

F2B002C - Introduction aux Réseaux

Christophe LOHR

Automne 2018

Sommaire

I	Introduction aux réseaux	5
1	Introduction	5
2	Les concepts fondamentaux	8
2.1	Multiplexage	8
2.2	Notion de connexion	16
2.3	Délais et QoS	21
2.4	Détection et correction d'erreur	23
2.5	Des protocoles	30
3	Modélisation et standardiasation	35
3.1	Standardisation	35
3.2	Principe de la modélisation	35
3.3	Le modèle ISO (OSI)	37
4	Les couches basses	49
4.1	Problèmes de la couche physique	49
4.2	Exemples de couches physique	52
4.3	Exemple de couche liaison de données	56
4.4	Exemple de couche réseau	60
II	Réseaux locaux	65
5	Introduction	65
6	La technologie Ethernet	70
6.1	Fondements d'Ethernet	70
6.2	Les adresses MAC	76
6.3	Les VLANs	86
6.4	Évolutions	89
6.5	Dénomination	91
6.6	Câblage	92

7 Les autres techniques autour des LANs	98
7.1 Le sans fil IEEE-802.11 (WiFi : Wireless Fidelity)	98
7.2 Les courants porteurs	99
7.3 Autres	99
7.4 La couche LLC	100
III IP	105
8 Standardisation, modélisation	105
9 Le datagramme IP	109
9.1 Le datagramme IPv4	109
9.2 Le datagramme IPv6	112
10 Adressage	114
10.1 Les adresses IPv4	114
10.2 Les adresses IPv6	120
11 Protocole ARP / NDP	123
12 Principe du routage	125
13 Les routeurs	134
14 Protocoles de routage	135
15 Gestion des erreurs	142
16 Services pour IP	143
IV TCP - UDP	147
17 Introduction	147
18 TCP	148
19 UDP	164
20 Protocoles applicatifs	167

Cadre du cours

Objectifs

- Appréhender les concepts fondamentaux des Réseaux
 - Multiplexage, paquetisation, délais, protocoles, modélisation, ...

- Montrer comment s'appliquent ces concepts à quelques exemples de réseaux fortement utilisés en entreprise
 - Les réseaux locaux et la technologie Ethernet
 - Concepts et réalité
 - Les réseaux IP
 - Concepts, principes de fonctionnement, routage...
 - Vous devriez savoir, en fin de cours, ce qu'est un routeur, un commutateur et connaître les principes de l'architecture d'un réseau d'entreprise

Plan

- Première partie : Concepts fondamentaux des réseaux
 - Notion de protocole et modélisation
 - Les couches 1, 2 et 3 dans le contexte connecté
- Deuxième partie : les réseaux locaux
 - La standardisation
 - La technologie Ethernet
 - Topologie et supports physique, Adressage, Notion de VLAN
- Troisième partie : Les réseaux IP
 - IP : concepts, adressage, routage
 - TCP,UDP : concepts, notion de port, contrôle de flux
 - Quelques protocoles applicatifs

Bibliographie sommaire

Références

- [1] A. Tanenbaum. *Réseaux*. Eyrolles, 2003
- [2] G. Pujolle. *Les réseaux*. Eyrolles, 2003
- [3] P. Rolin, G. Martineau, L. Toutain, A. Leroy. *Les réseaux*. Hermes, 1997
- [4] P. Toutain. *Réseaux locaux et Internet*. Hermes, 2003
- [5] <http://www.iec.org/online/tutorials>
- [6] <http://www.techfest.com/networking/index.htm>

Et bien d'autres...

(Voir D. Comer, W.R. Stevens,...)

Introduction aux réseaux

1 Introduction

Les Réseaux...8/307

Partout et pour tout...

- ▶ L'épine dorsale de l'entreprise, les synapses de ses neurones...
- ▶ Les entreprises spécialisées
 - ▶ Opérateurs de télécommunication
 - ▶ L'offre aux particuliers et aux entreprises
 - ▶ Téléphone
 - ▶ Internet
 - ▶ Les équipementiers
 - ▶ Les fournisseurs de service
 - ▶ Les applications et les services à valeur ajoutée
- ▶ Enjeux économiques majeurs

synapse : *nom féminin* (grec *sun*, avec, et *aptein*, joindre) Neurol. Région de contact entre deux neurones. (c)Larousse.

- ▶ Le Système d'Information (le SI) : le cœur de l'entreprise
 - ▶ Des bases de données
 - ▶ Le réseau permet l'accès à ces bases
 - ▶ Par l'infrastructure : câbles, matériels, machines...
 - ▶ Par les applications
 - . Ne pas oublier les applications
 - . Les logiciels de groupware (travail en groupe)
 - . Courrier, agenda partagé, workflow (suivi de documents électroniques), ...
- ▶ L'entreprise répartie
 - ▶ Agences dispersées, il faut les relier au siège
 - ▶ Dispersion à l'échelle mondiale parfois

- ▶ Le monde des télécommunications
 - ▶ Histoire plus que centenaire
 - ▶ Une culture : le téléphone
 - ▶ Des principes forts, une normalisation très structurée
- ▶ Le monde de l'informatique communicante
 - ▶ Histoire récente
 - ▶ Au départ pour l'entreprise, en interne, pas de facturation
 - ▶ Des acteurs divers
 - ▶ Fournisseurs de matériels et de logiciels, Utilisateurs, Chercheurs
 - ▶ Une réactivité et innovation fortes
 - ▶ Standardisation rapide et légère
- ▶ Les deux mondes convergent

Un choc des cultures...

On sait facturer dans un monde, dans l'autre on ne se pose même pas le problème... Mais vient l'heure des comptes... C'est un exemple de différence des cultures, il y en a d'autres, portant sur les principes mêmes du fonctionnement des réseaux : le mode connecté du monde des télécommunications et le mode datagramme du monde de l'informatique (spécialement dans les réseaux locaux et Internet). Nous aborderons ces concepts dans le cours.

Les deux mondes ont eu du mal à coopérer.

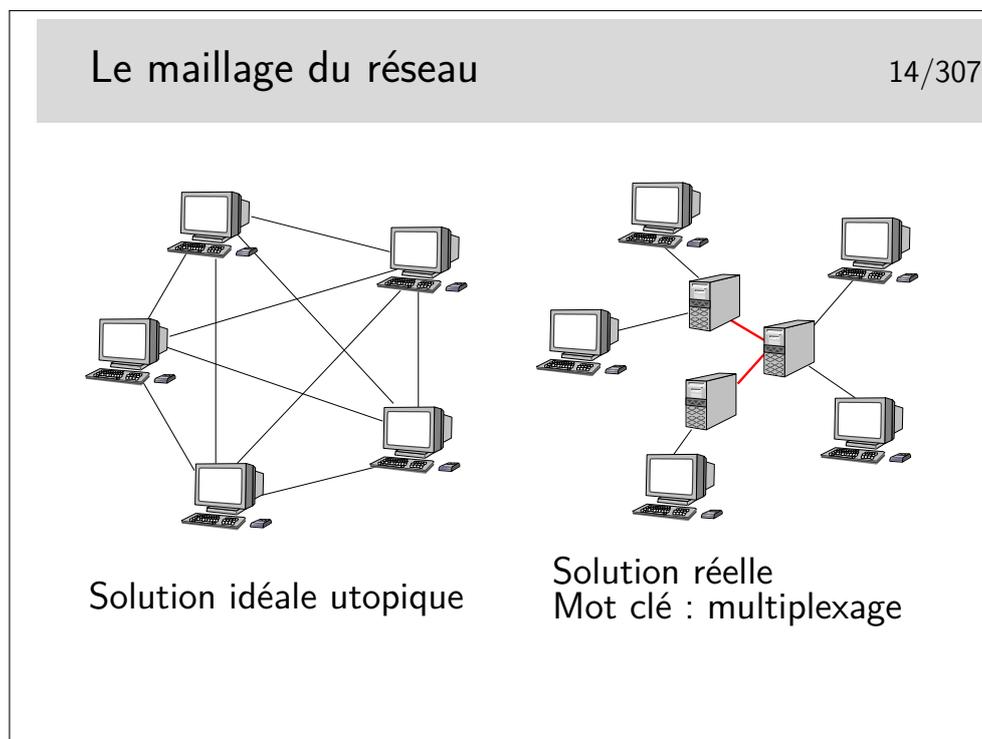
L'objectivité qui caractérise les scientifiques cède parfois la place à une «légère» subjectivité...

Réseaux : l'explosion du nombre de services 11/307

- ▶ Les services logiciels : ce sont les applications
 - ▶ Logiciels pour ordinateurs de bureau (portables ou non)
 - ▶ Logiciels embarqués
 - ▶ Sur les portables de type téléphone
 - ▶ Sur les portables de type PDA
 - ▶ La combinaison des appareils et des services
 - . Le téléphone portable qui devient appareil photo et PDA,
 - . média-center / média-renderer, domotique, M2M, etc.
- ▶ Les services matériels
 - ▶ La technologie va très vite
(à peine avons nous le temps de nous habituer au modem V92 que l'ADSL est arrivé, et maintenant le FTTH pointe son nez)
 - ▶ Les technologies sans fil s'imposent rapidement
(IEEE-802.11{abg}, Bluetooth, Zigbee)

2 Les concepts fondamentaux

2.1 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)

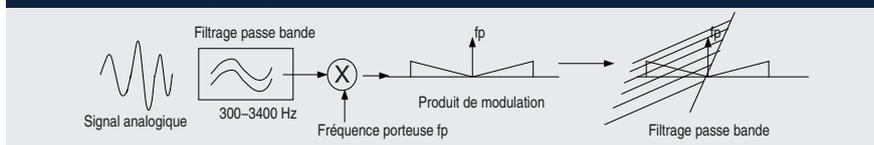
Différents types de multiplexes

- ▶ 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

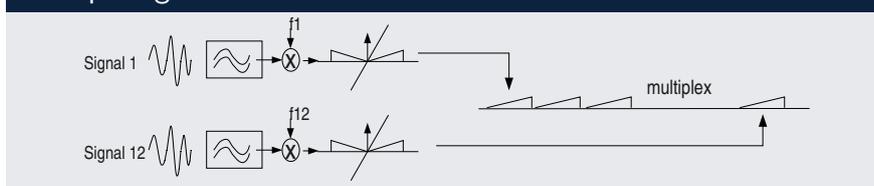
Multiplexage en fréquence

Exemple du multiplexage de canaux téléphoniques : Modulation en amplitude

Modulation d'un canal



Multiplexage

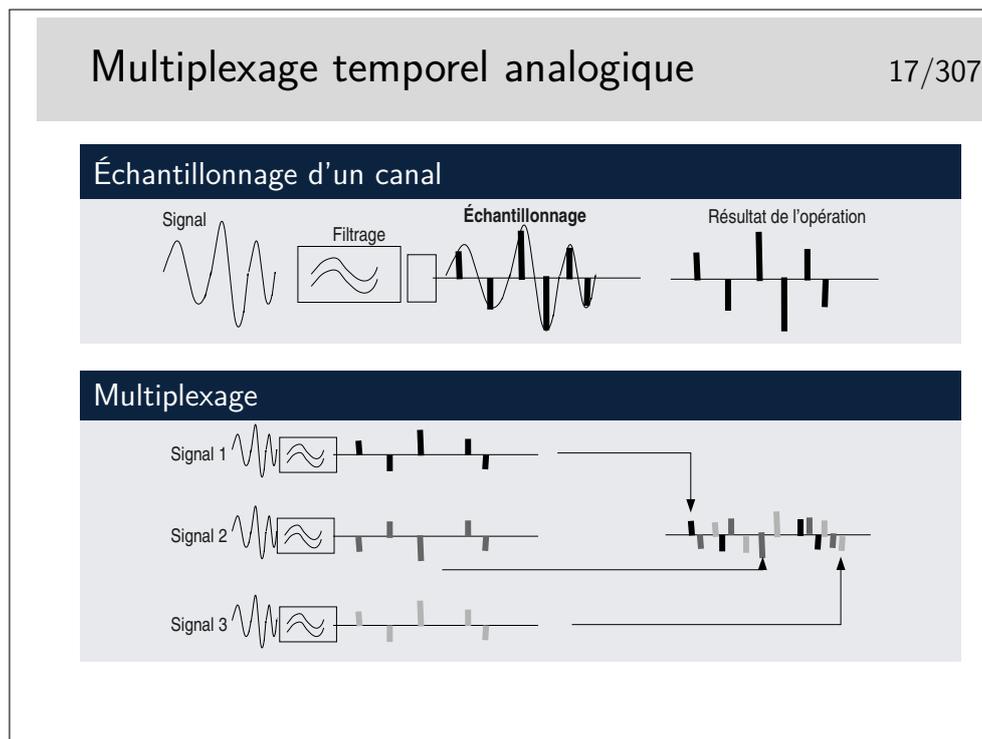


Cas du mutilpexage de canaux téléphoniques :

- 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 f_p-3400 à f_p-300 puis de f_p+300 à f_p+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 f_p à f_p+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.

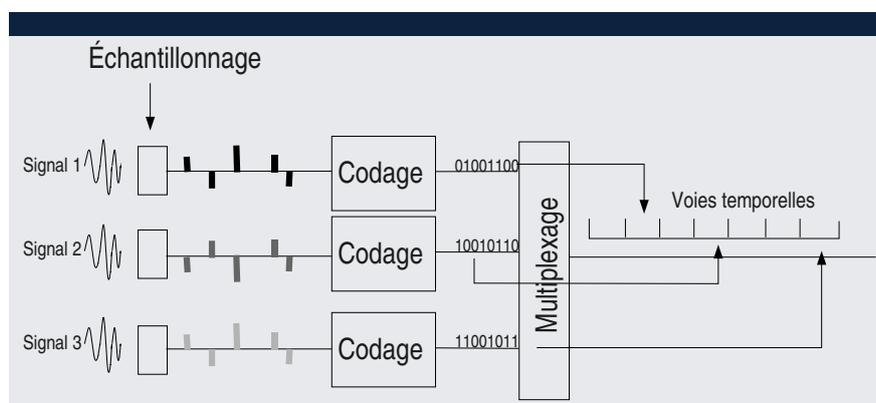


Chaque signal est échantillonné un certain nombre de fois par secondes, chacun avec un décalage dans le temps par rapport aux autres de telle manière qu'ils peuvent être véhiculés sur le même support sans se mélanger après multiplexage.

La fréquence d'échantillonnage est telle que le signal doit pouvoir être reconstitué sans déformations majeures. Shannon a montré que la fréquence d'échantillonnage devait 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 multiplex 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 capacitatives et selfiques du support, ils sont bruités et déformés et risquent de se mélanger.

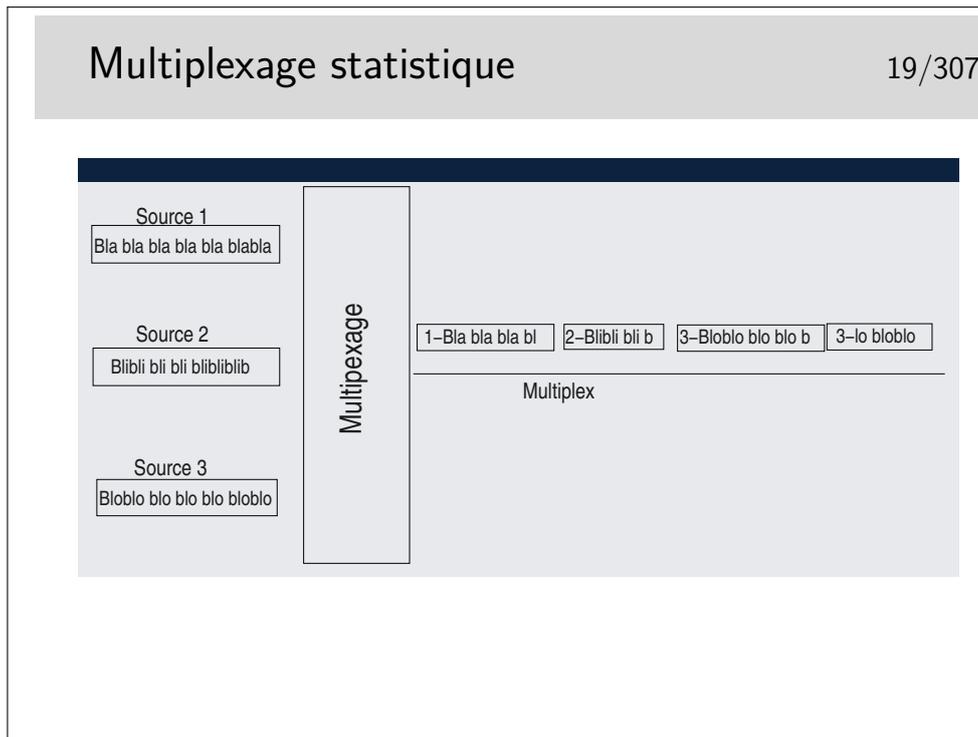
La solution réside dans la numérisation.



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 $125\mu\text{s}$. Elles divergent sur les points suivants :

- Loi européenne dite loi A (A_{law}) :
 - méthode de codage spécifique
 - $3,9\mu\text{s}$ 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 (μ Law)
 - 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) (terme à ne pas confondre avec les cellules des réseaux de téléphonie de type GSM où les cellules sont des zones géographiques desservies par une antenne émettrice/réceptrice).

Analogique

Une variation d'un signal physique, la pression d'air générée par un son, l'intensité de lumière et de couleur générée par la lumière éclairant une scène, un paysage, est transformée en un signal électrique représentatif du signal physique. Le signal électrique varie avec les variations du signal physique. Ses variations sont «analogues» à celles du signal physique, c'est un signal analogique.

Numérique

Un signal électrique analogique est échantillonné (on considère de courts instants du signal analogique, des «échantillons»). Les échantillons sont alors mesurés sur une certaine échelle et à chaque pas de cette échelle correspond un nombre. L'échantillon est remplacé par le nombre, nombre lui-même codé en base 2 c'est à dire une suite de 0 et 1. L'échantillon est numérisé, il devient un élément d'un signal numérique.

Pour qu'un signal échantillonné puisse être reconstitué dans de bonnes conditions de restitution il faut qu'il ait été échantillonné à une fréquence deux fois plus élevée que sa fréquence maximale. Ainsi pour un signal de fréquence f , il faut qu'il soit échantillonné à une fréquence $2 \times f$. Intuitivement on comprend qu'on doit au moins prendre un échantillon de chaque alternance (Shannon a montré cela de manière beaucoup plus rigoureuse que la simple intuition).

Quelques remarques sur l'échantillonnage 22/307

- ▶ Nyquist (1889-1976) a montré que pour restituer correctement un signal après échantillonnage il faut que la fréquence de celui-ci soit au moins le double de la fréquence maximale à échantillonner
- ▶ Exemple en téléphonie :
 - ▶ Bande analogique : 300Hz-3400Hz
 - ▶ Fréquence d'échantillonnage minimale : 6800Hz
 - ▶ Par précaution on prend 8000Hz

Questions :

- Pour un échantillonnage de type téléphonique, quelle sera la période de l'échantillonnage ?
- Quelle sera la durée d'un échantillon si on désire multiplexer 32 signaux sur un même support physique ?
- On code numériquement chaque échantillon sur un mot de 8 bits, quel débit numérique en résulte par voie ?
- Le codage du son en mode wav (origine Microsoft, fichiers avec l'extension .wav) correspond typiquement à un échantillonnage à la fréquence de 44100Hz, chaque échantillon étant codé sur 16 bits. Quelle est la bande passante nécessaire en b/s pour transmettre un fichier .wav contenant un son stéréophonique (2 canaux) ?

Sur Nyquist : <http://www.geocities.com/bioelectrochemistry/nyquist.htm>

Avantages et inconvénients des différents types de multiplexage

I 23/307

- ▶ Multiplexage temporel
 - ▶ 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
- ▶ Multiplexage en fréquence
 - ▶ mêmes avantages et inconvénients

Avantages et inconvénients des différents types de multiplexage

II 24/307

- ▶ 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.
 - ▶ La voix et la vidéo sont mal adaptées à cette technique car les délais et la bande passante ne sont pas garantis

- ▶ 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

2.2 Notion de connexion

- ▶ 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 opérations 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

28/307

▶ 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

29/307

▶ 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

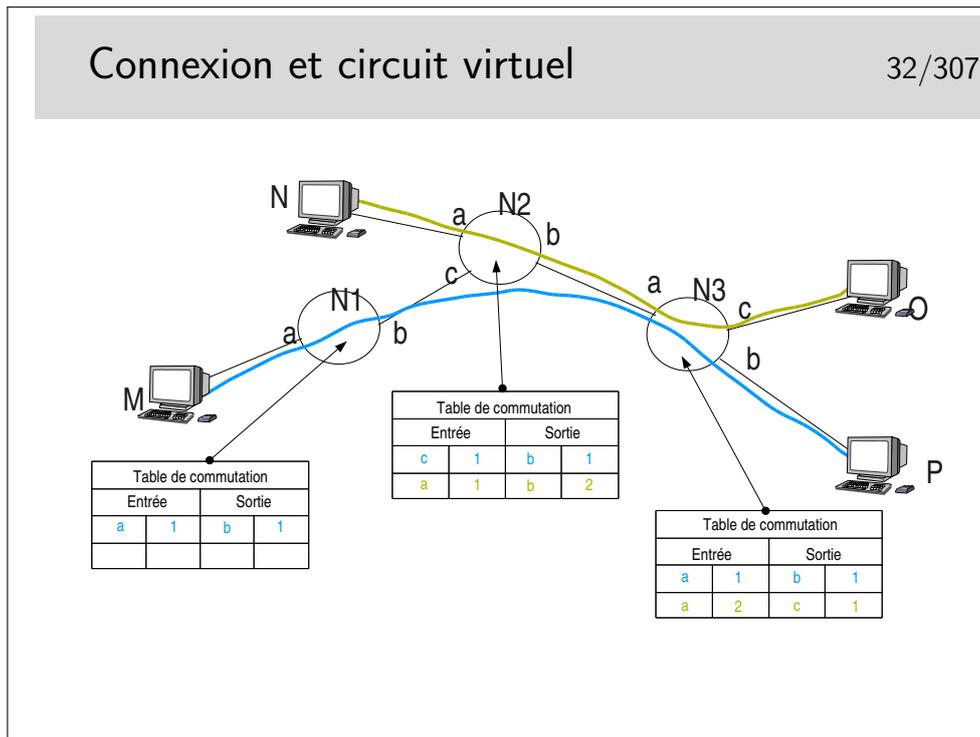
▶ Inconvénients

- ▶ 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

- ▶ **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...)

Caractéristiques des réseaux orientés connexion

- ▶ Les nœuds acheminent les unités de données (les paquets) entre les entrées et les sorties en effectuant des opérations 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'étiquettes réservées pour une communication sur chaque lien constitue un **circuit virtuel**

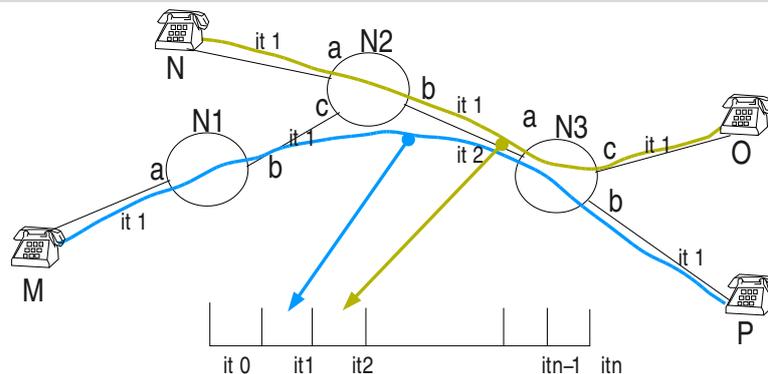


Les paquets issus de M et à destination de P sont munis d'une étiquette 1 en M. Elle reste 1 après passage dans tous les noeuds. Sans doute la communication M-P a-t'elle été établie la première.

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.

- ▶ 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 $125\mu s$)
- ▶ 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



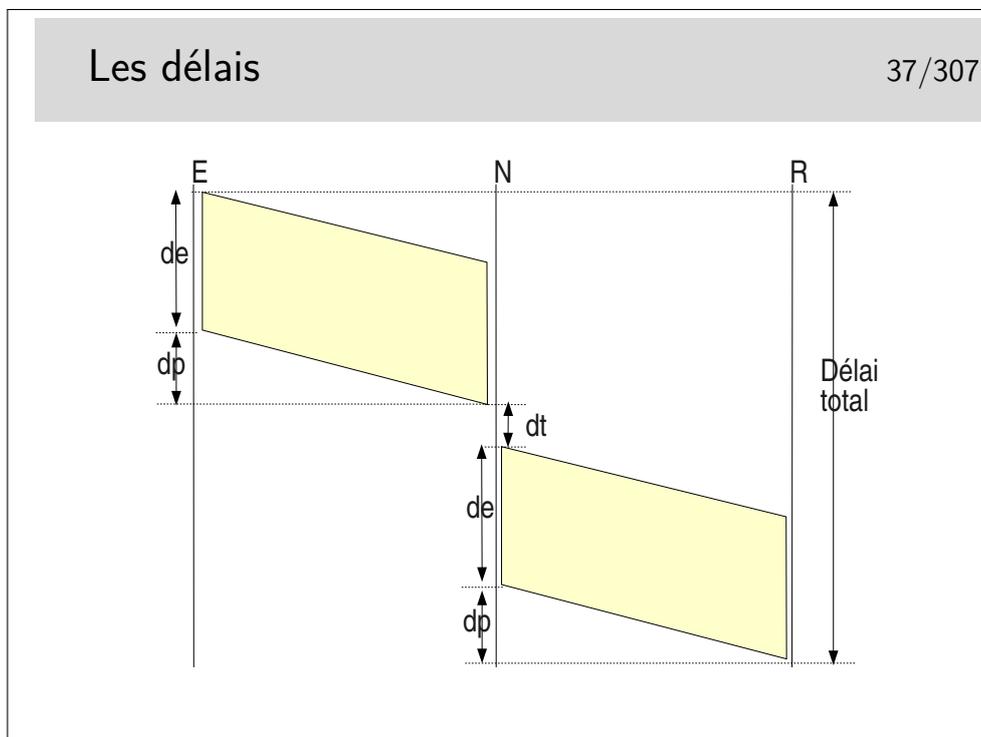
- ▶ Entre $N2$ et $N3$, les intervalles de temps 1 et 2 sont attribués aux communications $M - P$ et $N - O$, respectivement, et ce, pour la durée des communications
- ▶ Si la communication $M - P$ se termine, la ressource $it1$ est libérée, la communication $N - O$ n'en profite pas
- ▶ Si la communication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

2.3 Délais et QoS

Les délais36/307

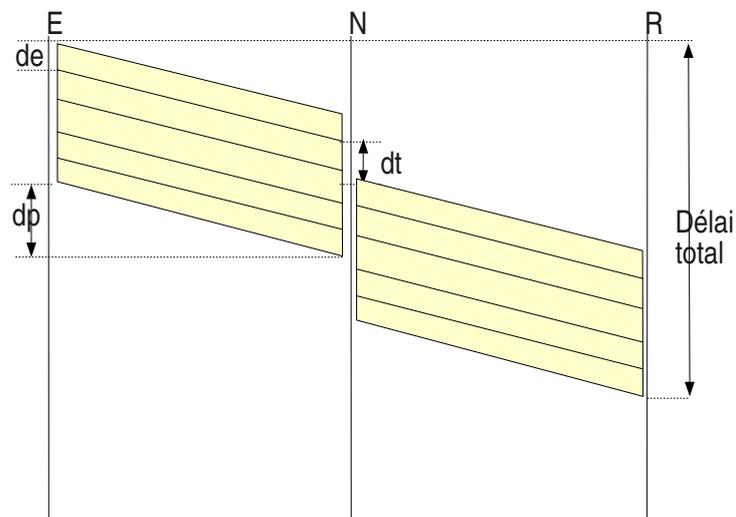
- ▶ Entre un émetteur et un récepteur, les données vont subir différents délais
 - ▶ Durée d'émission : inversement proportionnelle au débit, proportionnelle à la longueur des segments de donnée
 - ▶ Le délai de propagation : au minimum le délai de la lumière dans le vide ($c \approx 300000Km/s$), dépend par ailleurs du support ($2/3c$ dans le verre et le cuivre).
Exemple : $250\mu s$ 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) :
 $\lambda f = c$



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 38/307



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 39/307

- ▶ Pour le téléphone
 - ▶ Délai de 50ms : bon confort de communication
 - ▶ Au delà de 200ms (communications par satellites geostationnaires) : qualité très moyenne, 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)

2.4 Détection et correction d'erreur

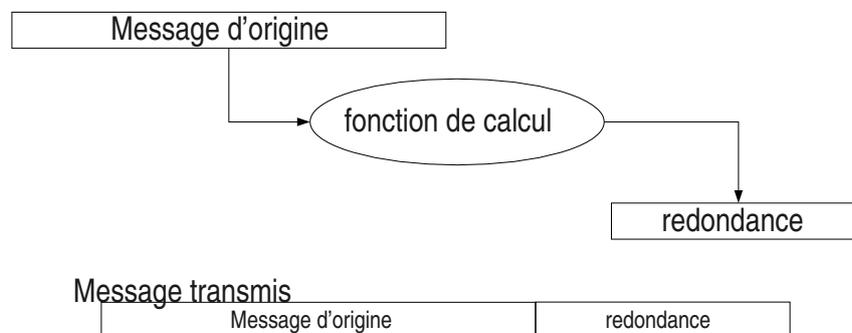
Détection et correction d'erreur

41/307

- ▶ 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

42/307



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

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'al-

gorithme, 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

43/307

- ▶ 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

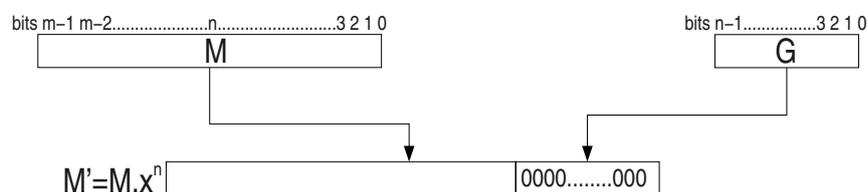
1	0	0	1	0	0	1	1
0	0	0	0	1	1	0	0
 - ▶ parité impaire

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

44/307

- ▶ 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 45/307

- ▶ 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)
- | | | |
|--------------------|--|-------------|
| $M' = M \cdot x^n$ | | 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 $R = 0$

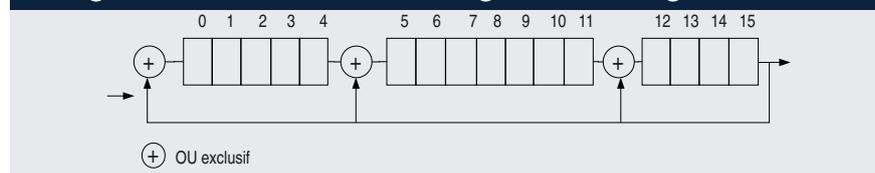
Le reste de la division est appelé CRC du nom de la méthode.

Quelques CRC types et le calcul par registres à décalage

46/307

- ▶ 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 télécommunications des réseaux et de l'informatique, un certain nombre de polynôme de CRC ont été standardisés, offrant soit une *bonne capacité* à détecter les erreurs, soit étant faciles à implémenter (tout est affaire de compromis). Voir

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

47/307

(ou *Checksum*)

- ▶ Soit un message formé par la suite de mots $A, B, C, D, \dots, R, S, T$
 - ▶ 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 : $Z = A + B + C + D + \dots + R + 0 + T$
 - ▶ Le résultat Z est inversé, on obtient \bar{Z}
 - ▶ On transmet $ABCD \dots R\bar{Z}T$
 - ▶ À la réception on fait la somme
 $Z' = A + B + C + D + \dots + R + \bar{Z} + T$
 - ▶ S'il n'y a pas d'erreur alors $Z' = Z + \bar{Z} = 1111 \dots 1111$
 - ▶ 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

48/307

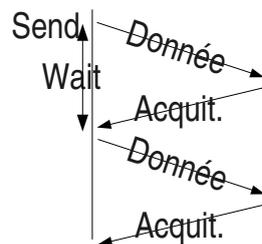
- ▶ Si la redondance est suffisante et l'algorithme suffisamment puissant il est possible de détecter les erreurs et de les corriger.
 - ▶ On peut le montrer sur un exemple simple avec le mécanisme du bit de parité.
Soit le bloc suivant, chaque ligne et chaque colonne est munie d'un bit de parité paire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

Correction d'erreur par retransmission I 49/307

- ▶ 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



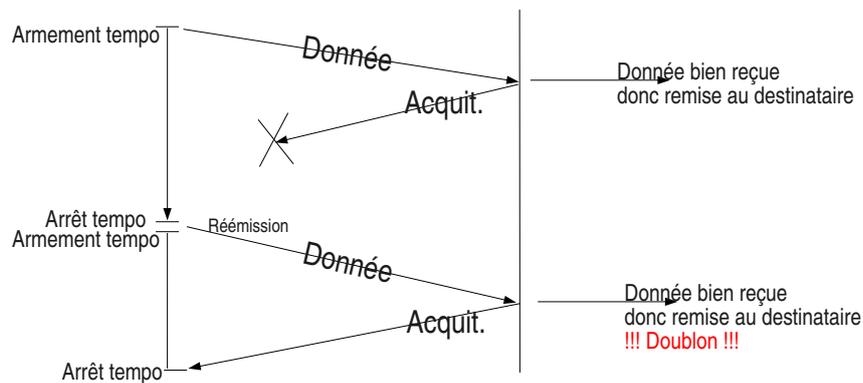
Ce mécanisme est appelé «Send & wait»

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

Si l'acquittement se perd ?

Correction d'erreur par retransmission II 50/307

- ▶ Utilisation de temporisateur (timer)

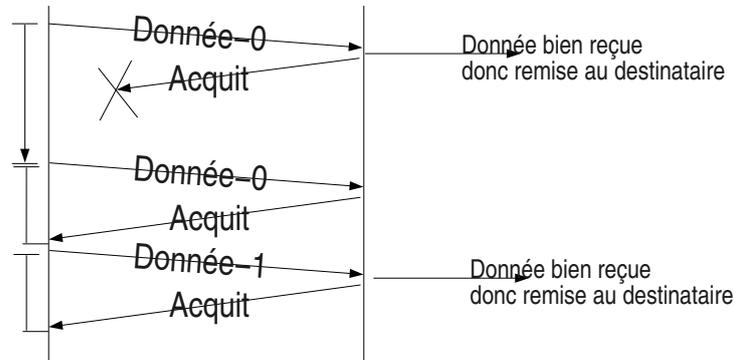


Où comment se créer d'autres problèmes en voulant en régler un... Et maintenant comment on fait pour éviter les doublons ?

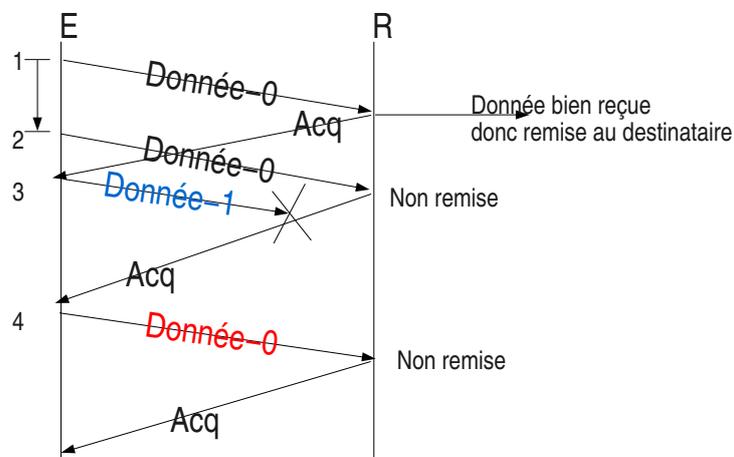
C'est une règle, hélas, très générale, en Réseaux, que de se créer de nouveaux problèmes en apportant des solutions à d'autres...

