas d'accusé de réception avec le drapeau à 1, il renvoie donc la trame 1 avec le drapeau 0. B reçoit donc cette trame avec un drapeau à 0 alors qu'il attend une trame avec un drapeau à 1 (puisqu'il a envoyé un accusé de réception avec un drapeau 1), il "en déduit" que l'accusé de réception précédent n'est pas arrivé à destination : il ne tient pas compte de la trame reçue et renvoie l'accusé de réception avec le drapeau à 1. Ensuite, le processus peut se poursuivre normalement.

Dans certaines situations, le protocole de bit alterné ne permet pas de récupérer les trames perdues, c'est pour cela que ce protocole est aujourd'hui remplacé par des protocoles plus efficaces, mais aussi plus complexes.