IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

UE PRIP

Principe des Réseaux Informatiques par la Pratique

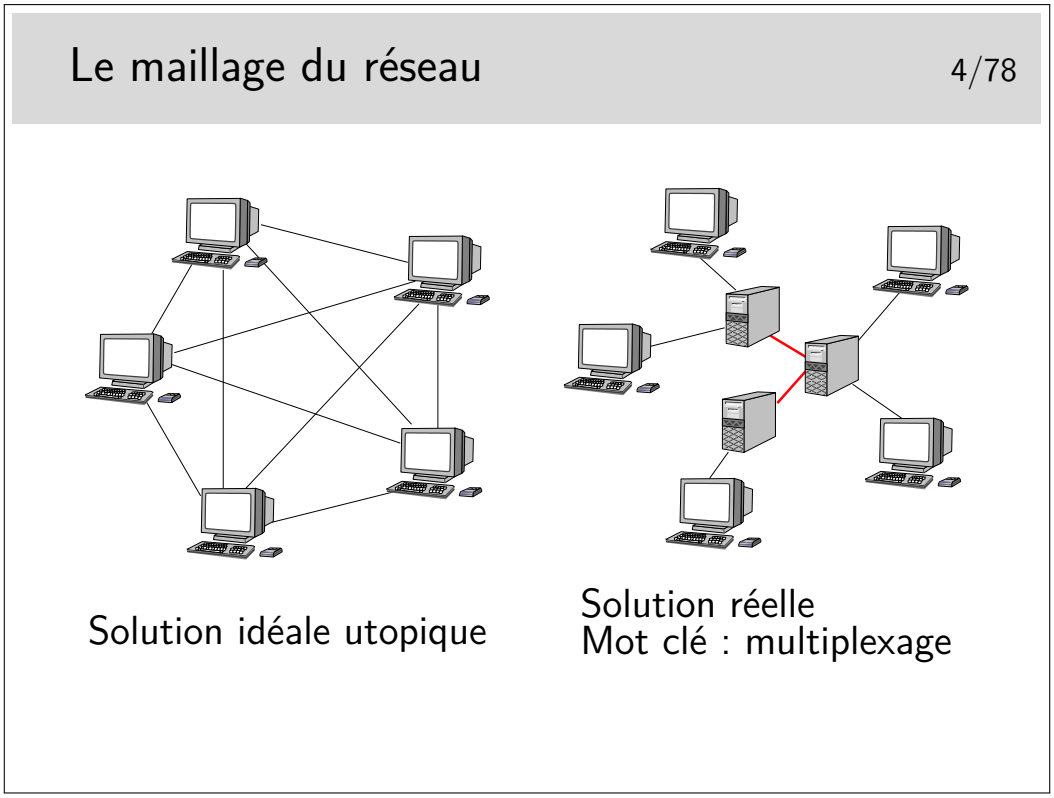
Le transport de l'information

Christophe Lohr

Automne 2023

1 Les concepts fondamentaux

1.1 Multiplexage



Les terminaux ne peuvent pas être tous interconnectés. Cette solution serait trop coûteuse en nombre de liaisons et ces dernières seraient la plupart du temps sous utilisées.

Les terminaux sont reliés à des machines intermédiaires, des relais, qui concentrent le trafic et acheminent les divers flux d'information sur des supports qui les relient.

Les divers flux de trafic sont acheminés sur les liens inter-noeuds de manière simultanée ou quasi simultanée. On dit qu'ils sont *multiplexés* et les liens entre les noeuds sont appelés de *multiplex*. Une des fonction des noeuds est d'assurer le *multiplexage* des informations sur les liens.

Un multiplex est donc une voie de communication sur laquelle on véhicule plusieurs «communications» à la fois. (Notez les guillemets, il reste à définir ce qu'est une «communication», ce n'est pas si simple)

Multiplexage et multiplexes 5/78

Différents types de multiplexes

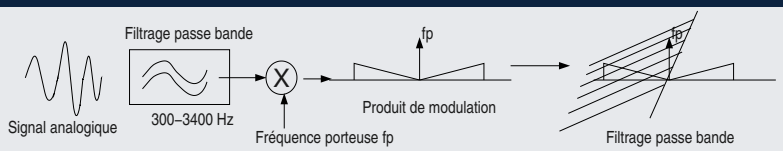
- Le multiplexage en fréquence
- Le multiplexage temporel analogique
  - Par échantillonnage du signal origine
- Le multiplexage temporel numérique
  - Par échantillonnage et numérisation
- Le multiplexage statistique
  - Par acheminement sur canal commun de segments d'informations appartenant à diverses communications

de la téléphonie... aux réseaux

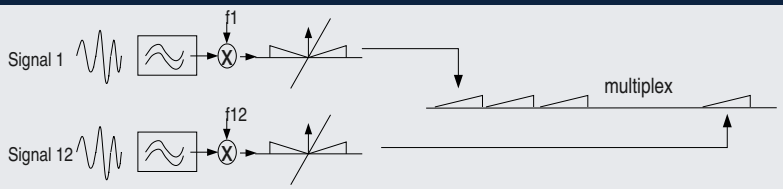
Multiplexage en fréquence 6/78

Exemple du multiplexage de canaux téléphoniques : Modulation en amplitude

Modulation d'un canal



Multiplexage



Cas du mutilpexage de canaux téléphoniques :

— Chaque canal est d'abord filtré dans la bande standard 300-3400 Hz. Le signal filtré vient moduler une porteuse en amplitude. Le résultat de cette modulation produit deux bandes de fréquences contenant l'information utile. Ces deux bandes sont centrées autour de la valeur de la porteuse : de fp-3400 à fp-300 puis de fp+300 à fp+3400.

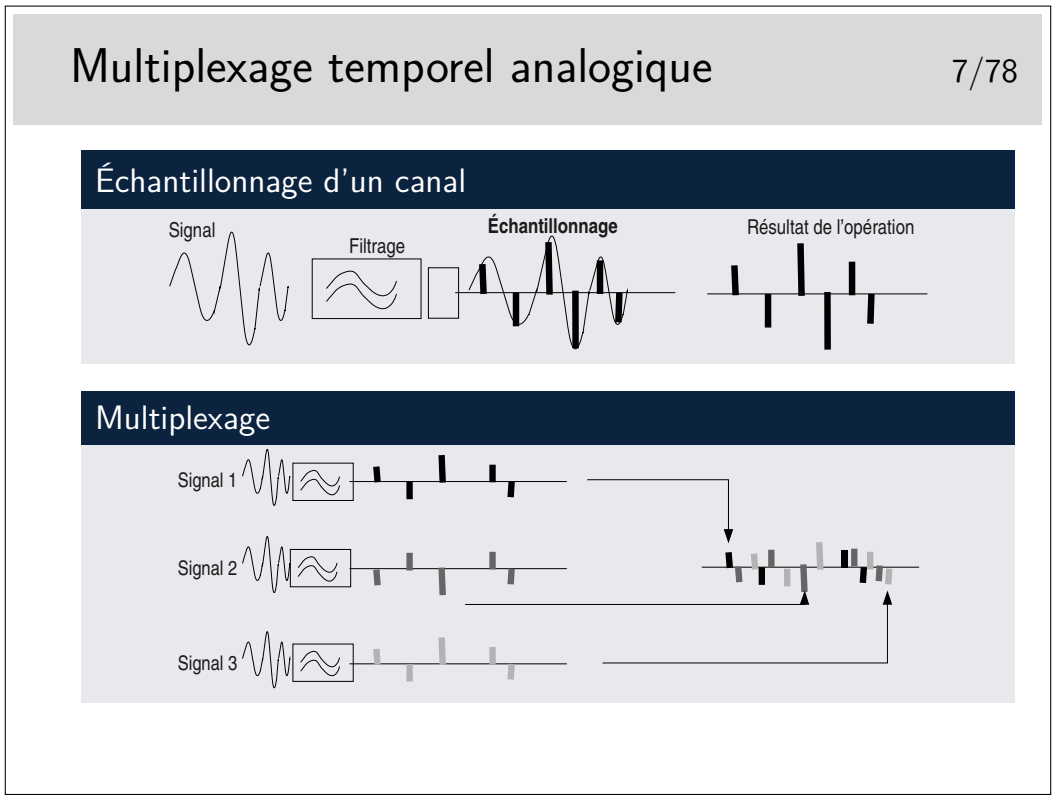
— Ce résultat de l'opération de modulation est filtré pour ne garder qu'une bande inférieure ou supérieure (ici supérieure) de largeur 4000 Hz (de fp à fp+4000).

— Le multiplexage consiste à placer sur le même support le résultat de la modulation et

filtrage de 12 canaux avec 12 fréquences porteuses différentes espacées chacune de 4000 Hz.

— On constitue ainsi ce qu'on appelle un groupe primaire qui à son tour peut être considéré comme un signal analogique et qui peut venir moduler une fréquence porteuse supérieure. Plusieurs groupes primaires peuvent ainsi être multiplexés pour constituer un groupe secondaire et ainsi de suite jusqu'à quatre niveaux.

Le multiplexage sur fibre optique est réalisé de manière similaire. Des communications différentes peuvent être transportées par une même liaison optique, mais sur des longueurs d'onde (couleurs) différentes : c'est le multiplexage en longueur d'onde ou WDM (*Wavelength-Division Multiplexing*).



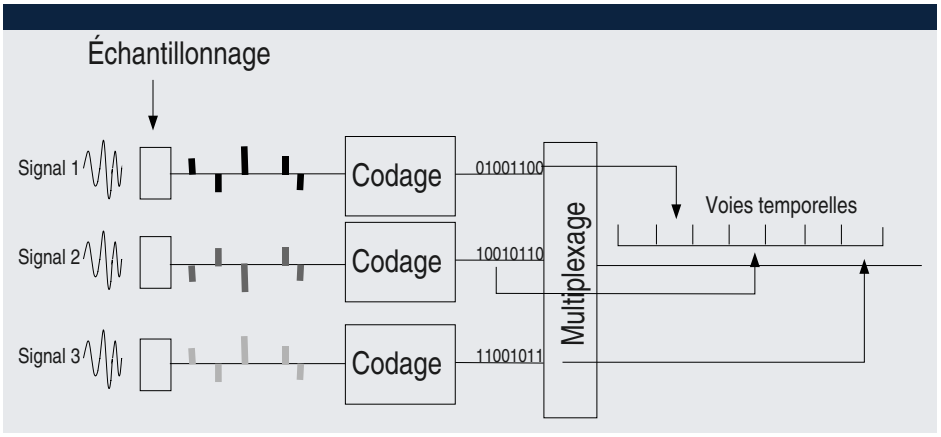
Chaque signal est échantillonné un certain nombre de fois par secondes, chacun avec un décalage dans le temps par rapport aux autres de telle manière qu'ils peuvent être véhiculés sur le même support sans se mélanger après multipexage.

La fréquence d'échantillonnage est telle que le signal doit pouvoir etre reconstitué sans déformations majeures. Shannon a montré que la fréquence d'échantillonnage devait être être le double de la fréquence maximum du signal à échantillonner. Ainsi, en téléphonie, la bande de fréquence est de 300-3400 Hz, on prend 0-4000 Hz par excès, donc la fréquence d'échantillonnage doit être de 8000 Hz.

