LI310 - Cours
Réseaux

PLISSET Charline & BARON Benjamin

Table des matières

Chapitre 1

Traitement du signal

1.1 Généralité sur les supports de transmission

Une **ligne de transmission** raccorde deux équipements de communication. On désigne généralement par le terme émetteur l'équipement qui envoie les données et par récepteur celui qui les reçoit. Les équipements de communication peuvent parfois être chacun à leur tour ou simultanément récepteur et émetteur (c'est le cas généralement des ordinateurs reliés en réseau). Différents types de support de transmission :
– support métallique ;
– support optique ;
– support radio.

Supports de transmission utilisés pour la communication entre machines :
– téléphone : paire torsadée ;
– internet : fibre optique, paires torsadées ;
– télévision : câble coaxial ;
– sans fil : air, ...

Critères de comparaison : bande passante, atténuation, sensibilités diverses, coût, facilité d'installation, ...

1.2 Propagation du signal et ondes

La transmission de données sur un support physique se fait par propagation d'un phénomène vibratoire.
Il en résulte un signal ondulatoire dépendant de la grandeur physique que l'on fait varier :
– Onde électrique dans le cas de la tension ou de l'intensité d'un courant électrique ;
– Onde électromagnétique dans le cas d'un champ électrique et d'un champ magnétique couplés (lumière, onde radio, ...).

Définition (Onde)**.** Une onde est une fonction de la variable temps t et de la variable espace x.

Exemple. Le champ électrique d'une onde électromagnétique qui se propage suivant un axe (Ox) peut être représenté par une fonction sinusoïdale de la forme :

$$E(x,\, t) = A \sin\left(2\pi f_0 \left(t - \frac{x}{c}\right) + \varphi\right)$$

où le paramètre c représente la vitesse de propagation de l'onde, encore appelée célérité (c est égal à 300 000 km/s pour une onde électromagnétique qui se propage dans le vide ou dans l'air).
La période T de ce signal et sa fréquence f sont :

$$T = \frac{c}{f_0}, \qquad f = \frac{1}{T} = \frac{f_0}{c}$$

Remarque. Pour la fonction $s\colon \nu \longrightarrow A\sin(2\pi a\nu)$, la plus petite période T telle que

$$A\sin(2\pi a(\nu+T)) = A\sin(2\pi a\nu)$$

est $T = \frac{1}{a}$.

Pour une valeur de x fixée, cette onde est donc représentée par un signal sinusoïdal $A\sin(2\pi a\nu + \varphi)$, caractérisé par trois paramètres :

– L'**amplitude** A qui est la valeur crête du signal dans le temps, la puissance moyenne est elle égale à

$$\frac{A^2}{2}$$

– La **fréquence** f_0 qui est la vitesse à laquelle le signal se répète, exprimée en nombre de cycles par seconde ou en Hertz (Hz) (l'inverse de la fréquence est appelé période T du signal et se mesure en secondes :

$$T = \frac{1}{f_0}$$

– La **phase** φ qui est une mesure de la position relative dans le temps à l'intérieur d'une période du signal, exprimée en radians (rad).

Remarque. En pratique, le signal qui se propage est constitué de plusieurs composantes sinusoïdales de fréquences , amplitudes et phases différentes.

Exemple. Soit s le signal ayant pour équation :

$$s(t) = \sin(2\pi f_0 t) + \frac{1}{3}\sin(2\pi(3f_0)t)$$

Il a pour représentation graphique :