Retransmission : le problème des doublons 51/307

- ▶ 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... 52/307



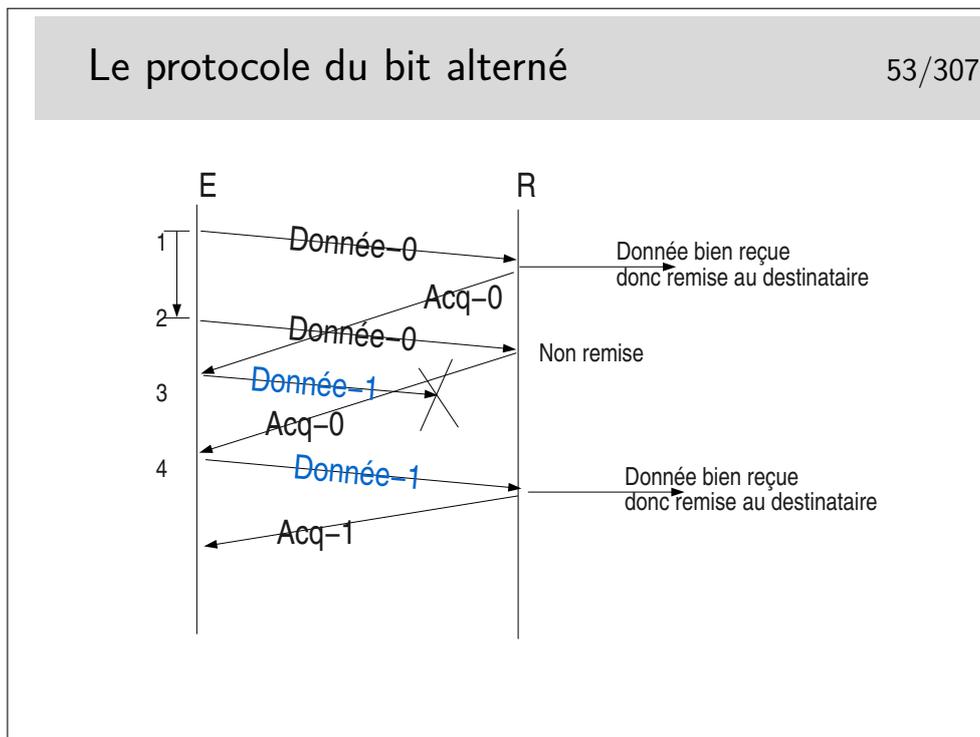
1. émission de **Donnée-0**, bien reçue, elle est remise à l'entité destinatrice
2. le timer expire trop tôt, **Donnée-0** est réémise et est bien reçue, elle n'est pas remise car elle porte le même numéro que la donnée précédente. Cependant, le récepteur renvoie un acquittement car il pense que s'il 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'acquiescement de **Donnée-1**.

À chaque fois qu'il reçoit un acquiescement 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 livrées 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 acquiescent. 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).

2.5 Des protocoles

Notion de protocole I

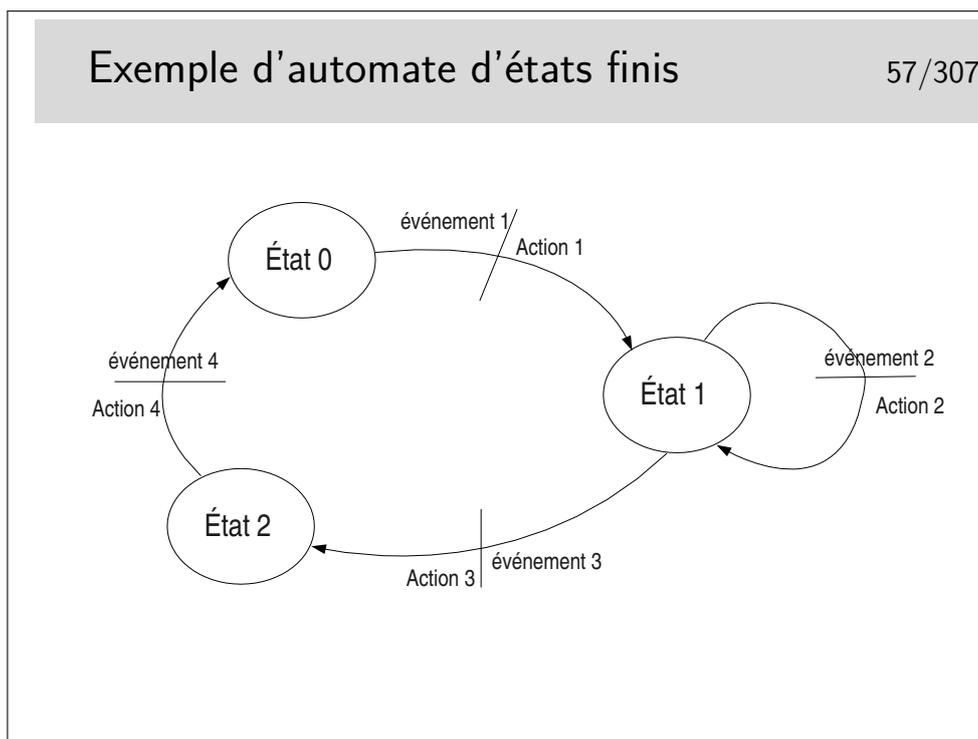
55/307

- ▶ 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 rajoute un autre
 - ▶ Il faut que la série correction/nouveau problème converge

Notion de protocole II

56/307

- ▶ 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 communication
- ▶ Pour mettre en œuvre et gérer les mécanismes 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

58/307

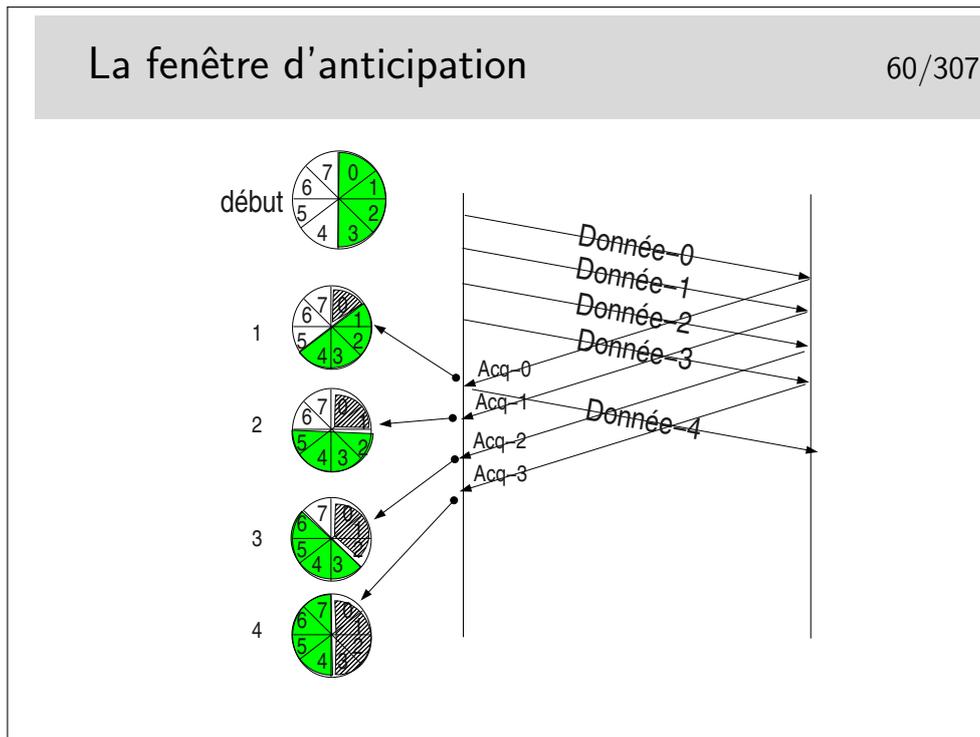
Événement État	Demande d'envoi de donnée	Arrivée Ack 0	Arrivée Ack 1	Expiration Timer
État 0 nouvel état	Envoyer Data-0 Armer Timer Attente Ack 0			
Attente Ack 0 nouvel état		Arrêt timer État 1		Envoyer Data-0 Armer Timer Même état
État 1 nouvel état	Envoyer Data-1 Armer Timer Attente Ack 1			
Attente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

Les protocoles à anticipation

59/307

- ▶ 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.



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.

- ▶ 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 destinataires 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».

3 Modélisation et standardisation

3.1 Standardisation

Les organismes de standardisation

64/307

- ▶ 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

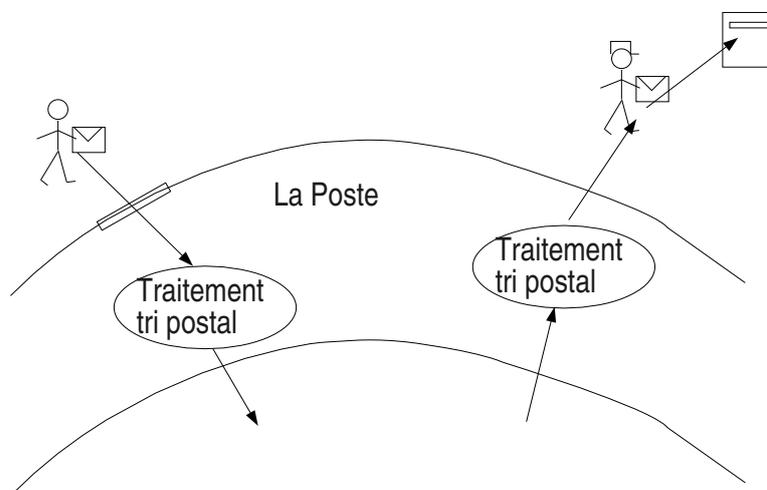
3.2 Principe de la modélisation

La modélisation de la problématique réseau

66/307

- ▶ 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?»*

- ▶ En tant qu'utilisateur :
 - ▶ Je dois savoir où est situé le bureau de poste, l'adresse de mon correspondant et avoir une boîte 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.



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 boîte 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 boîte aux lettres du destinataire ?

Réponse : oui et non!!!

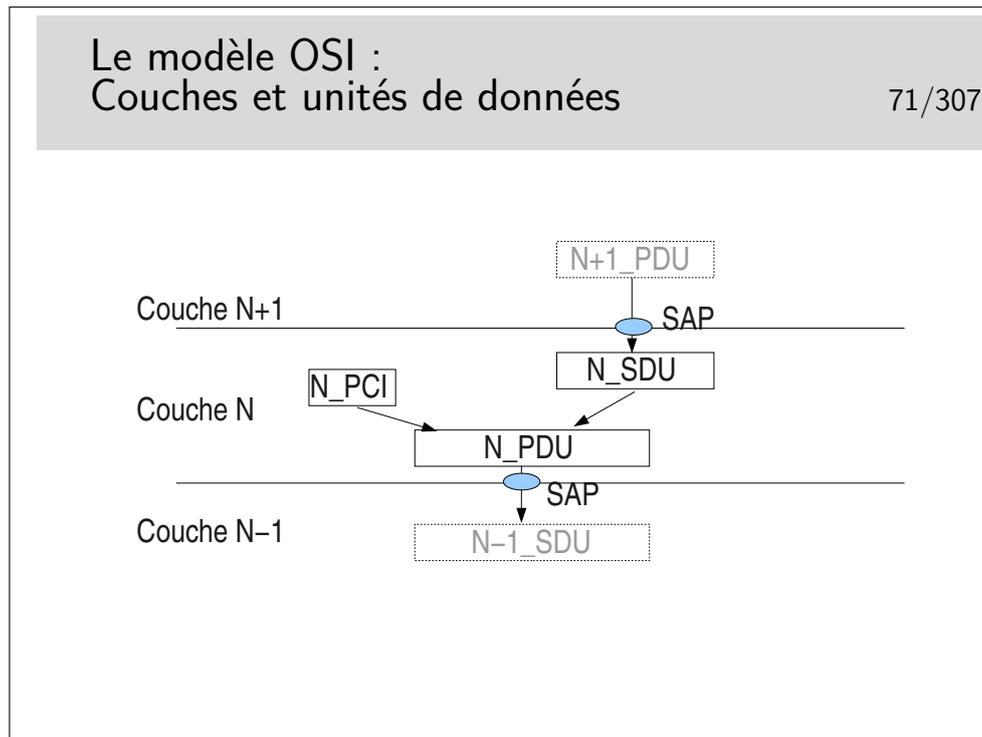
- Oui pour le service postal.
- Non pour le destinataire final. Il peut s'agir de la boîte aux lettres de la famille Dupont. La lettre peut être pour Philippe Dupont et non pour Jacques Dupont. Le SAP «adresse des Dupont» n'est pas le seul SAP à considérer. Le SAP «Jacques» et le SAP «Phillipe» sont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

3.3 Le modèle ISO (OSI)

Le modèle ISO (OSI)

70/307

- ▶ 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

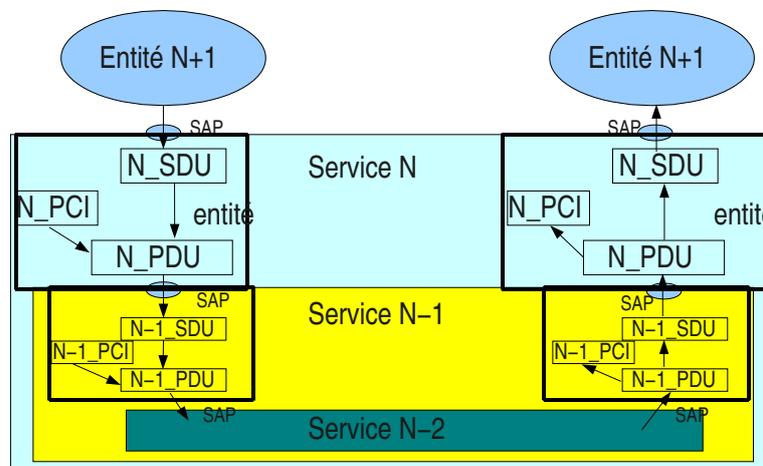


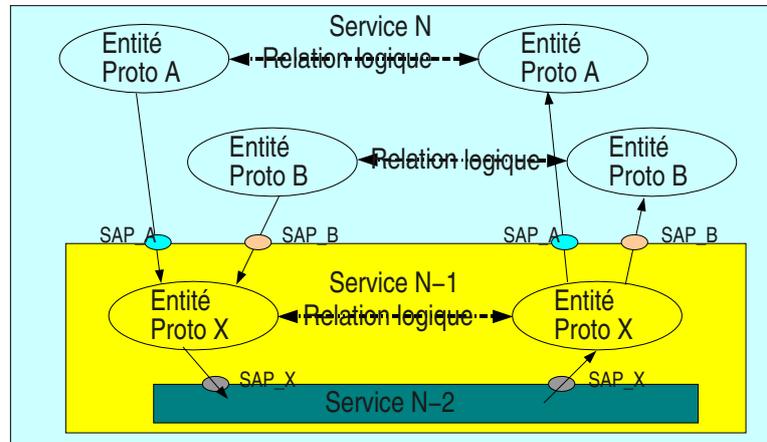
L'unité de donnée fournie par l'utilisateur est une unité de donnée à SERVIR. C'est une UNITE de DONNEE de SERVICE : une *SERVICE DATA UNIT* (SDU) en anglais.

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).

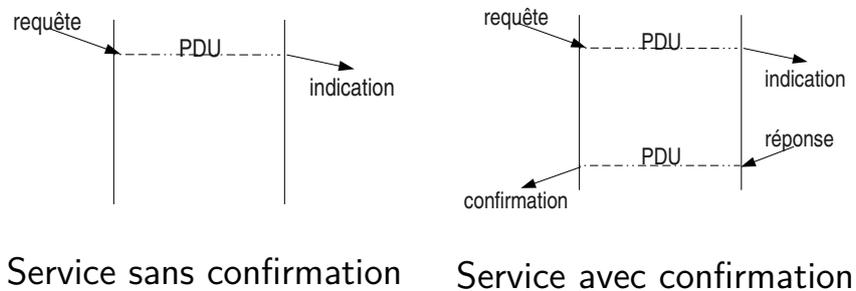
- ▶ Une entité protocolaire est un «programme informatique», une application ou un module opérationnel du système d'exploitation mettant en œuvre un protocole
- ▶ Une même couche peut être composée de plusieurs entités mettant en œuvre des protocoles différents pour assurer un même service
 - ▶ Exemples :
 - ▶ 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)





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 .



Le concept de primitive de service peut s'apparenter au concept de fonction de programme informatique ou mieux encore d'objet au sens «langage orienté objet» tel C++ ou Java.

Mais nous sommes face à un problème car si les concepts sont voisins, ils n'en ont pas moins des représentations différentes. Les primitives définies pour une couche protocolaire sont indépendantes de tout langage informatique. Fort heureusement d'un côté. Sous un autre angle de vue cela pose problème. En effet, il faut pouvoir, à la fin, implémenter les protocoles dans un langage et sur un système d'exploitation. Comment traduire en C/C++ sous Unix/Linux, Windows ou MacOS le paradigme «Requête/indication» présenté ci-dessus? Il faut noter en outre que la requête peut être codée en C et l'indication en Java.

Dans la pratique, il y a une réelle différence entre les outils à notre disposition pour développer des applications et les définitions formelles des primitives. C'est vrai pour les applications standards, c'est encore plus différent pour les entités protocolaires développées dans les noyaux des systèmes d'exploitation.

Lorsqu'on étudie les documents de spécification d'un protocole on est toutefois confronté au concept de primitive. Il faut alors absolument savoir ce que cela représente, en faisant abstraction, en première lecture, de toute idée d'implémentation. Par la suite peut se poser le problème de l'implémentation.

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

76/307

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*)

77/307

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*)

78/307

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*)

79/307

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*)

80/307

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*)

81/307

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*)

82/307

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.

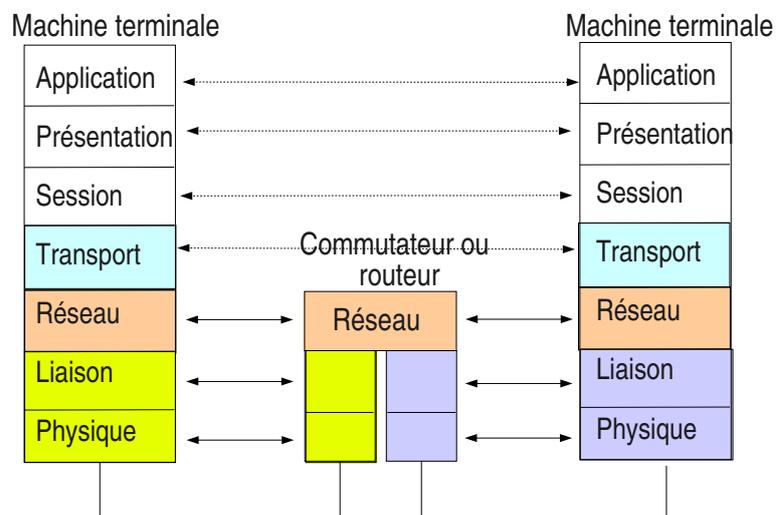
Les 7 couches du modèle OSI : pour résumer...

83/307

- ▶ 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 ?

84/307



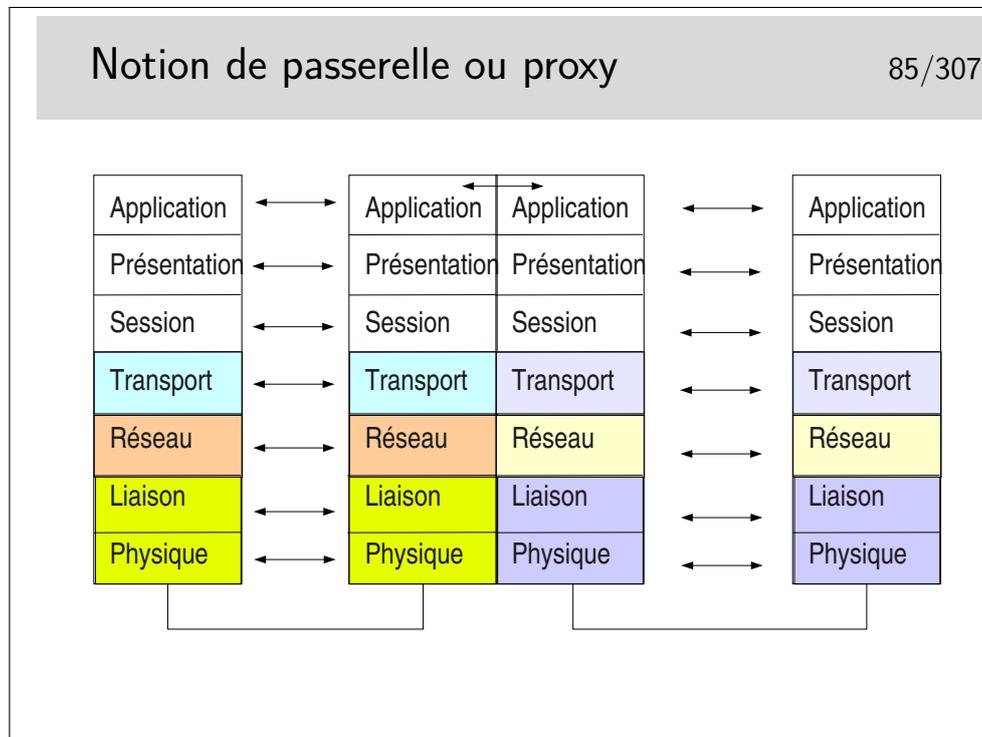
Toutes les couches sont implémentées dans les machines terminales.

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 employant la technologie Sonet/SDH.

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

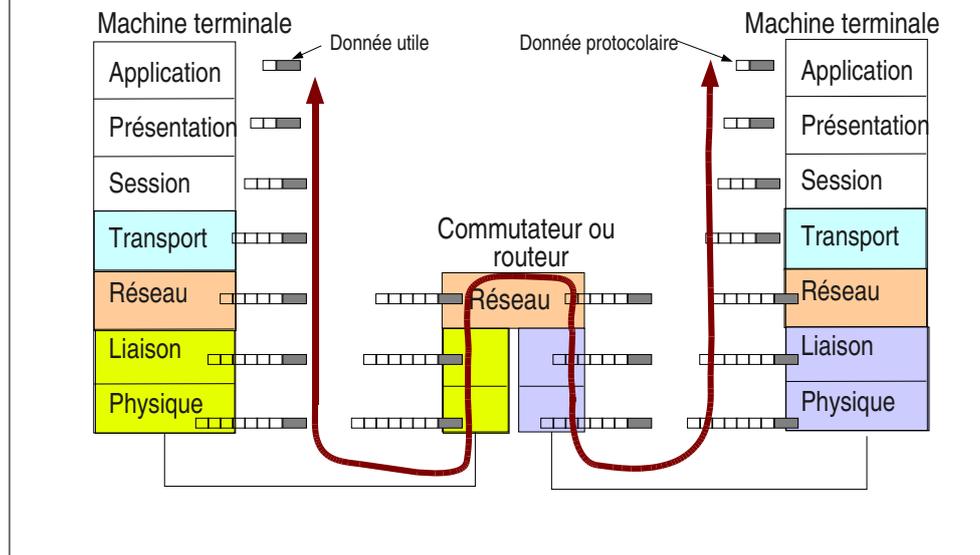


Toutes les couches sont implémentées dans la machine intermédiaires. La couche application réalise une traduction des données pour les adapter à l'application destinatrice. Les protocoles mis en oeuvre à gauche peuvent être tous différents de ceux de droite.

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

86/307



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 !

4 Les couches basses

4.1 Problèmes de la couche physique

Problématiques

89/307

- ▶ Fonction première : transport des bits sur un support analogique
 - ▶ Modulation de signal sinusoïdal (fréquence porteuse) en amplitude et en phase par modems
 - ▶ Codage en tension : le plus simple mais limité en distance et en débit. Fragile face aux perturbations électromagnétiques
 - ▶ Codage en courant : 2 fils par signal, plus résistant aux perturbations électromagnétiques. Permet de monter très haut en débit (100Mb/s et plus)
 - ▶ Codage photonique : sur fibre optique, des impulsions de lumière
- ▶ Problème connexe : synchronisation parfaite des horloges des récepteurs sur les horloges des émetteurs

Synchronisation des récepteurs

I

90/307

- ▶ L'idéal :
 - ▶ l'horloge du récepteur doit fonctionner à la même fréquence que celle de l'émetteur
 - ▶ l'horloge du récepteur doit fonctionner avec un décalage de phase constant avec celle de l'émetteur (décalage de π pour que le front montant coté récepteur ait lieu en milieu de bit reçu)

- ▶ La réalité physique
 - ▶ les conditions énoncées ci-dessus sont impossibles à réaliser
 - ▶ le récepteur fonctionne avec une horloge qui lui est propre. Il est impossible physiquement de fournir une horloge unique pour tous les émetteurs et récepteurs du réseau. La fréquence de l'horloge récepteur sera naturellement légèrement différente de celle de l'émetteur et en plus la différence variera en fonction du temps et des conditions physiques (température du milieu, perturbations électromagnétiques du milieu).
 - ▶ même si les horloges étaient parfaitement à la même fréquence, le problème de la phase demeurerait

- ▶ Le principe fondamental pour remédier au problème :
 - ▶ La suite des bits de données est transcodée pour permettre de véhiculer simultanément un signal en phase avec l'horloge d'émission.
 - ▶ L'électronique de réception est capable de déduire l'horloge émission du flux binaire reçu. Elle génère alors un signal qui vient synchroniser parfaitement l'horloge du récepteur.

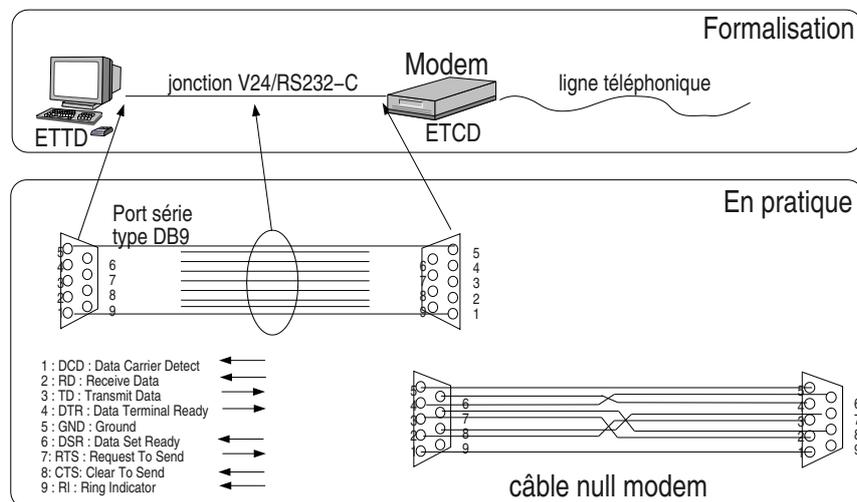
- ▶ Forts divers. Ils doivent permettre de respecter les règles suivantes :
 - ▶ pas de longues suites de 1 ou de 0, sinon on perd la synchronisation,
 - ▶ pas de composante continue dans le flux émis (la décomposition spectrale du signal fait apparaître un courant ou une tension à la fréquence 0 : courant continu). Ce point est finalement identique au précédent, des suites de 0 ou des suites de 1 sont analogues à des séquences de signal continu.

- ▶ HDB3 : (haute densité binaire d'ordre 3), les UNs sont codés alternativement +V et -V. Les ZEROs sont codés 0. Si une suite de 3 ZEROs arrive, un UN est inséré à leur suite. C'est un faux UN, pour le reconnaître on lui donne la même polarité que le UN normal qui a précédé (viol de bi-polarité). Multiplex téléphoniques à 32 voies (2,048 Mb/s).
- ▶ AMI : Alternate Mark Inversion, même principe sauf que le bit de poids fort est toujours à 1 et qu'il n'y a pas de longue suite de 0 à priori car on porte de la voix en PCM
- ▶ bipolaire (manchester) : chaque bit est codé par une transition +V-V ou -V+V suivant la valeur du bit. Il ne faut pas croiser les fils car les bits seraient décodés inversés (les UN seraient vus comme des ZEROs et réciproquement). Réseau Ethernet.

- ▶ bipolaire différentiel (manchester différentiel) : même principe que ci-dessus mais le codage d'un bit (une transition) dépend de la valeur du bit précédent. Le problème précédent n'existe alors plus. Réseau Token Ring.
- ▶ 4B/5B : on transmet 5 bits pour 4 bits utiles. À toute configuration de 4 bits de données on fait correspondre un mot de 5 bits (16 combinaisons, 32 mots possibles). Certains mots de 5 bits sont interdits de par leur configuration. Certains mots sont réservés pour un usage particulier de signalisation sur la liaison
- ▶ 8B/10B : principe analogue au précédent

4.2 Exemples de couches physique

Jonction ITU-T V24 – ANSI RS232-C Exemple de couche physique



ETTD : Equipement Terminal de Circuit de Données (en anglais DTE : Data terminal Equipment). Désigne le coté terminal de la jonction, en pratique, très souvent, l'ordinateur.

ETCD : Equipement Terminal de Circuit de Données (en anglais DCE : Data Circuit

Equipment). Désigne le coté Réseau de la jonction. Souvent en pratique il s'agit du modem. Mais cela peut être aussi une interface série d'une machine.

Ce sont de vieux termes, encore utilisés dans le cas des liaisons de type série, à bas débit.

En pratique votre ordinateur est relié à un modem standard (non ADSL), via le port COM1 ou COM2 (appelés encore «ports série»), par une jonction conforme à la recommandation V24/RS232-C.

Le câble *NULL Modem* permet de raccorder deux ordinateurs entre eux via leur port série. On dit encore de ce câble qu'il est «croisé» car, en particulier, la broche 2 de l'un est reliée à la broche 3 de l'autre et réciproquement.

Voir : <http://www.loop-back.com/null-mod.html>

Dans ce type de jonction on trouve les supports pour la transmission de l'information utile mais aussi des supports pour de l'information de service («présence porteuse», «terminal prêt», etc.). Ce type d'information est aussi appelé «signalisation».

Le codage des bits est réalisé par des niveaux de tension. Un seul fil est nécessaire par sens de transmission (plus une «terre» commune, c'est d'ailleurs plutôt un «retour commun» qu'une terre).

Jonction ITU-T V24 – ANSI RS232-C
Codage de l'information 98/307

- ▶ Selon la recommandation ITU-T V28
 - ▶ Niveaux de tension
 - ▶ $V > +3v \Rightarrow 0$
 - ▶ $V < -3v \Rightarrow 1$

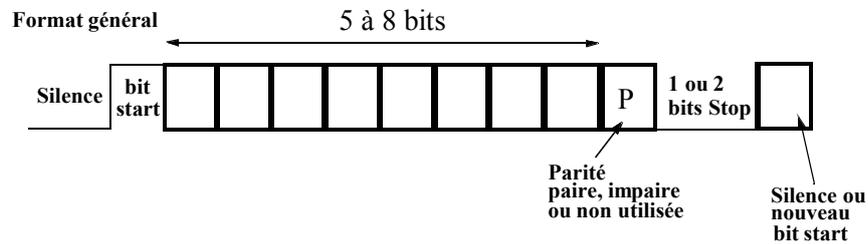
Ceci est un exemple, il en existe d'autres. Le codage par niveau de tension est simple à réaliser mais il est peu performant en termes de distance (quelques dizaines de mètres au plus) et de débit (environ 100Kb/s max).

Pour les plus grandes distances et les plus forts débits on privilégie les niveaux de courants. Pour cela il faut deux fils par jonction (2 fils par signal). Ce mode de codage est plus résistant aux perturbations électromagnétiques.

Jonction ITU-T V24 – ANSI RS232-C Structuration de l'information

99/307

Structure orientée octet



Exemple : transmission de 2 caractères en mode parité paire avec 2 bis Stop

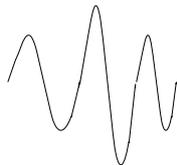


Les schémas ci-dessus présentent des «1» sous formes d'impulsions positives (créneaux vers le haut), contrairement à la réalité V24/V28 où les «1» sont codés par des tensions négatives.

Modems RTC Exemple de couche physique

100/307

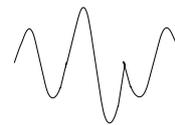
► Modulation en amplitude et phase



Modulation en amplitude



Modulation de phase



Modulation en amplitude et en phase

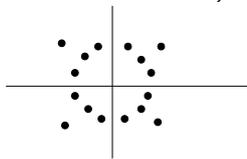
► Codage des bits

- une fréquence,
- une amplitude et une phase est un état
- x amplitudes, y sauts de phase, N états
- un état code $\log_2 N$ bits (exemple : pour 16 états on peut coder 4 bits par état)

Modems RTC : bits et bauds Codage de l'information

101/307

- ▶ Le nombre de changement d'états par seconde s'appelle la **rapidité de modulation**, elle s'exprime en **bauds**
- ▶ Le nombre de bits par seconde s'appelle le débit
- ▶ La relation entre le débit et la rapidité de modulation s'exprime par la relation suivante :
 - ▶ $D = R \log_2 N$ (N est le nombre d'états)
- ▶ Ce type de modulation se nomme «modulation d'amplitude à quadrature de phase» ou QAM (Quadrature Amplitude Modulation) et peut se représenter ainsi :



Phases espacées de 30° (12 phases)

Une amplitudes pour toutes les phases sauf 45° , 135° , 225° et 315° qui ont deux amplitudes.

16 états, 4 bits par états.

Si $R = 2400$ bauds, $D = 9600$ b/s

Une horreur à ne pas dire : «des bauds par seconde!» car alors ce n'est plus un débit mais une accélération.

Modems RTC : modulation et débit Codage de l'information

102/307

- ▶ Le débit est exprimé en bits par secondes
- ▶ Sa valeur maximum dépend des caractéristiques physiques du canal et de la capacité de l'électronique de réception de reconnaître le signal utile dans le signal composite formé par le signal lui même et les perturbations électromagnétiques incontournables (ce qu'on appelle le bruit)
- ▶ Les caractéristiques physiques du canal permettent de définir une bande passante maximum : W (coupure à 3db)
- ▶ La puissance du bruit est estimée par rapport à la puissance du signal lui même dans le rapport Signal sur Bruit (S/B)
- ▶ Résultat fondamental (Shannon 1948) $D_{max} = W \log_2(1 + \frac{S}{B})$

4.3 Exemple de couche liaison de données

HDLC – High-Level Data Link Control

Exemple de couche liaison 104/307

▶ La trame HDLC

Délimiteur	Adresse	Contrôle	Données	CRC	Délimiteur
1 octet	1 octet	1 ou 2 octets	n octets	2 octets	1 octet

- ▶ Fournir une structure en trame dans un «flot de bits» (ou d'octets)
- ▶ Le champ délimiteur est aussi appelé le fanion
- ▶ Le champ «Contrôle» est aussi appelé «Commande»
- ▶ Pour que la structure du fanion ne se retrouve pas dans les données, chaque série de cinq «1» successifs est suivie par un «0» rajouté. Technique dite du «bit stuffing»

HDLC : High-level Data Link Control (ne pas trop chercher le sens profond de cet acronyme... Se souvenir cependant fermement de son nom et de ce qu'il représente...).

Le champ adresse est aujourd'hui très mal nommé car ce type de structure de données n'est généralement utilisé que pour des liaisons point-à-point. Donc, ici, l'adresse ne sert pas à l'adressage... Il sert à distinguer des trames de commandes et des trames de réponses. Une trame de commande est une trame émise sans qu'elle soit demandée, l'émetteur décide seul de l'émettre. Une trame de réponse est une trame émise en «réponse» à une trame de commande.

Les protocoles des niveaux supérieurs ne prévoient pas de délimitation de leurs PDUs. Chaque fois qu'on ne sait pas comment faire pour véhiculer de telles unités de données on a recours à HDLC, en le simplifiant parfois. C'est par exemple le cas du protocole PPP (Point to Point Protocol) utilisé pour véhiculer les datagrammes IP sur les liaisons de type série, telles que celles utilisées par les particuliers pour se connecter à Internet via des modems.

Dans le cas de PPP sur lignes séries de type PC (entre le PC et le modem et ensuite via la modulation du modem) les trames HDLC sont découpées octet par octet et émises en mode asynchrone avec, par octet, un bit START et un ou deux bits STOP (plus, éventuellement, un bit de parité). Le fanion est constitué par l'octet 0x7E, donc comme dans le cas synchrone. Si cet octet se retrouve dans les données, il est remplacé par deux octets.

- ▶ Une spécialisation des trames HDLC (porter X.25)
- ▶ Adresse : 0xC0 ou 0x80 suivant le type Commande ou Réponse et le sens de la trame (ETTD→ETCD ou l'inverse)
- ▶ Différents types de trames :
 - ▶ I (*Information*) : trames portant des numéros et des acquittements
 - ▶ le champ contrôle est sur 1 octet (numérotation modulo 8) ou sur 2 octets (numérotation modulo 128)
 - ▶ S (*Supervision*) : trames ne portant que des acquittements
 - ▶ le champ contrôle est sur 1 ou 2 octets suivant le modulo de numérotation
 - ▶ U (*Unnumbered*) : trames de commande
 - ▶ le champ contrôle est sur 1 octet
 - ▶ trames servant à établir et rompre la relation (communication) entre les deux extrémités de la liaison

LAPB signifie : «*Link Access Protocol, Balanced*», ce qui traduit en français donne «protocole d'accès au lien en mode équilibré». En fait il utilise le Asynchronous Balanced Mode (ABM) de HDLC, mode dans lequel il n'y a pas de maître/esclave, les deux parties peuvent accéder indifféremment au lien.