Le multiplexage par échantillonnage analogique seul ne permet pas de bonnes performances. Le mutiplex ne doit pas être très long, quelques dizaines de centimètres par exemple dans un tout petit autocommutateur téléphonique. Si le support est plus long les échantillons se détériorent à cause des caractéristiques capacitives et selfiques du support, ils sont bruités et déformés et risquent de se mélanger.

La solution réside dans la numérisation.

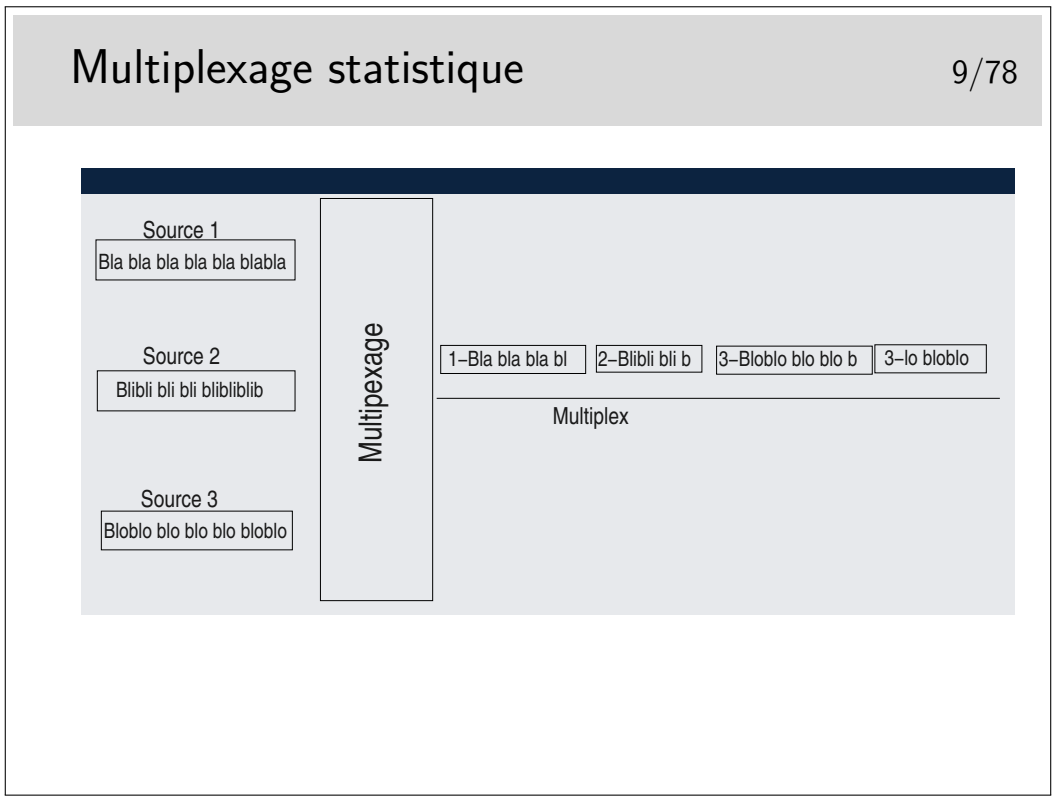
Multiplexage temporel numérique 8/78



Chaque échantillon est transformé en un nombre binaire pouvant être de 12 bits par exemple, ramenés par des techniques de compression à 8 bits.

En téléphonie par il existe deux techniques principales de codage standardisées par le document ITU-T G711, toutes deux respectent la loi de Shannon édictant qu'il faut échantillonner à 8KHz, donc un échantillon toutes les 125µs. Elles divergent sur les points suivants :

- Loi européene dite loi A (Alaw) :
  - méthode de codage spécifique
  - 3,9µs par voie, (8 bits, 488ns par bit)
  - chaque canal a un débit de 64 Kb/s (calculez...)
  - 32 voies par multiplex à 2,048 Mb/s
  - une voie pour la synchronisation du multiplex
  - une voie pour la signalisation des voies téléphoniques
  - 30 voies utiles (ou 31 si la voie de signalisation n'est pas utilisée)
  - multiplex appelé E1 pour le téléphone
- Loi nord américaine et japonaise dite loi µ (µLaw)
  - méthode de codage spécifique
  - 24 voies par mutiplex
  - 24 voies utiles, certains bits pris sur les voies téléphoniques servent à la synchronisation et à la signalisation → 56 Kb/s par voie
  - multiplex appelé T1



Les données sont découpées en séquences de *longueur variable* qui sont ensuite injectées sur le même support, les unes à la suite des autres. Il n'y a pas d'ordre à priori. Si une seule source émet elle a toute la bande passante du support à sa disposition. Si n sources émettent simultanément elles disposent chacune de 1/n de la bande passante (si leurs données sont de taille égales).

Les séquences ne doivent pas être trop longues car alors le support est dédié trop longtemps à une seule source.

Les séquences doivent pouvoir être identifiées pour pouvoir les reconnaître à la réception et les remettre à leur bon destinataire. Elles sont munies d'une «étiquette» d'identification qui peut être un numéro de canal ou une adresse de réception.

Le canal est virtuel, il n'existe que si des données sont présentes.

Ce mode de multiplexage est utilisé pour le transfert de données. Il est universellement répandu : ethernet, IP (donc Internet), etc...

Les séquences de données sont appelées «paquets».

Parfois les paquets de données sont de taille constante, on les appelle alors des cellules (réseau de type ATM par exemple).

Avantages et inconvénients des différents types de multiplexage I

- Multiplexage temporel
  Multiplexage en fréquence
  Multiplexage en longueur d'onde
  - ressource (la voie temporelle) réservée pour la durée de la communication, même si celle-ci est silencieuse : mauvaise utilisation de la ressource
  - bande passante et délais de transfert garantis

Avantages et inconvénients des différents types de multiplexage II

- Multiplexage statistique
  - utilisation optimale du canal, si des sources sont silencieuses, le canal peut être utilisé par d'autres
  - bande passante globale partagée entre toutes les sources, pas de garantie de réservation (sauf Frame Relay et ATM au prix de complexité supplémentaire pour la réservation et le contrôle de l'utilisation)
  - pas de délai de transfert garanti
  - gigue (variation des délais) pouvant être importante
  - C'est actuellement le moyen le plus utilisé pour les données
  - Les applications temps réel sont mal adaptées à cette technique car les délais et la bande passante ne sont pas garantis (sauf surdimensionnement)

La paquetisation dans les réseaux de données 12/78

- Multiplexage statistique
  - Pendant qu'une source émet sur un multiplex, les autres sources doivent être silencieuses
  - Pour rendre équitable l'utilisation du multiplex, une source ne peut pas le monopoliser trop longtemps, il faut limiter sa durée d'émission
  - Les données sont segmentées en unités appelées «paquets»
  - Les paquets ont une taille maximale et parfois une taille minimale
  - Les paquets doivent être munis d'une entête, sorte d'étiquette, qui permet de les reconnaître et ainsi de savoir en réception vers quel destinataire acheminer le paquet

1.2 Notion de connexion

Communications avec ou sans connexions 14/78

- Avec connexion, comme le téléphone ?
  - Il faut chercher un chemin dans le réseau entre la source et la destination puis le réserver et l'établir
  - Lorsque la communication est terminée il faut libérer le chemin
  - Ces opération nécessitent des échanges d'informations spécifiques que l'on appelle la signalisation
  - Le chemin est appelé circuit
- Sans connexion, comme à la poste ?
  - On munit les données «d'enveloppes» contenant l'adresse de la destination et le réseau achemine ces données en les routant dans chaque nœud en fonction de cette adresse
  - Les unités de données véhiculées sont appelées des datagrammes

Avantages et inconvénients des modes avec et sans connexion I

- Mode orienté connexion
  - Avantages
    - Le chemin est toujours le même pour la durée de la connexion, les données sont reçues dans l'ordre ou elles ont été émises
    - La signalisation nécessaire à l'établissement de la connexion peut permettre de véhiculer des informations de demande de qualité de service
    - Les délais de traitement dans les nœuds sont généralement courts
  - Inconvénients
    - Il faut un certain délai d'établissement et de rupture de la connexion
    - Le chemin est préétabli et si une maille du réseau devient inutilisable (panne d'un nœud, rupture de la maille) la communication est rompue

Avantages et inconvénients des modes avec et sans connexion II

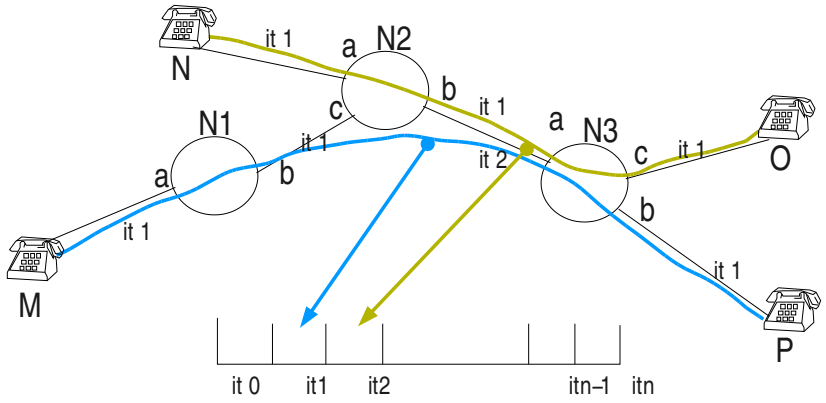
- Mode sans connexion
  - Avantages
    - Pas de nécessité de signalisation pour établir des chemins
    - Reroutage facilité des données en cas de rupture d'un lien dans le réseau
  - Inconvénients
    - Il n'y a pas de chemin préétabli, deux unités de données successives peuvent arriver dans le désordre si le chemin de la seconde a été plus court que le chemin de la première en cas de modification de parcours entre les deux unités
    - On peut envoyer des données vers des destinations inexistantes

Réseaux avec et sans connexion 17/78

- Orientés connexion
  - X25 : le réseau de données des années 80 (né vers 1976)
  - Frame Relay, très utilisé aujourd'hui pour interconnecter des unités dispersées de mêmes entreprises
  - ATM : dans le cœur de réseau des opérateurs (mais aussi dans votre modem ADSL...)
  - MPLS (MultiProtocol Label Switching) : dans le cœur des réseaux d'opérateurs (au service des entreprises)
- Orientés sans connexion
  - Les réseaux locaux d'entreprises : Ethernet, Token-Ring
  - IP : Internet
  - Réseau Cyclades (l'ancêtre de IP, abandonné en 1978 par choix politique... [1])

1. https://www.franceculture.fr/emissions/superfail/pourquoi-la-france-a-t-elle-invente-le-minitel-plutot-quinterne

Connexion et circuit réel 18/78



- Circuit réel : information dans des *canaux* distincts, réservés
- Entre $N2$ et $N3$, les intervalles de temps 1 et 2 sont attribués aux communications $M-P$ et $N-O$, respectivement, et ce, pour la durée des communications
- Si la communication $M-P$ se termine, la ressource $it1$ est libérée, la communication $N-O$ n'en profite pas
- Si la communication $M-N$ est silencieuse, la bande passante non utilisée est perdue

