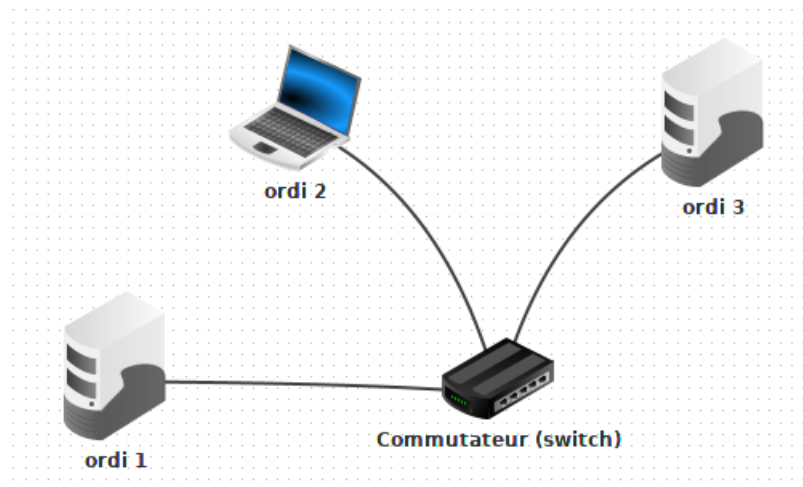


## 1 Architecture des réseaux

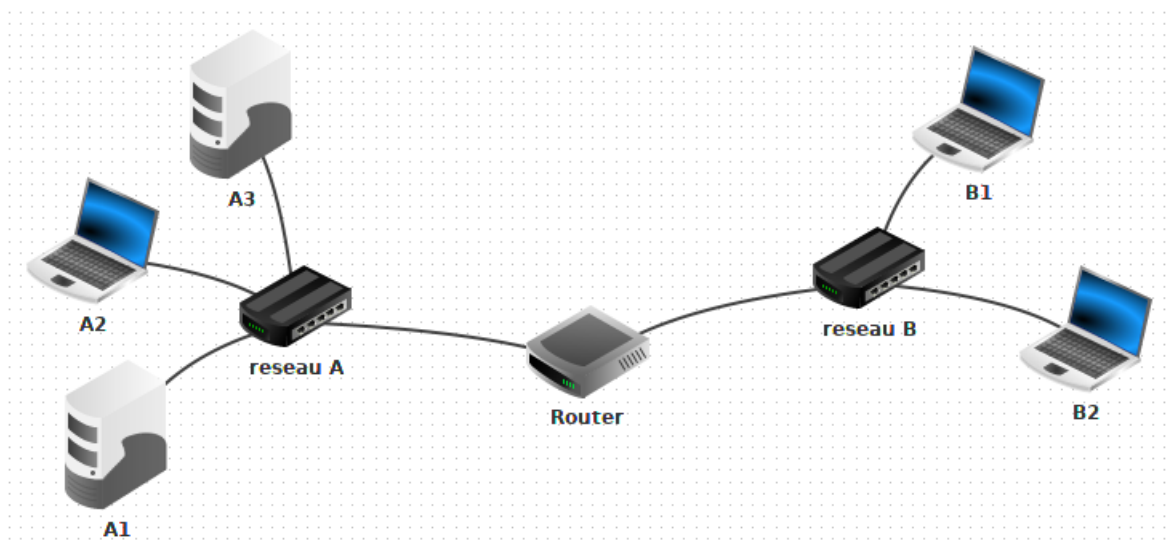
Les réseaux permettent de transmettre de l'information entre machines.

Pour mettre en réseau plus de 2 machines, il existe plusieurs topologies (ex : anneau, bus) mais la plus utilisée est la topologie en étoile qui repose sur l'utilisation d'un **commutateur (switch)**, dont le rôle est de distribuer un message envoyé par une machine source à la bonne machine de destination.



*Remarque :* Les connexions peuvent être filaires (câble Ethernet, prise RJ45), mais la communication peut également se faire par onde radio comme avec la technologie wi-fi (wireless fidelity).

Plusieurs réseaux locaux peuvent être reliés entre eux à travers l'utilisation de **routeurs**, dont le rôle est de transférer les données d'un réseau vers un autre.