On ne sait pas trop comment ces appellations ont été définies et par quel normalisateur (il ne s'en vante pas celui-là). Ce n'est pas très important... Il est, par contre, très important de savoir que cela existe et se nomme ainsi et aussi comment cela fonctionne, ce que nous allons voir dans ce qui suit.

La numérotation des trames et les acquittements permettent d'utiliser le mécanisme de la fenêtre d'anticipation.

L'utilisation du champ adresse est parfaitement défini dans la recommandation ITU-T X25, le lecteur se référera à ce document pour les détails techniques si nécessaire. Ces détails n'apportent rien de fondamental au concept général, on ne fournira pas, ici, d'explications complémentaires.

Protocole HDLC-LAPB

Structure du champ «contrôle»

106/307

TABLEAU 2-7/X.25

Commandes et réponses LAPB – Fonctionnement de base (modulo 8)

Format	Commandes	Réponses	Codage							
			1	2	3	4	5	6	7	8
Transfert d'information	I (Information)		0	N(S)			P	N(R)		
Supervision	RR (Prêt à recevoir)	RR (Prêt à recevoir)	1	0	0	0	P/F	N(R)		
	RNR (Non prêt à recevoir)	RNR (Non prêt à recevoir)	1	0	1	0	P/F	N(R)		
	REJ (Rejet)	REJ (Rejet)	1	0	0	1	P/F	N(R)		
Non numéroté	SABM (Mise en mode asynchrones symétrique)		1	1	1	1	P	1	0	0
	DISC (Déconnexion)		1	1	0	0	P	0	1	0
		DM (Mode déconnecté)	1	1	1	1	F	0	0	0
		UA (Accusé de réception non numéroté)	1	1	0	0	F	1	1	0
		FRMR (Rejet de trame)	1	1	1	0	F	0	0	1

Les trames I sont aussi dites «trames numérotées». Elles sont munies, à cet effet, d'un champ N(S) appelé «numéro de séquence en émission», en pratique c'est le numéro de la trame (ici modulo 8, mais on peut numéroter modulo 128 ou 32768, le champ «contrôle» est alors codé sur 2 ou 4 octets).

Le champ N(R) est appelé «numéro de séquence en réception», en pratique il s'agit de l'acquittement. Pour une valeur x , il signifie : «vous pouvez m'envoyer la trame $n^{\circ}x$ ».

Le fait que les trames d'informations peuvent porter des acquittements s'appelle «piggy backing»

Les trames de supervision servent à porter uniquement des acquittements et à réaliser la fonction de contrôle de flux :

RR : Receiver Ready avec $N(R) = x$: «je suis prêt, vous pouvez m'envoyer la trame x », sous-entendu : «j'ai bien reçu toutes les trames jusqu'à $x - 1$ »

RNR : Receiver Not Ready avec $N(R) = x$: «j'ai bien reçu la trame $x - 1$, mais je ne peux plus recevoir de trames momentanément, arrêtez l'envoi»

REJ : Reject avec $N(R) = x$: «Hop là!!! Erreur, vous m'envoyez n'importe quoi alors que j'attends la trame $n^{\circ}x$!»

SABM : Set Asynchronous Balanced Mode (inutile de chercher à traduire) : établissement de la connexion

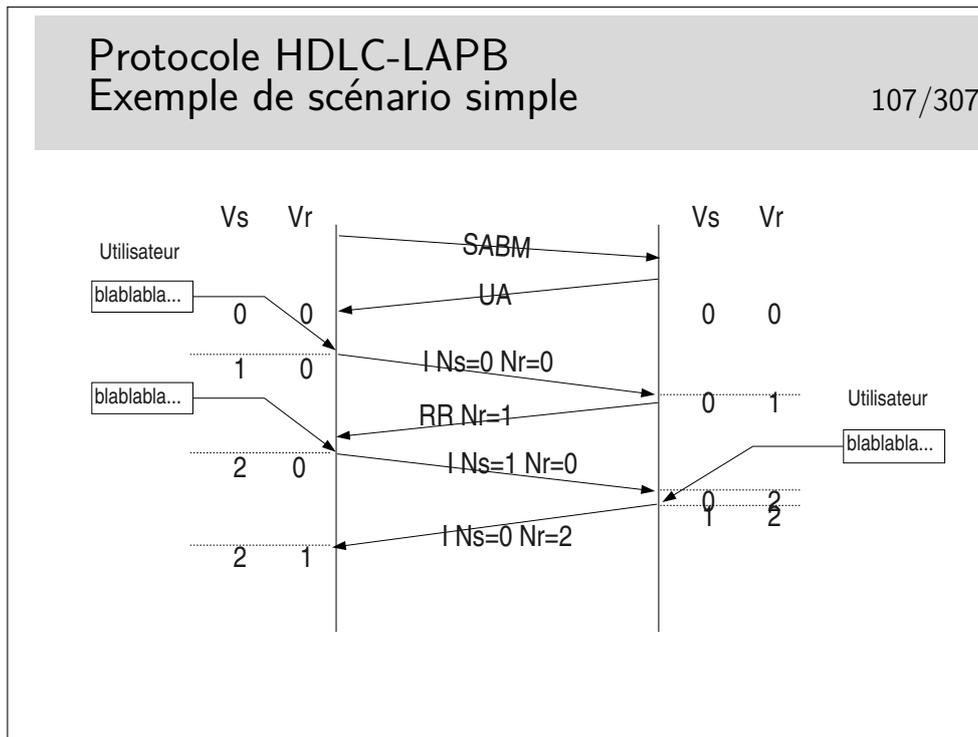
UA : Unnumbered Acknowledge (acquittement non numéroté)

DISC : Disconnect (déconnexion)

On ne s'attardera pas ici sur le rôle du bit P/F...

Protocole HDLC-LAPB Exemple de scénario simple

107/307



Les entités protocolaires d'extrémités (donc des logiciels qui implémentent le protocole LAPB) sont munies de deux compteurs **Vs** et **Vr**. **Vs** sert à numérotter les trames à émettre, **Vr** sert à vérifier que les trames que l'on reçoit portent le bon numéro.

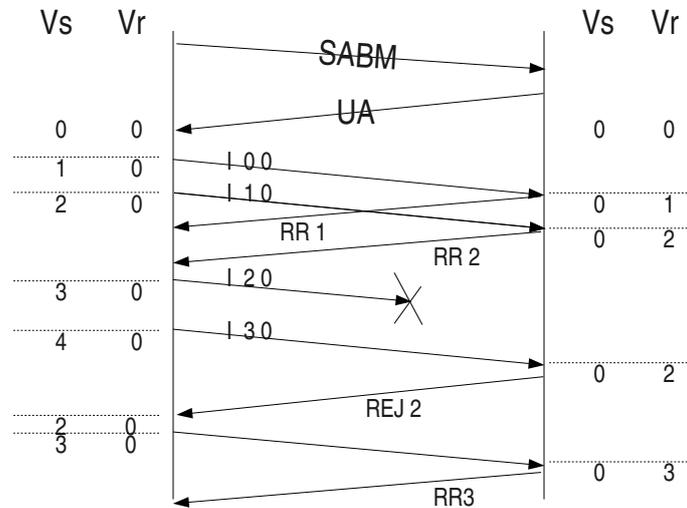
Au départ, lors de l'initialisation de la connexion par l'échange **SABM/UA**, les compteurs sont mis à 0. L'utilisateur coté gauche désire envoyer des données. Celles-ci sont encapsulées dans une trame. Le champ **Ns** de celle-ci va être rempli par la valeur de **Vs**, le champ **Nr** par la valeur de **Vr**. On recopie tout simplement **Vs** et **Vr** dans **Ns** et **Nr**. Un fois la trame envoyée on se préparera à envoyer une trame suivante si nécessaire. Celle-ci verra son numéro progresser de 1. On prépare donc cela en faisant progresser **Vs** de 1. Lors de la seconde émission on recopie **Vs** et **Vr** dans **Ns** et **Nr** de cette seconde trame, puis on l'émet et on incrémente **Vs** de 1.

Lorsque le récepteur (droite) reçoit la première trame il vérifie que son numéro de séquence **Ns** est bon en le comparant avec son compteur **Nr**. Les deux valeurs étant égales, la trame est acceptée. Le compteur **Vr** droit progresse de 1, une trame d'acquiescement est envoyée avec son **Nr** = 1 (recopie du **Vr**).

Plus tard, le coté droit désirera parler à son tour et cet événement surviendra juste au moment de la réception d'une trame (dans notre scénario). Il émettra donc directement une trame **I** portant le bon **Ns** et bien sur le bon **Nr**. (Piggy backing)

Protocole HDLC-LAPB Exemple de scénario plus complexe

108/307

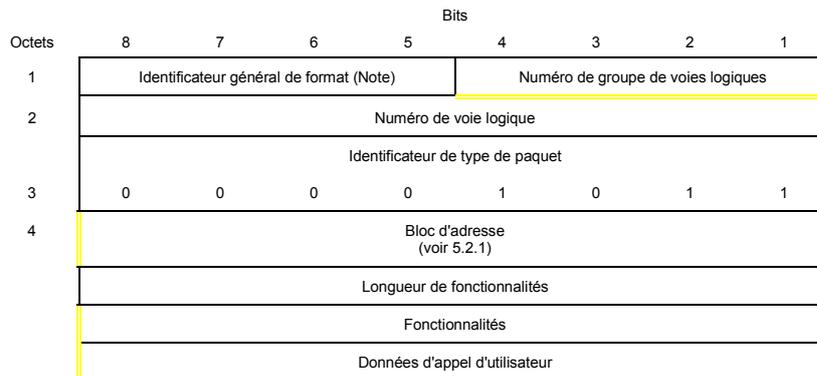


Pour quelle raison la trame numéro 2 (I 2 0) peut-elle ne pas être reçue ?

4.4 Exemple de couche réseau

X.25 commutation de paquets en p-à-p Les paquets d'appels

110/307



(Modulo 8 et modulo 128)

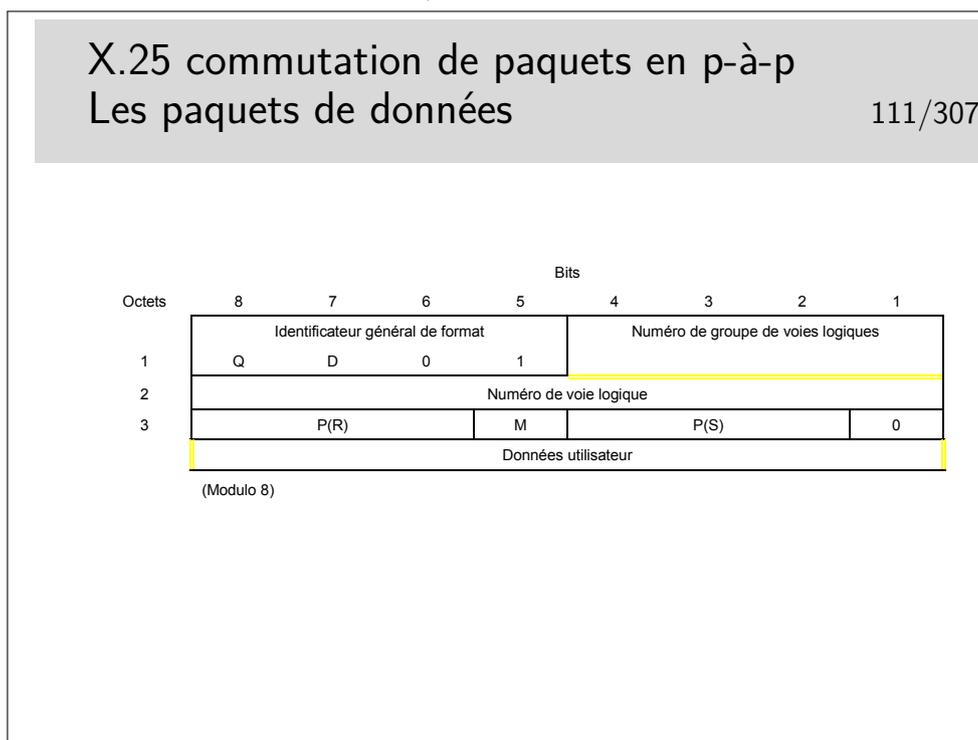
NOTE - Codé XX01 (modulo 8) ou XX10 (modulo 128).

La recommandation X.25 «suggère» que pour un appel sortant (coté demandeur), l'ETTD choisisse le numéro de voie logique le plus haut disponible. Pour un appel entrant (coté demandé) il est suggéré que l'ETCD choisisse le numéro de voie logique le plus bas

disponible.

On rappelle que l'ETTD (Équipement Terminal de Traitement de Données, ou DTE en anglais) est l'équipement «terminal» (l'ordinateur) et que l'ETCD (Équipement terminal de circuit de données, ou DCE en anglais) est l'équipement «réseau» (le commutateur de raccordement).

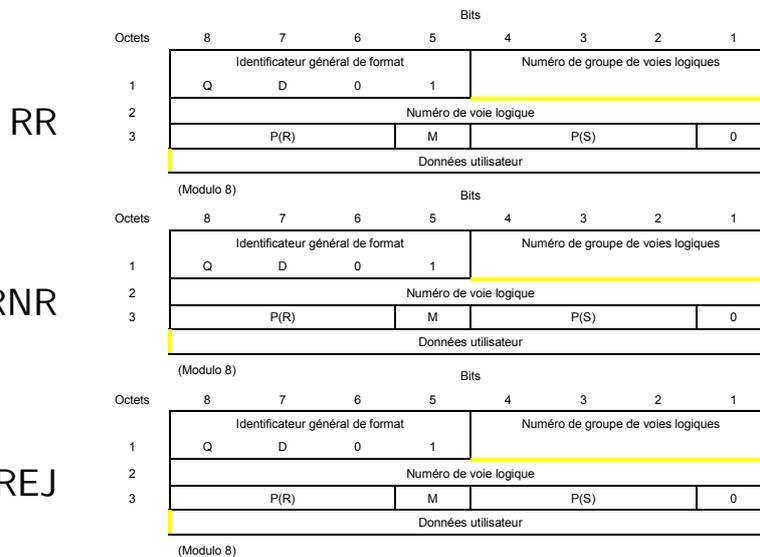
Le paquet d'appel est muni d'un bloc d'adresse contenant l'adresse de destination et possiblement l'adresse source. Il se comporte dans le réseau comme un datagramme. Il n'y a pas de système de signalisation (protocole pour l'établissement, la rupture et la supervision des communications) comme dans d'autres types de réseau à commutation (téléphone, ATM, Frame Relay, MPLS)



Les paquets de données sont numérotés (champ **P(S)** : numéro de séquence en émission). Ils contiennent aussi un champ **P(R)** permettant de transporter les acquittements (piggy backing)

X.25 commutation de paquets en p-à-p Les paquets de supervision

112/307



Comme au niveau 2 (LAPB), nous avons des paquets **RR** (Receiver Ready), **RNR** (Receiver Not Ready), **REJ** (Reject) permettant de transporter les acquittements, de réaliser le contrôle de flux et la reprise sur erreur.

X.25 commutation de paquets en p-à-p Tous les types de paquets

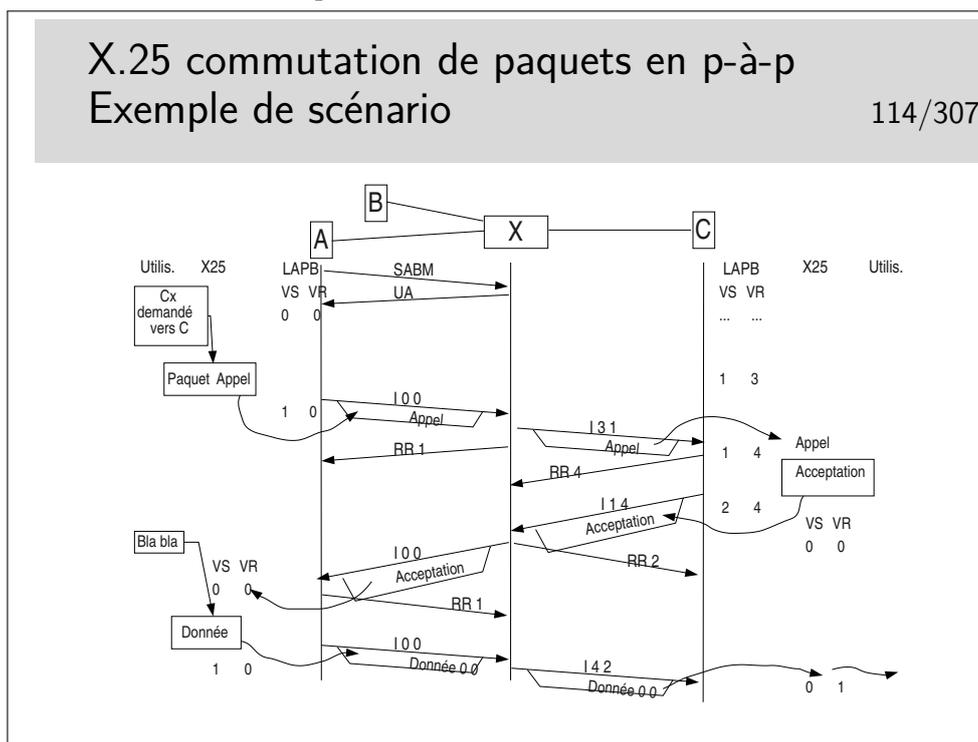
113/307

Type de paquet		Service	
de l'ETCD vers l'ETTD	de l'ETTD vers l'ETCD	VC	PVC
<i>Etablissement et libération des communications (Note 1)</i>			
Appel entrant	Demande d'appel	X	
Communication établie	Communication acceptée	X	
Indication de libération	Demande de libération	X	
Confirmation de libération par l'ETCD	Confirmation de libération par l'ETTD	X	
<i>Données et interruption (Note 2)</i>			
Données de l'ETCD	Données de l'ETTD	X	X
Interruption par l'ETCD	Interruption par l'ETTD	X	X
Confirmation d'interruption par l'ETCD	Confirmation d'interruption par l'ETTD	X	X
<i>Contrôle de flux et réinitialisation (Note 3)</i>			
RR par l'ETCD	RR par l'ETTD	X	X
RNR par l'ETCD	RNR par l'ETTD	X	X
	REJ par l'ETTD ³⁾	X	X
Indication de réinitialisation	Demande de réinitialisation	X	X
Confirmation de réinitialisation par l'ETCD	Confirmation de réinitialisation par l'ETTD	X	X
<i>Reprise (Note 4)</i>			
Indication de reprise	Demande de reprise	X	X
Confirmation de reprise par l'ETCD	Confirmation de reprise par l'ETTD	X	X
<i>Diagnostic (Note 5)</i>			
Diagnostic ⁵⁾		X	X
VC Communication virtuelle (<i>virtual call</i>)			
PVC Circuit virtuel permanent (<i>permanent virtual circuit</i>)			
³⁾ N'est pas nécessairement disponible dans tous les réseaux.			
NOTES			
1 Voir 4.1 et 6.16 pour les procédures, et 5.2 pour les formats.			
2 Voir 4.3 pour les procédures, et 5.3 pour les formats.			
3 Voir 4.4 et 6.4 pour les procédures, et 5.4 et 5.7.1 pour les formats.			
4 Voir 3.3 pour les procédures, et 5.5 pour les formats.			
5 Voir 3.4 pour les procédures, et 5.6 pour les formats.			

VC : Virtual Channel (canal virtuel ou circuit virtuel), établi à la demande par un paquet d'appel.

PVC : Permanent Virtual Channel : circuit virtuel permanent, établi par les gestionnaire du réseau à l'aide de procédures propres (commandes dans les commutateurs).

La liste ci-dessus est placée ici pour votre «culture générale», vous n'avez absolument pas besoin de connaître cette liste parfaitement.



Machines terminales (ETTD) : A, B et C.

Commutateur (ETCD) : X

On suppose dans ce scénario qu'une communication est déjà établie entre B et C. On ne s'en préoccupe pas mais le niveau 2 X-C est établi pour supporter cette communication. Des trames ont donc déjà été échangées entre X et C. Au moment où nous arrivons, les compteurs Vs et Vr du contexte LAPB en C sont, respectivement à 1 et 3.

Entre A et X, le niveau 2 n'est pas établi au début du scénario.

Deuxième partie

Réseaux locaux Ethernet et 802.3

5 Introduction

Les réseaux locaux

117/307

- ▶ Les réseaux de l'entreprise
- ▶ Caractéristiques :
 - ▶ Topologie
 - ▶ Bus, avec ou sans fil
 - ▶ Anneau
 - ▶ Étoile
 - ▶ Bande de base ou large bande
 - ▶ Caractéristiques physiques des supports (les média)
 - ▶ Cuivre
 - ▶ Fibre optique
 - ▶ radio
 - ▶ Méthode d'accès au médium
 - ▶ Comment une station peut-elle émettre ses données sur le réseau

Les différents types de réseaux

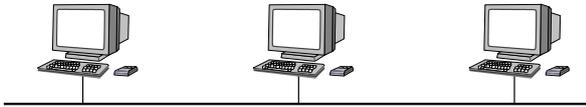
118/307

- ▶ Les réseaux locaux : LAN : Local Area Network
 - ▶ Réseaux d'entreprise
 - ▶ Courtes distances : de quelques centaines de mètres à quelques kilomètres
- ▶ Les réseaux métropolitains : MAN : Metropolitan Area Network
 - ▶ Interconnexion de réseaux locaux
 - ▶ Quelques dizaines à quelques centaines de kilomètres
- ▶ Les réseaux grande distance : WAN : Wide Area Network
 - ▶ Les réseaux nationaux et internationaux
 - ▶ Les réseaux d'opérateurs

Les topologies des réseaux locaux

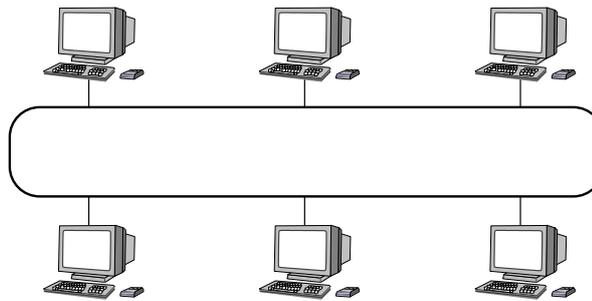
119/307

Bus



Remarque : un canal radio partagé peut être considéré comme un bus

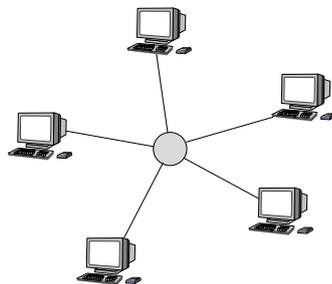
Anneau



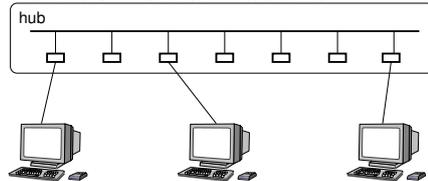
Les topologies des réseaux locaux

120/307

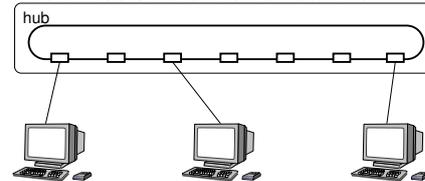
Étoile



Le bus devient étoile



L'anneau devient étoile



La topologie en étoile repose sur un nœud actif central qui contrôle le «droit de parole» des stations et achemine l'information. Cette structure n'a jamais vraiment été fortement déployée à grande échelle, on peut noter cependant les réseaux locaux basés sur le protocole X25, au début des années 80 et les ceux basés sur la technologie ATM dans les années 90 (mais c'était pour émuler Ethernet, c'est à dire un bus).

Les topologies à succès sont le bus (Ethernet, qui a gagné la bataille) et l'anneau (Token Ring, d'IBM, en cours d'abandon).

Au cours du temps, le réseau en bus Ethernet a, cependant, évolué vers une structure en étoile, où les nœuds (appelés «hubs») sont des boîtiers électroniques qui émulent un bus. Le «hub» fonctionne comme un bus.

La technologie Token Ring a, elle, été très rapidement implémentée dans des boîtiers électronique renfermant la structure d'anneau.

Aujourd'hui, donc, la topologie est en étoile, mais l'ensemble fonctionne comme un bus (ou encore parfois comme un anneau).

Caractéristiques des canaux

121/307

- ▶ Transmission en bande de base
 - ▶ Un seul canal physique pour toutes les stations
 - ▶ Problème d'accès concurrent
- ▶ Transmission en large bande
 - ▶ Plusieurs canaux sur le médium
 - ▶ Un canal est caractérisé par une bande de fréquence
 - ▶ Les signaux émis par les stations sont transposés dans la bande de fréquence qui est assignée aux stations
 - ▶ Un canal donné peut être vu comme un canal en bande de base (émulation d'un bus Ethernet par exemple)

Les méthodes d'accès au médium

122/307

Très liées à la topologie (topologie logique)

- ▶ Bus
 - ▶ Tout le monde partage le canal sans priorité
 - ▶ Chaque station peut, à priori, émettre lorsqu'elle le désire
 - ▶ Il en résulte des collisions
 - ▶ Il faut gérer le problème des collisions
 - . Algorithme spécifique de contention (CSMA CD ou CA...)
 - . Émulation d'un anneau logique
- ▶ Anneau
 - ▶ Le droit d'émettre est transmis sur l'anneau dans une trame spécifique contenant un bit spécial appelé «jeton»
 - ▶ Les trames circulent dans un sens donné

- ▶ Menée à bien par l'organisme IEEE
 - ▶ Plus particulièrement le comité 802 de l'IEEE
 - ▶ Chaque topologie et les divers protocoles et caractéristiques sont étudiés et standardisés par un sous-comité 802
 - ▶ Exemples : 802.3 pour Ethernet, 802.5 pour Token Ring
- ▶ Modèle en couches spécifique
 - ▶ Comparable au modèle OSI pour ses deux premières couches
 - ▶ Trois couches
 - ▶ La couche physique (comme pour OSI)
 - ▶ La couche «Medium Access Control» (MAC)
 - ▶ La couche «Logical Link Control» (LLC)

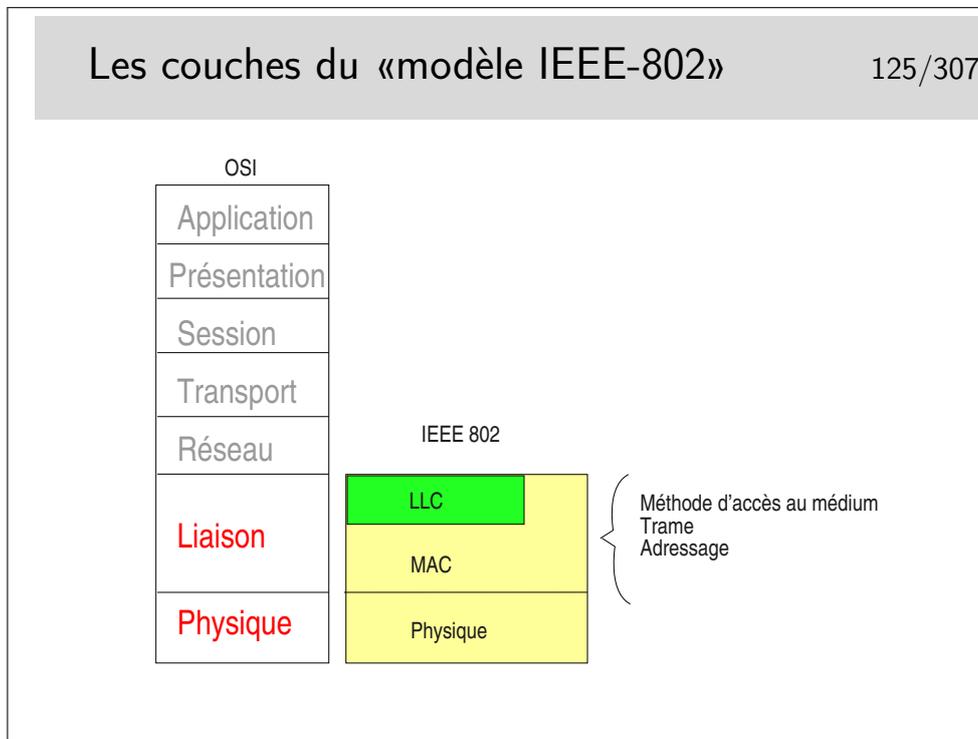
- ▶ 802.1 : architecture des réseaux locaux
 - ▶ Architecture générale, interconnexion (niveau 2), QoS, etc...
- ▶ 802.2 : la couche LLC
- ▶ 802.3 : Ethernet
 - ▶ 802.3u : Ethernet 100Mb/s
 - ▶ 802.3ab, z : Ethernet 1Gb/s
 - ▶ 802.3ae : Ethernet 10Gb/s
- ▶ 802.4 : le bus à jeton
- ▶ 802.5 : l'anneau à jeton (Token Ring)
- ▶ 802.11 : les réseaux sans fils (WiFi)
- ▶ 802.15 : les WPAN : Wireless Personal Area Network

Beaucoup de sous comités :

- 802.1d : techniques de pontage
- 802.1p/q : classes de service, techniques des VLANs (Virtual LAN)
- 802.3u : Ethernet
- 100Mb/s 802.3z : Ethernet 1Gb/s
- 802.3ab : Ethernet 10Gb/s

- 802.11a : sans fils, 5Gz (projet)
- 802.11b : sans fils, 11Mb/s
- 802.11g : sans fils, 54Mb/s

Note : WiFi (Wireless Fidelity) est une dénomination commerciale pour le 802.11



La couche LLC permet de définir différents types de liaisons (avec/sans connexion et avec/sans acquittement). Elle n'est pas toujours obligatoire, en Ethernet elle est optionnelle, certains protocoles de niveau 3 l'utilisent, d'autre pas. Par exemple IP, sur Ethernet, n'emploie pas, par défaut, la couche LLC (mais peut le faire, le choix se fait par paramétrage au niveau du système d'exploitation).

La couche MAC est centrale, elle définit l'algorithme gérant l'accès concurrent au support. Elle définit aussi une structure de trame ainsi qu'un mécanisme d'adressage (les adresses MAC, nous verrons cela plus loin).

6 La technologie Ethernet

6.1 Fondements d'Ethernet

Les méthodes d'accès sur Bus

128/307

- ▶ L'ancêtre : ALOHA de l'université de Hawaï
 - ▶ Tout le monde a le droit d'émettre quand il veut
 - ▶ Les collisions sont nombreuses
- ▶ Améliorations CSMA Carrier Sense Multiple Access
 - ▶ On écoute le canal, s'il est silencieux on peut émettre
 - ▶ Les collisions ne sont pas absentes, elles sont moins nombreuses
- ▶ Méthodes de contention des collisions
 - ▶ CA : Collision Avoidance
 - ▶ On envoie une trame test (TRTS : Request To Send), si elle ne collisionne pas, on peut émettre
 - ▶ CD : Collision Detection
 - ▶ On émet et on écoute, il y a collision si le signal écouté est différent de celui qu'on émet, on arrête l'émission qu'on retente après un temps aléatoire

Une autre méthode consiste à créer un anneau virtuel. Les stations s'échangent un jeton dans un ordre donné (la station A passe le jeton à la station B qui le passe à C, etc. La station qui désire émettre doit attendre de voir arriver le jeton). Ce type de réseau a été développé par des industriels des mondes de l'automobile et de l'aviation américains. Il fait partie des «réseaux locaux industriels». Il a été standardisé sous le numéro 802.4 par l'IEEE. Il ne s'est pas développé de manière importante.

- ▶ La technologie impérialiste, elle a écrasé toutes les autres (ou les autres s'adaptent à elle, exemple le WiFi)
- ▶ Topologie logique : le bus
 - ▶ Aujourd'hui, topologie physique en étoile avec hubs et commutateurs (switches)
- ▶ Méthode d'accès
 - ▶ CSMA-CD : Carrier Sense Multiple Access with Collision Detection
- ▶ Origine : Intel, Xeros, Digital, première idée en 1976 (Bob Metcalfe)
- ▶ Standardisation générale : IEEE-802.3
- ▶ Des débits divers : 10, 100, 1Gb/s, voir 10Gb/s

La technologie WiFi (IEEE-802.11) n'a rien à voir, à priori, avec Ethernet, mais on commence à l'appeler couramment «l'ethernet sans fil», c'est dire... Elle s'adapte, en effet, très bien à Ethernet via des ponts simples à mettre en place.

La technologie Ethernet évolue de manière souple où chaque étape d'évolution ne remet pas en cause les versions antérieures. On peut raccorder une interface 10Mb/s sur une interface 100 ou 1000Mb/s. Les matériels d'interconnexion sont compatibles avec les trois débits.



- ▶ La station A écoute le réseau, il n'y a pas de signal, elle peut émettre
- ▶ Le signal se propage
- ▶ La station B écoute le réseau, le signal de A ne lui est pas encore parvenu, elle décide d'émettre
- ▶ Les deux signaux vont collisionner, le signal résultant va se propager de part et d'autre et va parvenir aux deux stations qui **continuent d'écouter**
- ▶ Chaque station continue à émettre quelques instants pour renforcer la collision et s'arrêtent
- ▶ Chacune tire un temps aléatoire au bout duquel elles tentent une ré-émission

- ▶ Si une station émet pendant au moins 1 RTT alors il n'y aura plus de collision non détectée
- ▶ Si une collision est détectée, chaque station en cause arrête son émission
 - ▶ Chaque station considère alors 2 intervalles de temps de valeur RTT et tire aléatoirement 1 ou 2 et ré-émet tout de suite (si 1 est tiré) ou un RTT plus tard (si 2 est tiré)
 - ▶ Si une nouvelle collision intervient, on considère alors 4 RTT, et on tire aléatoirement entre 1 et 4. On tente une émission au début de l'intervalle de temps tiré
 - ▶ En Ethernet, on peut tenter jusqu'à 16 réémissions mais on ne multiplie par 2 que jusqu'à 10 fois le nombre de RTT

Bus Ethernet : quel est le diamètre du réseau ?

132/307

Ou encore : quelle est la distance maximale entre deux stations ?

- ▶ Pour détecter une collision il faut que les deux stations soient encore en émission lorsque le signal de collision leur revient
- ▶ Cas le plus défavorable : elles sont situées aux deux extrémités du réseau et la station B émet un court instant avant que le signal de A ne lui parvienne. La collision a lieu près de B
 - ▶ Pour que A détecte la collision il lui faut attendre un temps égal à la durée de propagation de A jusqu'à B et retour
 - ▶ On appelle ce temps le Round Trip Delay, le Round Trip Time (RTT), la «tranche canal» ou encore la fenêtre de collision
 - ▶ Ce temps dépend : de la taille du segment de données (trame), de la durée d'émission de cette trame et de la distance entre les deux stations les plus éloignées

Quelle est la dimension du réseau (son diamètre ou encore la distance maximale entre deux stations) si la taille minimale des trames est de 512 bits (64 octets), le débit de 10Mb/s et la célérité du signal sur le câble de 200Km/s (c'est sous évalué...)?

Vous devez trouver 5,12Km, ce qui est trop. Dans la réalité il faut tenir compte de l'affaiblissement en ligne qui ne permet pas de propager un signal sur une telle distance sans pertes de puissance et distorsion en ligne. Ces phénomènes obligent à placer des or-

ganes régénérateurs à des distances bien plus courtes. On appelle ces organes de répéteurs car ils «répètent» les bits, ils les régènèrent. Ce ne sont pas des amplificateurs car alors les distorsions seraient elle-mêmes amplifiées.

Les premiers réseau Ethernet pouvaient compter des segments de 500m maximum, reliés par des répéteurs.