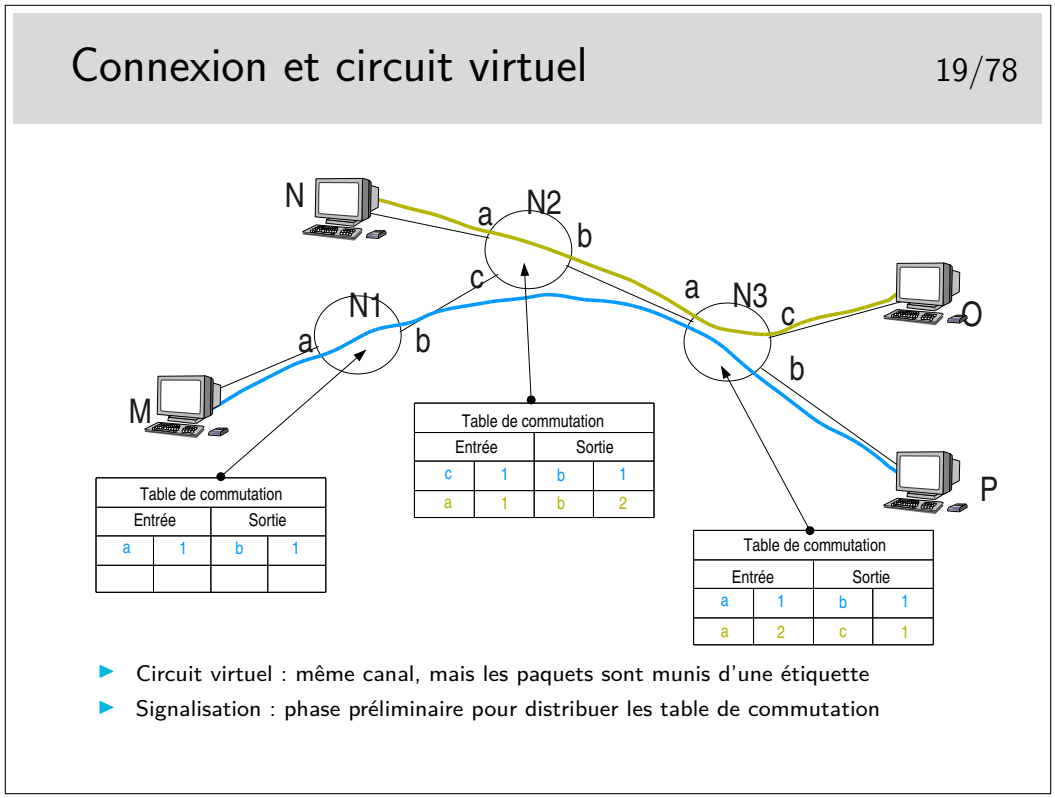
— Il n'y a pas d'étiquette associée aux données

— les données sont véhiculées dans des canaux spécifiques, réservé lors de la phase d'appel, établis lors de l'établissement de la connexion

— les canaux peuvent être des intervalles de temps, régulièrement espacés (en téléphonie numérique européenne : 32 intervalles de temps tous les 125µs)

— les noeuds du réseaux sont des commutateurs, ils *commutent* les données en fonction des informations contenues dans leurs tables de commutation

— les tables de commutation donnent la correspondance entre le canal entrant et le canal sortant



Les paquets issus de M et à destination de P sont munis d'une étiquette 1 en M. Elle reste 1 après passage dans tous les noeuds. Sans doute la communication M-P a-t'elle été établie la première.

Les paquets issus de N à destination de O sont munis d'une étiquette 1. En sortie du premier noeud traversé cette étiquette devient 2 car l'étiquette 1 est déjà attribuée à une communication.

La commutation de circuit virtuel consiste à échanger des étiquettes dans les noeuds et à orienter les unités de données (les paquets) vers les bonnes interfaces de sortie. Le chemin est un chemin d'étiquettes. Celles-ci (les étiquettes) sont réservées lors de l'établissement de la communication.

Caractéristiques des réseaux orientés connexion 20/78

- Les nœuds acheminent les unités de données (les paquets) entre les entrées et les sorties en effectuant des opération de commutation
  - Les nœuds sont des commutateurs
  - Les paquets sont munis d'étiquettes qui les identifient
  - Les étiquettes sont attribuées lors de la phase d'établissement de la connexion. Elles identifient les paquets sur chaque lien.
  - Elles identifient aussi la communication, le canal, le circuit.
  - La série d'étiquette réservées pour une communication sur chaque lien constitue un circuit virtuel

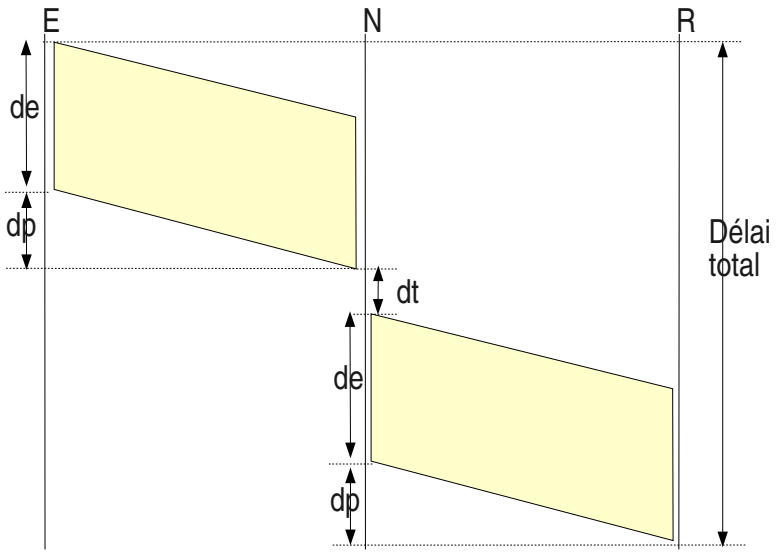
1.3 Délais et QoS

Les délais 22/78

- Entre un émetteur et un récepteur, les données vont subir différents délais
  - Durée d'émission : inversement proportionnelle au débit, proportionnelle à la longueur des segments de donnée
  - Le délai de propagation : au minimum le délai de la lumière dans le vide ($c \approx 300000 Km/s$), dépend par ailleurs du support ($2/3c$ dans le verre et le cuivre).
    Exemple : 250ms pour une liaison par satellite géostationnaire (terre - terre)
  - Le délai de traitement dans les nœuds : délai de traitement réel plus temps passé dans les files d'attente en attente de traitement et en attente de ré-émission

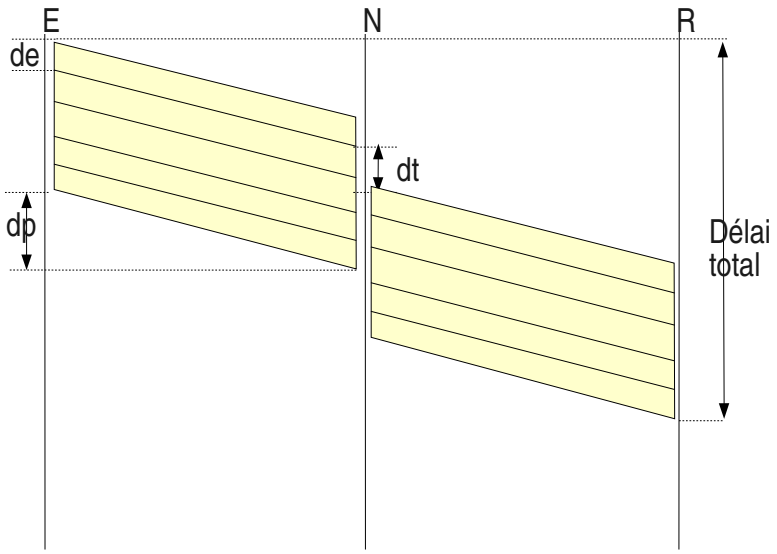
Relation célérité de la lumière dans le vide (c), fréquence (f) et longueur d'onde (λ) : $\lambda f = c$

Les délais 23/78



Délai total = $2de + 2dp + dt$ si un seul nœud intermédiaire et si les débits sont les mêmes sur les tous les liens.

Réduisons les délais avec de petits paquets 24/78



Les interfaces des nœuds fonctionnent en parallèle. Une interface peut recevoir pendant qu'une autre émet. Il reste cependant le temps de traitement, c'est à dire le temps mis par l'unité centrale du nœud pour déterminer vers quelle interface de sortie il faudra acheminer le paquet, plus le temps passé dans les mémoires files d'attente.

Notion de qualité de service liée aux délais 25/78

- Pour le téléphone
  - Délai de 50ms : bon confort de communication
  - Au delà de 200ms (communications par satellites geostationnaires) : qualité très moyene, nécessité d'annulation d'écho
  - 400ms : toute dernière extrémité à ne pas dépasser
- Pour les données
  - Dépend fortement du type d'application
- Variation des délais
  - Inhérente aux réseaux de transmission de données
  - Inadapté aux services de type téléphone ou vidéo (mais on tente quand même avec succès, p.ex. : VoIP)

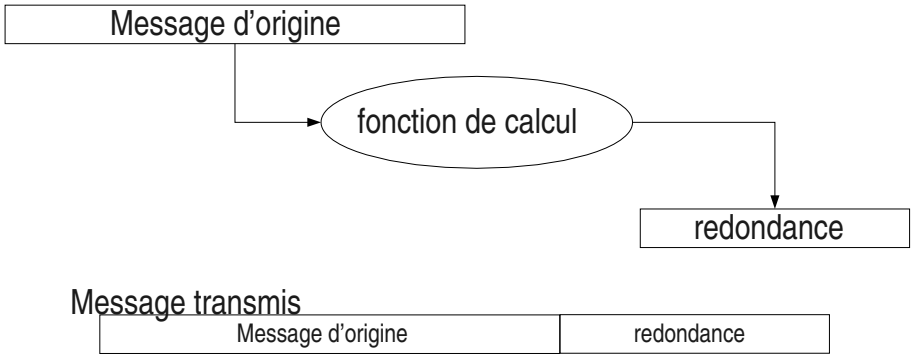
1.4 Détection et correction d'erreur

Détection et correction d'erreur 27/78

- Une transmission binaire ne se fait pas sans erreur : des «1» peuvent devenir des «0» et réciproquement
  - Comment détecter des erreurs ?
  - Comment les corriger ?
- Détection (principe général)
  - transmet l'information par bloc plus une séquence de redondance
  - la séquence de redondance est calculée par une fonction spécifique
  - le même calcul est fait à l'arrivée et le résultat est comparé
- Correction
  - directe si la redondance est suffisante et possède des propriétés de correction
  - par retransmission

Détection d'erreur : par séquence de redondance

28/78



Message transmis

À la réception l'opération est reconduite, le résultat est comparé à ce qui est reçu et le message est accepté ou non.

Exemple : bit de parité, CRC (*Cyclical Redundancy Code*), somme de contrôle (*Checksum*), ...

Dans les cas très répandus où les opérations de détection d'erreur sont simples et où les algorithmes ne permettent pas la correction immédiate, on ne peut pas dire si les erreurs de transmission portent sur le message lui même ou la redondance, ou les deux. On jette tout simplement le message (on l'ignore).

On ne se pose pas, à ce stade, le problème de la correction. Ce n'est pas l'affaire de l'algorithme, on s'en remet pour cela à des mécanismes situés dans des couches protocolaires supérieures (si on en a besoin).