On peut distinguer aussi les réseaux par leur étendue :

- **LAN** : Local Area Network : réseau local (adapté à une entreprise ou le lycée par exemple).
- **WAN** : Wide Area Network : réseau étendu (adapté à une zone géographique de grande envergure comme Internet par exemple).

## 2 Protocoles de communication

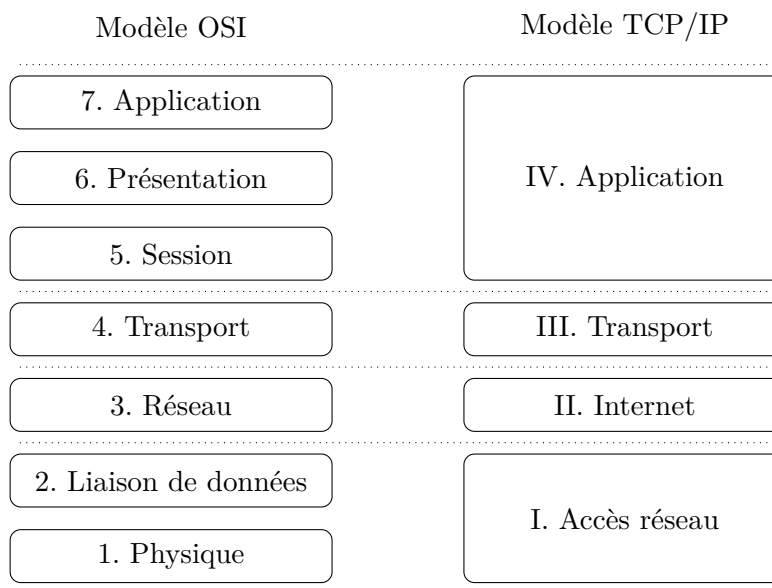
Pour communiquer et transférer des données de façon automatique d'une machine à une autre, il faut établir des **protocoles de communication**.

Les **protocoles** correspondent à un **ensemble de règles** précisant le format des informations échangées, la manière de les échanger, d'établir la communication et de la terminer.

Dans les années 1970's, un modèle théorique a été développé pour normaliser la communication entre machines : c'est le modèle OSI. Le modèle OSI est un modèle en couches où chaque couche rend un service particulier pour les autres couches, et s'adresse suivant des protocoles spécifiques à une couche identique d'une autre machine.

**Ce modèle OSI à 7 couches peut se simplifier en un modèle en 4 couches pour la description des échanges sur internet : c'est le modèle TCP/IP.**

Le nom TCP/IP vient des 2 protocoles principaux de ce modèle.



La communication d'une machine à une autre peut se structurer ainsi :

- **couche IV. Application** : les données sont mises en forme par une application (mail, page web...). Un protocole mis en jeu est par exemple le protocole HTTP pour le Web. Les applications sont identifiées par leur **numéro de port** sur une machine.
- **couche III. Transport** : les données de la couche Application sont remises à la couche inférieure, la couche Transport, qui les coupe en plusieurs paquets avec un identifiant.

Pour des raisons de gestion du réseau (congestion, fiabilité...) il est plus intéressant de transporter des quantités assez faibles de données à la fois.

La couche transport s'occupe en particulier de gérer la bonne communication avec l'autre machine, de contrôler par exemple que les paquets sont bien transmis, de les réordonner à la reçoit si nécessaire.... Le protocole principal de cette couche est le **protocole TCP**. On trouve aussi un protocole voisin UDP.

- **couche II. Internet** : ces paquets sont remis à la couche Internet, qui s'occupe principalement de l'acheminement des données d'une machine à une autre en identifiant de manière **logique** les machines sur le réseau à l'aide de leur **adresse IP**. Le protocole essentiel est ici le **protocole IP**.
- **couche I. Accès réseau** : enfin, les données sont **transmises physiquement** sur le réseau par la couche Accès réseau. À ce niveau les machines sont identifiées par l'**adresse physique unique** de leur carte réseau (adresse MAC).

Le problème de la communication entre machines est donc divisé en sous-problèmes et hiérarchisé en couches. Chaque couche rend un service à sa couche supérieure, et la couche supérieure se moque de savoir concrètement comment ce service est effectué. La couche Internet par exemple ne se préoccupe pas de savoir comment les données seront physiquement transmises (ondes radio, câbles...). Et si les protocoles de la couche Accès réseau sont modifiés, cela ne change pas son propre fonctionnement.