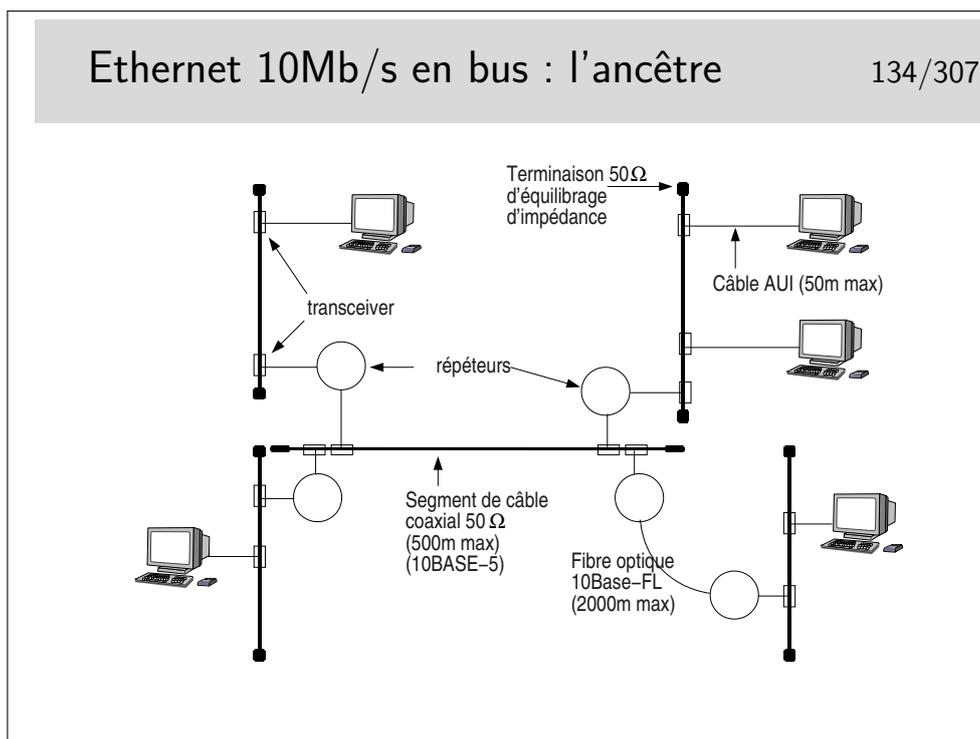
Les répéteurs apportent un retard, les câbles de raccordement de ces répéteurs au câble principal aussi. Le bilan des délais est tel que la taille maximale est de 2,5km pour les valeurs suivantes :

- taille minimale des trames : 512 bits
- débit d'émission : 10 Mb/s
- célérité sur les segments de câble principaux : $0,77c$
- 5 segments de 500 m maximum et 4 répéteurs maximum entre deux stations.

Ethernet : les caractéristiques générales au débit de 10Mb/s

133/307

- ▶ Taille minimale de la trame : 64 octets (512 bits)
- ▶ Taille maximale : 1518 octets
 - ▶ 1500 octets de charge utile (SDU) pour Ethernet pur
 - ▶ 1497 ou 1496 ou même 1492 en format 802.3 où la couche LLC est nécessaire
- ▶ Taille minimale : 64 octets
 - ▶ Bourrage dans le champ «données» si la longueur de celles-ci est inférieur à 46 octets
- ▶ Silence inter-trame de $9,6\mu s$
- ▶ Tentatives de réémission en cas de collision : 16



Le médium principal est constitué par des segments de câble 50Ω de 500m maximum, reliés entre eux par des répéteurs. Il ne peut y avoir plus de 5 segments entre deux stations (4 répéteurs). Les câbles sont terminés par un «bouchon» constitué par une résistance de 50Ω permettant d'équilibrer l'impédance caractéristique du support.

Un répéteur est un organe qui «répète» sur tous les ports les bits qui arrivent sur un port. Ce n'est pas une amplification car si cela était les altérations du signal seraient amplifiées. Les bits entrant dans un port sont reconnus et régénérés sur tous les autres ports et même sur les fils émission du port sur lesquels ils arrivent.

Les stations et les répéteurs sont raccordés au câble principal via des câbles de 50m maximum. Le câble arrive sur un organe de raccordement directement fixé sur le médium appelé le «transceiver».

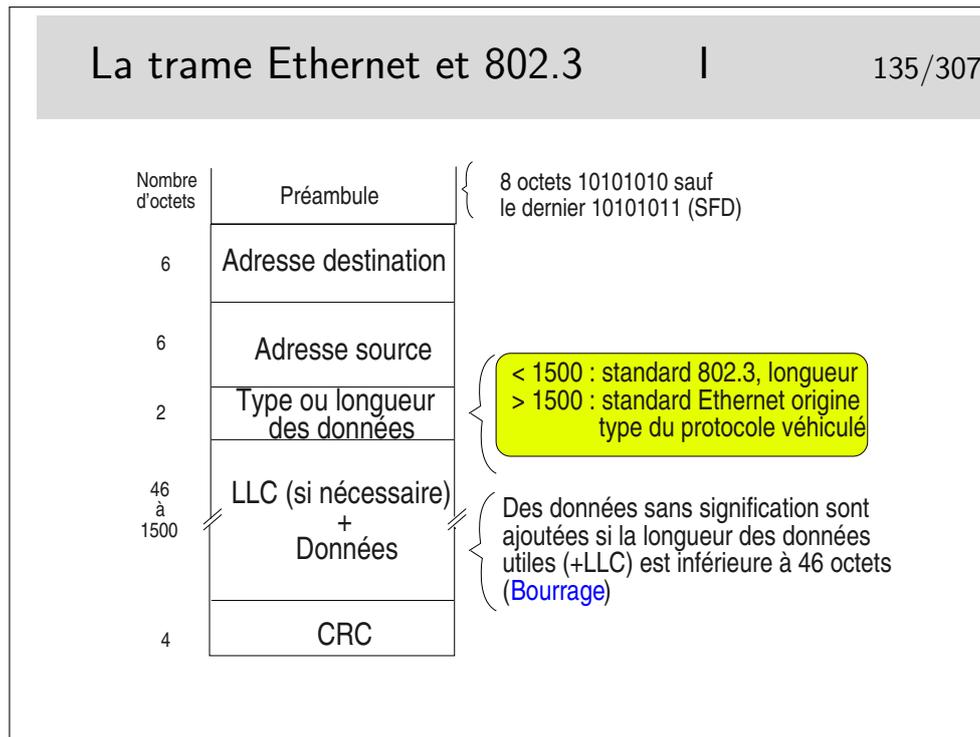
Les transceivers doivent être placés à des distances multiples de 2,5m pour des contrer le phénomène des ondes stationnaires. Un point ou un anneau de couleur noire sur le câble (qui lui est jaune) indique les emplacements possibles.

Deux segments principaux peuvent être reliés par une fibre optique de longueur 2000m maximum. Les deux répéteurs aux extrémités de la fibre ne comptent alors que pour 1.

Ce type de réseau est aujourd'hui abandonné. Les segments de câble sont remplacés par des hubs (ou des switches) et l'architecture physique est devenue ainsi une étoile tout en continuant à fonctionner avec une technique de bus. Le hub peut être comparé à ne boîte dans laquelle on aurait enfermé le câble et les transceivers.

Les dénominations 10BASE-5 et 10BASE-FL sont explicitées plus loin.

Il existe une version «cheaper net» comportant des câbles coaxiaux fins (de couleur noire), avec un raccordement par prises de type BNC. Ces câbles font 185m max (10BASE-2).



SFD : Start Frame Delimitor

Les créateurs d'Ethernet (Bob Metcalfe et Intel/Xeros/Digital)) ont défini le champ «Type» pour porter l'identité du protocole véhiculé dans les données (le champ type est le SAP). Pour des soucis d'interopérabilité avec des réseaux locaux dont les trames n'ont pas ce champ type (Token Ring 802.5), le comité 802 a décidé qu'il serait obligatoire d'utiliser la couche LLC pour porter l'identité du protocole véhiculé et a transformé le rôle de ce champ en indicateur de longueur des données. Ce n'est pas une mauvaise idée en soi.

Et en pratique ?...

En pratique les deux coexistent sur les mêmes supports. Une machine peut émettre en 802.3 pour certaines applications et en Ethernet pur pour d'autres.

Par défaut IP est véhiculé en Ethernet pur (type 0800hexa), il peut être véhiculé en mode 802.3 en paramétrant l'interface.

- ▶ C'est un datagramme
 - ▶ Elle contient l'adresse de la station destinatrice ainsi que l'adresse de la station qui émet
 - ▶ Elle contient un CRC, on peut donc vérifier son intégrité en réception
 - ▶ Si cette vérification montre une altération, la trame est jetée
- ▶ Il n'est pas prévu à ce niveau (MAC) d'échange préalable pour établir une relation entre l'émetteur et le récepteur
 - ▶ Il n'y a pas de connexion
 - ▶ Il y a pas de contrôle de flux
 - ▶ Il n'y a pas de récupération sur erreur

6.2 Les adresses MAC

- ▶ Définies dans la couche MAC (adresses MAC)
 - ▶ 6 octets
- ▶ 3 types d'adresses
 - ▶ Les adresses de stations, dites aussi «unicast» :
 - ▶ une adresse unique par interface matérielle (une machine peut avoir plusieurs interfaces matérielles)
 - ▶ L'adresse d'une interface est affectée par son constructeur
 - ▶ L'adresses globale, dite aussi «broadcast»
 - ▶ Permet d'envoyer une trame à toutes les stations du réseau, en une seule opération
 - ▶ Les adresses de groupes, dites aussi «multicast»
 - ▶ Permettent d'adresser un groupe de stations

En terminologie anglo-saxonne les interfaces matérielles sont appelées NIC pour *Network Interface Card*.

Une interface voyant passer une trame de broadcast (diffusion) doit prendre en compte cette trame.

Une interface voyant passer une trame de multicast ne la prend en compte que si elle a été paramétrée pour cela.

Format des adresses Ethernet 139/307

- ▶ **Format IEEE-48**
 - ▶ 6 octets
 - ▶ Les 3 premiers affectés au constructeur par l'IEEE (OUI : Organisation Unit Identifier)
 - ▶ Les 3 derniers affectés par le constructeur

1er bit émis (Ethernet)
1 : adresse de groupe
0 : adresse individuelle

1 : OUI non IEEE
0 : OUI officiel IEEE

Les adresses de broadcast et de multicast 140/307

- ▶ **Le groupe total : la diffusion ou broadcast**
 - ▶ ff:ff:ff:ff:ff:ff: les 48 bits à 1
- ▶ **Les groupes restreints : le multicast**
 - ▶ Le bit de poids faible du premier octet est à 1
 - ▶ Quelques exemples :
 - ▶ 01:00:5E:xx:xx:xx: multicast IP
 - ▶ 01:80:C2:00:00:00: spanning tree (protocole de gestion automatique des ponts et commutateurs)
 - ▶ 09:00:4E:00:00:02: multicast IPX

Information sur les OUI et adresses multicast :
<http://standards.ieee.org/regauth/oui/oui.txt>

Exemples d'OUI, d'adresses multicats, de types

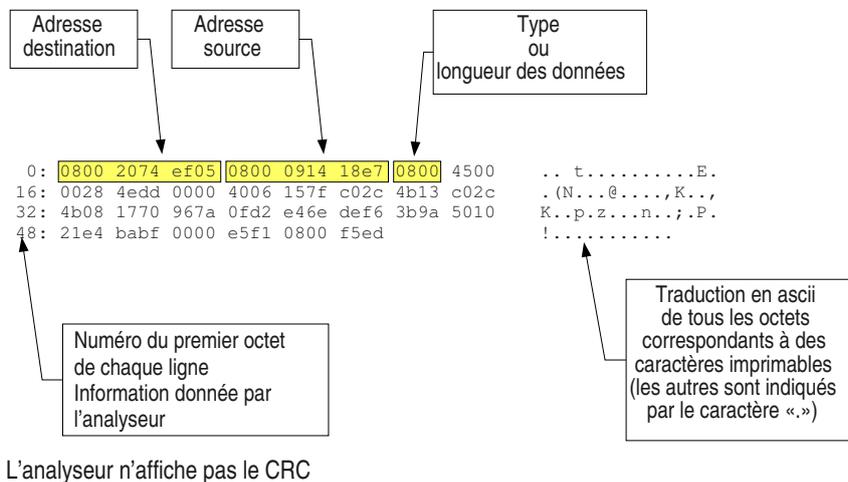
141/307

- ▶ Quelques OUI
 - ▶ 08-00-07 Apple
 - ▶ 00-00-0C Cisco
 - ▶ 08-00-08 HP
 - ▶ 08-00-20 Sun
- ▶ Quelques adresses multicast
 - ▶ 01-00-5E-xx-xx-xx Multicast internet
 - ▶ 01-80-C2-00-00-00 spanning tree

<http://standards.ieee.org/regauth/oui/oui.txt>

Une trame Ethernet capturée par un analyseur

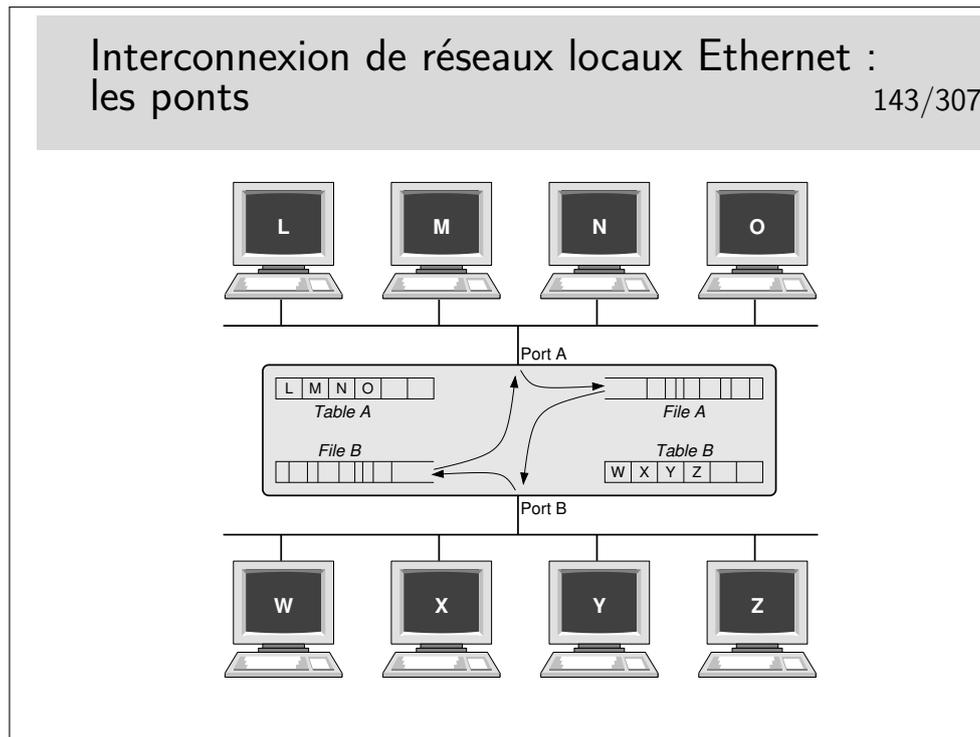
142/307



Cette trame est au format 802.3 ou Ethernet pur ?

Interconnexion de réseaux locaux Ethernet : les ponts

143/307



Le pont fonctionne tout seul sans paramétrage préalable. Il écoute le réseau sur son port A et son port B et «voit» passer des trames. Il enregistre les adresses sources de celles-ci et détermine ainsi que L, M, N et O sont du côté A. Ces adresses sont «appprises» par le pont et enregistrées dans sa table A. De même il «apprend» que, coté B, existent les stations W, X, Y et Z. Ces dernières adresses sont stockées dans la table B.

Si une trame est émise par L à destination de O, il reconnaît que ces deux stations sont du côté A, il ne fait rien. De même pour des trames de M vers N ou X vers Z par exemple.

Par contre si une trame est émise de A à destination de Z, alors cette trame est enregistrée dans la file A puis une tentative d'émission de celle-ci sera effectuée coté B. La trame ne sera mémorisée dans la file A que si elle n'a pas collisionné. Elle pourra peut-être collisionner coté B lors de sa ré-émission de ce coté mais la collision ne sera pas détectée coté A.

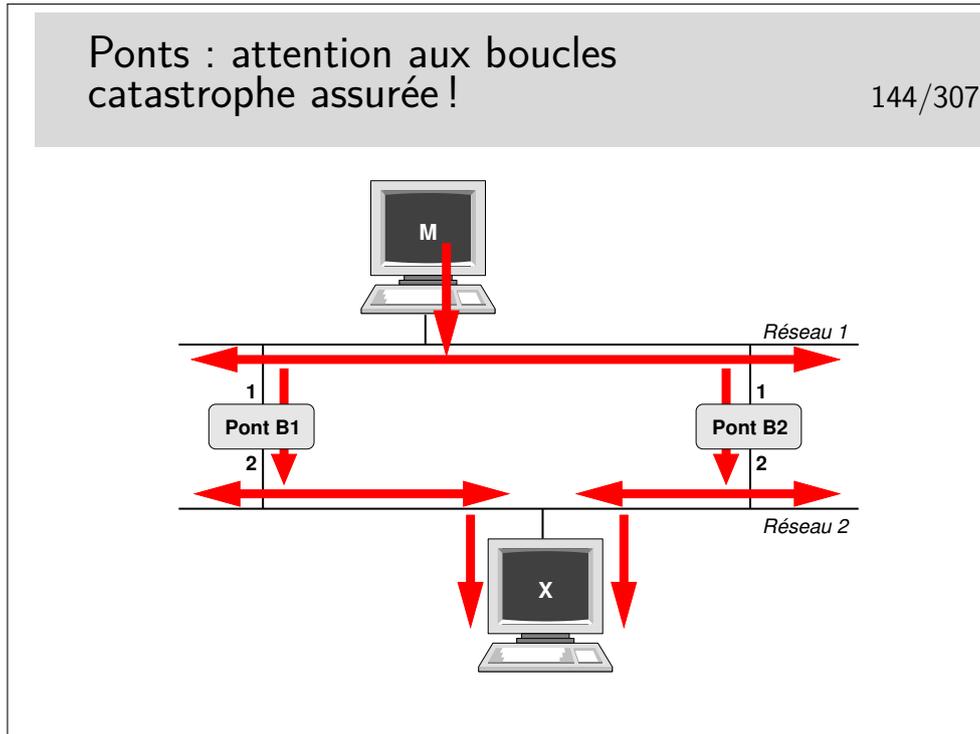
Si le pont ne connaît pas la station destinatrice (elle n'a rien émis encore) il retransmet les trames vers cette destination sur le port opposé de celui sur lequel il reçoit ces trames.

Avantages :

- le trafic est segmenté
- la connectivité totale est maintenue
- le réseau est divisé en «domaines de collisions»
- le diamètre du réseau peut être doublé

Ponts : attention aux boucles catastrophe assurée !

144/307



Pour des raisons de fiabilité on peut être amené à doubler les ponts entre deux réseaux. Se pose alors le problème de boucles comme l'exprime l'exemple présenté ici.

La machine M émet une trame à destination de la machine X.

Premier point négatif : multiplication des copies de trames...

Le pont B1 enregistre la trame et la réémet sur son port 2. La trame arrive à destination. Le pont B2 fait de même. Une seconde copie arrive à destination.

Second point négatif : les trames font des boucles...

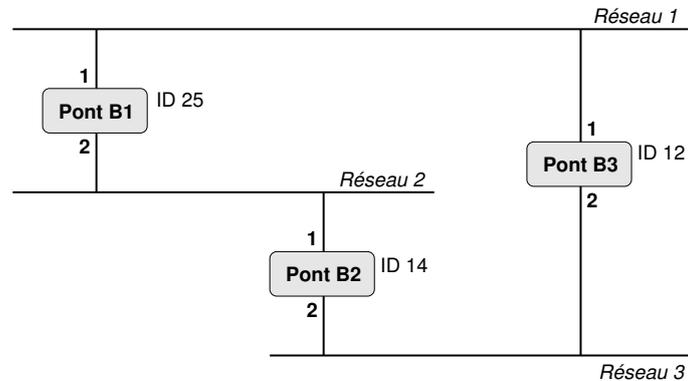
Si la station X n'a rien émis encore les ponts ne la connaissent pas. La station M émet une trame vers X. Les ponts ne connaissant pas X retransmettent cette trame comme précédemment. Il y a donc deux copies. Mais...

La copie faite par B1 arrive à destination et aussi en B2 port 2. B2 croit alors que la machine M a changé de réseau, elle est maintenant en bas. Comme il ne connaît pas X il recopie cette trame sur son port 1. Cette trame sera vue par B1 port 1 qui va la relayer vers son port 2 et va à nouveau arriver en X puis en B2 port 2 et ainsi de suite. Mais une trame tournera aussi dans le sens contraire car au début la copie faite par B2 arrive à destination.. etc... etc... Une seule trame émise suffit alors à saturer le réseau.

Pour éviter ce phénomène il faut transformer le réseau. De graphe avec des boucles il faut le transformer en arbre complet, en anglais en «spanning tree».

Éviter les boucles : Le mécanisme du «spanning tree»

145/307

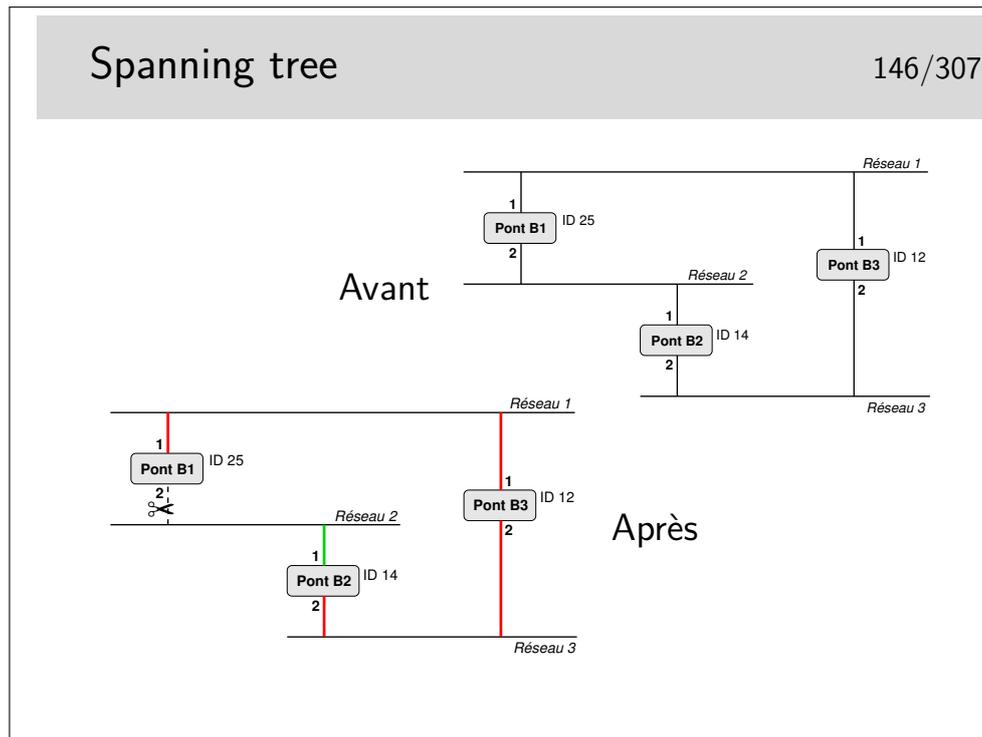


Les ponts (ou les commutateurs Ethernet) administrables peuvent mettre en œuvre cet algorithme. Sur chacun de ses ports, chaque pont annonce un identificateur qui lui est propre via une trame de type multicast (seuls les ponts sont configurés pour prendre en compte ce type de trame). Cet identificateur signifie en quelque sorte «c'est moi le pont racine». Tous les ponts commencent par annoncer qu'ils sont racine. Lorsqu'un pont reçoit un message comportant un identificateur «inférieur» au sien (on dira «meilleur»), il s'aperçoit qu'il n'est pas «racine» mais qu'un autre pont l'est plus certainement et que celui-ci est accessible via l'interface par laquelle est parvenue ce message «meilleur».

L'ensemble des ponts exécute le même algorithme, celui-ci converge finalement car un seul pont se trouve être le «meilleur», la racine (celui dont l'identificateur est le plus petit). Chaque pont sait par quelle interface atteindre le pont racine par le «plus court chemin», seule cette interface reste active pour le trafic de données, les autres interfaces permettant, elles aussi d'atteindre la racine, mais avec un chemin plus long sont inhibées pour le trafic de données. Les boucles sont ainsi évitées.

Les messages échangés s'appellent des BPDU (Bridge_PDU) et comportent les informations suivantes : **<id_pont_supposé_racine, coût_supposé, id_pont_émetteur, port_émission>**.

Les messages du protocole Spanning tree sont portés par des trames en adressage multicast. Leur adresse destination est (en standard) **01:80:c2:00:00:00**. Ce sont des trames de type 802.3. Les DSAP et SSAP de la couche LLC supérieure (voir plus loin) sont égaux à **0x42**. Les messages sont émis toutes les 2 secondes par défaut. Les identificateurs, les délais sont généralement paramétrables.



B1 émet : $\langle 25, 0, 25, 1 \rangle$ sur son port 1 et $\langle 25, 0, 25, 2 \rangle$ sur son port 2.

Il reçoit $\langle 12, 0, 12, 1 \rangle$ par son port 1 et $\langle 14, 0, 14, 1 \rangle$ par son port 2. Il constate qu'il n'est pas racine et que la racine doit être 12 (B2) accessible par son port 1. B1 considère que son port 1 est «root port».

Il émet alors : $\langle 12, 1, 25, 1 \rangle$ sur son port 1 et $\langle 12, 1, 25, 2 \rangle$ sur son port 2.

B2 émet $\langle 14, 0, 14, 1 \rangle$ sur son port 1 et $\langle 14, 0, 14, 2 \rangle$ sur son port 2.

Il reçoit : $\langle 12, 0, 12, 2 \rangle$ port 2 et $\langle 25, 0, 25, 2 \rangle$ port 2 dans une première phase indiquant ainsi que le pont racine doit être accessible par son port 2 plutôt que par son port 1. B2 considère que son port 2 est «root port».

Il émet alors : $\langle 12, 1, 14, 2 \rangle$ sur son port 2 et $\langle 12, 1, 14, 1 \rangle$ sur son port 1.

B3 ne reçoit pas de «meilleure» configuration que celles qu'il émet. Il est donc la racine.

Comment vont alors se départager B1 (port 2) et B2 (port 1) ?

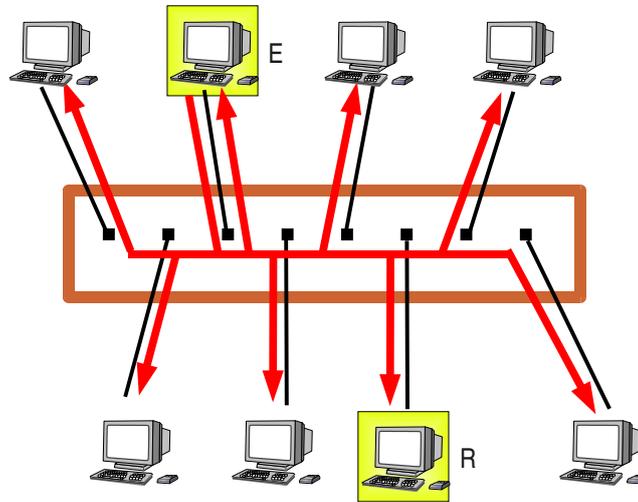
B1 reçoit port 2 : $\langle 12, 1, 14, 1 \rangle$ ce qui est meilleur que ce qu'il émet sur ce même port $\langle 12, 1, 25, 2 \rangle$. Il inhibe son port 2 (pour le trafic de données, pas pour les messages de spanning tree). B2 reçoit sur son port 1 un message «moins bon» que ce qu'il émet sur ce même port, donc B2 port 1 reste actif.

Excellent tutorial animé à :

http://www.cisco.com/warp/public/473/spanning_tree1.swf

Le bus devient étoile : raccordement par hubs

147/307



On prend le câble et les transceivers, on ramasse le tout dans un petit boîtier et le tour est joué. D'une topologie en bus nous passons à une topologie en étoile.

En fait ce n'est pas si simple car les raccordement changent, ce ne sont plus des transceivers mais des prises de type RJ45 et les câbles sont de type 10/100 BASE-T, à paire torsadée, de longueur max 100m.

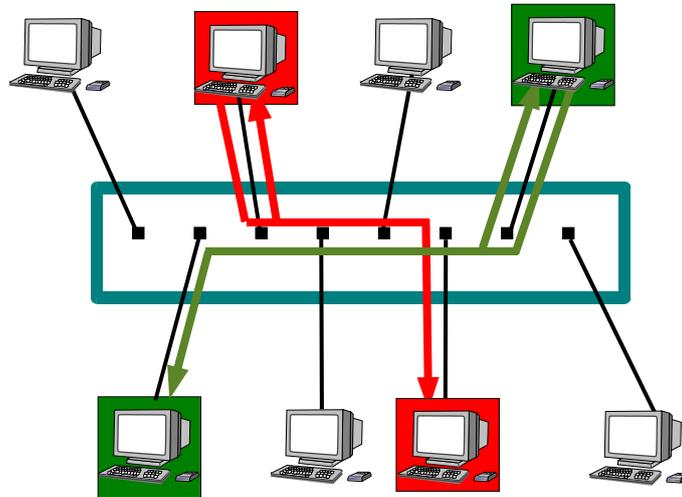
Exemple de fonctionnement : La station E émet vers la station R. Toutes les stations reçoivent le signal, même E pour des raisons de détection de collision.

Un hub se comporte donc comme un bus.

Il est très facile d'espionner tout ce qui passe sur le hub à partir d'une machine raccordée sur un des ports.

Les hubs sont des **répéteurs**. Ce sont des organes de la **couche physique** (selon le modèle OSI), ils ne s'intéressent qu'au niveau bit.

Le bus devient étoile :
raccordement par commutateurs (switch) 148/307



Le commutateur ressemble à un hub mais se comporte comme un pont. Il connaît les adresses (MAC) des machines qui sont raccordées sur chacun de ses ports (raccordées directement ou via d'autres d'autres commutateurs ou hubs). C'est un pont multiport.

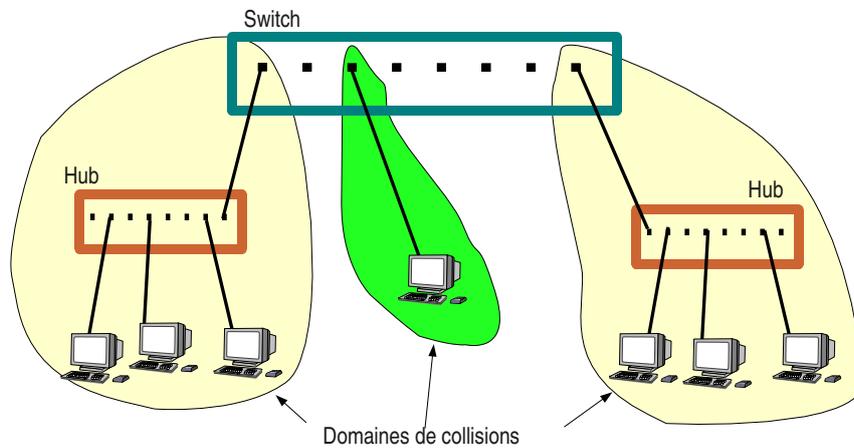
Il ne recopie une trame que sur le port qui mène vers la destination.

Il peut faire ce travail pour plusieurs trames simultanément à condition qu'elles n'aillent pas vers le même port de sortie, auquel cas elles sont traitées les unes après les autres. Il se comporte donc comme s'il était capable d'établir des circuits entre deux ports pour de courts instants, d'où son nom de commutateur (switch en anglais).

C'est une **commutation de niveau 2** (plus exactement de niveau MAC), à ne pas confondre avec la commutation de niveau 3 qu'on rencontre dans d'autres réseaux comme X25 ou ATM.

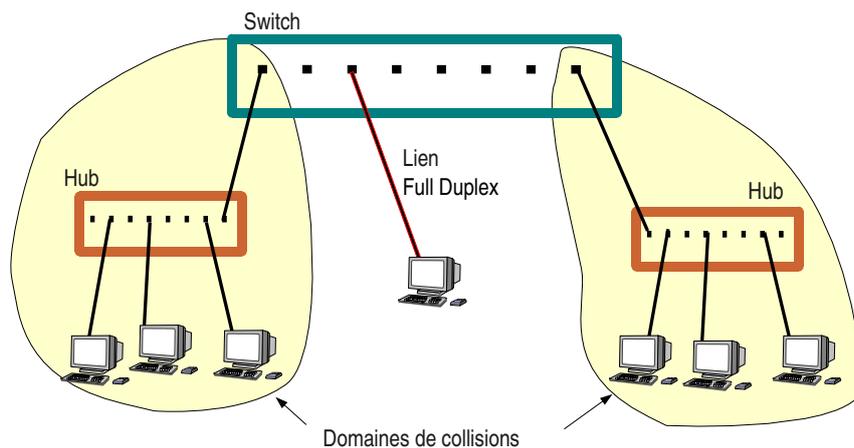
L'espionnage sur ce type d'appareil est plus difficile, sur un port donné on ne peut voir que les trames destinées à ce port ou les trames de *broadcast*, l'intérêt pour «l'espion» est donc très limité. Le «malfaisant» persévérant et cultivé en Réseaux peut cependant trouver des solutions...

Un switch est un pont multiports



Un switch se comporte comme un pont. Sur chacun de ses ports il enregistre les adresses sources des trames qu'il voit passer. Chaque port se comporte comme un port de pont. Chaque port délimite donc un domaine de collision.

On voit sur la figure ci-dessus qu'un de ces domaines ne comporte que deux machines (un port du switch et une machine seule), Voir le transparent suivant pour une utilisation plus efficace du lien en question.



Lorsqu'une machine terminale est reliée directement à un switch le domaine de col-

lision est restreint à la machine et au port du switch sur lequel elle est raccordée. Si on considère en plus que le lien physique de raccordement est un câble comportant un circuit différent pour chaque sens (deux fils par pour l'émission, deux autres fils pour la réception, ou deux fibres optiques), il devient alors intéressant d'inhiber le mécanisme de détection et contention de collision pour tirer partie pleinement de la paire émission et de la paire réception présente dans le câble de raccordement en les utilisant simultanément pour véhiculer des données utiles.

On peut ainsi passer de 10/100/1000 Mb/s à l'alternat à 10/100/1000 Mb/s dans chaque sens simultanément (10 ou 100 ou 1000 MB/s selon le matériel).

Attention, en général il faut gérer le full duplex, c'est à dire qu'il faut se connecter au switch pour l'administrer (sur son port console via un PC et l'application HyperTerminal ou via telnet en IP) et configurer les ports qu'on désire voir fonctionner dans ce mode. Il faut aussi vérifier sur la machine terminale qu'elle est bien en full duplex et, éventuellement, la forcer dans ce mode. Si le full duplex n'est pas positionné des deux cotés il en résultera un fonctionnement très ralenti, une des deux extrémités détectant alors des collisions qui n'en sont pas.