Détection d'erreur : la méthode du bit de parité

29/78

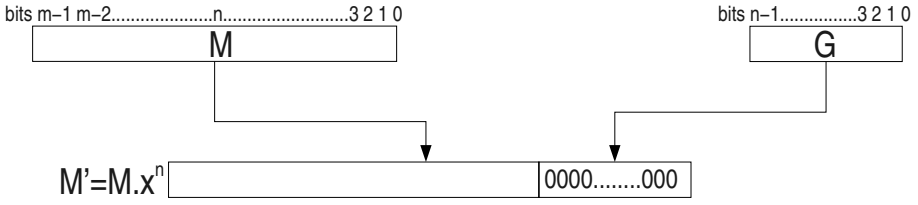
- Un bit de parité est rajouté à une séquence de bits
  - le nombre de bits à 1 résultant doit être pair (parité paire) ou impair (parité impaire)
  - Exemples :
    - parité paire

      1 0 0 1 0 0 1 **1**
      0 0 0 0 1 1 0 **0**
    - parité impaire

      1 0 0 1 0 0 1 **0**
      0 0 0 0 1 1 0 **1**
  - Ne permet pas de détecter des erreurs doubles
  - Méthode utilisée par exemple dans les interfaces séries de nos PC (interfaces de type ANSI RS232-C (ITU-T V24))

Détection d'erreur : la méthode de la division par polynôme I

- Un message M est une suite de bits, donc un nombre. On peut l'assimiler à un polynôme. Exemple : la suite de bits 101101 peut être représentée par le polynôme $x^5 + x^3 + x^2 + x^0$
- Le message M a au plus m bits, il est de degré $m - 1$
- Considérons un polynôme G de n bits (degré n-1) avec $n < m$
- On multiplie M par x^n (2^n). Cela revient à «décaler» M de n bits vers la gauche et à ménager ainsi n bits vides à droite



Cette méthode est appelée CRC (*Cyclical Redundancy Code*)

Détection d'erreur : la méthode de la division par polynôme II

- Le polynôme M' est divisé par G (division modulo 2)
- Alors : $M' = G \times Q + R$ (Q polynôme quotient, R reste)
- Le polynôme $T = M' - R$ est construit
  - le reste R vient se placer dans l'espace droit de M' (la soustraction binaire modulo 2 est équivalent à une addition)

    $M' = M.x^n$ | 0000..0000

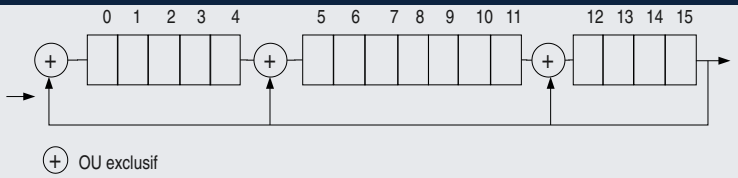
    $T = M' - R$ | R
  - $T = M' - R = G \times Q$ donc T est divisible par G
- On transmet T, à la réception on divise le polynôme reçu par G, si on a reçu T, pas d'erreur, alors $R = 0$

Le reste de la division est appelé CRC du nom de la méthode.

Quelques CRC types et le calcul par registres à décalage

- CRC-12 $= x^{12} + x^{11} + x^3 + x^2 + x + 1$
- CRC-16 $= x^{16} + x^{15} + x^2 + 1$
- CRC-CCITT $= x^{16} + x^{12} + x^5 + 1$
- CRC-32 $= x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

Codage du CRC-16 à l'aide d'un registre à décalage



Facile à implémenter matériellement...

Dans l'histoire des télécomunications des réseaux et de l'informatique, un certain nombre de polynôme de CRC ont été standardisés, offrant oist une *bonne capacité* à détecter les erreurs, soit étant faciles à implémenter (tout est affaire de compromis). Voir `http://en.wikipedia.org/wiki/Cyclic_redundancy_check#Common_polynomials`.

Existe aussi en solutions logicielles, mais plus le message est long plus il y a de temps de calcul. Voir les implémentations en C du CRC-32, par exemple ici :
`http://www.cl.cam.ac.uk/Research/SRG/bluebook/21/crc/crc.html`

Des fonctions existent toutes faites, par exemple en PHP : `int crc32(string str)`
voir : `http://fr2.php.net/crc32`

Détection d'erreur : la méthode de la somme de contrôle

33/78

(ou *Checksum*)

- Soit un message formé par la suite de mots A, B, C, D, ..., R, S, T
  - Les mots sont des ensemble de 8 ou 16 bits ou plus
  - Un mot contiendra la somme de contrôle dans le message, soit S ce mot. Il est mis à 0 pour le calcul
  - Calcul : $Z = A + B + C + D + \ldots + R + 0 + T$
  - Le résultat Z est inversé, on obtient $\bar{Z}$
  - On transmet $ABCD \ldots R\bar{Z}T$
  - À la réception on fait la somme $Z' = A + B + C + D + \ldots + R + \bar{Z} + T$
  - S'il n'y a pas d'erreur alors $Z' = Z + \bar{Z} = 1111 \ldots 1111$
  - le résultat est inversé comme à l'émission et donc $\bar{Z'} = 0$

Rappel (s'il en est besoin) : si un mot Z vaut 1010, alors son inverse (on dit aussi son complément à 1) vaut 0101 et peut être noté $\bar{Z}$ (Z barre).

Détection et correction d'erreur

34/78

- Si la redondance est suffisante et l'algorithme sufisamment puissant il est possible de détecter les erreurs et de les corriger.
  - On peut le montrer sur un exemple simple avec le mécanisme du bit de parité.
    Soit le bloc suivant, chaque ligne et chaque colonne est munie d'un bit de parité paire (dernier bit).

	1	0	1	0	1	0	1	0
	1	0	1	0	0	0	0	1
Cherchez l'erreur :	0	0	0	1	0	1	0	0
	1	1	1	1	0	0	0	0
	1	1	1	0	0	1	1	1

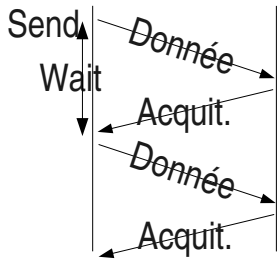
  - Autres techniques : codes de Reed-Solomon, Turbo-codes (origine ENST Bretagne), etc.

— Reed-Solomon : `http://en.wikipedia.org/wiki/Reed%E2%80%93Solomon_error_correction`

— Turbo-codes : `http://en.wikipedia.org/wiki/Turbo_codes`

Correction d'erreur par retransmission I 35/78

- Un récepteur détecte une erreur, il jette le segment de données erroné
- Comment l'émetteur peut-il détecter qu'il y a eu une erreur ?
  - Impossible sans l'utilisation d'un mécanisme d'acquittement

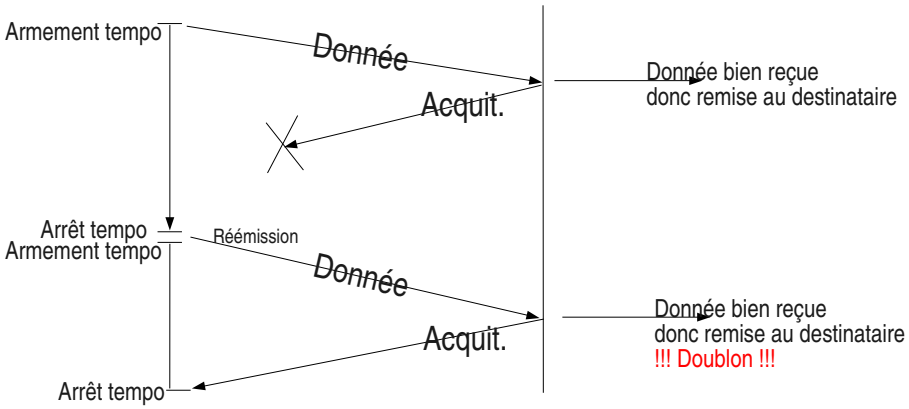


Ce mécanisme est appelé «Send & wait»

Que se passe-t'il si la donnée se perd ?
Si l'acquittement se perd ?

Correction d'erreur par retransmission II 36/78

- Utilisation de temporisateur (timer)

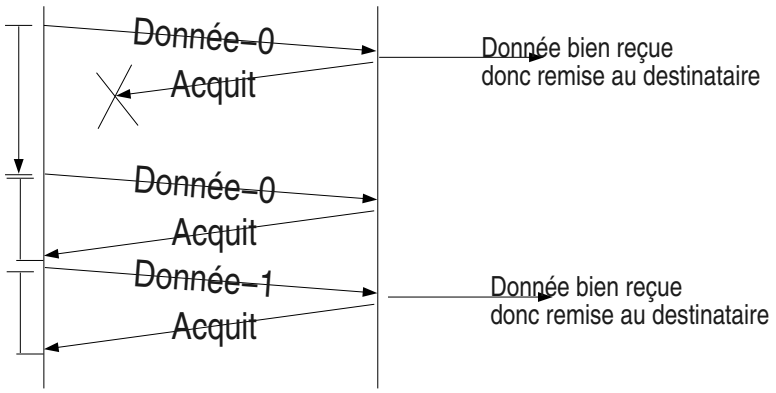


Où comment se créer d'autres problèmes en voulant en régler un... Et maintenant comment on fait pour éviter les doublons ?

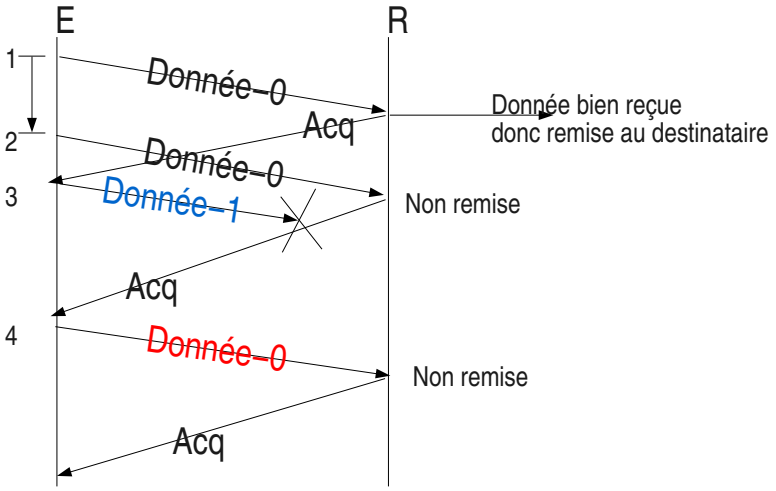
C'est une règle, hélas, très générale, en Réseaux, que de se créer de nouveaux problèmes en apportant des solutions à d'autres...