On peut proposer l'analogie suivante pour imaginer le fonctionnement global :  
"Je souhaite déménager de Caen à Marseille".

- **couche Application** : je range bien ma maison et place toutes les affaires du déménagement à transporter dans une pièce dédiée au déménagement.
- **couche Transport** : les déménageurs cannaïis rangent tout en cartons et leur apposent des étiquettes, puis les confient au service de livraison postale. À Marseille, l'autre équipe de déménageurs saura ainsi tout remettre à la bonne place.
- **couche Internet** : les facteurs ajoutent les adresses postales de destination, et de source pour le cas d'un retour à l'expéditeur, et confie les paquets aux transporteurs. Souvent les paquets passeront par plusieurs relais (chacun avec son adresse postale) avant d'arriver à l'adresse de destination finale.
- **couche Accès réseau** : les transporteurs empruntent la route (ou le train, ou l'avion) d'un relais à l'autre jusqu'à arriver à destination.

### Compléments sur les protocoles TCP et IP :

- Le protocole TCP (**Transfer Control Protocol**) gère la préparation des données à envoyer en paquets et contrôle que l'ensemble des données est bien transféré. La route prise par un paquet est indépendante des routes prises par les autres paquets sur le réseau, et ces derniers peuvent arriver dans un ordre aléatoire : leur identifiant permet de remettre le fichier complet dans le bon ordre.

Un fichier découpé en petits morceaux permet de ne pas engorger une liaison du réseau et permet aussi de ne renvoyer que les paquets qui ont disparu en cas de pertes. La réception d'un paquet à travers le réseau n'est pas garantie. Le protocole TCP gère ces situations en attendant un **accusé de réception** pour chaque paquet. Si l'accusé de réception n'arrive pas à la machine source, alors le protocole entreprend de renvoyer à nouveau le même paquet.

*Remarque* : il existe aussi le **protocole UDP** qui peut remplacer TCP. Sa différence principale est qu'il ne s'assure pas que les paquets arrivent bien à destination : ce protocole est donc plus rapide (utilisé par exemple en streaming où la perte de quelques données n'est pas gênante).

- Le protocole IP (**Internet Protocol**) gère l'acheminement des données d'une machine à une autre, en s'appuyant sur leur **adresse IP**. Le routage IP sera étudié en Terminale.

### 3 Principe d'encapsulation

Lorsque les données passent d'une couche vers la couche inférieure, elles sont **encapsulées** en ajoutant l'en-tête de la couche inférieure qui **contient les informations spécifiques à cette couche utiles aux protocoles**.

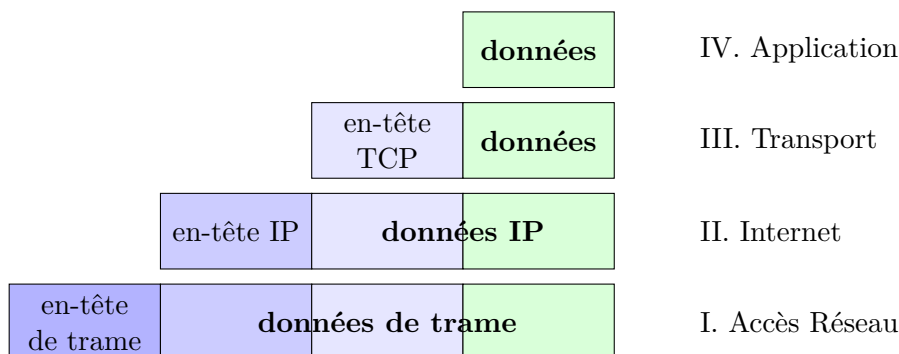
Un processus de **décapsulation** est alors nécessaire vers les couches de niveaux supérieurs lorsque les données "remontent" vers l'ordinateur de destination.

Au niveau de la couche transport, TCP encapsule les données de l'application dans un **segment** en ajoutant un en-tête indiquant entre autres les numéros de port de communication (source et destination) et le numéro du paquet.