6.3 Les VLANs

Le concept de LAN virtuel ou VLAN | 152/307

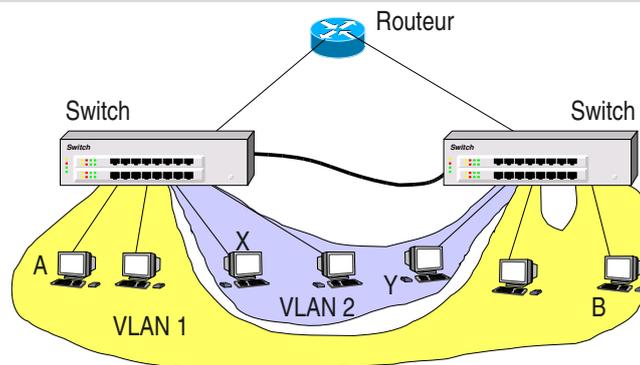
- ▶ Un VLAN est construit à l'aide des commutateurs dont on restreint les possibilités de commutation à des groupes de ports, la commutation peut être totale entre les membres d'un groupe mais devient impossible entre les membres de groupes différents

Un groupe définit un **domaine de broadcast**
domaine de broadcast \Leftrightarrow VLAN (plus petite commune définition)

- ▶ suivant les possibilités matérielles des switches, un VLAN peut être défini par **port**, par **adresse MAC**, par **adresse IP** (dans ce dernier cas, ce n'est toutefois pas assimilé à de la commutation niveau 3)

- ▶ un VLAN peut être réparti sur plusieurs commutateurs reliés entre eux
- ▶ routeur obligatoire entre VLANs, même s'il n'y a qu'un commutateur
- ▶ l'attribution d'un élément (port, adresse MAC, adresse IP) à un VLAN est réalisée par une opération de gestion
- ▶ l'interconnexion de switchs impose de marquer les trames sur les liens d'interconnexion afin de les associer à un VLAN.
Exemple figure suivante : les trames échangées entre A et B ou entre X et Y doivent être différenciées sur le lien entre les deux switchs. Solution : étiquetage des trames ; normes IEEE et solutions propriétaires

Les LANs virtuels : VLANs

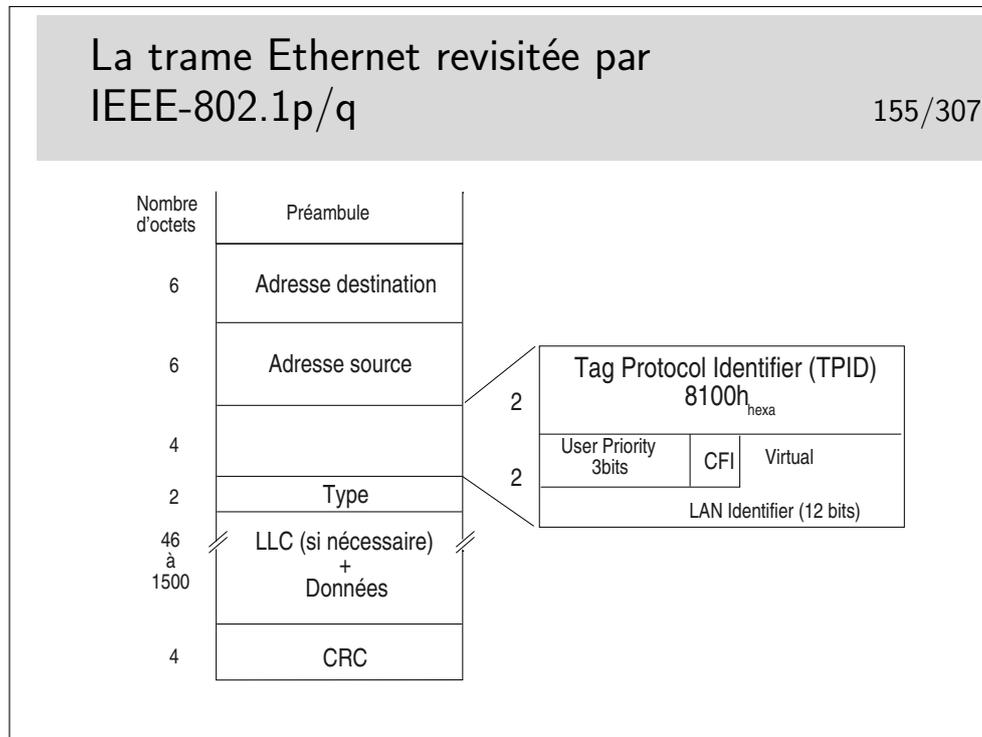


- ▶ Sur un commutateur ou plusieurs interconnectés
 - ▶ On subdivise l'espace d'interconnexion en sous espaces étanches
 - ▶ On construit ainsi une architecture logique sur une topologie physique
 - ▶ Standardisation : IEEE-802.1p et 1q

Imaginons que la machine A veuille émettre vers toutes les machines de son VLAN, elle doit émettre une trame de *broadcast* qui devra être relayée par le switch de gauche vers le switch de droite. Cependant, le switch de droite devra connaître l'identité du VLAN vers lequel diffuser la trame. Il ne faudra pas, par exemple, émettre vers la machine Y du VLAN2. Pour cela, sur le lien entre les switchs, la trame devra être complétée par un champ porteur de l'identité du VLAN.

Le transparent suivant indique comment est modifiée la trame ethernet de base.

Le lien d'interconnexion des commutateurs est parfois un «trunk». Selon les constructeurs et les types de switches, des ports spécifiques peuvent être réservés pour le «trunking».



Le champ TPID est tout simplement une valeur de champ Type spécifique. Le champ «User Priority» apporte une possibilité de privilégier certains flux.

Le bit CFI (Canonical Format Indicator), s'il est à 1, indique que le champ information comporte des indications de routage par la source.

Le champ «Virtual LAN Identifier» indique à quel VLAN appartient la trame.

- ▶ GARP : Generic Attribute Registration Protocol
 - ▶ mécanisme de signalisation permettant aux stations de fournir des indications (par des valeurs d'attributs) affectant les paramètres de filtrage des switches.
 - ▶ 3 attributs déjà définis :
 - ▶ groupe d'adresses MAC, facilite les mécanismes du multicast,
 - ▶ mode de filtrage des ports,
 - ▶ VLAN
- ▶ GMRP : Garp Multicast Registration Protocol
 - ▶ mécanisme permettant aux stations terminales et aux switches de s'enregistrer (et se retirer) comme participants à un groupe multicast et de diffuser cette information à l'ensemble de switches du réseau,
 - ▶ les switches utilisent cette information comme paramètre de filtrage,
 - ▶ utilise GARP comme protocole support.

6.4 Évolutions

- ▶ Années 80 : le 10Mb/s, 802.3, en bus sur câbles coaxiaux
- ▶ Années 90 :
 - ▶ 10Mb/s sur paires torsadées, raccordement sur hub
 - ▶ 100Mb/s
- ▶ Fin des années 90 : le 1000Mb/s
- ▶ Années 2000 : la 10Gb/s arrive
 - ▶ Support physique revisité, utilisation du support Sonet (OC192)
 - ▶ Distances visées : 40Km et plus
- ▶ Interopérabilité totale ascendante

Sonet (Synchronous Optical Network) est un type de multiplexage mis au point par la société américaine Bellcore et standardisée par l'ITU-T sous l'appellation SDH (Synchronous Digital Hierarchy).

À l'origine, Sonet/SDH est destiné au transport des voies téléphoniques numériques.

Les concepts mis en œuvre permettent un multiplexage et un démultiplexage facilité par rapport aux techniques plus anciennes. C'est le support privilégié pour les réseaux de type ATM.

Les débits courants sont de 155Mb/s (Sonet : OC3, SDH : STM1) sur cuivre (100m) ou fibre optique et 622Mb/s. Le débit de 1,2 Gb/s est possible. Un bond technologique est réalisé pour l'Ethernet 10Gb/s.

Le câblage cuivre, les différents types de câbles

159/307

- ▶ Le standard : ANSI/TIA/EIA 568 B
- ▶ Câbles à 4 paires torsadées, connecteurs RJ45
- ▶ Catégories
 - ▶ 5 : 100MHz, 100Mb/s sur 100m
 - ▶ 5e : 100 Mhz
 - ▶ 6 : 250MHz : 1Gb/s sur 100m
 - ▶ 7 : 600MHz : 10Gb/s sur 100m (15Gb/s sur 15m)
- ▶ Blindés ou écrantés
 - ▶ UTP : Unshielded Twisted Pair : non blindé
 - ▶ FTP : Foilded TP : écranté
 - ▶ STP : Shielded TP : blindé
 - ▶ SFTP : écranté et blindé

Un lien : <http://cablingdb.com/GlossaryPages/GlossaryC/CGlossary.asp>

6.5 Dénomination

Les dénominations des différentes versions d'Ethernet

I 161/307

- ▶ Format : x BASE/BROAD y
 - ▶ x : le débit
 - ▶ BASE : bande de base, BROAD : large bande
 - ▶ y : indication sur la topologie (et/ou la longueur du câble)
- ▶ Bande de base :
 - ▶ Le 10 Mb/s
 - ▶ 10 BASE-5 : 10 Mb/s, topologie en bus constitué de segments de 500m
 - ▶ 10 BASE-2 : 10Mb/s, topologie en bus constitué de segments de 185m
 - ▶ 10 BASE-T : 10Mb/s, topologie en étoile, câbles en paires torsadées (T pour *Twisted pair*), longueur 100m

Les matériels à 10Mb/s tendent à disparaître aujourd'hui (2004). Les cartes interfaces sont toutes en 100-BASE-T, même en premier prix.

Les dénominations des différentes versions d'Ethernet

II 162/307

- ▶ Le 100Mb/s
 - ▶ 100BASE-T4 : 4 paires utilisées, catégorie 3 à 5
 - ▶ 100BASE-TX : 2 paires torsadée, catégorie 5
 - ▶ 100BASE-FX : 2 fibre optique
- ▶ Le 1000Mb/s
 - ▶ 1000BASE-LX : fibre optique, grande (Long) longueur d'onde
 - ▶ 1000BASE-SX : fibre optique, courte (Short) longueur d'onde
 - ▶ 1000BASE-CX : paire torsadée, 25m max
 - ▶ 1000BASE-TX : 4 paires torsadées de catégorie 5

6.6 Câblage

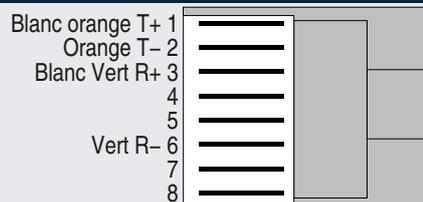
Les câblages : les câbles catégorie 5 et leurs connecteurs

164/307

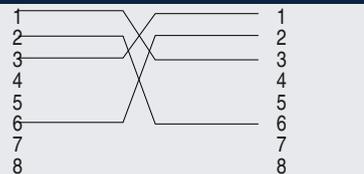
- ▶ Les câbles sont classés par catégories en fonction de leurs caractéristiques (en particulier les débits maximum)
- ▶ Aujourd'hui on recommande la catégorie 5 voire 5e ou 6 pour des câblages cuivre permettant d'atteindre le 100Mb/s

Les câbles pour le 10BASE-T et 100BASE-T (EIA/TIA 568 A)

La prise RJ45



Câble croisé (interconnexion de machines hubs switches)



Un document bien illustré sur le câblage... En italien, mais Mhz et nm et Km s'écrivent comme en français... <http://www.garr.it/ws4/pdf/Montessoro.pdf>

Un câble croisé permettra de raccorder :

- deux machines directement entre-elles
- deux hubs ou deux switches entre-eux
- un hub et un switch

Les hubs et les switches peuvent être munis de ports directement croisés, on les repère généralement par la mention «Up Link». Un port peut aussi avoir deux prises RJ45, une droite et une croisée «up-link».

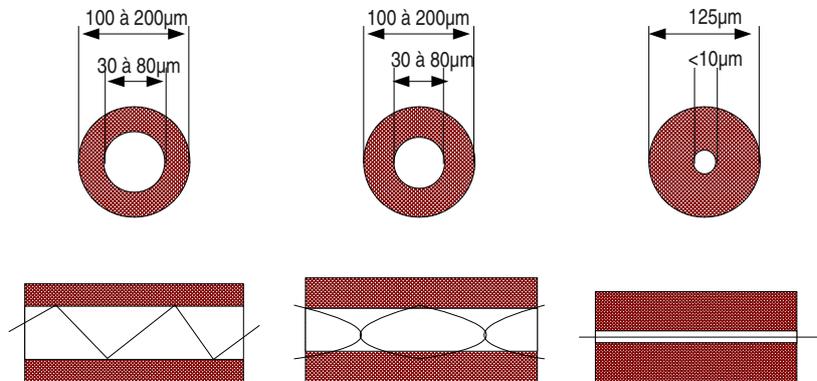
Un port peut être muni d'un petit commutateur appelé **MDI-MDIX** permettant de positionner le port en mode normal (MDI) ou en mode croisé (MDIX).

Et enfin, la fonction MDI-MDIX peut être à auto-détection : le switch détecte automatiquement s'il a à faire à un câble droit ou croisé.

Multimode

Gradient d'indice

Monomode



La fibre et Ethernet gigabit

Transceiver	Fiber Diameter (microns)	Bandwidth (MHz*km)	Minimum Range (meters)
1000BASE-SX	MM 62.5	160	2-220
	MM 62.5	200	2-275
	MM 50	400	2-500
	MM 50	500	2-550
1000BASE-LX	MM 62.5	500	2-550
	MM 50	400	2-550
	MM 50	500	2-550
	SM 9	NA	2-5000

Transceiver	Fiber Diameter (microns)	Wavelength (nm)	Minimum Range (meters)
1000BASE-LH (Extended distance)	SM 9	1310	1 km - 49 km
1000BASE-LH (Extended distance)	SM 9	1550	50 km - 100 km

- ▶ 1000 BASE-SX : (S pour Short), $\lambda = 850nm$, codage en ligne 8B/10B
- ▶ 1000 BASE-LX : (L pour long), $1270nm < \lambda < 1355nm$, codage en ligne 8B/10B
- ▶ 1000 BASE-LH (Long Haul), fibre mono mode (SM : Single Mode), 9μ

Interopérabilité des débits

167/307

- ▶ Les hubs et les switches acceptent aujourd'hui les deux débits de base 10Mb/s et 100Mb/s
- ▶ Les switches comportant des ports à 1Gb/s se généralisent
 - ▶ Un mécanisme d'autonégociation entre les extrémités dce la liaison permettent de détecter le débit maximal accepté (10 ou 100, 1000)
 - ▶ L'autonégociation permet aussi de détecter la possibilité du full-duplex et de contrôle de flux

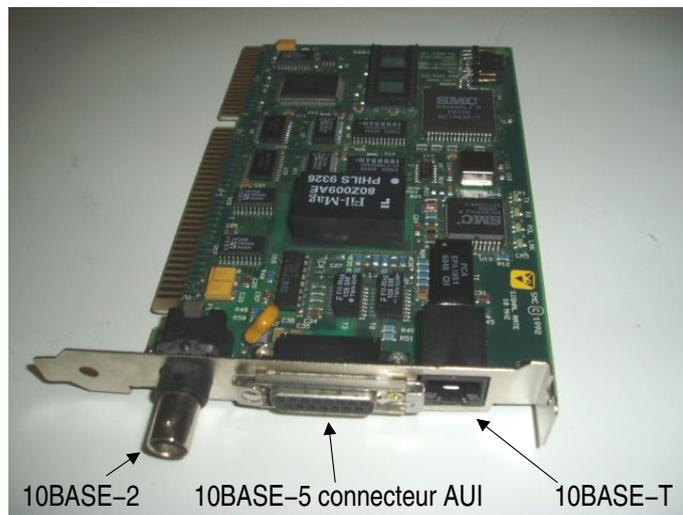
La fonction contrôle de flux

168/307

- ▶ Disponible sur certains commutateurs
- ▶ Trames «pause»
 - ▶ Adresse multicast 01-80-c2-00-00-01
 - ▶ Type 0x8808
 - ▶ Information : sur deux octets (<65536), contient le nombre de «pause_times» (périodes de 512 bits)
 - ▶ Indiquent la durée pendant laquelle l'extrémité ne désire pas recevoir de nouvelle trame

Une carte interface «multimedium»

169/307



Technologie des débuts 90.

Aujourd'hui on ne trouve plus que du 10/100/1000 BASE-T, à quelques Euros...

Et en plus il y a beaucoup moins de composants...

Câbles et connecteurs – Cuivre et fibre

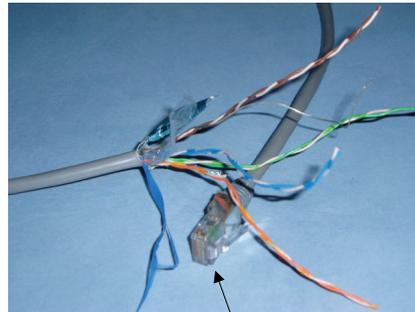
170/307

Fibre optique



Connecteurs SC et TS

Paires torsadées

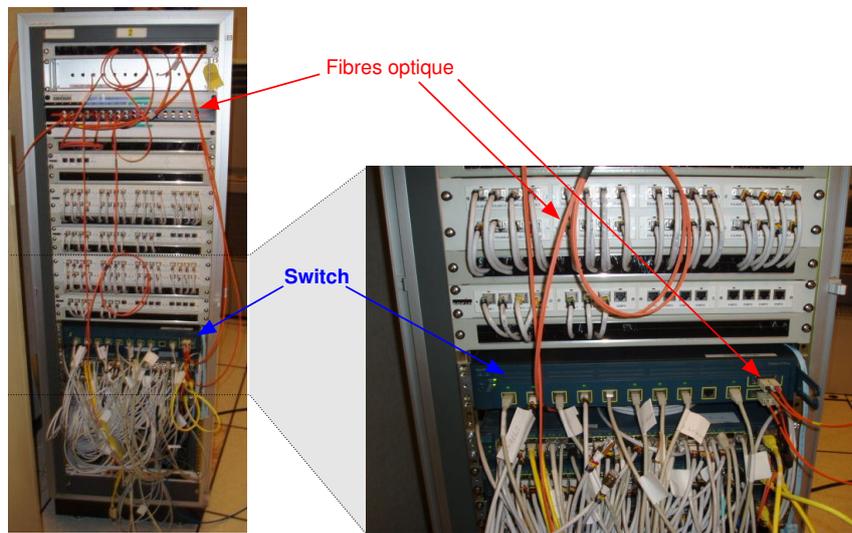


Connecteur RJ45



Une baie de brassage de câbles

171/307



L'envers du décor :



- ▶ Les commutateurs de bas de gamme ne s'administrent pas
- ▶ On peut se connecter sur les autres
 - ▶ Par port série dédié (et hyper-terminal sous Windows ou minicom sous linux)
 - ▶ Par telnet sur une adresse IP configurée dans l'appareil
 - ▶ Par page Web sur une adresse IP configurée dans l'appareil
 - ▶ Par SNMP
- ▶ Les actions d'administration peuvent être
 - ▶ Mettre en place ou inhiber la fonction spanning-tree et gérer cette fonction (priorités du switch, QoS, ...)
 - ▶ Mettre en place et gérer des VLANs
 - ▶ Monitorer des ports
 - ▶ Surveiller les adresses MACs enregistrées
 - ▶ Surveiller le trafic

La fonction «monitoring» consiste à rediriger le trafic normal d'un port vers un port dédié à la fonction et sur lequel on place une machine munie d'un analyseur de flux. Toutes les trames entrantes et sortantes du port surveillé sont recopiées sur le port de monitoring.

7 Les autres techniques autour des LANs

7.1 Le sans fil IEEE-802.11 (WiFi : Wireless Fidelity)

IEEE-802.11 - WiFi	175/307
<ul style="list-style-type: none">▶ Canal radio à 2,4GHz (5GHz pour 802.11a)<ul style="list-style-type: none">▶ 11b : jusqu'à 11Mb/s – 11g : jusqu'à 54 Mb/s▶ Distance relativement courte : une centaine de mètres dans de bonnes conditions de propagation, ce qui est rarement le cas en intérieur▶ Méthode d'accès au médium : CSMA/CA<ul style="list-style-type: none">▶ Les collisions sont évitées par un mécanisme de délai avant d'émettre▶ Contrôle d'accès<ul style="list-style-type: none">▶ Sécurité possible avec chiffrement : Wired Equivalent Privacy (Wep)▶ Type d'application :<ul style="list-style-type: none">▶ raccordement de mobiles à des réseaux de type LAN dans les entreprises et lieux publics	

Un tutorial correct :

<http://www.intelligraphics.com/articles/80211.article.html>

Un lien sur la sécurité en 802.11 :

<http://www.iss.net/wireless/WLAN.FAQ.php>

Autre lien intéressant :

http://wireless.ictp.trieste.it/school.2002/lectures/ermanno/HTML/802.11_Architecture.pdf#search='802%2011%20ssid'

7.2 Les courants porteurs

Technique Courant Porteur	177/307
<ul style="list-style-type: none">▶ CPL : courant porteur en ligne<ul style="list-style-type: none">▶ Le réseau d'alimentation électrique est le medium▶ Interfaçage simple avec Ethernet ou USB : le câble catégorie 5/RJ45 est relié à une prise spéciale enfilant une prise de courant▶ Débits annoncés : 5-20 Mb/s<ul style="list-style-type: none">▶ Des produits arrivent annonçant 200Mb/s (théorique)!▶ Modulation en ligne OFDM (certains utilisent CDMA)▶ Solutions propriétaires, pas de compatibilité entre produits<ul style="list-style-type: none">▶ Pourtant un standard : Home Plug▶ Des travaux à l'ETSI▶ Applications : «émulation Ethernet», raccordement de quartiers résidentiels, d'entreprises, etc...	

Un lien intéressant <http://vlan.org/breve123.html>

7.3 Autres

Autres techniques Réseau	179/307
<ul style="list-style-type: none">▶ Les PANs : Personal Area Network<ul style="list-style-type: none">▶ Très courtes distances▶ Typiquement :  Bluetooth  ZigBee<ul style="list-style-type: none">▶ Interconnexion de petits portables (téléphones, PDAs, etc.), domotique (capteurs, actionneurs)▶ Standardisation en cours : IEEE-802.15▶ Les bus spécialisés<ul style="list-style-type: none">▶ USB▶ IEEE-1394  (FireWire ou iLink selon les constructeurs)<ul style="list-style-type: none">▶ Raccordement d'appareils multimédia (caméra vidéo numériques, audio)▶ Canaux synchrones et asynchrones▶ 800Mb/s▶ Distance : 100m annoncée entre nœuds et hubs (4,5m en standard)▶ En standard sur les PCs aujourd'hui	

Les canaux synchrones de IEEE-1394 (FireWire/iLink) sont adaptés à la transmission de flux sons et images.

Ces réseaux utilisent des protocoles spécifiques pour les couches supérieures, par exemple AMDTP (Audio and Music Data Transmission Protocol. IEC61883-6) pour IEEE-1394. IP n'est pas implémenté en standard sur ces réseaux.

Une émulation Ethernet existe pour IEEE-1394 en utilisant les canaux asynchrone.

FireWire : IEEE-1394 chez Apple.

ilink chez Intel

7.4 La couche LLC

La couche LLC – IEEE-802.2

I

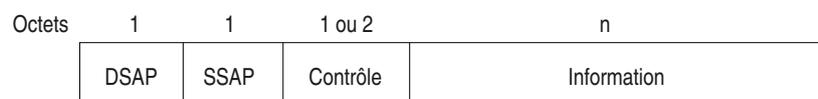
181/307

- ▶ Permet de combler les manques de la couche MAC par rapport au niveau 2 OSI standard (si nécessaire)
- ▶ Permet d'indiquer le SAP du protocole véhiculé dans les données utiles de la trame MAC

► Trois types LLC

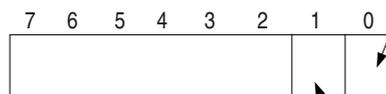
- LLC type 1 : mode datagramme, sert uniquement à véhiculer le SAP des données utiles
- LLC type 2 : mode connecté, en plus de la fonctionnalité du type 1 on assure des contrôles identiques au HDLC LAPB (avec numérotation des trames modulo 128 : champ contrôle sur 2 octets), sert à véhiculer des paquets X25 par exemple
- LLC type 3 : mode datagramme avec acquittement. Prévu pour les réseaux industriels

Format du PDU LLC



Codage des DSAP et SSAP

Bits



SSAP : si 1, indique une trame de commande ou réponse

DSAP : 0 → SSAP unique
1 → SSAP de groupe

0 : SAP local attribué par le gestionnaire du réseau

1 : SAP global attribué par un organisme officiel

— DSAP : Destination Service Access Point

— SSAP : Source Service Access Point (i.e. *comment* interpréter le champ contrôle)

Et donc ... il reste combien de SAP possibles, donc de protocoles identifiables ?

64! C'est bien peu et c'est un problème...

LLC : quelques SAP

184/307

- ▶ 0x06 : IP
- ▶ 0x42 : Spanning Tree
- ▶ 0x7E : X25
- ▶ 0xE0 : Novell IPX
- ▶ 0xAA : SNAP (voir plus loin)

La notation 0x... est celle du langage C ou Java pour les nombres en hexadécimal.
Le SAP 0x7E ne vous rappelle rien concernant X25 et surtout sa couche 2 standard ?

LLC : le champ contrôle

185/307

- ▶ LLC type 1
 - ▶ Le champ contrôle est codé sur un octet et vaut 0x03
- ▶ LLC type 2
 - ▶ Format identique au champ contrôle (ou commande) de HDLC LAPB
 - ▶ Trames numérotées modulo 128
 - ▶ Longueur du champ : 2 octets pour les rames I, RR, RNR et REJ
 - ▶ La trame d'établissement de la connexion s'appelle SABME (Set Asynchronous Balanced Mode *Extended*)

- ▶ SNAP : Sub Network Access Protocol
- ▶ Les champs DSAP et SSAP standards sont trop courts
 - ▶ Le champ type de la trame Ethernet pur est très bien...
 - ▶ Si on le réutilisait... En le plaçant après le champ contrôle ?
 - ▶ Oui mais sa longueur n'est que de deux octets !
 - ▶ Le tout ferait donc 5 octets (Contrôle sur 1 octet), ce n'est pas optimum pour une architecture 32 bits (l'architecture de la plupart de nos machines encore pour l'instant)
 - ▶ Et si on complétait à 8 octets en rajoutant l'OUI du concepteur du protocole ?

SNAP : même traduit un français, ça ne veut rien dire (humour (?!!!))

OUI : Organizational Unit Identifier

Octets					
1	1	1	3	2	n
DSAP 0xAA	SSAP 0xAA	Contrôle 0x03	OUI	Type	Données

- ▶ DSAP = SSAP = 0xAA
- ▶ Contrôle = 0x03 (trame UI : Unnumbered Information)
- ▶ OUI : code attribué par l'IEEE à la société créatrice du protocole. Identique aux 3 octets de début des adresses Ethernet. Souvent à 0
- ▶ Type : identifie le protocole : identique au champ Type de la trame Ethernet

Mais non!... Ce n'est pas si compliqué!...

Et en plus tout peut être justifié... Il y a une bonne raison pour qu'il en soit ainsi. Simplement il faut chercher cette raison parfois profondément...

Exemples de trames LLC

188/307

```
0: 0900 07ff ffff 0040 9c00 0294 0022 aaaa .....@....."..
16: 0308 0007 809b 001a 0000 0000 03e8 ffe1 .....
32: 0101 0103 e808 e103 e880 03e8 8201 9400 .....

0: 0180 c200 0000 0000 1d07 2c63 0026 4242 .....,c.&BB
16: 0300 0000 8000 0000 0000 0000 0000 0000 .....
32: 0000 0000 0000 00c0 0f00 0050 0000 0000 .....P....
48: 0000 0000 ....
```

Décodons... Et répondons aux questions :

- pourquoi peut-on dire que ces trames sont de type multicast ?
- que nous enseignent les adresses destination utilisées dans ces trames ?
- pourquoi peut-on dire qu'elles sont au format 802.3 et non Ethernet pur ?
- quels sont les protocoles véhiculés ?

Troisième partie

Le protocole IP et les protocoles associés

8 Standardisation, modélisation

L'organisation de l'Internet	191/307
<ul style="list-style-type: none">▶ Organisme de tutelle : l'ISOC (Internet SOCIety), qui regroupe<ul style="list-style-type: none">▶ Internet Architecture Board (IAB)<ul style="list-style-type: none">▶ Responsable final des travaux de l'IETF▶ Internet Engineering Steering Group (IESG)<ul style="list-style-type: none">▶ Contrôle les travaux de l'IETF▶ Internet Engineering Task Force (IETF)<ul style="list-style-type: none">▶ L'organisme de standardisation▶ Internet Research Task Force (IRTF)<ul style="list-style-type: none">▶ Responsable de la recherche à long terme▶ Internet Assigned Numbers Authority (IANA)<ul style="list-style-type: none">▶ 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 responsabilités et celles de IANA qui continue d'exister.

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

- ▶ 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)

- ▶ 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 renvoi 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
194/307

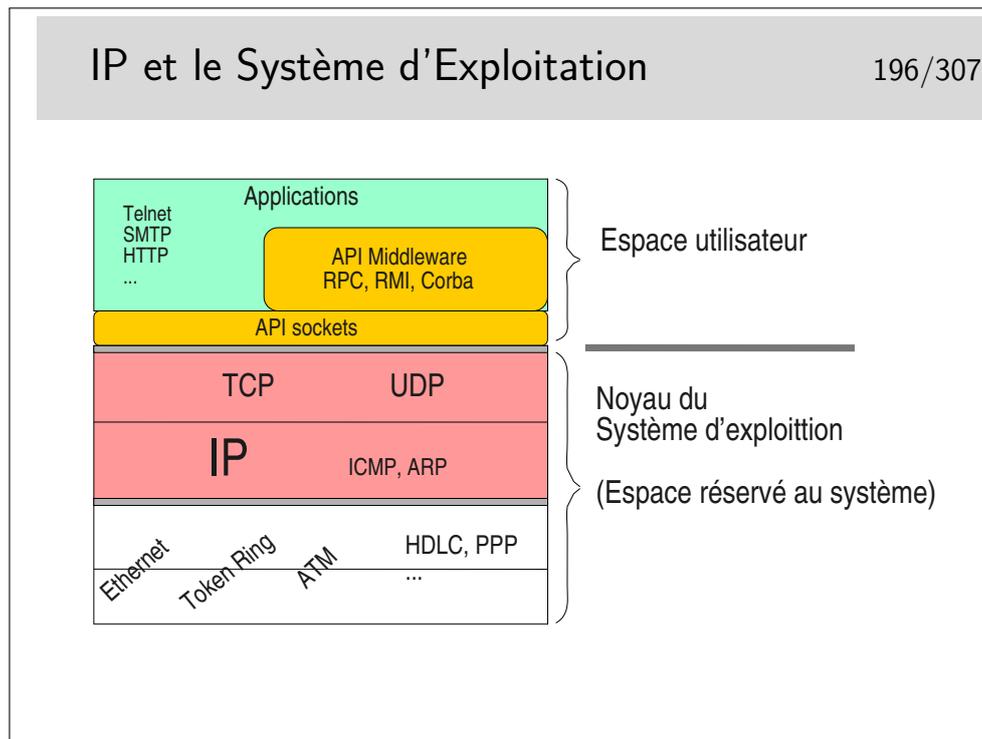
- ▶ Certains RFCs standards sont classées sous l'appellation std, le std 5 par exemple correspond au rfc791 définissant IP
- ▶ Certains sont classées sous l'appellation FYI : For Your Information
- ▶ Quelques RFCs :
 - ▶ 791 : IP (std5)
 - ▶ 793 : TCP (std7)
 - ▶ 763 : UDP (std6)

IP et le modèle OSI
195/307

Application	<div style="display: flex; justify-content: space-between;"> <div style="width: 30%;">Telnet SMTP HTTP ...</div> <div style="width: 60%; text-align: center;">Applications</div> </div>
Présentation	<div style="display: flex; justify-content: space-between;"> <div style="width: 30%;"></div> <div style="width: 60%; text-align: center;">API Middleware RPC, RMI, Corba</div> </div>
Session	<div style="text-align: center;">API sockets, ...</div>
Transport	<div style="display: flex; justify-content: space-around;"> TCP UDP </div>
Réseau	<div style="display: flex; justify-content: space-around;"> IP ICMP, ARP </div>
Liaison	<div style="display: flex; justify-content: space-between;"> Ethernet Token Ring ATM HDLC, PPP </div>
Physique	<div style="display: flex; justify-content: space-between;"> Ethernet Token Ring ATM ... </div>

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 incluses dans le noyau. Les applications (les programmes des utilisateurs) y accèdent via des interfaces qui les masquent.

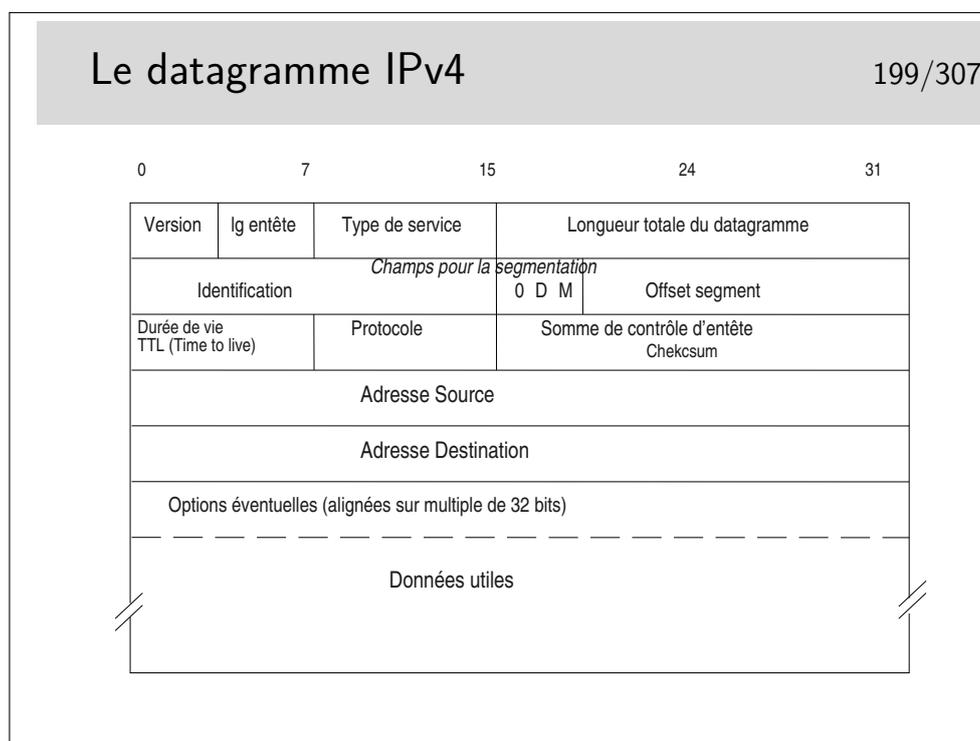
Ces interfaces sont constituées par des bibliothèques de fonctions formant des APIs utilisables par les programmeurs d'applications. La plus répandue est la bibliothèque socket tout à fait bien adaptée aux protocoles Internet (TCP, UDP, IP). Les couches protocolaires sont vues comme des fichiers qu'on 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.

9 Le datagramme IP

9.1 Le datagramme IPv4



Longueur d'entête : en nombre de mots de 32 bits ; si supérieur à 5, indique la présence d'options (40 octets max d'options : 10 mots max de 4 octets)

Type de service : peu utilisé par le passé, permet aujourd'hui, entre autres, de coder le DSCP (DiffServ Code Point). Voir pages suivantes.

Champs pour la segmentation : on recommande d'éviter la segmentation avec TCP. Elle n'existe plus que pour UDP si les unités de données fournies ont une taille supérieure au MTU de l'interface

bit D : *Don't fragment* : interdit la fragmentation du segment s'il est à 1

bit M : *More* : indique la présence de fragments complémentaires, si le segment d'origine a été fragmenté.

Si le champ Offset est à 0 ainsi que le bit M, le segment est celui d'origine

TTL : durée de vie; décrémente de 1 par chaque routeur. Si le résultat passe à 0, le datagramme est jeté et un message ICMP est envoyé à la machine émettrice

Protocole : indique le protocole supérieur véhiculé (TCP, UDP, ICMP voire même IP en cas d'encapsulation IP dans IP pour tunnelling)

Somme de contrôle d'entête : contient la somme des mots de 16 bits constituant l'entête. Permet à l'arrivée de vérifier l'intégrité de celle-ci. Si une erreur est détectée le datagramme est jeté sans autre forme d'action. On ne peut même pas envoyer un message d'erreur à l'émetteur car l'adresse source qui indique ce dernier peut être corrompue. Comme l'indique le nom de ce champ le contrôle d'intégrité n'est fait que sur l'entête.

Adresses : source et destination, sur 32 bits; information fondamentale, au moins en ce qui concerne la destination car elle sert au routage.

- ▶ Encore appelé ToS (*Type of Service*) *rfc1349*

Priorité (précédence) Par défaut à 0	Type de service -délai +débit +fiabilité -coût				0
--	---	--	--	--	---

- ▶ Utilisation non généralisée
 - ▶ Précédence utilisable pour marquer des flux dans les nœuds du réseau (routeurs)
 - ▶ Les bits Type de Service peuvent être positionnés par les applications terminales (API socket sous Windows et linux)
 - ▶ Sous Linux ils peuvent être positionnés via la commande iptables selon divers critères

111 - Network Control (protocole de type HELLO)

110 - Internetwork Control (protocoles de routage)

101 - CRITIC/ECP

100 - Flash Override

011 - Flash

010 - Immediate

001 - Priority

000 - Routine (priorité la plus faible)

RFC 791 et 795

Le champ Type de Service en version DiffServ

201/307

DSCP						0	0
1	2	3	4	5	6	7	8

- ▶ Le champ DSCP sert à coder le *PHB (Per Hop Behavior)*, paramètre fondamental de DiffServ (rfc 2474)
- ▶ Exemples de PHBs :
 - ▶ Expedited forwarding (pertes faibles, latence faible, gigue faible, bande passante assurée)
 - ▶ Assured forwarding (groupe de PHBs)
 - ▶ Best effort
 - ▶ Network control

DiffServ signifie Différentiation de Service (en anglais aussi). C'est un mécanisme relativement récent permettant de marquer les paquets afin qu'ils soient traités de manière différenciée dans les routeurs. C'est un mécanisme de qualité de service.

Les options IPv4

202/307

- ▶ Format :
- ▶ Les options :
 - ▶ LSR : *Loose Source Route* : permet d'indiquer la route
 - ▶ SSR : *Strict Source Route*, comme précédemment en plus rigoureux
 - ▶ RR : *Record Route* : les routeurs traversés rajoutent leur adresse
 - ▶ RTALT : *Router Alert*, permet de passer le paquets aux couches hautes des routeurs traversés
 - ▶ etc.

L'option LSR permet d'indiquer la route à suivre mais si un routeur intermédiaire ne sait pas comment utiliser une des indications contenue dans l'option, il peut alors utiliser sa table de routage normale.