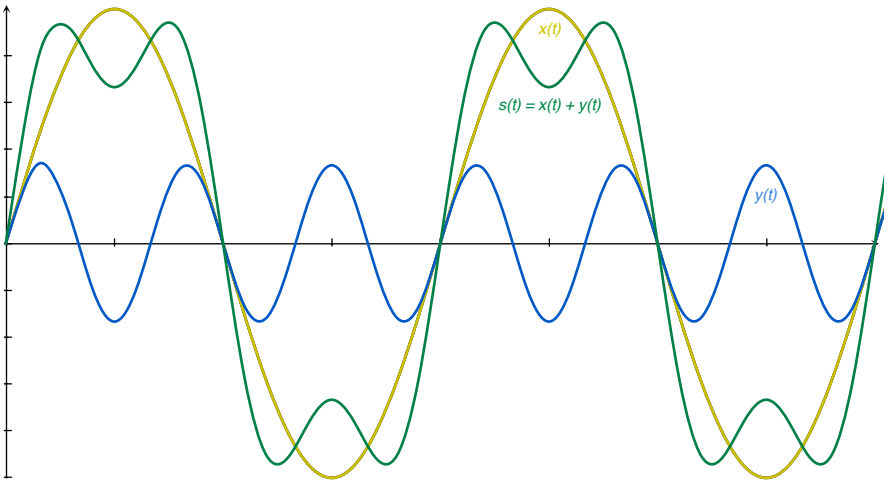


FIGURE 1.1: Représentation graphique de l'onde $s(t)$

Remarque. De manière générale si l'on additionne à un signal périodique de période T un autre signal périodique de période T/n, le signal résultant est périodique de période T.

1.3 Domaine temporel / Domaine fréquentiel

1.3.1 La décomposition en séries de Fourier

Toute fonction périodique $g(t)$ de période T peut se décomposer en une somme (éventuellement infinie) de fonctions sinus et cosinus.

$$g(t) = c_0 + \sum_{n=1}^{\infty} a_n \cos(2\pi n f_0 t) + \sum_{n=1}^{\infty} b_n \sin(2\pi n f_0 t)$$

où

$$\begin{aligned} a_n &= \frac{2}{T}\int_{\tau}^{\tau+T} g(t)\cos(2\pi n f_0 t)\ dt \\ b_n &= \frac{2}{T}\int_{\tau}^{\tau+T} g(t)\sin(2\pi n f_0 t)\ dt \\ c_0 &= \frac{1}{T}\int_{\tau}^{\tau+T} g(t)\ dt \end{aligned}$$

avec :
- f_0 la fréquence fondamentale,
- a_n et b_n sont les amplitudes sinus et cosinus de la $n^{\text{ième}}$ harmonique,
- c_0 est la composante continue du signal.

Une telle décomposition est appelée **Série de Fourier**.

Remarque. On a aussi :

$$g(t) = c_0 + \sum_{n=1}^{\infty} \sqrt{a_n^2 + b_n^2}\cos(2\pi n f_0 t + \varphi_n)$$

avec $\varphi_n = \arg(a_n + ib_n)$

1.3.2 Spectre fréquentiel et largeur spectrale

Le spectre fréquentiel associé à une fonction périodique $g(t)$ est un *spectre de raies*, chaque raie correspondant à la fréquence d'une harmonique de la décomposition en série de Fourier de $g(t)$.

Il existe trois types de spectres fréquentiels :
- Le **spectre d'amplitude** pour lequel la hauteur de chaque raie est égale à $\sqrt{a_n^2 + b_n^2}$ (leurs carrés sont proportionnels à la puissance transmise à la fréquence correspondante),
- Le **spectre de puissance** où chaque raie a une hauteur égale à $a_n^2 + b_n^2$,
- Le **spectre de phase** où chaque raie possède une hauteur égale à $\arctan({}^{b_n}/_{a_n})$.

1.4 Bande passante

1.4.1 Supports de transmission

Les supports physiques de transmission sont les éléments permettant de faire circuler les informations entre les équipements de transmission.

On classe généralement ces supports en trois catégories, selon le type de grandeur physique qu'ils permettent de faire circuler, donc de leur constitution physique :
- Les supports **métalliques** permettent de faire circuler une grandeur électrique sur un câble généralement métallique ;

– Les supports **optiques** permettent d'acheminer des informations sous forme lumineuse ;
– Les supports **aériens** désignent l'air ou le vide ; ils permettent la circulation d'ondes électromagnétiques diverses.