Retransmission : le problème des doublons 37/78

- On peut éviter les doublons en numérotant les données
  - Exemple : les données sont munies d'un numéro 0 et 1 alternativement



Timer trop court et malchance... 38/78

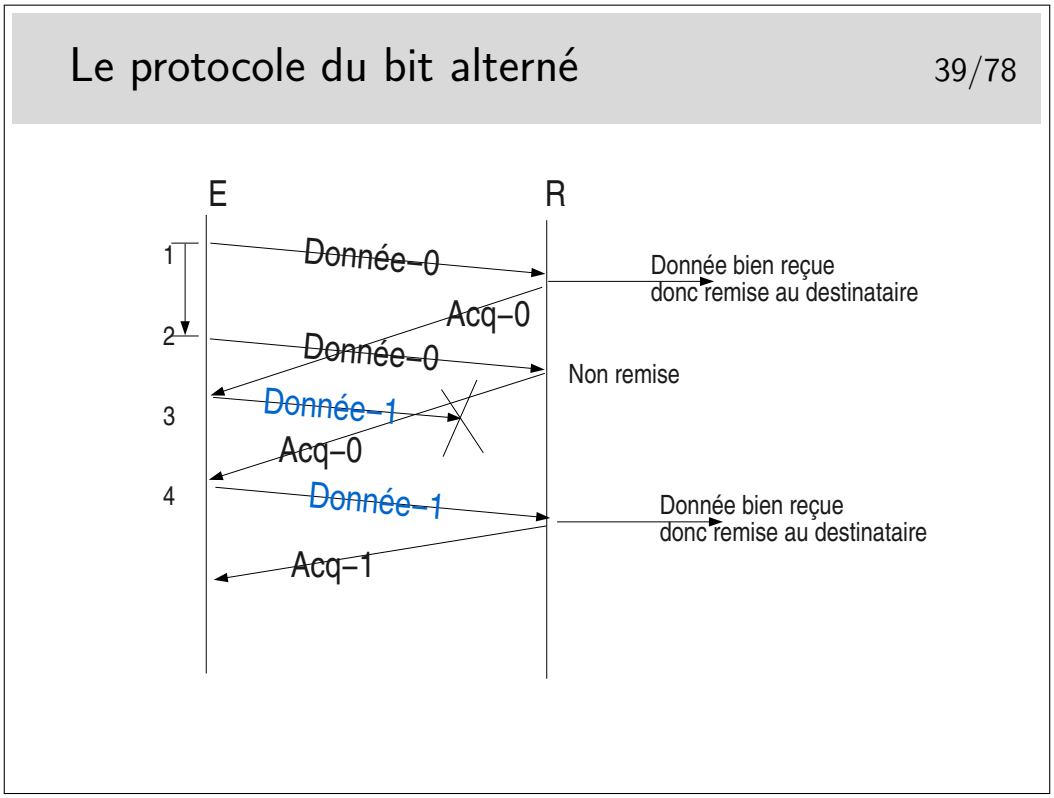


1. émission de `Donnée-0`, bien reçue, elle est remise à l'entité destinatrice
2. le timer expire trop tôt, `Donnée-0` est réémise et est bien reçue, elle n'est pas remise car elle porte le même numéro que la donnée précédente. Cependant, le récepteur renvoie un acquittement car il pense que s'il a reçu à nouveau `Donnée-0` c'est que son acquittement précédent s'est perdu..
3. un acquittement arrive, l'émetteur peut alors émettre une nouvelle donnée : `Donnée-1`
4. un acquittement arrive, l'émetteur pense qu'il s'agit de l'acquittement de `Donnée-1`.

À chaque fois qu'il reçoit un acquittement il pense que celui-ci concerne les données précédemment émises, il vide les tampons mémoires qui les contenaient.

Ainsi, en 4, s'il y a une nouvelle donnée à envoyer ce sera `Donnée-0` (pas le même contenu qu'au début du scénario), or le récepteur voyant arriver à nouveau un numéro 0 le rejettera. Le numéro est censé représenter les données, on ne compare pas celle qu'on reçoit avec les précédentes. D'ailleurs celles-ci ont été livrée à l'entité de destination, elles ne sont pas gardées en mémoire.

Dans ce scénario le segment `Donnée-1` n'est pas reçu mais il est considéré par l'émetteur comme bien reçu. Le segment `Donnée-0` suivant est bien émis et bien reçu mais il n'est pas remis à son destinataire final... Tout va mal dans ce scénario !



Les acquittements portent les numéro des données qu'ils acquittent. Ainsi, en 4, on reçoit à nouveau `Acq-0`, alors qu'on attendait `Acq-1`, il y a un problème, le contenu de `Donnée-1` précédent est toujours en mémoire (on ne le vidra que sur réception de `Acq-1`), on peut donc réémettre `Donnée-1` (le même message que précédemment).

1.5 Des protocoles

Notion de protocole I

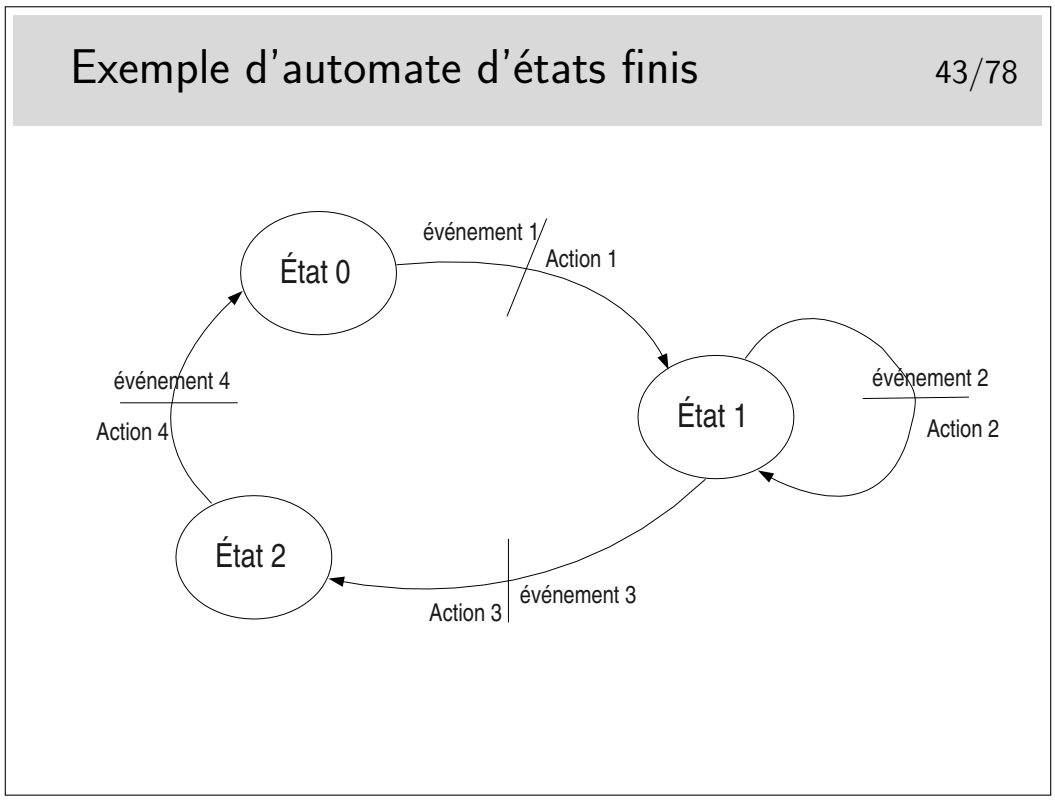
41/78

- Dans ce qui précède nous avons développé des moyens de transmettre des données sans erreurs en rajoutant de l'information et des messages spécifiques
  - À chaque ajout de mécanisme on corrige un problème, on en rajout un autre
  - Il faut que la série correction/nouveau problème converge

Notion de protocole II

42/78

- En rajoutant un protocole au dessus d'une voie de communication on modifie les propriétés de cette voie de telle manière que l'on obtient une nouvelle voie de comunication
- Pour mettre en œuvre et gérer les mécanimes il faut construire un automate logiciel comportant des états et des transitions d'états sur événements



Au départ l'automate est dans un certain état, disons l'état 0 par exemple. Seul l'arrivée de l'événement 1 peut faire que l'automate passe dans l'état 2. Au passage dans ce nouvel état l'action 1 sera effectuée. Un événement ne fait pas toujours changer d'état, on l'illustre ici par l'état 1, dans lequel on reste après que soit survenu l'événement 2 et que l'action 2 ait été effectuée.

Cette représentation est proche de la formalisation mathématique des automates par les «réseaux de Petri».

La représentation schématique est intéressante car elle est souvent plus facile à comprendre (lorsqu'il n'y a pas trop d'états ni d'événements). Cependant elle ne permet pas de s'assurer que tous les cas possibles de relation événement/transition soient envisagés. Il faut alors recourir à la représentation de l'automate sous forme de tableau.

L'automate du protocole du bit alterné 44/78

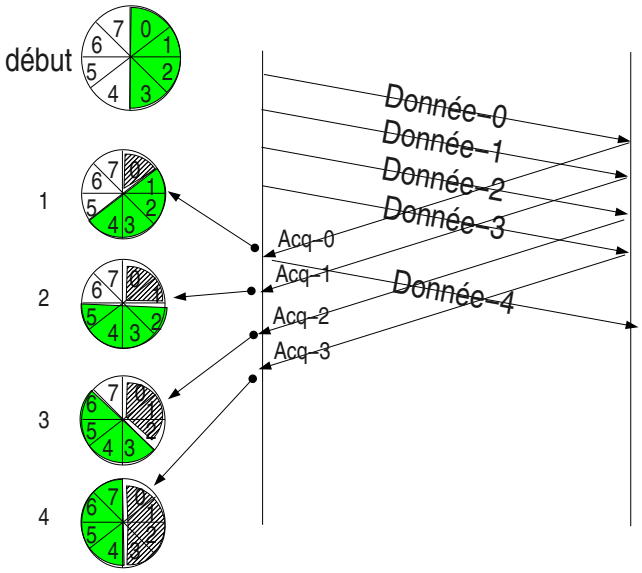
Événement État	Demande d'envoi de donnée	Arrivée Ack 0	Arrivée Ack 1	Expiration Timer
État 0 nouvel état	Envoyer Data-0 Armer Timer Attente Ack 0			
Attente Ack 0 nouvel état		Arrêt timer État 1		Envoyer Data-0 Armer Timer Même état
État 1 nouvel état	Envoyer Data-1 Armer Timer Attente Ack 1			
Attente Ack 1 nouvel état			Arrêt timer Etat 0	Envoyer Data-1 Armer Timer Même état

Les protocoles à anticipation 45/78

- le mécanisme du *Send & Wait* conduit à une mauvaise utilisation de la bande passante, quand on attend on n'utilise pas la bande passante alors disponible
- Il est judicieux d'anticiper l'émission sans attendre les acquittements
  - Dans une certaine mesure...
  - Une limite, appelée fenêtre d'anticipation, permet de ne pas trop anticiper et bloque l'émission si les acquittements n'arrivent pas
  - Les acquittements existent toujours
  - La fenêtre «tourne» ou «glisse» lorsqu'un acquittement arrive

On dit que la fenêtre est «tournante» ou «glissante» selon la représentation qu'on en fait. Voir le transparent suivant.

La fenêtre d'anticipation 46/78



Paramètres : les segments de données sont numérotés modulo 8. La fenêtre d'anticipation est de 4.

On ne peut pas émettre plus de 4 segments à la suite si on ne reçoit pas pendant ce temps un acquittement. Lorsqu'un acquittement arrive, la fenêtre «tourne» d'un pas, le tampon mémoire contenant la donnée correspondant à l'acquittement est libéré, la donnée est considérée comme bien envoyée, on l'efface. En pratique on ne l'efface pas, on considère son emplacement mémoire comme libre. Sur le schéma ci-dessus, on indique cette «libération mémoire» par une zone grisée dans la fenêtre.