Le segment TCP est alors encapsulé au niveau la couche Internet sous forme de **paquet** avec un en-tête qui précise par exemple les adresses IP de la machine source et de la machine de destination. Enfin, au niveau de la couche accès réseau, le paquet IP est encapsulé dans une **trame** Ethernet avec un en-tête spécifiant les **adresses MAC** (Media Access Control) des machines qui communiquent. Ce sont des adresses codées sur 6 octets (*exemple* : 3E:49:82:79:43:60) **qui permettent d'identifier chaque carte réseau (au niveau mondial) de manière unique**.

Les en-têtes contiennent d'autres informations, notamment pour permettre de s'assurer de la bonne transmission et de la **correction** des données transmises.

On peut finalement donner le schéma de l'encapsulation :



### 4 Adressage logique : IPv6/IPv4

Pour **identifier de manière logique des machines sur un réseau**, celles-ci doivent posséder **une adresse** (analogue de l'adresse postale pour nos maisons) : c'est l'**adresse IP**.

Les adresses au format IPv6, codées sur 128 bits (16 octets), permettent d'identifier de manière unique chaque machine au monde. Pourtant l'identification des machines avec ce format n'est pas encore pris en compte sur tous les réseaux.

*Exemple d'adresse IPv6* : 2a01:cb05:0039:1300:b83a:3d63:ce44:7f30 (notation hexadécimale).

Le format IPv4 est donc encore largement utilisé.

Une adresse IPv4 est codée sur 32 bits (4 octets), notée de façon décimale sous la forme **a.b.c.d**, où les nombres **a**, **b**, **c** et **d** sont compris entre 0 et 255.

*Exemple d'adresse IPv4* : 90.46.210.146 (notation décimale pointée).

Une machine d'un réseau local ne peut entrer en communication directement qu'avec une machine du même réseau.

L'adresse IP d'une machine contient donc l'information du réseau auquel elle appartient en plus de l'information qui l'identifie au sein de ce réseau.

On utilise ainsi la notation complète (notation CIDR) pour une adresse IPv4 sous la forme  $a.b.c.d/n$ , où  $n$  est un nombre qui indique le nombre de bits consacrés à l'adresse du réseau.

Exemples :

- $10.15.122.235/8$  : identifie une machine du réseau  $10.0.0.0$   
8 bits (1 octet) pour décrire la partie réseau et les 3 autres pour décrire la partie machine.
- $172.16.12.125/16$  : identifie la machine du réseau  $172.16.0.0$   
16 bits (2 octets) pour décrire la partie réseau et les 2 autres pour décrire la partie machine.
- $192.168.1.135/24$  : identifie la machine du réseau  $192.168.1.0$   
24 bits (3 octets) pour décrire la partie réseau et le dernier pour décrire la partie machine.

## 5 Un exemple de protocole de récupération de perte de paquets

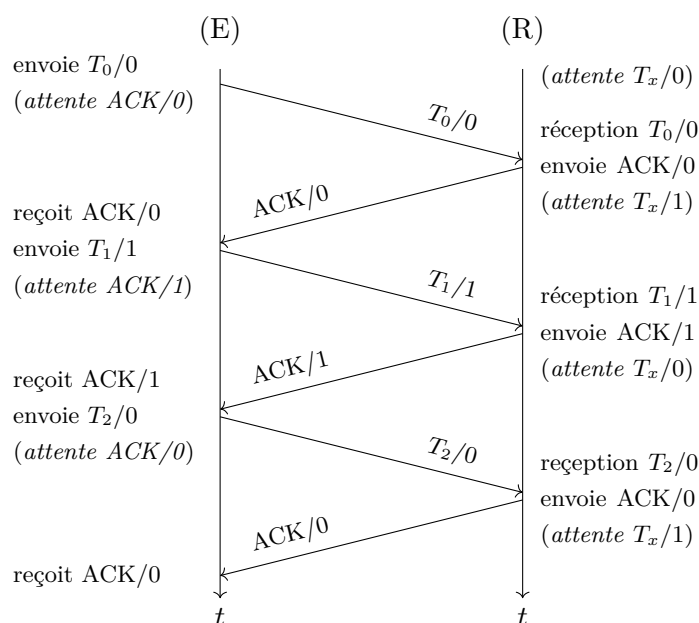
Le **protocole du bit alterné** (au niveau de la couche *Accès réseau*) permet de s'assurer (dans la grande majorité des cas) que les trames sont bien acheminées et reçues correctement.