L'option SSR est plus stricte car le paquet est rejeté si un routeur ne sait pas l'orienter

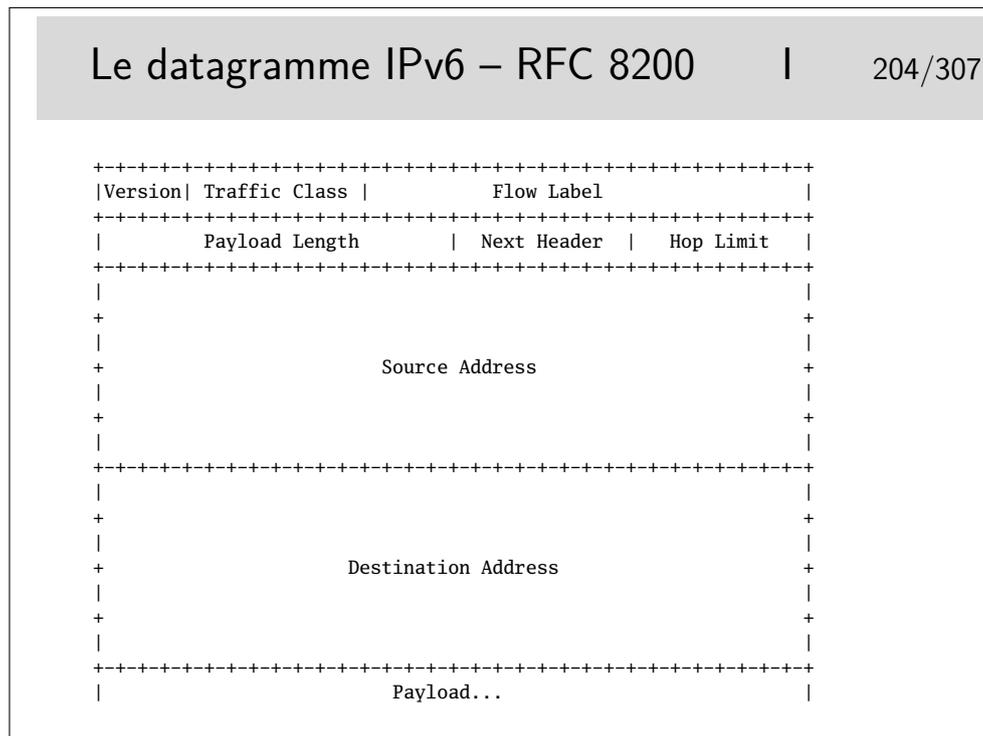
vers une direction indiquée.

Il y a encore l'option Traceroute, obsolète car dangereuse, EEOL et NOP pour indiquer la fin de la liste et du bourrage d'alignement sur 32 bits.

L'option RTALT est utilisée par IGMP et par RSVP, ces protocoles nécessitent des traitements particuliers dans les routeurs. Ces traitements sont effectués par des entités applicatives des routeurs. Ces derniers ne doivent donc pas effectuer le routage rapide des paquets munis de cette options mais ils doivent les diriger vers les entités applicatives.

<http://www.iana.org/assignments/ip-parameters>

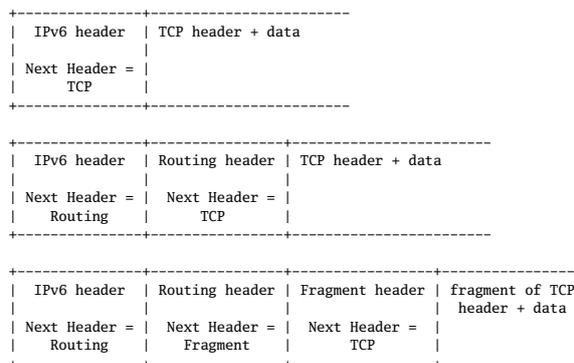
9.2 Le datagramme IPv6



- ▶ Version : 6 (!)
- ▶ Traffic Class : typiquement Differentiated Services et Explicit Congestion Notification, comme le TOS IPv4
- ▶ Flow Label : pour tagger les paquets d'un même flux; visiblement peu utilisé...
- ▶ Payload Length : comme son nom l'indique... (Note : dans IPv4 on a la taille *totale* du datagramme)
- ▶ Next Header : remplace le champ Protocole IPv4 et ajoute un mécanisme un peu novateur pour gérer les options (cf. slides suivants)
- ▶ Hop Limit : l'équivalent du champ TTL IPv4 (Note : contrairement à IPv4 qui compte en secondes, ici c'est bien en nombre de sauts)
- ▶ Source & Destination **Addresses** : sur **128 bits** (Ouf!)

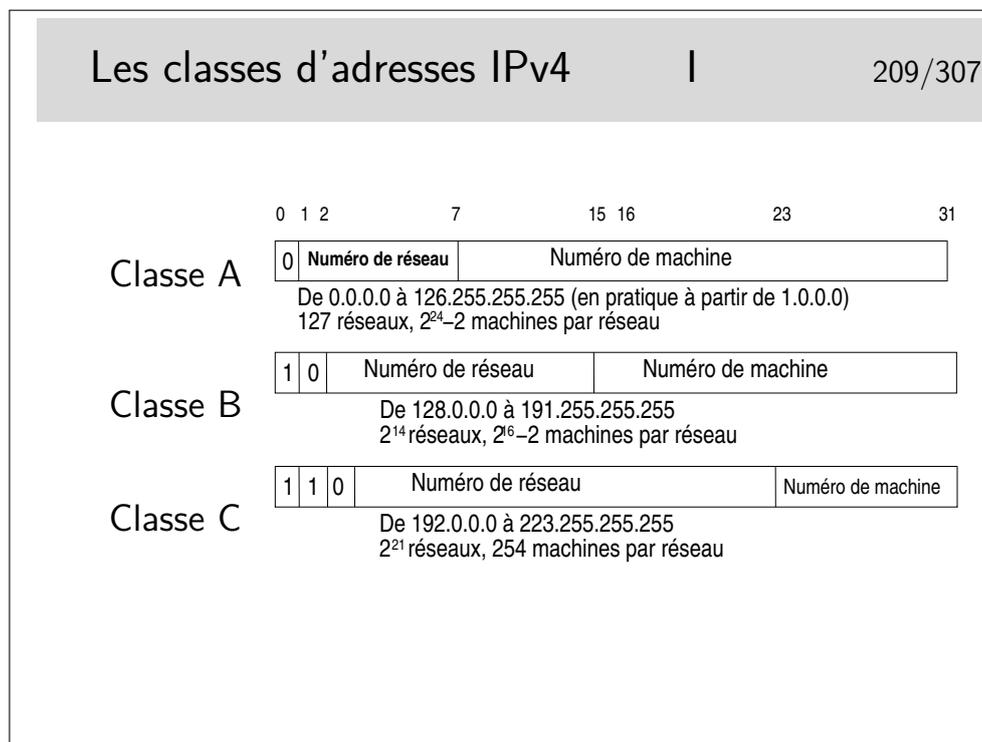
Next Header

- ▶ Les headers s'enchaînent dans le payload
- ▶ ...les options IPv6 diverses, puis le protocole transporté effectivement (TCP, UDP, ...)



10 Adressage

10.1 Les adresses IPv4

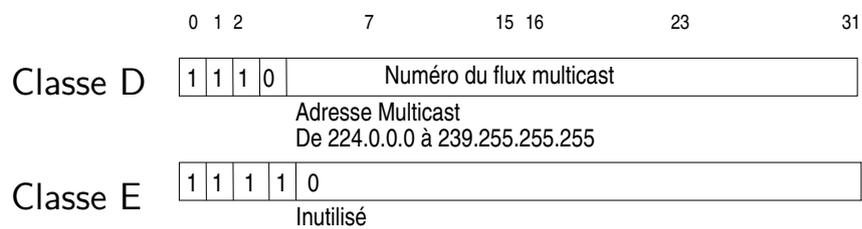


Une interface IP doit être munie d'une adresse pour pouvoir fonctionner. En général, pour des machines de bureau standard en réseau local, l'interface IP correspond à une carte réseau de type Ethernet.

L'interface IP peut aussi être correspondre à une couche complètement logicielle, c'est le cas des interfaces de niveau 2 réalisées avec le protocole PPP qui s'interpose entre IP et une interface physique comme un port série ou un port USB relié à un modem ADSL (l'empilement des couches protocolaires est alors très complexe).

Une interface matérielle peut être munie de plusieurs adresses IP, sous Unix et Linux en particulier. On a alors, par exemple, les interfaces `eth0 :0`, `eth0 :1`, `eth0 :2`, etc.

A noter que les adresses de classe A de la gamme 0.0.0.0 à 0.255.255.255 ne peuvent pas être utilisées, bien qu'elles ne correspondent pas à une fonction particulière.



Le multicast...

Permet d'envoyer des paquets IP à un certain nombre de machines mais pas à toutes. Des machines sur le réseau peuvent être sources de trafic multicast, elles peuvent émettre par exemple à l'adresse 224.5.6.7. Il suffit que d'autres machines soient «au courant» de ce fait pour se mettre en écoute en lançant une application spécifique. Cette application demande explicitement à écouter le flux en envoyant un message du protocole IGMP (RFC-3376 pour la version 3) vers l'Internet. Ce message est relayé par les routeurs vers les sources, chaque routeur rencontré s'insère alors dans un arbre de diffusion multicast (préalablement configuré). Les paquets du flux applicatif peuvent arriver ainsi aux récepteurs ayant fait la requête.

Il faut que les routeurs soient configurés, ce n'est pas automatique.

Il existe un réseau mondial multicast, appelé le MBONE. Les sources potentielles diffusent des annonces de session à l'adresse SAP.MCAST.NET (224.2.127.254) (en UDP, port 9875).

L'outil sdr (pour unix/linux ou windows) permet d'écouter ce flux et afficher les informations de sessions.

(mot clé MBONE; un lien : <http://www-itg.lbl.gov/mbone/>)

- ▶ L'adresse de boucle locale : 127.0.0.1
 - ▶ Interface lo sous Linux
- ▶ Les adresses privées : rfc-1918
 - ▶ 10.0.0.0 à 10.255.255.255
 - ▶ 172.16.0.0 à 172.16.255.255
 - ▶ 192.168.0.0. à 192.168.255.255
 - ▶ Non routables dans l'Internet
 - ▶ Les machines munies de ces adresses peuvent cependant accéder l'Internet via des passerelles réalisant une fonction de translation d'adresse appelée **NAT** pour **Network Address Translation**
 - ▶ Routables dans les réseaux privés

La fonction de NAT est mise en œuvre dans un routeur muni d'une interface configurée avec une adresse officielle (pouvant donc être routée dans l'Internet). Les paquets sortant sont modifiés, le champ «adresse source» est remplacé par l'adresse officielle de l'interface de sortie du routeur. Ce dernier mémorise l'opération pour effectuer la modification inverse pour les paquets en retour.

Des extensions du concept peuvent affecter aussi les ports TCP ou UDP.

Le mécanisme pose un problème pour les protocoles tels que ftp. En effet, ftp demande la création d'une connexion «entrante» pour réaliser les transferts de fichiers, les paquets de demande de connexion doivent être corrélés avec la connexion sortante existante. Les traitements sont alors plus complexes que pour les connexions simples comme celle pour le Web par exemple). Ces concepts demandent de comprendre ce qu'est une connexion au niveau supérieur (il n'y a pas de connexion IP). Ce point sera abordé plus loin dans le chapitre sur TCP.

Sous linux, les mécanismes de translation d'adresse sont directement intégrés au noyau. Une commande permet d'en paramétrer les caractéristiques, il s'agit de iptables qui peut bien plus encore en ce qui concerne les opérations de filtrage pour la sécurité.

Les adresses particulières pour la diffusion 212/307

Différentes possibilités :

- ▶ *Broadcast* sur le réseau local : 255.255.255.255
 - ▶ Peu utilisée
- ▶ Partie réseau normale, partie machine à 255
 - ▶ Généralement l'adresse de broadcast mise en œuvre
 - ▶ Exemples :
 - ▶ 192.168.100.255
 - ▶ 172.16.255.255
 - ▶ Partie réseau normale, partie machine à 0
 - ▶ Ancienne adresse de broadcast pouvant encore être utilisées sur des machines SUN dont l'OS est SUNOS-4

Les adresses de réseaux 213/307

- ▶ Les réseaux avec un netmask standard
 - ▶ On indique les 4 octets (entre 0 et 255), comme pour une adresse normale, les derniers octets étant à 0 (le dernier pour une classe C, les deux derniers pour la classe B, les trois derniers pour la classe A)
 - ▶ Exemple : 192.168.100.0
- ▶ Les réseaux «subnettés»
 - ▶ On fait figurer tous les octets (entre 0 et 255), y compris les bits de l'extension
 - ▶ Exemple : 192.168.100.32 (netmask 255.255.255.224 par exemple)
 - ▶ Voir page suivante

- ▶ Extension de la partie «Réseau» de l'adresse en empruntant quelques bits de poids forts de la partie machine
- ▶ Par exemple 255.255.255.224 pour une classe C
 - ▶ Tout à 1 sauf la partie machine (224d = 1110 0000b)
- ▶ Le masque indique quels bits
- ▶ Permet de créer des «sous» réseaux
 - ▶ Les sous réseaux sont raccordés entre eux via des routeurs, comme des réseaux «normaux»
- ▶ Il y a toujours un netmask
 - ▶ Il est standard s'il ne comporte pas d'extension de bits par rapport à la classe d'adresses : 255.255.255.0 pour une classe C par exemple

Pardon pour l'Académie, mais «subnetting» est plus joli que «sous-réseautage»

Avec le masque 255.255.255.224 on prend 3 bits de la partie machine. On pourra créer 8 sous réseaux à partir du numéro 192.168.100.0 :

- 192.168.100.0
- 192.168.100.32
- 192.168.100.64
- 192.168.100.96
- 192.168.100.128
- 192.168.100.160
- 192.168.100.192
- 192.168.100.224

Question : à quel sous réseau appartient la machine de numéro 192.168.100.72 ?

- ▶ Les adresses sans classe
 - ▶ Concept CIDR (RFC-1519) : *Classless InterDomain Routing*
 - ▶ La frontière de l'adresse réseau n'est plus figée selon la loi des classes
 - ▶ Utile pour agréger des routes dans les tables de routage des routeurs
 - ▶ Utile pour les fournisseurs de service pour affecter un sous ensemble d'adresses à un client...
 - ▶ Le netmask doit être précisé avec les adresses
- ▶ Notation
 - ▶ Classique : 255.255.255.128 (25 bits de masque)
 - ▶ Notation CIDR : /25 : exemple 192.168.100.128/25

Dans le dernier exemple, l'adresse donnée est une adresse de réseau et non une adresse de machine. Avec une telle adresse (et son masque) on pourra adresser deux fois 128-2 machines : de 192.168.100.1 à 192.168.100.126 (broadcast 192.168.100.127) et de 192.168.100.129 à 192.168.100.254 (broadcast 192.168.100.255).

La notation classique (255.255...) reste très répandue. La notation CIDR peut se rencontrer sous quelques Unix (BSD, Linux, ???), elles est plus commode d'utilisation.

- ▶ «À la main»
 - ▶ Selon les outils offerts par le système d'exploitation
 - ▶ À l'aide d'interfaces graphiques d'administration
 - ▶ Via des commandes spécifiques
- ▶ Dynamiquement
 - ▶ Via le protocole DHCP (*Dynamic Host Control Protocol*)
 - ▶ Un serveur est configuré pour donner l'information
 - ▶ Sur connexion via liaison point à point et le protocole PPP (Point to Point Protocol). La machine à configurer contacte un serveur situé à l'autre extrémité de la liaison. Le serveur peut lui fournir son adresse

Le protocole PPP est utilisé par exemple dans les connexions aux fournisseurs de

service IP via des modems et le réseaux téléphonique général.

Dans les connexions via ADSL, c'est aussi PPP qui est utilisé pour affecter l'adresse à la machine qui se connecte mais le mécanisme est beaucoup plus complexe.

Le protocole DHCP permet de fournir une adresse IP, le netmask associé, le routeur par défaut pour l'interface en cours de configuration, le nom de domaine DNS ainsi que le ou les serveurs DNS. Tout ceci pour une période de temps configurée par le gestionnaire du serveur DNS. Les clients doivent renouveler leur demande d'adresse à la fin du temps alloué.

10.2 Les adresses IPv6

Adresses IPv6 – RFC 4291

218/307

- ▶ Adresses sur 128 bits
- ▶ Notation hexa par bloc de 16 bits
2001:db8:cafe:deca:0:0:0:1
- ▶ Compression de zéros
2001:db8:cafe:deca::1

- ▶ Deux parties :
 - ▶ **préfixe** ou **identifiant de sous-réseau** (subnet ID) : partie gauche de l'adresse
 - ▶ **identifiant de machine** (host ID) : partie droite
- ▶ notation CIDR :
 - ▶ 2001:db8:cafe:deca:a9e:1ff:fe6b:25c9/64
 - ▶ subnet correspondant:
2001:db8:cafe:deca::

- ▶ Préfixe de documentation 2001:db8::/32
- ▶ Préfixe link-local fe80::/10
- ▶ Préfixe multicast ff02::/10
- ▶ Exemple de préfixe «end-user»
2001:db8:fada:ba00:/56

Exemples d'adresses

221/307

- ▶ Unicast
2001:db8:cafe:deca:a9e:1ff:fe6b:25c9/64
- ▶ Link-local
fe80::a9e:1ff:fe6b:25c9/64
- ▶ Notation IPv4
2001:db8::cafe:192.168.0.1 égal à
2001:db8::cafe:c0a8:1
- ▶ Localhost ::1
- ▶ Tout les bits à 0 ::

Affectation d'adresses à une interface IP

222/307

- ▶ Il est *normal* d'avoir plusieurs adresses IPv6 à une interface (adresse de portée *lien*, de portée *globale*)
- ▶ Configuration «à la main» : plutôt rare
- ▶ Dynamiquement
 - ▶ Auto-configuration, grâce au préfixe annoncé par le router
 - ▶ Via DHCPv6

11 Protocole ARP / NDP

Relation entre adresse IP et adresse MAC dans les réseaux locaux

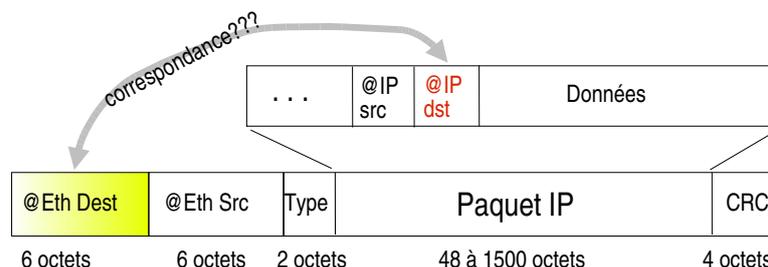
224/307

- ▶ Concernent les machines reliées à un réseau local de type Ethernet ou 802.11
- ▶ Les adresse IP se gèrent
 - ▶ Elles sont affectées «à la main» via des outils spécifiques du système d'exploitation des machines (interfaces graphiques ou commandes telles que ifconfig sous Unix/Linux)
 - ▶ Elles peuvent être affectées automatiquement via le protocole DHCP, mais cette possibilité est configurée elle aussi à la main
- ▶ Les adresses MAC ne se gèrent pas
 - ▶ Elles sont préaffectées par le constructeur de la carte interface que l'on achète ou qui est fournie avec la machine
 - ▶ Parfois le pilote (drivers) de la carte permet que cette adresse puisse être modifiée (via ifconfig)

...Adresses MAC et adresses IP...

225/307

- ▶ Problème : une machine M doit émettre un paquet IP vers une machine N voisine dont on ne connaît que le numéro IP (sur le même LAN)
- ▶ Si on est sur Ethernet le paquet IP sera véhiculé par une trame telle que celle-ci

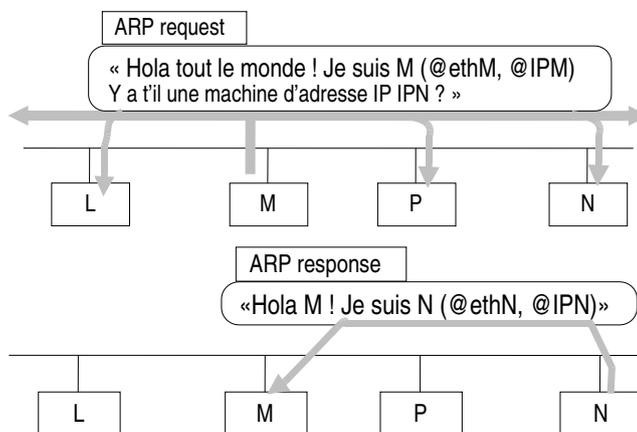


Comment la machine M peut elle retrouver l'adresse Ethernet de la machine N en ne connaissant que l'adresse IP de N ?

Adresses MAC et adresses IP : la solution ARP (IPv4)

226/307

► Address Resolution Protocol - rfc826



Soyez curieux : la commande **arp**

Sur un réseau local, sous Windows ou Unix/Linux, ouvrez une fenêtre de commande et tapez : **arp -a** pour voir la table de résolution arp instantanée. Si vous ne voyez pas la machine de votre voisin et que vous connaissez son adresse IP (ou son nom), faites un «ping» dessus : `ping 192.168.100.5` par exemple.

Si la réponse du ping est positive, refaites alors immédiatement **arp -a**, vous verrez apparaître la résolution concernant la machine voisine.

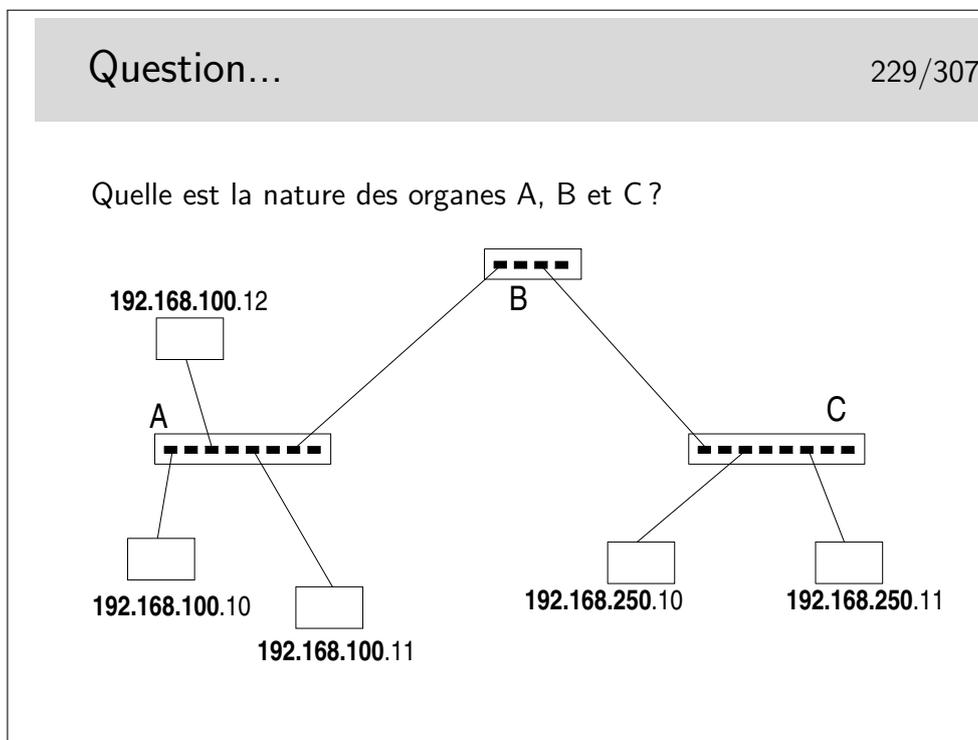
Adresses MAC et adresses IP : Neighbor Discovery Protocol (IPv6)

227/307

- Neighbor Discovery Protocol (NDP)
- Intégré dans Internet Control Message Protocol version 6 (ICMPv6) – rfc4443
- Fonctionne de manière similaire à ARP (IPv4)

Soyez curieux : la commande `ip neigh`

12 Principe du routage



Vous avez le choix entre trois types d'organes :

- des hubs,
- des switches ou commutateurs ou pont multiports
- des routeurs

Aide technique : le netmask est standard pour la classe d'adresses IP utilisée...

Et cette classe est la classe ??????....

???

???

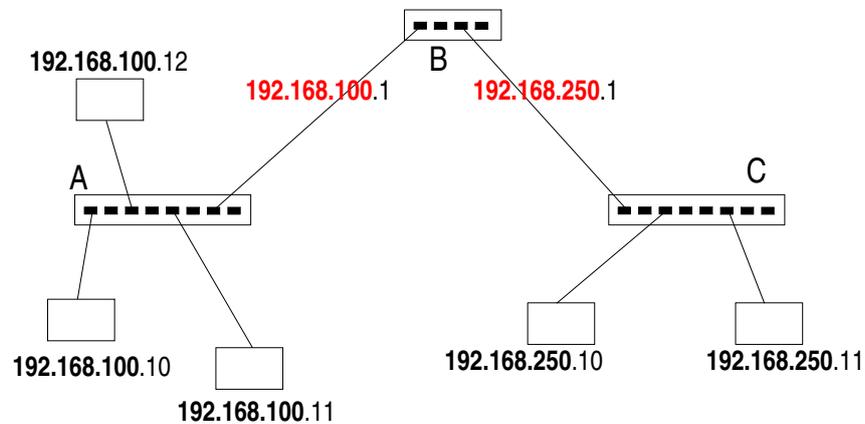
...C!

Ce sont des adresses de classe C...

Or les parties «Réseau» ne sont pas identiques entre la partie droite du schéma et la partie gauche...

Donc... Il ne s'agit pas de mêmes «Réseaux». Donc, forcément B est un... et A et C sont des ... ou des ...

Alors...



Réponse à la question... A et B sont des hubs ou des commutateurs (ou switches ou ponts multiports). Même si fondamentalement les fonctionnalités des hubs et des commutateurs sont différentes (hubs : répéteurs, organes physiques, niveau 1 ISO – switches : niveau 2 ISO) on ne peut pas faire la différence sur le schéma.

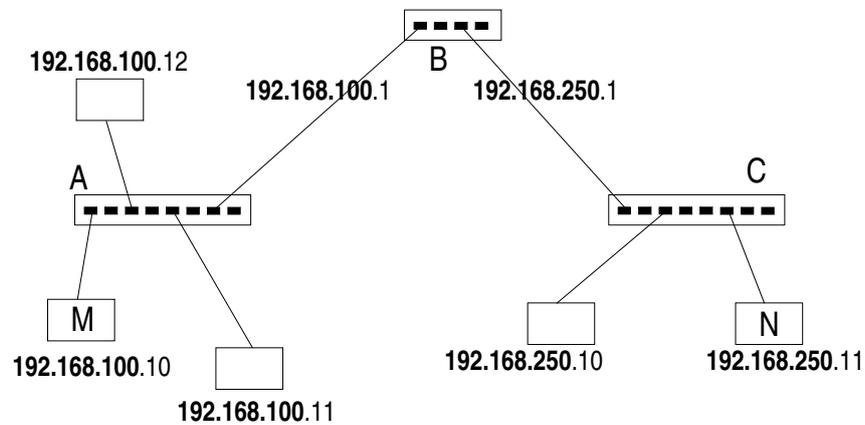
B est un routeur. Il raccorde des réseaux dont les machines sont identifiées par des numéros IP de classe C dont la partie réseau est différente. Il raccorde donc des réseaux. Il fonctionne au niveau 3 ISO.

B est un routeur, donc chacun de ses ports est identifié par une adresse IP dont la partie réseau identifie le réseau auquel ce port est raccordé.

Le premier port de B est raccordé sur le réseau 192.168.100.0, on lui donnera donc une adresse libre de ce réseau (ici .1). Un autre port est sur le réseau 192.168.250.0, on lui donnera par exemple l'adresse .1 sur ce réseau.

Ce n'est pas pour faire joli, c'est vraiment fonctionnel. Si on ne le fait pas le réseau ne peut pas fonctionner.

Comment envoyer un paquet de M vers N ?



Rien n'est magique... L'application sur M qui désire envoyer un message à une application sur N doit savoir que N existe. Sur M on doit connaître l'adresse IP de N.

On doit aussi savoir par où on passe. En M on doit savoir que pour atteindre N il faut passer par le routeur B.

Mais B est identifié par deux adresses IP!!!

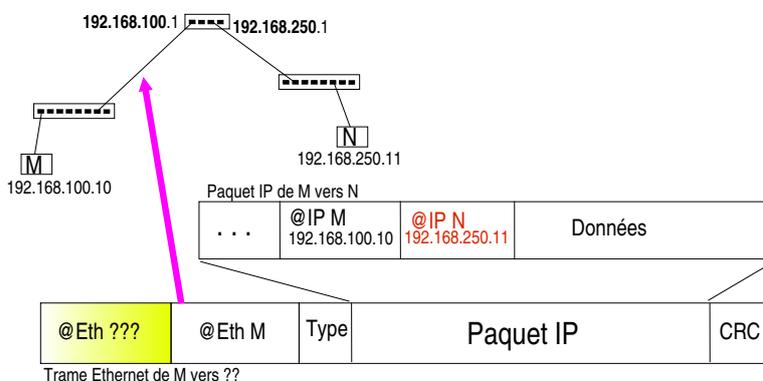
En M, nous sommes sur un réseau local, nous ne pouvons atteindre que des machines se trouvant sur le même réseau local que nous. Nous ne pouvons donc atteindre le routeur que par son port qui est situé sur le même réseau physique, celui identifié par le préfixe IP 192.168.100.

...Comment faire?...

À suivre...

Paquet IP de M vers N, et son porteur... | 232/307

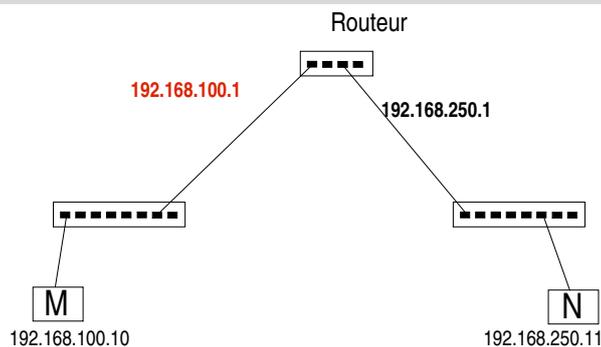
En réseau local



Comment, en M, déterminer l'adresse Mac de la trame Ethernet qui va emporter le paquet vers sa destination ?

Table de routage

| 233/307



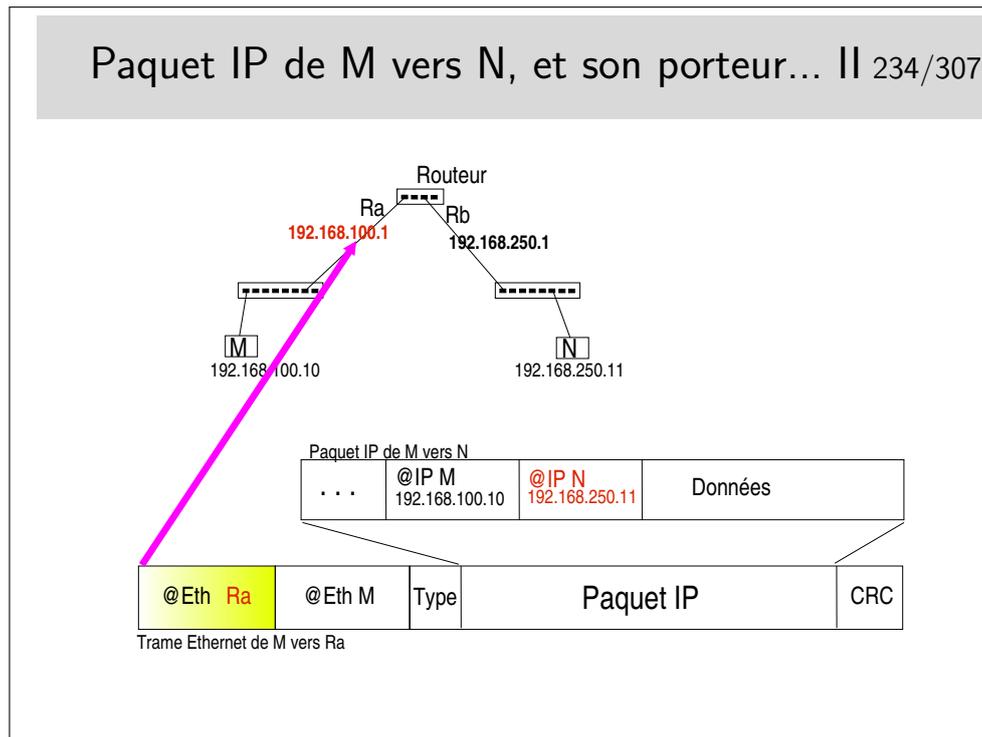
En M il faut une **table de routage** qui dit : « pour aller en 192.168.250.11 passer par 192.168.100.1 »
À l'aide de cette table il sera facile de faire une résolution ARP pour trouver l'adresse MAC du routeur. L'adresse destination de la trame Ethernet sera celle du routeur

Il faut une table de routage en M. Donc dans la machine terminale!!!

Chaque machine, quelle soit terminale ou intermédiaire, travaillant en IP incorpore des fonctions de routage.

Cela ne signifie pas que chaque machine terminale soit un routeur... Pour qu'une machine puisse jouer le rôle de routeur il faut qu'elle possède au moins deux interfaces munies d'adresses IP (autres que 127.0.0.1, correspondant à l'interface boucle locale) et que son module IP soit autorisé à effectuer la fonction de relaiage (forwarding).

Paquet IP de M vers N, et son porteur... II 234/307



Quand on disait qu'affecter une adresse à chaque interface de routeur n'était pas un effet de cosmétique...

Si en M on ne connaît pas l'adresse IP de l'interface du routeur qui est du même côté que M (sur le même réseau) alors la résolution ARP ne peut se faire.

Il ne suffit pas que l'interface du routeur et M soient «du même côté», il faut aussi qu'ils soient sur le même réseau local pour que ARP fonctionne (on rappelle que la requête ARP est transmise par broadcast Ethernet et que ce type de message ne passe pas les routeurs).

Ici la notion de réseau local est celle qui a été vue dans la partie précédente du cours, à savoir un réseau où la diffusion est possible vers toutes les machines. Ce pourrait être un VLAN car ce type de topologie définit des domaines de broadcast.

Lorsque des machines sont reliées entre-elles via des liaisons point-à-point, il n'y a pas de résolution ARP.

Chaque «entrée» dans la table contient au moins

- ▶ Une direction (réseau ou machine)
- ▶ Une indication de route
 - ▶ machine par laquelle les paquets doivent être acheminés
 - ▶ Cette machine doit être accessible
 - ▶ interface locale
- ▶ Un coût
 - ▶ Notion de «distance» ou de «coût»
 - ▶ Nombre de sauts nécessaires
 - ▶ Débit
 - ▶ ...

Les directions sont spécifiées par des numéros de réseaux indiqués sur 4 octets et accompagnés de leur netmask, par exemple 192.168.10.0 255.255.255.0, ou en notation CIDR 192.168.10.0/24. Attention, la notation CIDR peut ne pas être comprise par tous les routeurs. Elle l'est pour les machines Unix Linux ou BSD.

Une direction peut être une machine, dans ce cas l'adresse IP de la machine est indiqué directement et le netmask vaut 255.255.255.255.

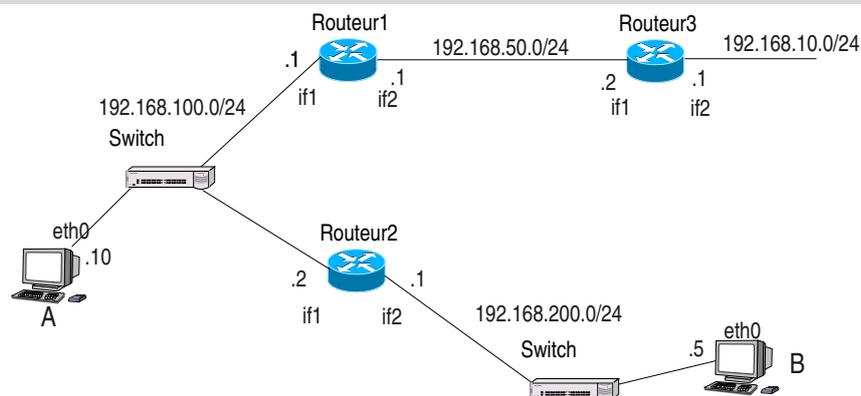
Les indications de route peuvent être données en indiquant l'adresse IP du routeur par lequel il faut envoyer les paquets pour la direction correspondante. Plus précisément il s'agit de l'adresse IP de l'interface du routeur immédiatement accessible. C'est absolument obligatoire dans le cas où l'interface du routeur en question est sur un réseau local classique.