Départ : la fenêtre englobe les numéros 0 à 4. On commence l'envoi. L'acquittement pour la donnée 0 arrive en 1. La fenêtre tourne et englobe 1 à 4. Le segment 4 devient éligible à l'émission. Il est émis si une donnée est à émettre. Le tampon mémoire contenant Donnée-0 est effacé. En 2 l'acquittement pour Donnée-1 arrive, la fenêtre tourne et englobe maintenant 2 à 5. Etc.

Le contrôle de flux 47/78

- Mécanisme permettant à un récepteur d'asservir la capacité à émettre de son correspondant en fonction de ses capacités de traitement
  - Permet d'informer l'émetteur qu'il doit réduire son débit
  - Permet de ne pas inonder les tampons mémoire du récepteur

Un émetteur et un récepteurs ne fonctionnent pas obligatoirement à la même vitesse (au même débit). Les tampons mémoire de réception se vident lorsque les applications destinatrices viennent y puiser les données reçues. Si la machine de réception est lente, si l'application réceptrice prend trop de temps à traiter les données et ne vient pas les retirer suffisamment rapidement des tampons de réception, ceux-ci se remplissent dangereusement. Lorsque les tampons sont pleins les données à recevoir seront perdues. Le contrôle de flux permet d'éviter ces pertes en évitant que les données ne soient envoyées.

Le contrôle de flux pourra être couplé à des mécanismes d'acquittement, ce sont les trames RR et RNR du protocole HDLC-LAPB que nous verrons plus loin : RR pour *Receiver Ready* ou encore «tout va bien, envoyez !», RNR pour *Receiver Not Ready* ou encore «OK, j'ai bien reçu vos données mais arrêtez vous quelque temps».

2 Modélisation et standardisation

2.1 Standardisation

Les organismes de standardisation 50/78

- Les organismes officiels nationaux et internationaux
  - OSI : Organisation de Standardisation Internationale (ISO en anglais), et ses branches nationales : AFNOR en France, DIN en Allemagne, ANSI aux USA
  - UIT-T : Union Internationale des Télécommunications, secteur des Télécommunications (il y a aussi le secteur Radio)
  - ETSI : European Telecommunications Standards Institute
- Les organismes de l'industrie et de la recherche
  - IEEE : Institut of Electrical and Electronics Engineers
- L'Internet
  - IETF : Internet Engineering Task Force, étudie et développe les protocoles et services au dessus du protocole IP

2.2 Principe de la modélisation

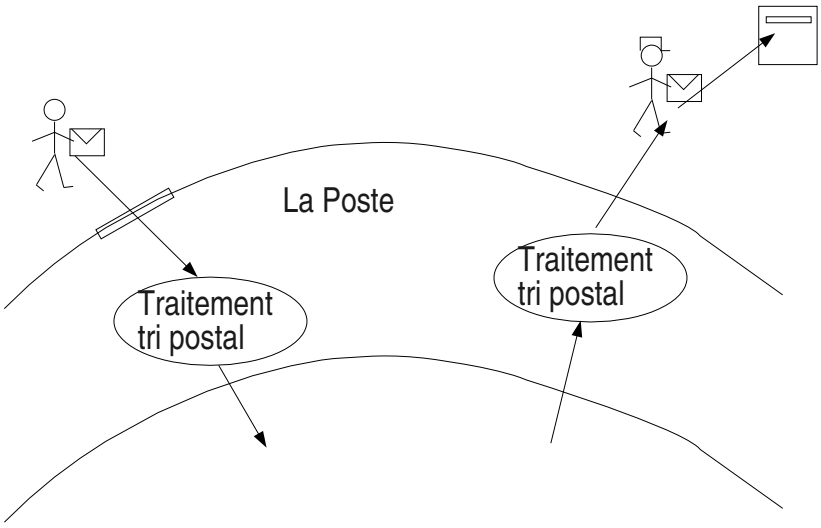
La modélisation de la problématique réseau 52/78

- Comment «voir» le réseau ?
  - Comme un utilisateur : «*Je me connecte, je ne sais pas comment ça marche, je ne veux pas savoir comment ça marche, mais ça marche ! Et j'utilise.*»
  - Comme un développeur d'application communicantes : «*Par quel moyen programmatique mes applications peuvent-t'elles communiquer ? Dois-je savoir comment fonctionne le réseau ? Tout le réseau ? Une partie du réseau ?*»
  - Comme le gestionnaire du réseau : «*Dois envisager le réseau dans son ensemble, du câble aux applications ?*»

Exemple d'un réseau existant : La poste 53/78

- En tant qu'utilisateur :
  - *Je dois savoir où est situé le bureau de poste, l'adresse de mon correspondant et avoir une boite aux lettres. Je dois pouvoir interagir avec le service offert par La Poste. La poste doit me fournir un service et des moyens d'accès à ce service.*
  - *Je n'ai pas à savoir comment fonctionne le service de La Poste de l'intérieur*
- En tant que postier :
  - *Je dois savoir traiter les lettres, les orienter vers les bons sacs postaux, placer les sacs postaux dans les bons camions, voitures trains, avions...*
  - *Je n'ai pas à savoir comment fonctionne le service de transport qui achemine physiquement les sacs postaux.*

Services et interfaces d'utilisation 54/78



La Poste offre un *SERVICE*

On accède à ce service par des moyens appelés, en terme «réseaux», des *PRIMITIVES DE SERVICE*.

On interagit avec le service via les primitives de service dont le paramètre principal est une sorte d'adresse : l'adresse où est situé le bâtiment de La Poste pour aller «poster» sa lettre, l'adresse du destinataire de la lettre pour que le postier sache dans quelle boite aux lettres déposer celle-ci. Cette sorte d'adresse est appelée en termes réseaux le *POINT D'ACCÈS AU SERVICE* (le Service Access Point ou SAP).

Notion de couche, d'interface entre couches et d'indépendance entre couche.

Question : le travail d'acheminement de la lettre est-il terminé lorsque le postier distributeur (le facteur) a déposé la lettre dans la boite aux lettres du destinataire ?

Réponse : oui et non !!!

- Oui pour le service postal.
- Non pour le destinataire final. Il peut s'agir de la boite aux lettres de la famille Dupont. La lettre peut être pour Philippe Dupont et non pour Jacques Dupont. Le SAP «adresse des Dupont» n'est pas le seul SAP à considérer. Le SAP «Jacques» et le SAP «Phillipe» sont aussi à prendre en compte, mais pas au niveau du service postal (on dira plus tard : «pas dans la même couche»).

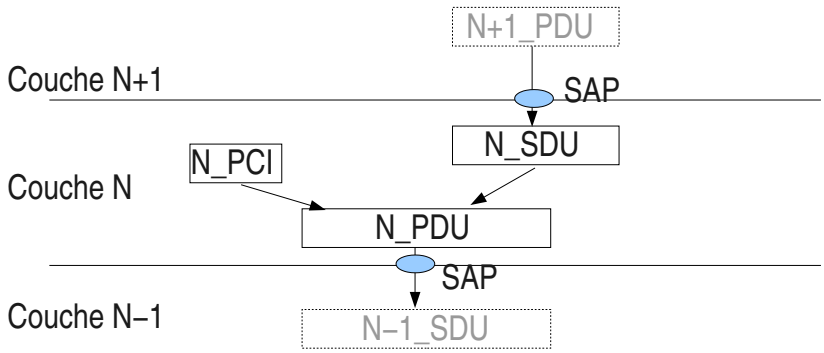
2.3 Le modèle ISO (OSI)

Le modèle ISO (OSI) 56/78

- Interconnexion de Services Ouverts (OSI en anglais)
- Définit la notion de service et de couche de service ainsi que les relations entre les entités distantes d'une même couche (les protocoles de communication)
- Définit aussi les relations entre couches (les primitives de service et les SAP)
- Définit les différentes couches, leur rôle ainsi que leurs protocoles

Le modèle OSI : Couches et unités de données

57/78



L'unité de donnée fournie par l'utilisateur est une unité de donnée à SERVIR. C'est une UNITE de DONNEE de SERVICE : une *SERVICE DATA UNIT* (SDU) en anglais.

La couche assurant le service utilise un certain mécanisme qui lui est propre, un *protocole* particulier, nécessitant un échange de données spécifiques. Ces données protocolaires n'ont rien à voir avec les données utiles, elles servent à la gestion du transfert de celles-ci.

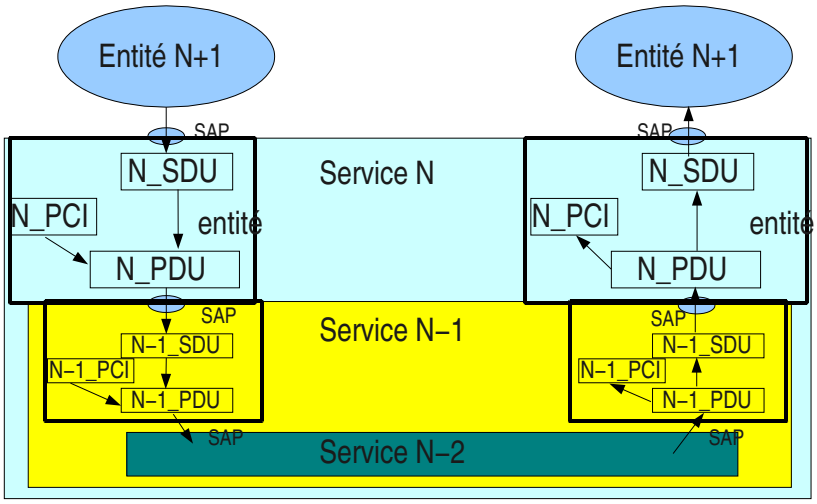
Les données protocolaires (Protocol Control Information) sont ajoutées au segment de données utiles (les données de service, la SDU) pour former une nouvelle unité : l'unité de données de protocole ou *PROTOCOL DATA UNIT* (PDU).

Notion d'entité protocolaire

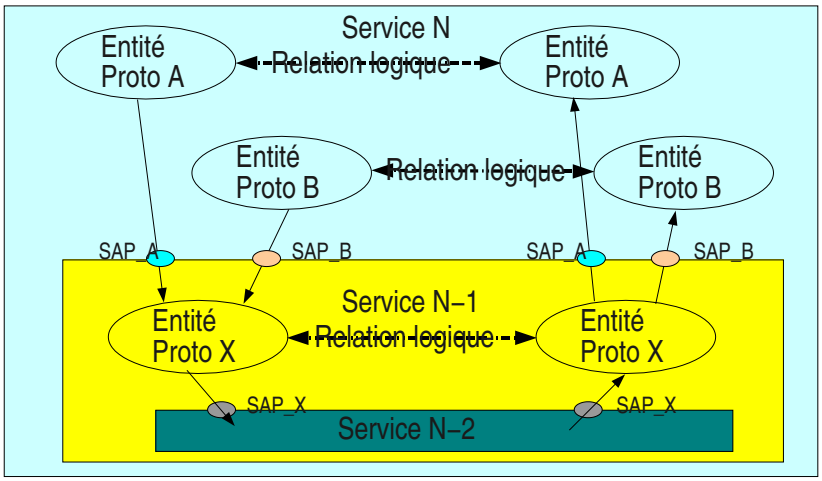
58/78

- Une entité protocolaire est un «programme informatique», une application ou un module opérationnel du système d'exploitation mettant en œuvre un protocole
- Une même couche peut être composée de plusieurs entités mettant en œuvre des protocoles différents pour assurer un même service
  - Exemples :
    - Le téléchargement de fichier peut être réalisé via le protocole ftp ou le protocole du web http
    - Des machines peuvent partager des fichiers via les protocoles NFS (monde Unix), SMB (monde Windows) ou AppleTalk (monde MacIntosh)

Les services et les couches 59/78



Relations entre entités de même niveau et niveaux différents 60/78



La figure ci-dessus est un exemple de ce qui est possible. Un service N met en œuvre deux types de protocoles différents pour assurer un service. Les entités protocolaires A peuvent communiquer entre elles. De même pour les entités protocolaires B. Les entités A ne peuvent communiquer avec les entités B car elles ne «parlent pas» le même langage (le même protocole).