L'ordinateur émetteur (E) ajoute un bit de contrôle (*flag*) à chaque trame envoyée. Ce bit de contrôle vaut **alternativement** 0 ou 1. Lorsque l'ordinateur récepteur (R) reçoit une trame, il envoie un accusé de réception ou acquittement (acknowledgement ACK) avec également le même bit de contrôle. Si, dans un délai défini par le protocole, l'émetteur ne reçoit pas de bit contrôle en retour (ou un bit de contrôle erroné), il renvoie à nouveau la même trame avec le même bit de contrôle.

Voyons quelques exemples de situations possibles

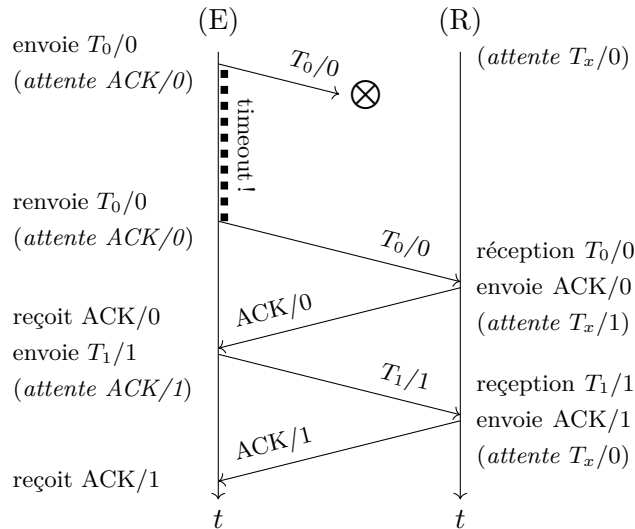
Le message  $M$  à transmettre est divisé en trames  $T_0, T_1$ , etc... La 1ère trame  $T_0$  est envoyée avec un bit de contrôle à 0.

1. Communication sans soucis :

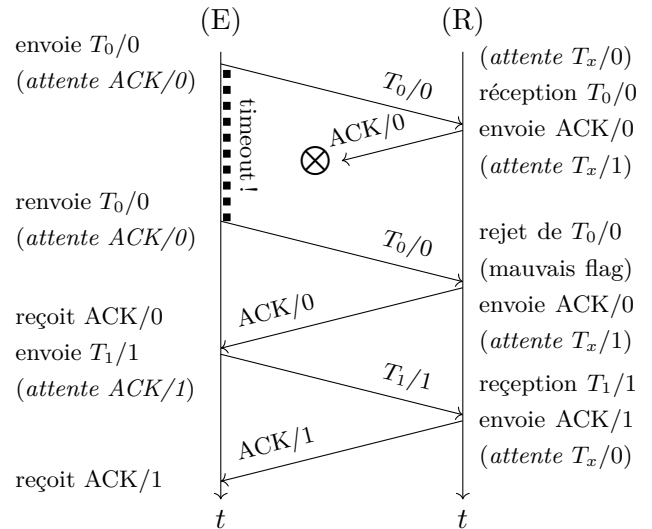


2. Quand arrivent les soucis :

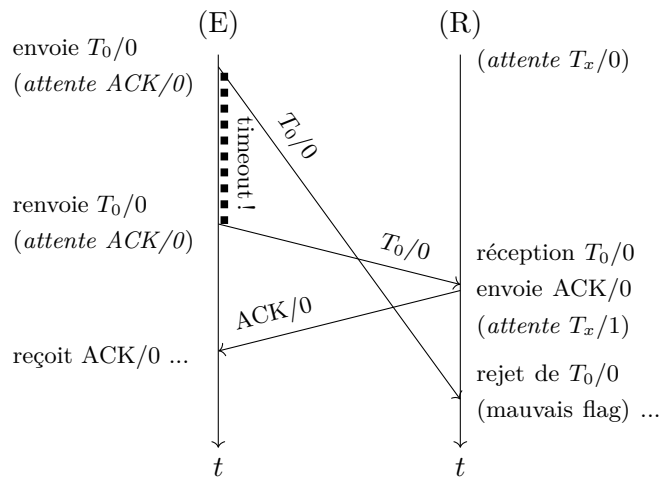
Perte d'une trame (donc pas de ACK)  
⇒ renvoie la trame après un délai *timeout* :



Perte de l'accusé de réception :



Chevauchement de trames :



Chevauchement d'accusés de réception :