Dans le cas où l'interface du routeur est sur une liaison point-à-point, on peut ne donner que le nom de l'interface locale. Voir transparent suivant.

Le coût correspond à la notion de «plus court chemin». Moins le coût est grand meilleur semble le chemin (ce n'est pas toujours vrai, un chemin peut être plus court en vraie distance (moins de sauts par exemple) et cependant moins efficace en débit).

Table de routage

III 236/307



Légende :

192.168.X.0/24 : adresse de réseau et netmask
 .x (.1 ou .2, etc.) : partie machine de l'adresse de l'interface
 if1, if2, eth0 : nom d'interface réseau sur la machine

Table de routage en A

direction	netmask	gateway	interface
192.168.100.0	255.255.255.0	192.168.100.10	eth0
192.168.200.0	255.255.255.0	192.168.100.2	eth0
default	0.0.0.0	192.168.100.1	eth0

Rq : ici la mention de l'interface n'est pas très importante, elle est redondante car il n'y a qu'une seule interface physique (il y a quand même l'interface boucle locale qu'on n'a pas fait figurer dans la table de routage, mais qui existe et provoque des entrées spécifiques dans la table)

Table de routage en Routeur1

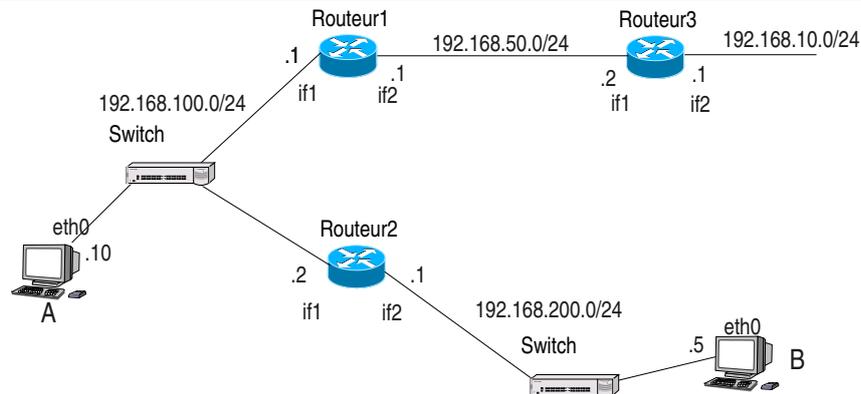
direction	netmask	gateway	interface
192.168.50.0	255.255.255.0	192.168.50.1	if2
192.168.100.0	255.255.255.0	192.168.100.1	if1
192.168.200.0	255.255.255.0	192.168.100.2	if1
default	0.0.0.0	192.168.50.2	if2

Rq : dans la ligne par défaut, la seule mention de l'interface suffirait car le lien entre les routeurs 1 et 2 est de type point-à-point.

Compléter pour le routeur 2 et la machine B...

Table de routage

IV 237/307



Considérons les tables de routage correctement servies partout sauf en A
 A peut atteindre if1 de Routeur3 (192.168.50.2), mais pas if2 (192.168.10.1)
 Pourquoi **ne peut-on pas** dire en A : pour aller en 192.168.10.0/24 passer par
 192.168.50.2 ? *Que faut-il dire ?*

On suppose la table de routage en A partiellement remplie. L'interface if1 du routeur2 est accessible, une commande ping, depuis A, vers l'adresse de cette interface fonctionne. Donc on peut supposer (et on a raison) que cette interface est «visible» depuis A. Cependant si on tente de créer en A une route via cette interface la commande doit échouer. Pourquoi ?

Exemple de table de routage sous Windows 238/307

```
C:\>route print
=====
Liste d'Interfaces
0x1 ..... MS TCP Loopback interface
0x1000003 ...00 01 02 6e 7c 46 ..... 3Com EtherLink PCI
=====
Itinéraires actifs:
  Destination réseau      Masque réseau    Adr. passerelle  Adr. interface  Métrique
    0.0.0.0                0.0.0.0         192.44.75.1     192.44.75.184   1
    127.0.0.0             255.0.0.0      127.0.0.1      127.0.0.1       1
    192.44.75.0           255.255.255.0  192.44.75.184  192.44.75.184   1
    192.44.75.184        255.255.255.255  127.0.0.1     127.0.0.1       1
    192.44.75.255        255.255.255.255  192.44.75.184  192.44.75.184   1
    224.0.0.0             224.0.0.0      192.44.75.184  192.44.75.184   1
    255.255.255.255     255.255.255.255  192.44.75.184  192.44.75.184   1
Passerelle par défaut: 192.44.75.1
=====
```

Quelle est le numéro de l'interface de boucle locale (interface interne non raccordée à un réseau physique) ?

- Quelle sont les adresses de broadcast possibles ?
Par quelle interface sont acheminés les paquets de broadcast ?
Que pouvez vous déduire de tout ceci concernant l'adresse de la machine en question ?
Que se passe-t'il si une application de cette machine envoie un paquet IP vers une autre application de la même machine ? Quel chemin est emprunté par le paquet ?
Quelle adresse code la destination par défaut ?

Et votre système dans tout ça...

239/307

Quel est votre paramétrage ?

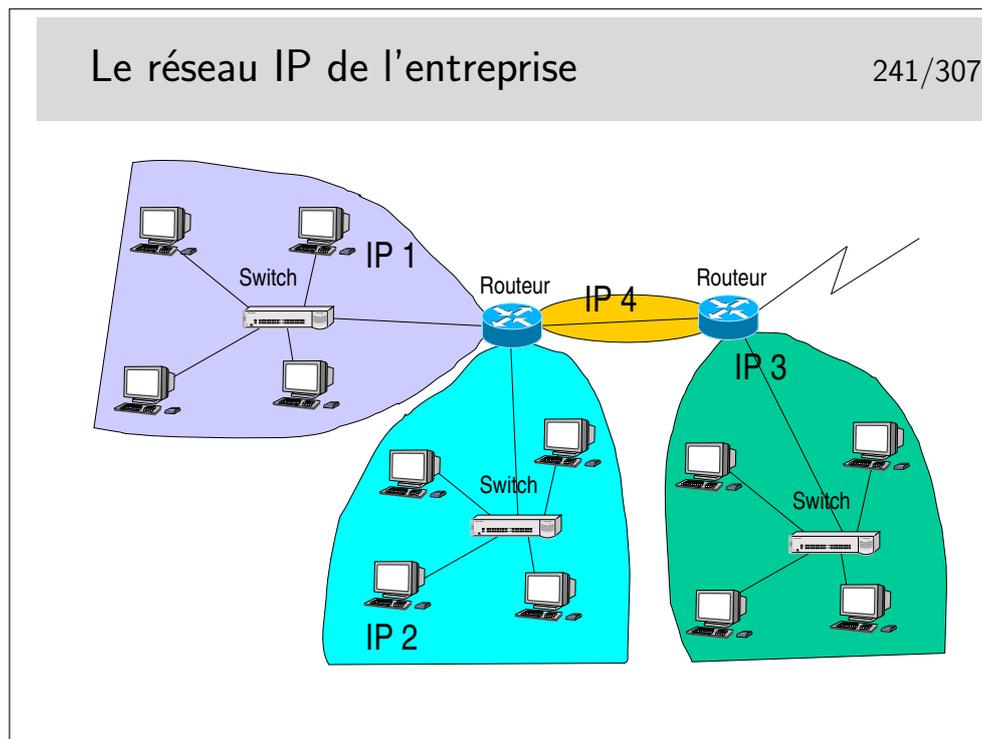
- ▶ **Sous Windows** (Ouvrez une fenêtre de commande et testez les commandes suivantes :)
 - ▶ `ipconfig` (`winipcfg` sous Windows 9x/ME) avec le sélecteur `/all`
 - ▶ `route print` pour afficher la table de routage
 - ▶ `nslookup` pour traduire des noms de machine en adresse IP et inversement
 - ▶ `arp -a` si vous êtes sur un LAN pour lire la table de traduction arp
 - ▶ `tracert`
- ▶ **Sous Linux** (Ouvrez un terminal et testez :)
 - ▶ `ifconfig` avec ou sans `-a`
 - ▶ `route` avec ou sans `-n`
 - ▶ `nslookup` ou `host`
 - ▶ `arp -a`
 - ▶ `traceroute` (avec ou sans `-n`)

Et n'oubliez pas la commande ping... C'est la première commande à utiliser quand le réseau ne va pas très bien... «Mon voisin est-il joignable? Alors :

`ping mon_voisin` (plutôt `adresse_IP_de_mon_voisin`)

Note : Vous remarquerez peut être que Windows affiche plus de routes que Linux. En fait Linux gère plusieurs tables de routage et a pour habitude de n'afficher que la table `main`, sauf si on lui demande gentiment. (Pour avoir la liste des tables de routage : `cat /etc/iproute2/rt_tables`; pour afficher la table principale : `ip route show table main`; pour afficher la table avec les règles pour le multicast et le broadcast : `ip route show table local`; etc.)

13 Les routeurs

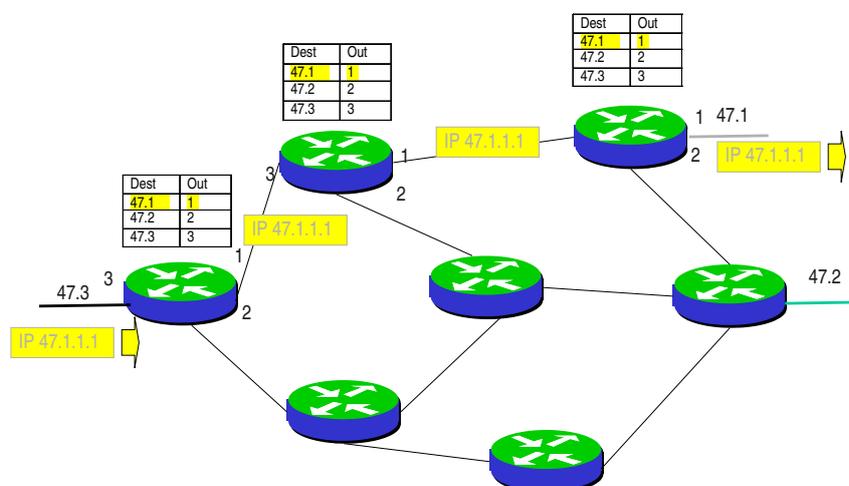


Ce que l'on va appeler le «réseau local» de l'entreprise est en réalité un réseau de niveau 3 composé de routeurs interconnectant des réseaux locaux séparés physiquement ou logiquement (VLANs).

L'adressage pourra être de type «privé», avec une seule adresse officielle en sortie et une fonction de type NAT dans le routeur de sortie.

Fonctions des routeurs 242/307

- ▶ Chaque paquet IP entrant dans le routeur voit son adresse destination examinée et comparée sur un certain nombre de bits avec le contenu d'une table (table de routage) associant des directions avec une interface de sortie
 - ▶ Cette fonction est aussi mise en œuvre dans les machines terminales
 - ▶ Le champ TTL est décrémenté par chaque routeur, le champ *checksum* doit être recalculé à chaque fois



14 Protocoles de routage

- ▶ Routage statique
 - ▶ La table est remplie «à la main» via une commande spécifique ou une interface graphique
- ▶ Routage dynamique
 - ▶ Via des protocoles appropriés
 - ▶ Permettent de détecter des ruptures de liens et de reconfigurer les tables de routage de manière dynamique

- ▶ Deux grands types de mise en oeuvre
 - ▶ Les protocoles internes
 - ▶ Pour les réseaux d'entreprises
 - ▶ Notion de système autonome (*autonomous system*)
 - ▶ Les protocoles externes
 - ▶ Pour les réseau d'opérateurs
 - ▶ A prendre en compte aussi par les entreprises pour leurs routeurs de raccordement aux opérateurs

Un AS (*Autonomous System*) est un grand système réseau (typiquement un fournisseur d'accès) qui gère de manière cohérente et autonome son routage. Les numéros d'AS (utilisés notamment dans le routage BGP, cf. plus loin) sont attribués par les RIR (*Regional Internet Registry*) sous l'autorité de l'IANA. Le RIR européen est RIPE-NCC (*Réseaux IP Européens - Network Coordination Centre*).

Par exemple en 2015 on dénombre 1208 "opérateurs" qui offrent des services sur le territoire français : <http://www.ripe.net/membership/indices/FR.html>. (Ne pas croire que les seuls opérateurs sont Orange SFR Bouygues...)

Ces grands systèmes ont un numéro d'AS : France Télécom (AS3215), Proxad (AS12322), LDCOM Networks (AS15557 gaoland.net), Cegetel (AS8228), 9TELECOM (AS12626 9tel.net), AOL France customers , SIRIS (AS3305 siris.fr), KAPTECH (AS6760 kapttech.net), BELGACOM France (AS6771), Nerim (AS13193), etc.

Ces AS sont interconnectés via des points dits de *peering* : France-IX, SFINX, PARIX, FreeIX, PanaP, LyonIX, LINX, AMS-IX, DE-CIX, etc. Voir http://en.wikipedia.org/wiki/List_of_Internet_exchange_points.

Par exemple, à quoi est raccordé SFR ?

Tapez `whois as15557`

<http://www.robtex.com/as/as15557.html>

Quelles sont les blocks d'adresse alloués à Free :

```
whois -r -i org ORG-PISP1-RIPE
```

À quoi est raccordé Télécom Bretagne? Voyez <http://www.robtex.com/dns/telecom-bretagne.eu.html#graph>

Un autre concept important est la séparation (conceptuelle) entre cœur et bordure d'Internet. On parle alors de la *Default-Free Zone* (DFZ). C'est l'ensemble des routeurs qui n'ont pas de *route par défaut*. Autrement-dit, ce sont les routeurs qui constituent *le*

cœur du réseau Internet (et ceci indépendamment du découpage d'Internet en AS). Par opposition, tous les autres routeurs qui ont à priori une route par défaut vers le reste d'Internet, constituent la *la bordure* d'Internet.

Le principe de fonctionnement du routage dynamique

I 247/307

- ▶ La configuration des interfaces d'un routeur «apprend» à celui-ci les numéros des réseaux sur lesquels il est raccordé
 - ▶ Le routeur connaît donc déjà quelques réseaux
 - ▶ Il peut communiquer cette connaissance aux autres autres routeurs situés sur les mêmes segments de réseau que lui, via un protocole approprié
 - ▶ Les autres routeurs peuvent faire de même

../..

Le principe de fonctionnement du routage dynamique

II 248/307

- ../.. ▶ Au bout d'un certain temps tous les routeurs connaissent tout sur tout les autres
 - ▶ Les routeurs continuent périodiquement à s'envoyer des informations. Si l'un d'eux tombe en panne, les autres, ne recevant plus ses informations, le considèrent comme hors service et inhibent les routes qui passaient par lui. Ils peuvent aussi déterminer des routes alternatives

- ▶ Deux grands types d'algorithmes
 - ▶ Le vecteur de distance
 - ▶ Les routeurs s'échangent leur table de routage (destination et distance ou coût) entre voisins
 - ▶ Protocole de type interne RIP (Routing Information Protocol), le plus répandu mais peu performant
 - ▶ L'état de liens
 - ▶ Tous les routeurs d'un domaine s'échangent leurs informations de routage
 - ▶ Protocole de type interne OSPF (Open Shortest Path First), plus performant que RIP mais complexe

- ▶ Tous les routeurs du monde ne doivent pas avoir la copie des tables de routage de tous les autres routeurs
- ▶ Concept de domaines
 - ▶ Un domaine est une entreprise, un campus, une entité restreinte
 - ▶ Des protocoles de routage sont adaptés au routage intra domaine
 - ▶ RIP, OSPF
- ▶ Routage inter domaines
 - ▶ Les informations sont agrégées
 - ▶ Notion d'autonomous système
 - ▶ Protocole fortement utilisé : BGP (Border Gateway Protocol)

Protocoles de routage les plus répandus I

251/307

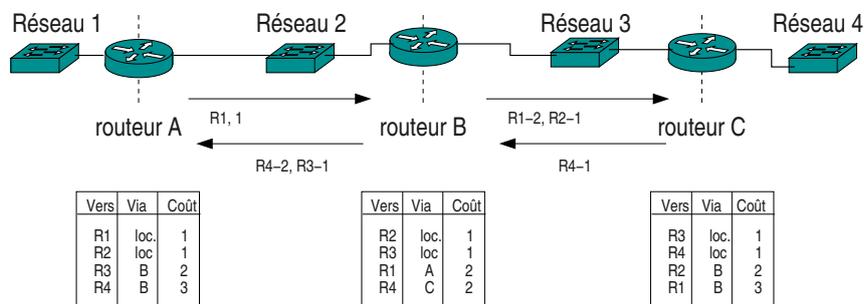
- ▶ RIP : Routing Information Protocol
 - ▶ Routage intra-domaine
 - ▶ Type vecteur de distance
 - ▶ Peu performant
 - ▶ Pauvre notion de coût (nombre de saut)
 - ▶ Facile à mettre en œuvre
- ▶ OSPF : Open Shortest Path First
 - ▶ Routage intra-domaine
 - ▶ Famille état des liens (link state)
 - ▶ Notion de coût plus riche (par défaut : 108/bande_passante)
 - ▶ Routage hiérarchique avec délimitation d'aires

Protocoles de routage les plus répandus II

252/307

- ▶ BGP : Border Gateway Protocol (rfc1771)
 - ▶ Routage inter-domaines
 - ▶ Prend en compte le routage sans classe (CIDR rfc1466)
 - ▶ Véhiculé par TCP

► Vecteur de distance : *direction, coût*



► Pourquoi les routeurs n'annoncent-ils pas tout le contenu de leur table ?

Chaque routeur annonce, systématiquement toutes les 30s, sa table de routage à ses voisins (seulement les réseaux connus et le coût pour les atteindre, ils n'annoncent pas le paramètre «via»). Chaque routeur prend donc «connaissance» de la «connaissance» des ses voisins.

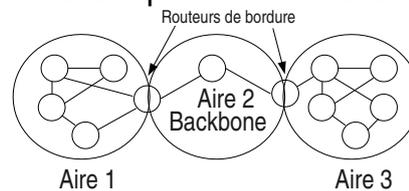
Lorsque l'algorithme a convergé chaque routeur doit connaître l'existence des tous les réseaux. Si une modification du routage apparaît dans un routeur (nouvelle destination, routeur voisin silencieux), le routeur en question recalcule sa table. S'il en résulte une modification (comme dans le cas de l'apparition d'une nouvelle route) il en informe immédiatement ses voisins sans attendre les 30s habituelles.

Le schéma ci-dessus montre les tables pour un réseau simple une fois atteinte la convergence. Pourquoi les routeurs n'annoncent-ils pas toute leur table? Par exemple, pourquoi A n'annonce-t'il pas R4? L'algorithme serait plus simple à implémenter.

En fait A n'annonce pas vers B qu'il connaît R4 car il apprend l'existence de R4 par B lui-même. Et si C tombait en panne, B ne recevant plus d'annonce de C placerait R4 à une distance infinie et le supprimerait de sa table. Puis il indiquerait vers A ce fait nouveau. Mais peut-être que A aurait le temps de communiquer à B sa connaissance de R4 avec un coût de 2. B pourrait alors croire que A connaît une route vers R4.

Cette non annonce, dans une direction, des routes que l'on apprend par elle s'appelle la technique de l'**horizon coupé** (*splitted horizon*). Elle n'est pas complètement satisfaisante car s'il existe plusieurs autres routeurs du coté de A et que ceux-ci forment des boucles, il se peut que l'algorithme ne puisse pas converger.

- ▶ rfc1583, 1587, 2328
- ▶ Routage hiérarchique : notion d'aires



- ▶ Chaque aire doit être reliée à l'aire backbone qui assure le transit entre deux aires
- ▶ Dans chaque aire les routeurs ont une connaissance complète du routage dans leur aire
- ▶ Les routeurs de bordure d'aires ont une connaissance des aires qu'ils raccordent, ils propagent dans chaque aire une connaissance agrégée des autres aires

Un exemple de découverte de route

Sous Windows :

```
C:\>tracert www.redhat.com
Détermination de l'itinéraire vers www.redhat.com [209.132.177.50]
avec un maximum de 30 sauts :
 1 <10 ms  16 ms <10 ms galaxie-75.enst-bretagne.fr [192.44.75.1]
 2 <10 ms <10 ms <10 ms 193.50.69.89
 3 <10 ms <10 ms <10 ms 193.48.78.17
 4 <10 ms  15 ms <10 ms PA0-Rennes2-TR.rrb.ft.net [195.101.145.25]
 5  16 ms <10 ms  16 ms peering-GIP.rrb.ft.net [195.101.145.6]
 6 <10 ms  16 ms <10 ms rennes-a0-0-2.cssi.renater.fr [193.51.181.126]
 7  16 ms  16 ms  15 ms caen-pos1-0.cssi.renater.fr [193.51.180.17]
 8  15 ms  16 ms  15 ms rouen-pos1-0.cssi.renater.fr [193.51.180.22]
 9  31 ms  16 ms  16 ms nri-a-pos6-0.cssi.renater.fr [193.51.179.21]
10  32 ms  15 ms  16 ms 193.51.185.1
11  15 ms  16 ms  15 ms P11-0.PASCR1.Pastourelle.opentransit.net [193.251.241.97]
12  93 ms  94 ms  94 ms P12-0.NYKCR3.New-york.opentransit.net [193.251.241.134]
13  94 ms  94 ms  94 ms ft-gw.n54ny.ip.att.net [192.205.32.137]
14  93 ms  94 ms  94 ms tbr1-p010401.n54ny.ip.att.net [12.123.3.57]
15 110 ms 125 ms 109 ms tbr1-cl1.cgcil.ip.att.net [12.122.10.2]
16 110 ms 109 ms 125 ms tbr2-cl2.cgcil.ip.att.net [12.122.9.134]
17 172 ms 172 ms 172 ms tbr2-cl7.sl9mo.ip.att.net [12.122.10.46]
18 125 ms 140 ms 125 ms tbr2-cl6.dlstx.ip.att.net [12.122.10.90]
19 172 ms 172 ms 172 ms gbr2-p40.phmaz.ip.att.net [12.122.10.86]
20 172 ms 172 ms 172 ms gar2-p370.phmaz.ip.att.net [12.123.142.49]
21 172 ms 172 ms 172 ms mdf1-gsr12-1-pos-7-0.phx1.attens.net [12.122.255230]
22 141 ms 156 ms 141 ms mdf1-bi8k-2-eth-2-1.phx1.attens.net [63.241.128.158]
23 * * * Délai d'attente de la demande dépassé.
```

La recherche de chemin n'aboutit pas, des filtrages sur certains routeurs en sont la cause.

Cette commande peut être rapide si on lui ajoute l'option -d qui empêche la traduction des adresses IP en noms.

La commande existe sous Unix/Linux sous le nom traceroute.

15 Gestion des erreurs

Le protocole ICMP (IPv4)

257/307

Internet Control Message Protocol (rfc 792)

- ▶ Sert à véhiculer des messages d'erreur ou de demande d'information
 - ▶ Demande d'écho et réponse (commande ping)
 - ▶ Destination non accessible
 - ▶ Le réseau ne peut être atteint
 - ▶ La fragmentation est nécessaire et le bit D est à 1 (PMTU discovery)
 - ▶ etc
 - ▶ Redirection, il existe une meilleure route
 - ▶ Durée de vie dépassé
 - ▶ etc

La commande **ping** n'utilise que ICMP (porté par IP).

La commande **tracroute** (**tracert** sous Windows) joue sur le dépassement de la durée de vie. Un premier paquet est créé «à la main» et son champ `ttl` est mis à 1. Il contient un paquet UDP à destination d'un port inconnu. Le paquet est routé, il atteint le premier routeur qui décrémente alors le `ttl`. Le résultat valant 0 le paquet est jeté et un message ICMP est émis vers la source du paquet jeté. Le message est véhiculé par un paquet IP comportant l'adresse du routeur, ce qu'on attend dans `tracroute`. On peut alors afficher l'identité de ce routeur ainsi que le temps mesuré entre l'envoi du paquet initial et le l'arrivée du message ICMP. On fait cela trois fois pour avoir une estimation du temps moyen d'aller et retour puis on recommence en mettant cette fois le `ttl` à 2.

Internet Control Message Protocol v6 (rfc 4443)

- ▶ Mêmes fonctions que ICMP (IPv4)
 - ▶ Erreurs diverses, ping, redirections, etc.
- ▶ En plus :
 - ▶ Neighbor Discovery Protocol (équivalent du ARP pour IPv4)
 - ▶ Router Solicitation, Router Advertisement :
Auto-configuration de l'adresse. Au démarrage, un host recherche s'il y a un routeur présent sur LAN ; le routeur diffuse le préfixe IPv6 à utiliser ; le host se choisit une adresse dans ce préfixe.

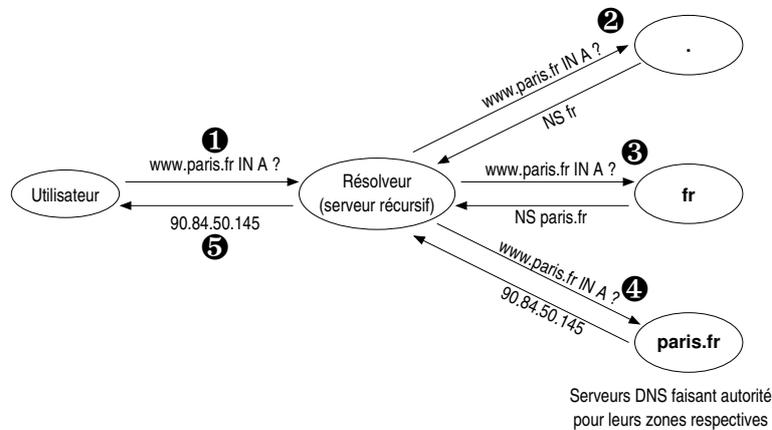
16 Services pour IP

- ▶ Le protocole DHCP : Dynamic Host Control Protocol (DHCPv4 rfc2131, DHCPv6 rfc3315)
 - ▶ Utilisé pour obtenir une adresse IP de manière dynamique ainsi que des informations diverses : serveur DNS, domaine, netmask, routeur par défaut
- ▶ Le service de nommage : le DNS
 - ▶ Domain Name Service (rfc 1034 et 1035)
 - ▶ Permet d'associer un nom à une adresse IP
 - ▶ Offre un protocole pour retrouver ce nom en partant de l'adresse ou l'adresse en partant du nom

- ▶ Service réparti de nommage
- ▶ Répartition mondiale
- ▶ Gestion indépendante de chaque domaine, pas d'autorité absolue de référence, chaque administrateur est libre de ses choix
- ▶ Permet de connaître :
 - ▶ les numéros de machine à partir de leur nom et inversement
 - ▶ les machines gestionnaires de courrier électronique pour les domaines
 - ▶ et bien d'autres informations

- ▶ Un domaine DNS peut être composé de plusieurs réseaux IP (identifiés par des numéros IP différents)
- ▶ Un domaine peut être subdivisé en sous domaines
- ▶ Les domaines forment une arborescence
- ▶ La racine de l'arborescence est subdivisée en «top level domains» : .edu, .org, .gov, .mil, .net, .com, .us pour les USA, .fr, .ca, .uk, .de, .jp, etc pour le reste du monde
- ▶ De nouveaux «top level» apparaissent ou vont apparaître : .int, ...

- ▶ Des *résolveurs* (forward les requêtes et cachent les réponses)
- ▶ Des *serveurs faisant autorité* (qui possèdent l'information originelle)



Soyez curieux...

Essayez **nslookup** sous Windows ou Unix/Linux.

Sous Linux il y a aussi **host** tout simplement ou **dig** qui est beaucoup moins simple.

Le DNS peut faire l'objet d'attaques (certaines effectives, d'autres fantasmées...) : voir la présentation de Stéphane Bortzmeyer (Afnic) aux journées JCSA 2012 <http://www.afnic.fr/fr/1-afnic-en-bref/actualites/actualites-generales/6171/show/succes-pour-la-journee-du-conseil-scientifique-sous-le-signes-de-la-resilience-8.html>

Quatrième partie

Les protocoles TCP et UDP

Quelques protocoles applicatifs

17 Introduction

Les protocoles de transport au dessus de IP_{266/307}

- ▶ TCP : Transmission Control Protocol rfc793
 - ▶ Transport en mode «connecté»
 - ▶ Contrôle de flux et détection d'erreur avec retransmission
- ▶ UDP : User Datagram Protocol rfc763
 - ▶ Mode datagramme
 - ▶ Pas de contrôles, pas d'assurance de délivrance
- ▶ Nouveaux, expérimentaux
 - ▶ DCCP : Datagram Congestion Control Protocol, rfc4340
 - ▶ SCTP : Stream Control Transmission Protocol, rfc4960
 - ▶ MPTCP : Multipath TCP, rfc6826

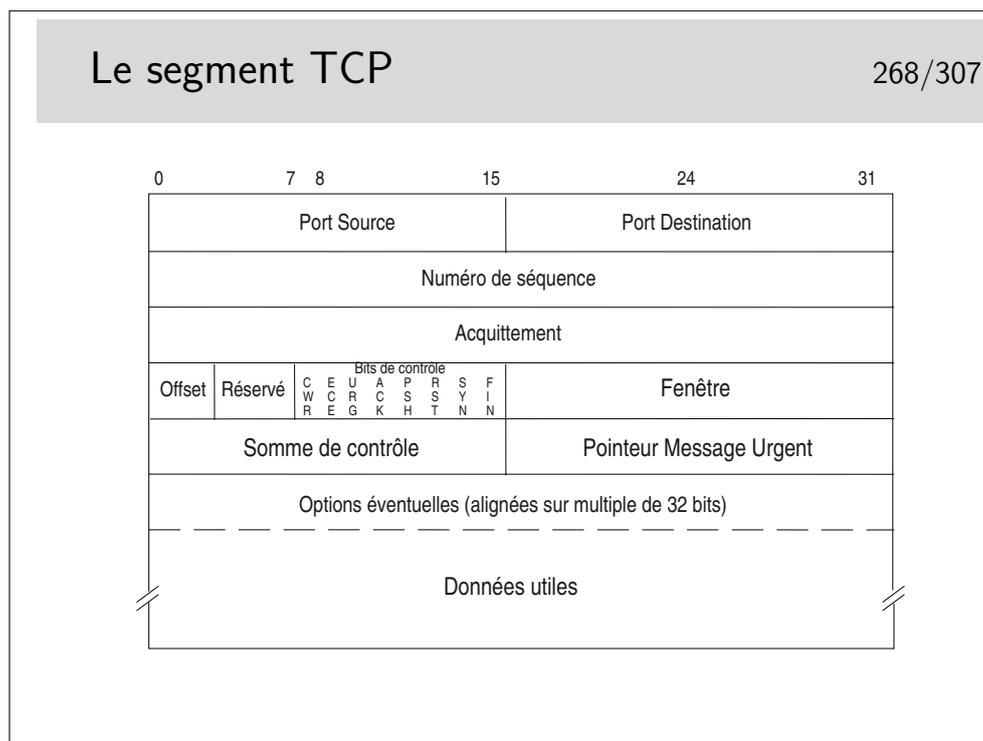
TCP et UDP, deux besoins extrêmes, mais relativement faciles à utiliser. De nombreux protocoles "universitaires" proposent des compromis, des services intermédiaires (p-ex. le protocole POC : Partial Order Connection), mais certains commencent à émerger sérieusement comme DCCP et SCTP.

DCCP : un genre d'UDP mais en mode connecté et avec contrôle de congestion. Pas de préservation de l'ordre des message ni de garantie de transmission des donnée, mais garantie sur les acquittement.

SCTP : un genre de TCP mais orienté *flux de messages* et non pas flux d'octets (flux : préserve l'ordre et garantie la transmission); gère le multi-flux au sein d'une même connection, et le *multihoming* (annoncer que l'on va changer d'adresse IP pour la suite de la session).

MPTCP : compatible avec le TCP classique (programme avec des sockets TCP); définit des nouvelles *options* pour annoncer des sous-flux, en parallèle ou en secours, rejoindre une autre IP sans clore la session (*multihoming*), etc.

18 TCP



Ports source et destination : identifient les applications en relation, généralement l'une d'elle est serveur, son port correspond alors au numéro du protocole (exemple : 80 pour les serveurs Web)

Numéro de séquence : numéro du premier octet des données. C'est le rang du premier octet véhiculé par ce segment en comptant depuis le début de l'échange. Le numéro de début est tiré aléatoirement entre 0 et 232-1.

Acquittement : numéro du prochain octet attendu

Offset : indique la longueur de l'entête en mots de 32 bits (si égal à 5 alors pas d'option)

Bits de contrôle (CWR, ECE, Urg, Ack, Psh, Rst, Syn, Fin)

Fenêtre : permet le contrôle de flux, le récepteur indique avec ce champ combien d'octets il est prêt à recevoir (une valeur de 0 indique que ses tampons mémoire sont pleins)

Somme de contrôle : permet de savoir si le segment a été altéré ou non pendant sa transmission

Pointeur de données urgentes : indique l'emplacement de ces données dans le segment (si le bit Urg est positionné, sinon ce champ est ignoré, bien qu'il existe toujours)

Les bits de contrôle I

269/307

CWR	ECE	URG	ACK	PSH	RST	SYN	FIN
-----	-----	-----	-----	-----	-----	-----	-----

- ▶ ECE Explicit Congestion Notification-Echo (rfc 3168), l'entité TCP recevant un segment contenant ce bit à 1 doit réduire son débit
- ▶ CWR Congestion Window Reduced (rfc 3168) réponse à la réception du bit ECE pour indiquer que la requête a bien été prise en compte
- ▶ URG indique que le champ *Urgent Pointer* est significatif.
- ▶ ACK indique que le champ *Acknowledgement Number* est significatif.
- ▶ PSH fonction *PUSH*. Les données doivent être immédiatement remises à la couche supérieure.
- ▶ RST *reset*, la connexion est rompue.
- ▶ SYN synchronisation des indicateurs numéro de séquence. Segments d'initialisation de connexion.
- ▶ FIN terminaison de connexion.

Les bits de contrôle II

270/307

Exemple : segment IP et TCP dont le bit ACK est à 1

```
0: 0800 2074 ef05 0800 0914 18e7 0800 4500
16: 0028 8954 0000 4006 db07 c02c 4b13 c02c
32: 4b08 1770 fdbe 0162 6e86 8e21 a873 5010 /* 0001 0000 */
48: 2210 bba3 0000 023a b3a1 1829
```

Les données «urgentes»

271/307

- ▶ Données appelées parfois "hors bande" ou *Out of Band* (OOB)
- ▶ Données à traiter en priorité par la couche réceptrice
- ▶ Elles sont véhiculées dans le flux normal en suivant le chemin normal. IP n'est pas sensible à ces données, leur caractère "urgent" est significatif seulement aux extrémités
- ▶ L'arrivée de ces données a un caractère aléatoire pour les applications destinataires
- ▶ Les applications ne lisent pas ces données dans le flux normal
- ▶ Une application devant pouvoir accepter de telles données doit avertir le système pour que celui-ci lui envoie une interruption (un signal logiciel) afin qu'elle puisse traiter en priorité la donnée. L'application doit prévoir une routine spéciale de traitement pour la lecture de ces données
- ▶ Sémantique mal définie : le RFC6093 recommande aux nouvelles applications de ne plus l'utiliser...