Il est possible que les entités A et les entités B utilisent le même protocole X sous jacent (de couche $N-1$). Lorsque l'entité A de gauche émettra un PDU vers l'entité A de droite, il faudra que ce PDU soit muni de l'identité du SAP entre l'entité A (niveau N) et l'entité X de niveau $N-1$ (donc SAP A). Sinon l'entité X de droite ne saurait pas vers quelle entité réceptrice A ou B envoyer le PDU. De même pour les entités B.

Les 7 couches du modèle OSI
La couche 1 (*Physical Layer*)

1 - La couche physique

- Définit les moyens pour transformer les bits constituant les données en information analogique transportable sur les liens physiques entre la machine et son nœud de raccordement au réseau
- Définit les caractéristiques matérielles et électriques (ou optiques) des supports physique

Les 7 couches du modèle OSI
La couche 2 (*Data Link layer*)

2 - La couche liaison

- Définit les moyens d'acheminer des données structurées au dessus du niveau physique, entre la machine et son nœud de raccordement. La structure de base est la trame
- Définit les moyens de contrôler la fiabilité de la trame en réception
- Peut définir des mécanismes de contrôle de flux et de récupération d'erreurs

Niveau physique et niveau liaison : on ne traverse pas le réseau.

Les 7 couches du modèle OSI
La couche 3 (*Network Layer*)

3 - La couche Réseau

- Définit les moyens pour acheminer l'information entre machines d'extrémités en traversant les nœuds du réseau
- Implique une nécessité d'identification des machines terminales : un adressage
- Implique de mettre en œuvre des mécanismes de routage dans les nœuds
- Les unités de données de ce niveau sont appelées des paquets

Rappel de l'épisode précédent (niveau physique et niveau liaison) : on ne traverse pas le réseau !

Épisode présent (niveau 3) : on traverse le Réseau !

Les 7 couches du modèle OSI
La couche 4 (*Transport Layer*)

4 - La couche transport

- Dernière des couches basses
- Interface entre les applications et la couche réseau
- Fournit une abstraction du réseau
  - Corrige les imperfections de la couche réseau
    Dernière chance pour que les applications soient assurées du bon transfert de leurs données
  - Le modèle OSI fournit 5 classes de fonctionnalités différentes pour corriger les imperfections du réseau. De la classe 0 pour le très bon réseau à la classe 4 pour le mauvais réseau.

Tout ceci est très théorique. En pratique, il faut considérer ce niveau transport par type d'architecture : IP, Novell IPX, AppleTalk, IBM SNA, Decnet de Digital (racheté par Compaq, racheté par HP).

Les 7 couches du modèle OSI
Les couche 5 (*Session Layer*)

5 - La couche session

- Une communication entre deux applications est considérée comme une session
- La session est organisée, contrôlée
  - Il est prévu des points de synchronisation du dialogue

Les 7 couches du modèle OSI
Les couche 6 (*Presentation Layer)*

6 - La couche Présentation

- Fournit les moyens de décrire les types d'information échangés entre les application : langage de description de types : ASN.1
- Fournit les moyens de coder ces types de données dans un format indépendant de celui des machines (problème de la représentation *big* ou *little endian*)

Les 7 couches du modèle OSI
Les couche 7 (*Application Layer)*

7 - La couche application

- Fournit des sous ensembles applicatifs pour établir et contrôler les communications.

Tout ceci est à nouveau très théorique. Il faut le replacer dans le contexte de chaque architecture (IP, IPX, etc.)

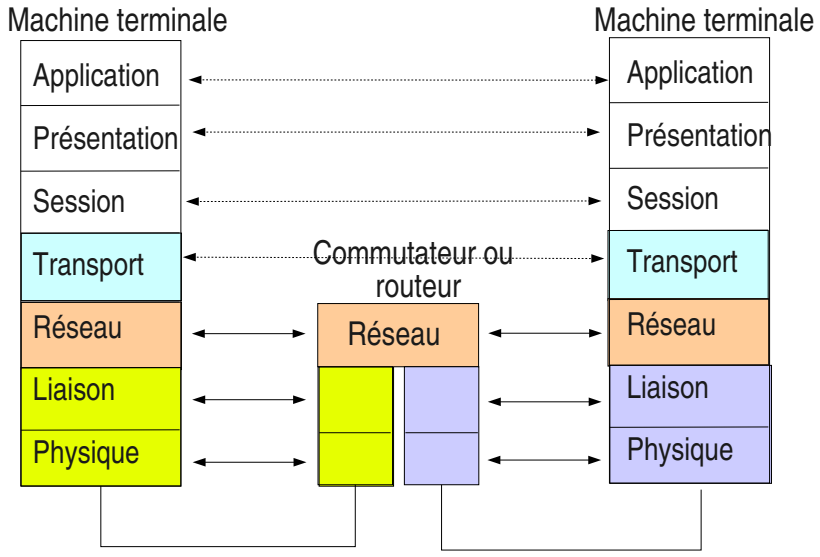
La couche application, qu'on «résume», ici, d'une seule phrase est la plus complexe de toutes dans la réalité des recommandations ISO et ITU-T. Même si ses mérites sont grands, sa complexité et le peu d'outils de développements (API) ont fait qu'il n'existe pas beaucoup de réalisations pratiques (courrier électronique X400, annuaire X500 et quelques autres). Ces applications ont vu le jour vers la fin des années 80, à une époque où l'Internet se répandait déjà beaucoup dans les réseaux du monde de l'enseignement et de la recherche. Le début des années 90 a été décisif, après ne courte période d'incertitude, l'ouverture des standards autour de IP, face à au monde OSI plus fermé à fait pencher la balance en faveur de IP.

Aujourd'hui, il reste cependant le modèle, incontournable, tout au moins pour les couches bases. Il nous aide à placer les concepts et les fonctionnalités des différents réseaux qui existent.

Les 7 couches du modèle OSI : pour résumer... 68/78

- Couches physique (1) et liaison (2) : de la machine au réseau
- Couche réseau (3) : de la machine à la machine en traversant le réseau
- Couche transport (4) : une abstraction du réseau (couches basses) pour l'applicatif
- Couches hautes (5 6 7) : des services, géré au niveau applicatif ou middleware

Où sont implémentées les couches ? 69/78



Toutes les couches sont implémentées dans les machines terminales.

Seules les couches 1 à 3 sont nécessaires dans les machines du cœur de réseau.

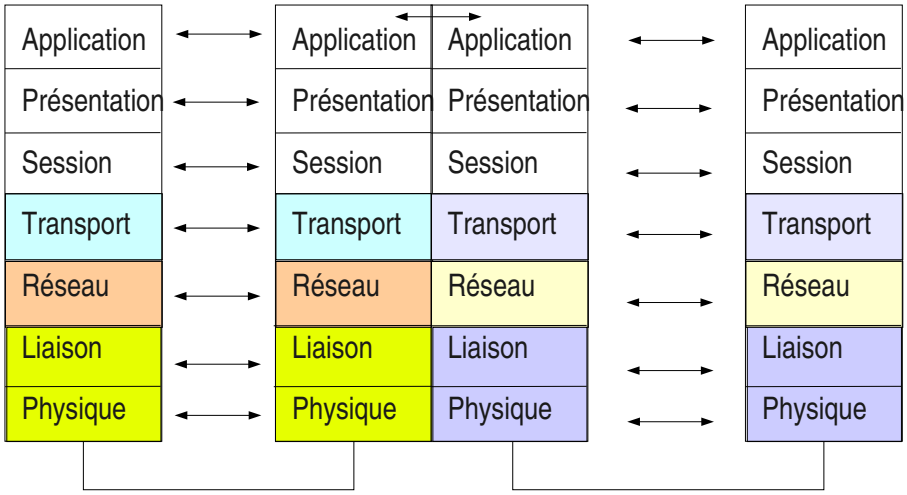
Il est très important de noter l'indépendance entre les couches. Voyez ici les couches physique et liaison concernant le lien coté gauche et le lien coté droit. Ces niveaux peuvent être totalement différents du point de vue physique mais aussi du point de vue couche liaison. On pourrait avoir, coté gauche un lien Ethernet sur paire torsadée et de l'autre un lien sur fibre optique.

Les débits peuvent aussi être différents. On pourrait avoir 1GB/s coté gauche et 155Mb/s coté droit.

Les couches Physique et liaison sont en général implémentées sur un même support matériel,

une carte d'interface du type de celle que vous pouvez avoir sur votre ordinateur personnel

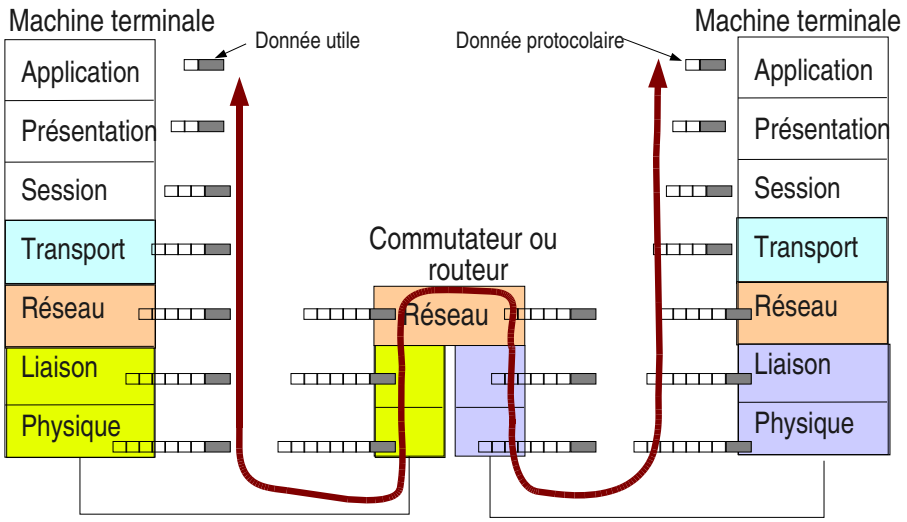
Notion de passerelle ou proxy 70/78



Toutes les couches sont implémentées dans la machine intermédiaires. La couche application réalise une traduction des données pour les adapter à l'application destinatrice. Les protocoles mis en oeuvre à gauche peuvent être tous différents de ceux de droite.