Le bit de contrôle RST

272/307

- ▶ Utilisé par une entité TCP connectée avec une autre entité TCP distante pour avertir d'un problème
- ▶ Une application se terminant normalement fait une fermeture sur le port TCP utilisé (souvent une *socket*), ceci se concrétise par un échange de segments avec le bit FIN positionné et la connexion est rompue
- ▶ Si l'application se termine brutalement sans fermer la connexion, l'entité TCP associée envoie vers l'autre extrémité un segment avec le bit RST positionné. L'autre extrémité est ainsi prévenue de la terminaison de la connexion

- ▶ La connexion TCP :
 - ▶ Relation établie point à point entre les deux extrémités
 - ▶ Normalement transparente aux routeurs (sauf si ceux-ci mettent en œuvre du filtrage ou des mécanismes de QoS)
 - ▶ Caractérisée par un contexte dans les machines d'extrémité (numéros IP local et distant, ports local et distant)

- ▶ La connexion est établie par un échange de paquets initiaux : le *three way handshake*
 - ▶ Ce n'est pas une connexion au sens strict du terme, il n'y a pas de chemin virtuel établi dans le réseau, les segments TCP sont portés par IP, protocole orienté datagramme
 - ▶ La connexion TCP se traduit par un contexte mémorisé dans les machines d'extrémité (le client et le serveur), ce contexte est le quintuplet :

[protocole, port local, port distant, @IP locale, @IP distante]

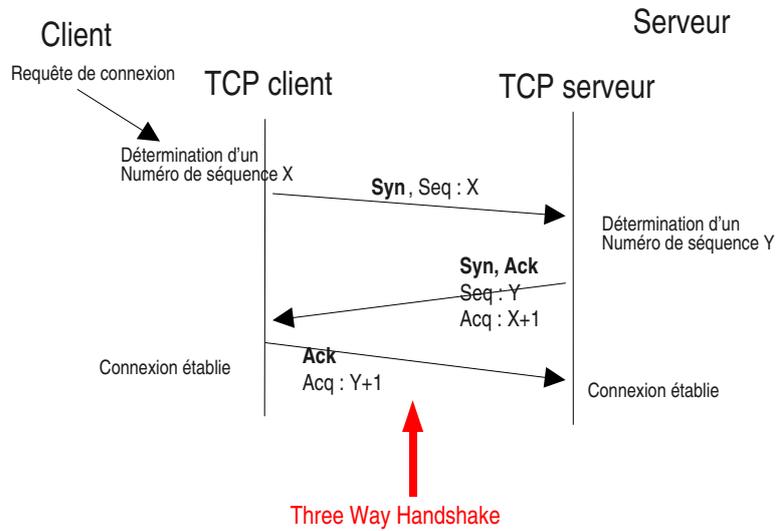
- ▶ Si les applications connectées n'ont rien à se dire, les entités TCP correspondantes ne s'envoient aucun segment, il n'y a plus alors de trafic
- ▶ Il est possible de faire en sorte que les entités s'échangent des segments d'alerte (sans données utiles puisqu'il n'y en a pas à envoyer) toutes les deux heures en positionnant une option dans TCP
- ▶ Si l'entité TCP distante ne répond pas au segment d'alerte ceci signifie que l'application associée s'est terminée sans prévenir ou que la machine s'est arrêtée. L'application locale sera prévenue lors de la prochaine lecture du port TCP

Cette option n'est pas une option au sens dans lequel ce mot sera vu plus loin. Il s'agit simplement d'un paramètre interne à l'entité TCP, positionné par l'application à l'aide d'une fonction spécifique (`setsockopt(SO_KEEPALIVE)`).

- ▶ Connexion passive
 - ▶ Ce n'est pas une connexion mais un point d'accès TCP ouvert par une application en fonction de [serveur](#)
 - ▶ Le contexte TCP créé attend une requête de connexion émanant d'un client
- ▶ Connexion active
 - ▶ La connexion réelle, initialisée par une application en mode [client](#)

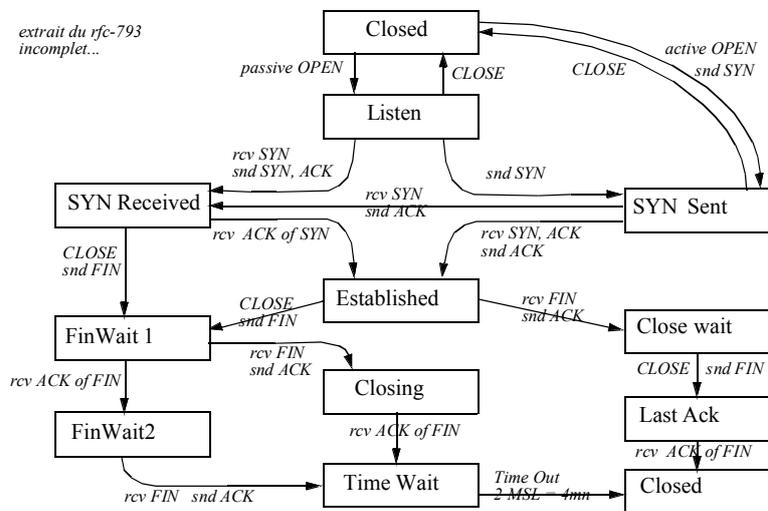
Établissement de la connexion

277/307

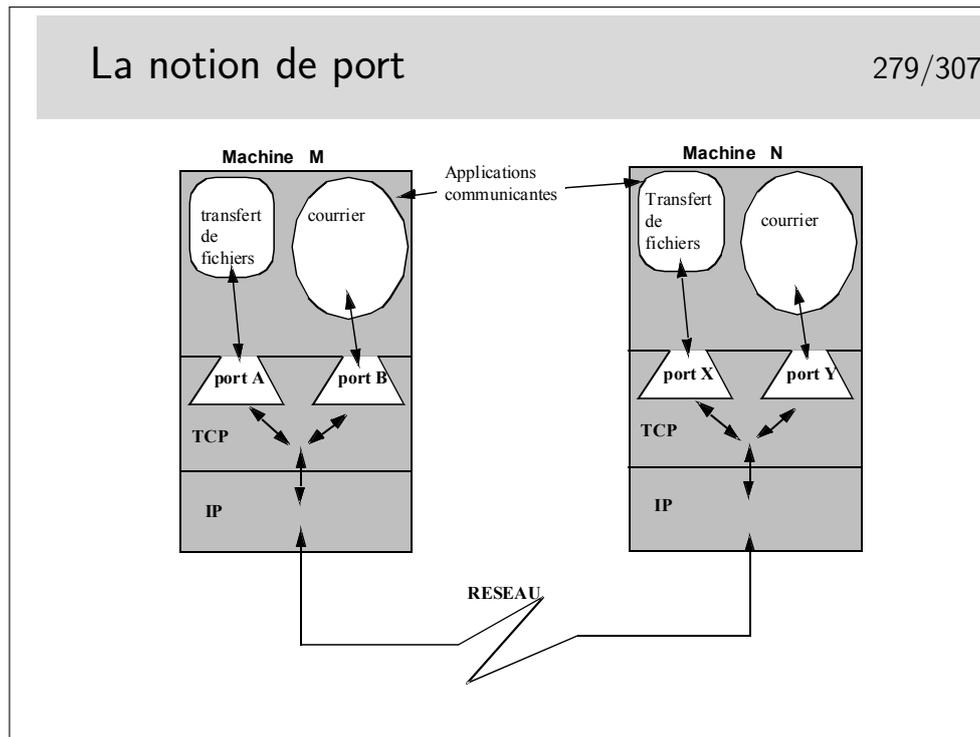


La machine d'états TCP

278/307



Ces états sont visualisables sous Windows dans une fenêtre de commande avec la commande `netstat -p tcp`. Sous Linux on fera `netstat -at`.



Les ports sont, en quelque sorte, les adresses des applications à l'intérieur des machines. Selon la terminologie OSI, un port est un SAP.

Quelques ports bien connus (voir aussi le fichier `/etc/services` sous Unix/Linux, ou `C:\WINNT\system32\drivers\etc\services` sous Windows)

21 : ftp (le port du canal de commande de ftp)

20 : ftp-data (le port pour la phase effective du transfert de fichiers par ftp)

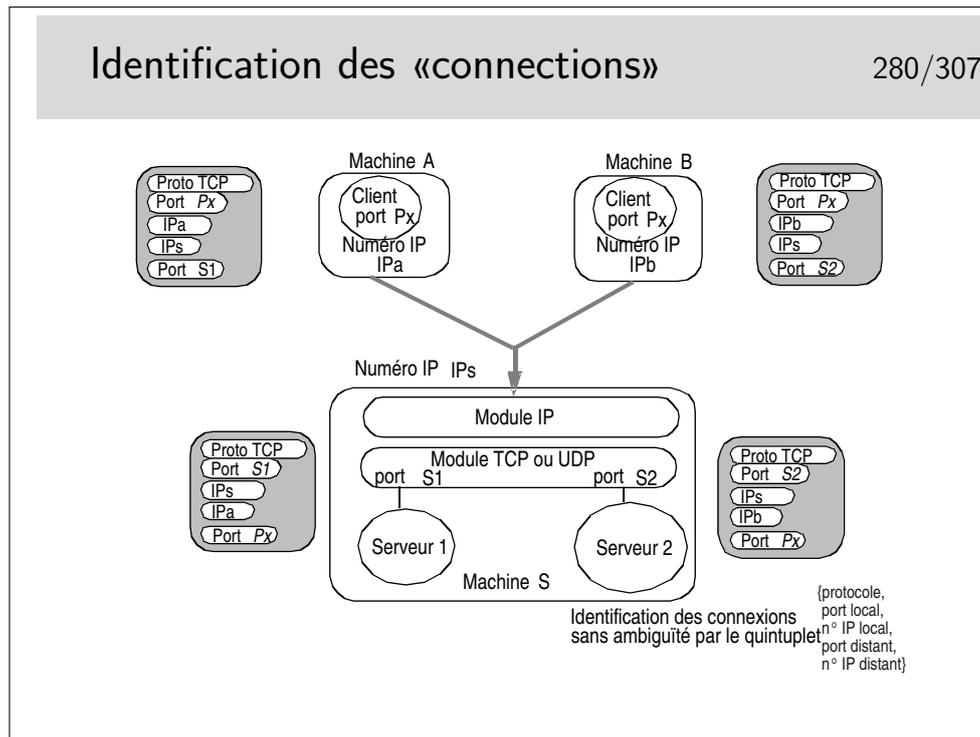
22 : ssh (les connexion à distance sécurisée par chiffrement)

23 : telnet : la même chose mais sans chiffrement

25 : le protocole d'envoi du courrier électronique Internet, SMTP

80 : le protocole http, donc le WEB

etc...



Pour résumer ce qu'est une «connexion TCP» :

- c'est une relation entre deux entités TCP résidant sur des machines différentes (ou entre deux contextes de la même couche TCP d'une machine où s'exécutent les applications en communication). En terminologie OSI, dans la couche Application on utilise le mot «association» plutôt que «connexion». Il serait plus approprié ici aussi.
- il n'y a pas de connexion associée dans le réseau
- dans chaque entité TCP en relation un contexte est créé et est associé à l'application ne cause de son coté
- ce contexte comprend :
 - une machine d'état finis gérant les états de la connexion
 - un quintuplet d'identification de connexion
 - des tampons mémoire d'émission et de réception
 - diverses variables complémentaires comme, entre-autre, le compteur associé à la gestion des congestions
- C'est l'application qui demande la création de ce contexte, à l'aide de fonctions d'une API spécifique telle que les Sockets (API issue du monde Unix, BSD au départ et ensuite reprise par tous les Unix et portée sous Windows)

▶ Caractéristiques

- ▶ Le champ offset de l'entête étant codé sur 4 bits (il indique le nombre de mots de 32 bits de l'entête), la longueur max de l'entête est de 15 mots de 32 bits (15dec = 1111bin). L'entête minimum étant de 5 mots de 32 bits (20 octets), la longueur du champ option est de 10 mots de 32 bits max.

▶ Options standards

- ▶ mss : *maximum segment size*, permet de négocier la taille maximale des segments TCP afin d'éviter des segmentations coûteuses. cette option est utilisées au moment de la connexion. Longueur 4 octets.
- ▶ pas d'opération (NOP) : option nulle, permet d'aligner la prochaine option sur un début de mot de 32 bits. Plusieurs NOP peuvent se suivre si besoin. (lg : 1 octet)
- ▶ fin de liste : permet d'aligner la fin des options sur un mot de 32 bits (lg 1 octet)

- ▶ Multiplicateur du champ fenêtre (window scale)
 - ▶ valeur permettant de multiplier la fenêtre par un multiple d'une puissance de 2 (jusqu'à 2^{14}) (rfc 1323). Ceci permet d'utiliser plus efficacement la bande passante pour des sources à haut débit et/ou des communications à grande distance en permettant un flux le plus continu possible
- ▶ Horodatage des données (rfc 1323)
 - ▶ L'option contient deux valeurs : un indicateur d'heure d'émission et un acquittement. Une source peut ainsi calculer le temps d'aller et retour sur un chemin en plaçant sa valeur d'horloge dans le premier champ. Lorsque l'acquittement du segment correspondant arrivera, cette valeur sera contenue dans le second champ de cette option. Il sera alors facile de retrancher cette valeur de la valeur courante de l'horloge.

- ▶ Négociation de l'acquittement sélectif (rfc 2018)
- ▶ Acquittement sélectif : la première option permet d'indiquer qu'une source TCP est capable d'utiliser cette fonctionnalité, la seconde met en œuvre ce type d'acquittement permettant de demander uniquement la retransmission de données perdues en évitant de retransmettre des données suivantes bien reçues

rfc-1323 et rfc 2018 (relevé avec tcpdump lors d'un début de requête web)

```
linux1.1082 > 206.132.41.203.www: S 2047350264:2047350264(0) win 16060 <mss
1460,sackOK,timestamp 43244276>
206.132.41.203.www > linux1.1082: S 2319802134:2319802134(0) ack 2047350265 win
32120 <mss 1460, sackOK, timestamp 190973015>
linux1.1082 > 206.132.41.203.www: . ack 2319802135 win 16060 <nop,nop,timestamp
43244311 190973015>
linux1.1082 > 206.132.41.203.www: P 2047350265:2047350896(631) ack 2319802135 win
16060 <nop, nop, timestamp 43244311 190973015>
206.132.41.203.www > linux1.1082: . ack 2047350896 win 31856 <nop, nop, timestamp
190973051 43244311>
206.132.41.203.www > linux1.1082: P 2319802135:2319803583(1448) ack 2047350896 win
31856 <nop, nop, timestamp 190973063 43244311>
linux1.1082 > 206.132.41.203.www: . ack 2319803583 win 14612 <nop,nop,timestamp
43244367 190973063>
206.132.41.203.www > linux1.1082: P 2319803583:2319805031(1448) ack 2047350896 win
31856 <nop, nop, timestamp 190973063 43244311>
```

S : bit Syn

ack : bit ack

P : bit Push

. : pas de flag (autre que ack)

sackOK : acquittements sélectifs en fonction

mss : maximum segment size

en italique : complément d'information fourni par tcpdump mais ne figurant pas réellement dans le paquet

numéro de séquence du dernier octet, correspondant à l'acquittement alors attendu.

Entre parenthèses : nombre d'octets du paquet

Les trois premières trames correspondent à une ouverture de connexion (Three Way Handshake)

- ▶ La commande netstat avec l'option `-a` (sous Unix/linux, ou `-p tcp` sous Windows)

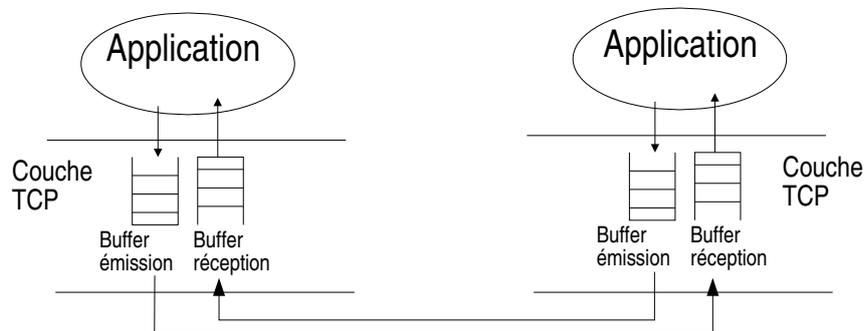
```
[linux30]$ netstat -atn
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale Adresse distante Etat
tcp    0      0 0.0.0.0:32768   0.0.0.0:*      LISTEN
tcp    0      0 127.0.0.1:32769 0.0.0.0:*      LISTEN
tcp    0      0 0.0.0.0:111    0.0.0.0:*      LISTEN
tcp    0      0 0.0.0.0:6000   0.0.0.0:*      LISTEN
tcp    0      0 0.0.0.0:22     0.0.0.0:*      LISTEN
tcp    0      0 0.0.0.0:631    0.0.0.0:*      LISTEN
tcp    0      272 192.168.100.30:22 192.168.100.18:1994 ESTABLISHED
```

L'option `-n` de netstat permet de ne pas traduire les adresses et numéros de ports (les services). L'affichage est direct et on ne perd pas de temps en requêtes DNS pour afficher les noms des machines.

L'option `-t` restreint l'affichage au seul protocole TCP. Avec l'option `-u` on affiche les services UDP en cours (les applications serveur UDP). Il n'y a pas de connexion en UDP.

Avec la seule option `-a` on affiche toutes les connexions (ou serveurs lancées pour UDP), y compris les services n'utilisant que la communication Unix (par exemple le serveur d'affichage X-Window local et ses applications clientes).

Voir `-p tcp` ou `-p udp` sous Windows



Les données sont bufferisées et TCP les émet selon ses propres règles de contrôle de flux. Les applications n'ont pas le contrôle.

Selon la bibliothèque de développement (sockets sous Unix/Linux ou Windows par exemple) il peut être possible de dimensionner les tampons mémoire d'émission et de réception.

- ▶ Contrôle de flux d'extrémité à extrémité : le champ window
 - ▶ Les routeurs ne mémorisent pas "la connexion" TCP, ils ne peuvent donc pas participer au contrôle de flux. Celui-ci ne peut être réalisé que par les extrémités via le champ fenêtre.
 - ▶ Les entités TCP d'extrémités peuvent enregistrer les données reçues dans un tampon mémoire de taille limitée (8, 16, 32 Ko, ...)
 - ▶ Si l'application réceptrice ne lit pas suffisamment vite les octets s'accumulent dans le tampon mémoire de réception.

.../...

- ▶ L'entité TCP réceptrice envoie des acquittements avec une valeur de fenêtre qui diminue pour venir à 0 s'il le faut.
- ▶ 5s après avoir reçu un tel acquittement ($win=0$), l'entité émettrice teste le récepteur en lui envoyant un octet et continue ainsi en doublant l'intervalle (jusqu'à une borne de 1 min). Ce moyen permet de forcer le récepteur à renvoyer un acquittement et donner ainsi sa fenêtre. La valeur de 5s est en réalité variable selon les implémentations.
- ▶ Dès que les octets reçus par l'entité TCP réceptrice seront consommés par l'application réceptrice, la fenêtre pourra revenir à une valeur différente de 0.

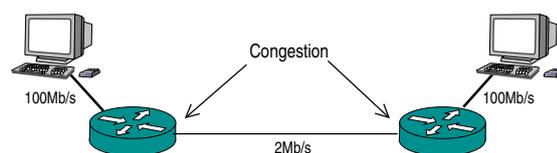
Sur Linux (noyau 2.6.xx), les paramètres liés à la gestion de la taille de la fenêtre de réception dans la pile TCP sont accessibles par la commande :

```
$ cat /proc/sys/net/ipv4/tcp_rmem  
4096 87380 2076672
```

Les trois nombres donnent respectivement la valeur minimale (4096 octets), par défaut (87380 octets) et maximale (2076672 octets) de la fenêtre d'une session TCP.

Le problème de la congestion

- ▶ Il n'y a pas que les entités d'extrémités en jeu, il y a aussi le réseau. Le mécanisme de fenêtre ne permet pas d'adapter le contrôle de flux aux congestions du réseau.
- ▶ S'il y a congestion dans le réseau, des paquets peuvent être perdus, les retransmissions qui en découleront participeront au renforcement de la surcharge totale.
- ▶ Le congestion peut être due à des liens saturés mais aussi à des différences de débits entre segments de réseau.



Comment adapter TCP ?

Adaptation de TCP pour le contrôle de congestion I

291/307

- ▶ Détermination du RTT (*Round Trip Time*)
 - ▶ Avec les options spécifiques
 - ▶ Permet de «régler» un temporisateur de retransmission

Adaptation de TCP pour le contrôle de congestion II

292/307

- ▶ Le mécanisme du «*slow start*»
 - ▶ Une fenêtre de congestion est définie : *cwnd* (*congestion window*). On ne peut pas émettre plus que *cwnd* octets.
 - ▶ Juste après la connexion *cwnd* vaut 1mss (*max segment size*).
 - ▶ Un RTT plus tard, ce nombre double, et double à chaque RTT jusqu'à un seuil fixé à l'origine à 65535 (on ne peut cependant émettre que $\min(cwnd, w)$, *w* étant la valeur de la fenêtre de l'entête TCP).
 - ▶ Lorsqu'une congestion a lieu, une perte se produit, le temporisateur expire et provoque une retransmission, le seuil est divisé par 2 et *cwnd* est remis à 1mss.
 - ▶ Si le seuil est dépassé, la progression de *cwnd* devient linéaire.

Les valeurs de RTT sont typiquement de :

10 ms entre sites intra Renater,

35 ms entre des sites intra Europe,

90 ms entre des sites situés en France et aux USA

210 ms entre des sites situés en Europe et en Asie

- ▶ Le mécanisme du «*fast recovery*» (entre autres) vient compléter le low start
 - ▶ Si on reçoit un même acquittement plusieurs fois, cela signifie deux choses : d'une part c'est qu'on a bien envoyé des segments et que ceux-ci ont été bien reçus (sinon on n'aurait pas reçu d'acquiescement du tout) et d'autre part il doit y avoir un «trou» dans la séquence de segments transmis. Il suffit alors de ne retransmettre que le segment qui semble perdu.
 - ▶ Lorsque que plusieurs pertes successives ont lieu, ce mécanisme ne suffit plus car il ne permet que de récupérer le premier segment perdu. Il faut alors utiliser le mécanisme des acquittements sélectifs.

Comment éviter la segmentation avec TCP 294/307

- ▶ Le mécanisme du *Path MTU Discovery* (rfc 1191)
- ▶ le problème
 - ▶ la couche TCP émission connaît le MTU de l'interface IP de sortie (1500 par défaut sur Ethernet). Sur le chemin vers le récepteur un lien peut avoir un MTU inférieur (700 par exemple), le routeur amont sur ce lien devra segmenter.
- ▶ la solution
 - ▶ les routeurs sont interdits de segmentation TCP (bit D à 1). Si un paquet se présente de taille trop grande il est rejeté et le routeur qui le rejette émet un paquet ICMP vers l'émetteur. Ce paquet contient une information significative. Voir exemple :

Exemple de PATH MTU Discovery

295/307

Exemple : extrait d'un relevé avec tcpdump

- ▶ émission du premier paquet (seq = 1, taille 1448=MTU(1500)-TCPhead(20)-TCPOptions(12)-IPhead(20))

```
Em > Rec P 1:1449(1448) ack 1 win 16060 <nop,nop,timestamp 830140 827243>
```

- ▶ réception du message ICMP émis par un routeur intermédiaire (celui qui devrait segmenter)

```
192.168.200.17 > Em: icmp: Rec unreachable - need to frag (mtu 700)
```

- ▶ On a bien compris... On émet des paquets de 648 octets (700-12-20-20=648)

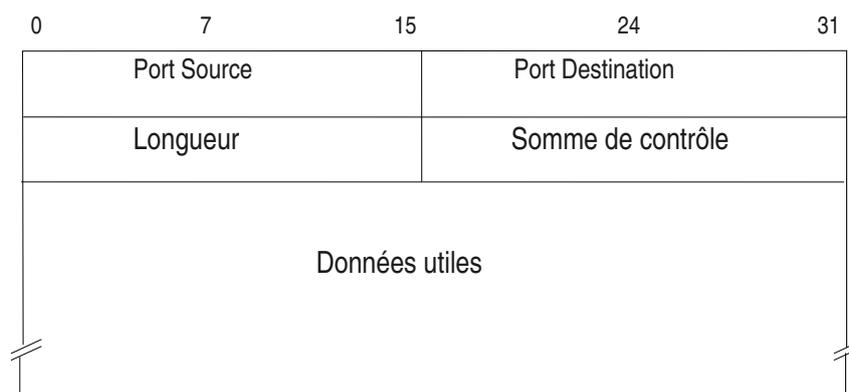
```
Em > Rec . 1:649(648) ack 1 win 16060 <nop,nop,timestamp 830140 827243>
```

En italique : ce que tcpdump nous indique concernant la longueur du SDU véhiculé.

19 UDP

Le datagramme UDP

297/307



- ▶ Pas de connexion
- ▶ Pas de contrôle de flux
- ▶ Pas d'assurance de la remise
- ▶ Mode message : alignement des données reçues sur les données émises

UDP n'apporte que peu de choses par rapport à IP. Il garde de TCP la notion de port qui permet d'identifier les applications.

(A noter le champ longueur qui n'existe pas dans TCP)

Il n'y a pas de contrôle de flux, pas d'assurance de remise.

Il n'y a pas de «connexion».

On a cependant l'assurance que, si un datagramme arrive, il correspond très exactement à ce qui a été envoyé en ce qui concerne l'alignement des données. En d'autres termes, on peut dire que UDP est orienté «message». Si un message est reçu, le début correspond au début envoyé, la fin à la fin...

Ce n'est pas vrai en TCP. Lorsqu'une application reçoit un buffer de données via TCP elle ne peut être certaine que cette réception correspond bien à l'émission. Certes les données sont les mêmes, elles ne sont pas alignées. TCP est orienté «flot d'octets», il assure que les octets sont bien transmis.

- ▶ TCP est orienté «flot d'octets», il ne permet pas d'assurer le «cadrage» des données reçues sur celui des données émises

Exemple de possibilité :



- ▶ UDP est orienté «Message»



TCP garantit que les octets sont bien transmis, il ne permet pas l'alignement de la réception sur l'émission. Autrement dit, un premier «message» peut voir son début partir dans un segment TCP et arriver dans un autre. Il faut pouvoir rassembler les octets appartenant à un message donné à l'arrivée. Pour cela on peut utiliser la technique TPKT recommandée dans le RFC 1006 qui consiste à commencer les messages par un octet de version (3), un octet inutilisé (sans doute pour faire joli, ou plus sûrement pour que l'ensemble soit aligné sur 4 octets), et deux octets de longueur de message. En réception il suffit de se synchroniser sur le premier message...

UDP ne garantit pas que les messages arrivent mais s'ils sont reçus, alors, on trouve tout d'un coup (à condition toutefois que la lecture prévoit de lire un nombre suffisant d'octets).

20 Protocoles applicatifs

Les protocoles applicatifs I

301/307

- ▶ Mis en œuvre par les applications
- ▶ Des API existent
 - ▶ Sockets : le réseau est vu comme un fichier (on écrit et on lit le réseau comme un fichier), le protocole de communication est à implémenter «à la main»
 - ▶ RPC : l'API masque les aspects réseau, on appelle des procédures distantes de la même manière que des procédures locales
 - ▶ CORBA : *Common Request Broker Architecture*, extension aux applications réparties du concept «programmation objet»

Les protocoles applicatifs II

302/307

- ▶ Codage des données
 - ▶ Protocoles «texte» : smtp, http (1.x), sip, sdp
 - ▶ Protocoles codés :
 - ▶ ASN.1/BER/PER : H323, snmp
 - ▶ XDR : NFS, NIS
- ▶ Notions de «session»
 - ▶ p.ex. protocole sip, sdp, beep, et même les *cookies* sur le web, etc.

smtp : simple mail transfert protocol (rfc-821)

http : hyper text transfert protocol (le protocole du Web, rfc-1945)

sip : session initiation protocol (un des protocoles pour la téléphonie sur IP, rfc-2543)

sdp : session description protocol (descripteur de flux multimedia, inspirateur de sip : rfc-2327)

beep : blocks extensible exchange protocol, un framework pour créer des protocoles applicatifs, incluant des mécanismes d'authentification et de gestion de session, rfc 3080 3081 3117

ASN.1 : Abstract Notation, version 1 (on espère qu'il n'y aura pas de version 2, ou alors plus simple...) : langage de spécification de types de données. Comme un langage informatique ou on ne définirait que des types et des variables de ces types.

On lui associe une syntaxe de transfert (en clair : des mécanismes de codage en ligne) tels que le BER (Basic Encoding Rules) ou le PER (Packet encoding Rules).

ASN.1/BER : utilisé pour snmp (simple network management protocol, rfc 1155 à 1158, 1212, 1213, etc)

ASN.1/PER est utilisé pour définir les données de signalisation pour la téléphonie sur IP dans l'architecture H323 (définition des PDUs H225 et H245, plusieurs centaines de pages, avec rien que du ASN.1 goûteux et savoureux).

On retrouve aussi ASN.1 dans les spécifications de la structure des certificats de sécurité X509.

Exemple d'échange HTTP : la requête

303/307

```
GET /index.fr.php HTTP/1.1
Host: www.enst-bretagne.fr
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.
Gecko/20030624 Netscape/7.1 (ax)
Accept: text/xml,application/xml,.....
Accept-Language: fr,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://www.enst-bretagne.fr/
```

```
HTTP/1.1 200 OK
Date: Fri, 09 Jul 2004 09:20:06 GMT
Server: Apache/1.3.22 (Unix) PHP/4.0.4pl1 mod_fastcgi/2.2.10 PHP/3.0.....
X-Powered-By: PHP/4.0.4pl1
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html
Content-Language: fr

<HTML>
<HEAD>
<TITLE>[ENST Bretagne] école formation ingénieur</TITLE>
<!-L'école nationale supérieure des télécommunications de Bretagne offre
un large choix de formation: ingénieur, mastères, télécom, thèse, DEA...->
<meta name="robots" content="noindex,nofollow">
<meta name="description" content="L'école nationale supérieure des télécommunications de Bretagne offre un
large choix de formation: ingénieur, mastères, télécom, thèse, DEA...">
<meta name="keywords" content="école ingénieur, formation ingénieur, école nationale supérieure
télécommunications, Bretagne, mastères, télécom, thèse, DEA, ENST, GET, étude télécommunication,
enseignement supérieur, technologie information, communication, TIC, concours, mines-ponts, alternance,
projets européens, recherche, laboratoire, grandes écoles, formation continue, diplôme, Brest, Rennes,
ECOLE INGENIEUR, FORMATION INGENIEUR">
<LINK rel="stylesheet" href="css/style.css">
...
```

Coté client, c'est au programmeur de l'application de préparer «à la main» les chaînes de caractères constituant les messages de requêtes du protocole. Coté serveur il faut savoir interpréter ces messages (le mot «savoir» étant traduisible en «implémentation de code»), il faut savoir aussi constituer les messages de réponse. Ces messages sont donc des suites de caractères ascii, placés dans des tampons mémoire qu'il suffit ensuite d'envoyer sur des sockets, coté émission. Coté réception il suffit de lire les sockets (comme on lirait des fichiers) et d'analyser en,suite ce qui est mémorisé dans les tampons de lecture.

Le langage est simple, il obéit à une grammaire (au sens informatique du terme) parfaitement spécifiée. Il est possible, à partir des spécifications de générer facilement des fonctions de traitement des messages. Il existe des outils adaptés aux traitements lexicaux, voire sémantiques, de telles grammaires : lex et yacc sous unix, flex et bison en logiciels libres.

(Exemple extrait du rfc-821)

```
S: MAIL FROM:<Smith@Alpha.ARPA>
R: 250 OK

S: RCPT TO:<Jones@Beta.ARPA>
R: 250 OK

S: RCPT TO:<Green@Beta.ARPA>
R: 550 No such user here

S: RCPT TO:<Brown@Beta.ARPA>
R: 250 OK

S: DATA
R: 354 Start mail input; end with <CRLF>.<CRLF>
S: Blah blah blah...
S: ...etc. etc. etc.
S: <CRLF>.<CRLF>
R: 250 OK
```

S: indique l'émetteur, **R:** indique le récepteur. (**S:** et **R:** ne font pas partie du protocole.)

MAIL FROM:, **RCPT TO:**, **DATA** sont des entêtes du protocole, de même que les séquences **<CRLF>** qui signifient simplement *retour à la ligne* pour **CR** (Carriage Return) et *saute de ligne* pour **LF** (Line Feed).

On peut noter que le cacatère **.** (point) est aussi partie intégrante du protocole.

Voici donc le protocole de base du courrier Internet si utilisé aujourd'hui. Ses spécifications datent d'août 1982!

Que pensez-vous des aspects sécurité d'un tel protocole ?

- ▶ Les utilisateurs ont une adresse de format standard
 - ▶ Par ex. :
 - ▶ nom_utilisateur@nom-du-serveur.domaine
 - ▶ ou plus concis :
 - ▶ nom_utilisateur@domaine
- ▶ Les outils mis en œuvre sont
 - ▶ Le client (MUA : *Mail User Agent* en langage OSI) : netscape, outlook, lotusNotes, etc...
 - ▶ Parle SMTP lors de l'envoi
 - ▶ Parle POP ou IMAP (sécurisé ou non) pour recevoir
 - ▶ Les serveurs (MTA : *Message Transfert Agent*) : ISS, sendmail, Postfix, etc...
 - ▶ Intermédiaires et final
 - ▶ Le DNS : requêtes MX (*Mail Exchanger*)

POP : Post Office Protocol

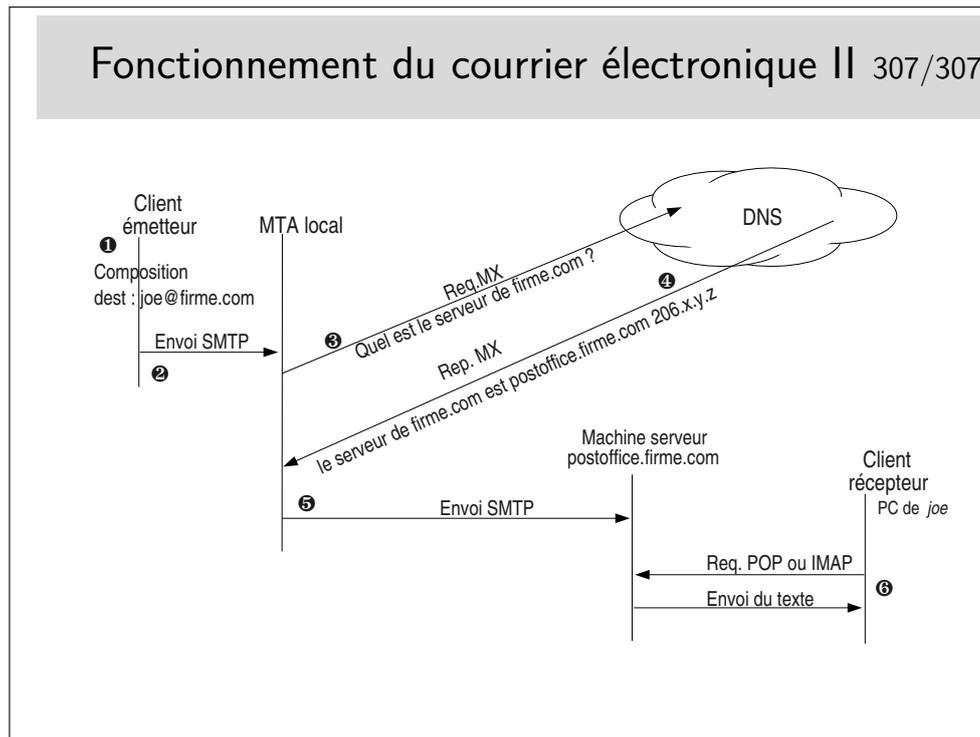
IMAP : Internet Message Access Protocol

Les protocoles POP et IMAP sont par défaut sans chiffrement et nécessitent l'échange de mots de passe utilisateur. Ces échanges se font par défaut en clair pour POP. IMAP supporte différents mécanismes d'authentification, notamment un système de challenge dans lequel le mot de passe est chiffré (pour autant, les messages ne le sont pas...) Les outils clients ainsi que les serveurs peuvent être configurés pour fonctionner avec chiffrement (POPS ou IMAPS : POP/IMAP + SSL).

POP permet de télécharger sur son poste client les courriers reçus sur le serveur final. Les messages sont totalement transférés. Ils peuvent cependant rester sur le serveur.

IMAP permet de ne charger que les entêtes, il permet aussi de rechercher les messages par mot clé, les messages peuvent rester stockés sur le serveur, et de gérer des sous-répertoires (*folders*) sur le serveur.

Fonctionnement du courrier électronique II 307/307



Il peut y avoir plus de machines intermédiaires que ne le montre la figure ci-dessus. En particulier, coté réception, la machine serveur (appelée ici **postoffice.firme.com**) peut renvoyer vers un serveur interne déservant le département de l'utilisateur joe, ce dernier paramètre alors son client pour qu'il se connecte par POP ou IMAP sur ce serveur particulier.

Il peut y avoir plusieurs serveurs d'arrivée, classés selon des priorités dans le DNS.

Il peut être intéressant de regarder l'entête complète des messages que l'on reçoit pour avoir la liste des MTA traversées par ces messages. Cette curiosité étant encore plus intéressante en cas de problèmes d'acheminement.

Les MTAs peuvent filtrer les messages, au départ comme à l'arrivée. Ils peuvent être munis de détecteurs de SPAMs et de virus.

Vous pouvez tester le DNS avec des requêtes MX de la manière suivante (sous Windows) :

```
c:\nslookup
> set type=MX
> microsoft.com
```

Ou sous Unix : `host -t MX microsoft.com` (la commande `nslookup`, bien qu'encore présente, est maintenant considérée comme rendue obsolète par la commande `host`).