Dans le cas d'un proxy de type Web, les protocoles sont les mêmes à gauche et à droite, mais on oblige toutes les requêtes à passer par l'application intermédiaire (on imagine par exemple un client Web à gauche et un serveur à droite). Cela permet de réaliser des caches permettant de mémoriser les documents déjà téléchargés, afin de les renvoyer plus rapidement sans encombrer le réseau à nouveau.

Chaque couche rajoute ses données protocolaires 71/78



Notion de rendement : rapport entre le volume des données utiles et le volume total de données véhiculées sur un niveau.

Par exemple, pour une application cliente telnet, sur TCP IP sur Ethernet (nous verrons ces protocoles plus loin dans le cours) il est courant d'envoyer les données octet par octet dans le sens client vers serveur (au rythme où sont frappées les touches du clavier). Pour chaque octet envoyé, telnet ne rajoute rien mais TCP (niveau 4) rajoute au moins 20 octets (plus 12 avec les options des implémentations TCP d'aujourd'hui), IP (niveau 3) rajoute 20 octets. On a donc au moins $1 + 20 + 20 = 41$ octets et au plus $1 + 20 + 12 + 20 = 53$ octets. Dans le cas 41 octets, Ethernet rajoute 5 octets de bourrage pour arriver à la taille minimale requise, donc 46 octets. Par ailleurs Ethernet rajoute 18 octets d'informations protocolaires donc $46 + 18 = 64$ octets dans un cas et $53 + 18 = 71$ octets dans l'autre.

Le rendement est de 1/64 dans un cas et 1/71 dans l'autre. Ce n'est pas très efficace mais c'est ainsi !

3 Modélisation et standardisation de IP

L'organisation de l'Internet 73/78

- Organisme de tutelle : l'ISOC (Internet SOCiety), qui regroupe
  - Internet Architecture Board (IAB)
    - Responsable final des travaux de l'IETF
  - Internet Engineering Steering Group (IESG)
    - Contrôle les travaux de l'IETF
  - Internet Engineering Task Force (IETF)
    - L'organisme de standardisation
  - Internet Research Task Force (IRTF)
    - Responsable de la recherche à long terme
  - Internet Assigned Numbers Authority (IANA)
    - La gestion des adresses, des numéros de protocoles, du DNS

Il existe aussi l'ICANN (The Internet Corporation for Assigned Names and Numbers) société de droit privé destinée à remplacer IANA. L'ICANN a été fondée en 1998 et on a parfois du mal à faire la distinction entre ses responsabilité et celles de IANA qui continue d'exister.

Voir un article de synthèse sur ce lien : `http://www.renater.fr/Projets/ICANN/`

La standardisation Internet 74/78

- Les membres des groupes de l'IETF travaillent par échange de courriers électroniques, sur des documents appelés *drafts*, dont la validité est de 6 mois
- Périodiquement ils se rencontrent dans des «meetings» pour valider des choix techniques
- Un *draft* peut évoluer vers une nouvelle version, valable 6 mois de plus
- Lorsque que le groupe de travail arrive à un consensus, le *draft* est promu au rang de RFC (Request For Comment)
  - Les RFCs sont les «normes» Internet, l'équivalent des recommandations ISO et ITU-T
  - Mais pas seulement...
- RFC et drafts sont en publications gratuite sur différents sites ftp et web (`www.ietf.org`)

Les RFC 75/78

- Format standard : ASCII pur et dur (voir rfc 2223)
- Plusieurs types forts
  - Proposed standard : draft présentant un fort consensus
  - Draft standard : pour un protocole dans cet état de standardisation, il existe au moins deux versions interopérantes
  - Standard : LE document final (mais pas obligatoirement figé pour l'éternité)
- Autres types
  - Expérimental : protocole encore en développement
  - Informatif : comme son nom l'indique
  - Historique
  - Best Current Practice (BCP)
  - Les rfc du premier avril... rfc1084, 1149, etc...

Les RFC sont en ASCII 7bits pur... Quelques activistes à l'IETF osent militer pour y introduire un peu de HTML (au moins pour le renvoie vers les sections ou la table des matières) : `http://www.catb.org/~esr/rfcs-in-html/index.html`

Les RFCs sont classés dans leur ordre de parution, il n'y a pas de classement thématique (hors les std).

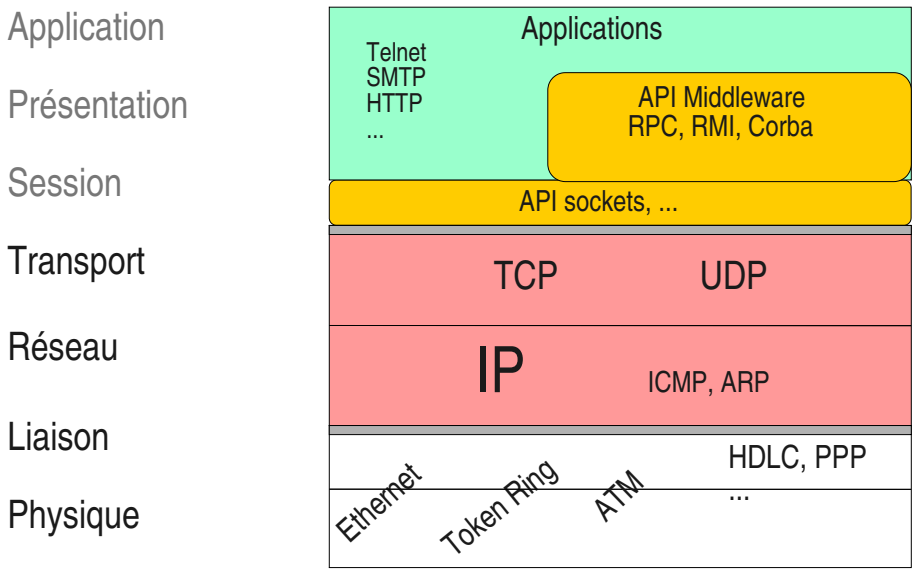
Parmi les RFCs du premier avril citons par exemple celui spécifiant le protocole permettant d'envoyer des messages subliminaux (rfc1097), celui spécifiant comment utiliser IP sur une couche liaison de type «pigeons voyageurs» ou plus exactement des «avian carriers» (rfc1149) le MTU des messages étant proportionnel à la longueur de la patte de l'oiseau...

Mais tous les RFC «premier avril» ne sont pas des plaisanteries... Le 777 est par exemple celui qui spécifie ICMP (Internet Message Control Protocol), et ce protocole est loin d'être humoristique...

Les classifications des RFCs 76/78

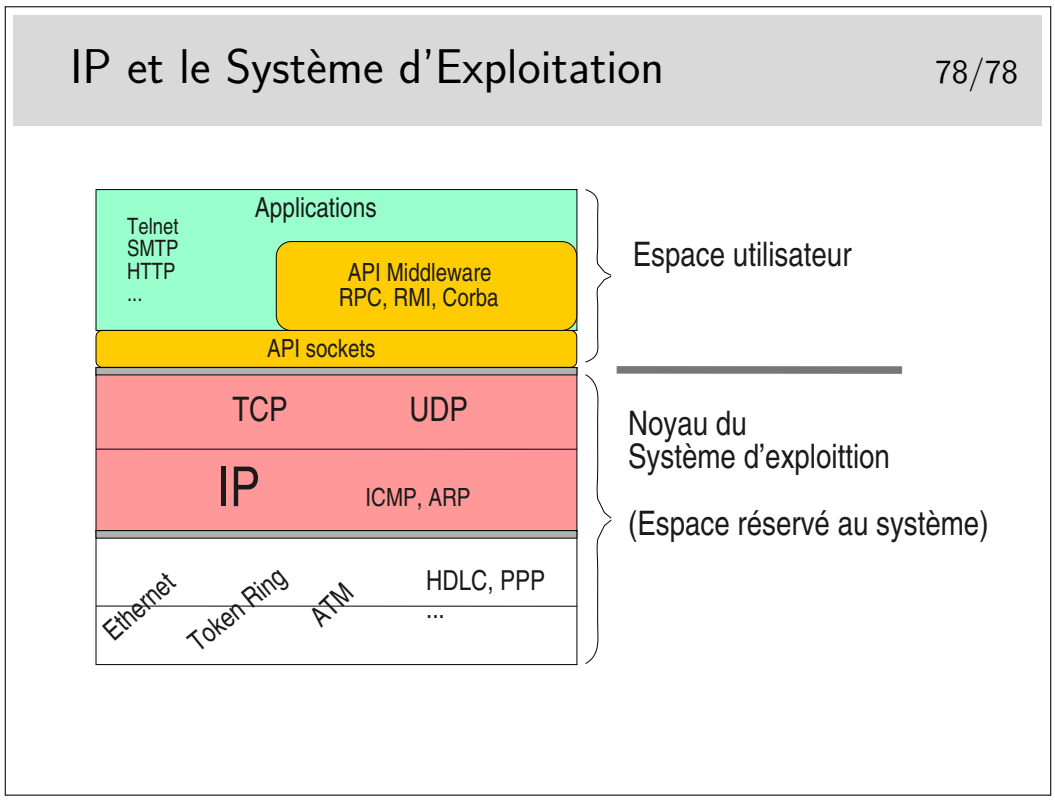
- Certains RFCs standards sont classées sous l'appellation std, le std 5 par exemple correspond au rfc791 définissant IP
- Certains sont classées sous l'appellation FYI : For Your Information
- Quelques RFCs :
  - 791 : IP (std5)
  - 9293 : TCP (std7)
  - 763 : UDP (std6)

IP et le modèle OSI 77/78



Clairement, IP est au niveau 3 et TCP/UDP au niveau 4.

Quand bien même ces protocoles ne sont pas conformes à ceux qui ont été spécifiés pour le modèle OSI, on peut leur trouver ces places dans le modèle.



Les couches protocolaires jusqu'au niveau 4 sont inclues dans le noyau. Les applications (les programmes des utilisateurs) y accèdent via des interfaces qui les masquent.

Ces interfaces sont constituées par des bibliothèques de fonctions formant des APIs utilisables par les programmeurs d'applications. La plus répandue est la bibliothèque socket tout à fait bien adaptée aux protocoles Internet (TCP, UDP, IP). Les couches protocolaires sont vues comme des fichiers qu'on écrit ou qu'on lit. Le transfert de données est aisé puisqu'il se fait comme avec des fichiers. Le portage est aisé vers le monde Windows qui implémente aussi cette interface programmatique issue des Unix de Berkeley (les sockets BSD).

Un autre type d'API est lui aussi très répandu puisqu'il est utilisé pour NFS et NIS. Au départ spécifié par SUN dans les années 80 il est depuis universel sous Unix. Il s'agit des Remote Procedure Calls ou RPC. La couche RPC offre une abstraction du réseau et permet d'appeler des fonctions en local alors qu'elles s'exécutent à distance.

Le concept a été poussé encore plus loin et adapté aux langages orientés objet comme C++ avec CORBA (Common Object Broker Architecture) qui permet d'appeler des méthodes sur des objets situés sur des machines distantes. Cette interface n'est cependant pas standard (du point de vue API) et plusieurs implémentations existent, la plupart commerciales et d'autre libres comme ORBit qui est utilisé par GNOME sous Linux.