Selon le type de support physique, la grandeur physique a une vitesse de propagation plus ou moins rapide.

Exemple. Le son se propage dans l'air à une vitesse de l'ordre de 300 m/s alors que la lumière a une célérité proche de 300 000 km/s

Un support de transmission ne peut transmettre que dans une bande de fréquence limitée.

Définition (Bande passante)**.** On appelle *bande passante* du support la bande de fréquences $[f_1,\, f_2]$ $(f_1 < f_2)$, que le support laisse passer sans (trop de) déformation.
La largeur de cette bande passante est généralement notée $B = f_2 - f_1$.

Or la transmission du signal complet requiert une largeur de bande infinie. La bande passante du support limite le débit qui peut être véhiculé sur le canal de transmission.

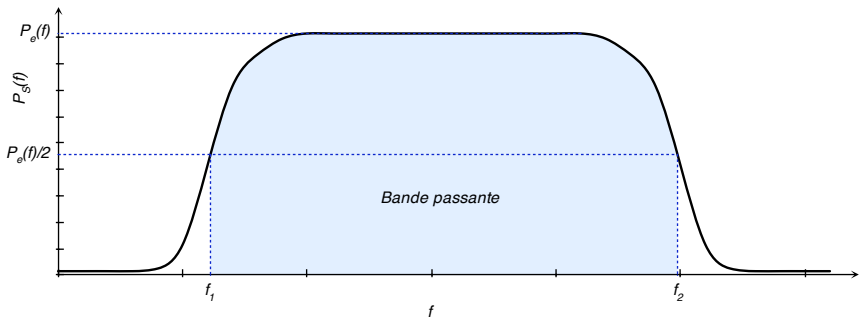


FIGURE 1.2: Représentation graphique de la bande passante $[f_1,\, f_2]$

1.4.2 Perturbations du signal

On distingue généralement 3 types de perturbations du signal :
– **Affaiblissement de propagation** : perte de signal en énergie dissipée sur le support : Puissance reçue < Puissance émise. Plus le signal se propage, plus il est affaibli.
– **Distorsion** : déformation subie par le signal – certaines composantes fréquentielles sont plus atténuées que d'autres.
– **Bruit** : perturbation aléatoire qui se rajoute au signal sur tout support. On différencie :
  – Les *bruits blancs* : perturbation uniforme du signal – ajoute au signal une petite amplitude de moyenne nulle ;
  – Les *bruits impulsifs* : petits pics de forte intensité et de faible durée ajoutés au signal.

Définition (Rapport Signal/Bruit)**.** Le rapport signal-à-bruit se calcule en faisant le quotient entre la puissance du signal P_S et la puissance du bruit P_N. Il est exprimé en décibels (dB) :

$$\left(\frac{S}{N}\right)_{dB} = 10 \cdot \log_{10}\left(\frac{P_S}{P_N}\right)$$

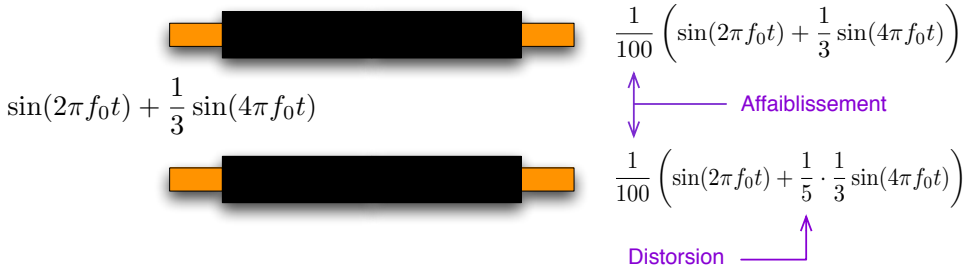


FIGURE 1.3: Exemples de perturbations

1.4.3 Bande passante et capacité

La loi de Shannon fournit le **débit binaire maximum** (*capacité*) auquel on peut théoriquement transmettre **sans erreur** sur un canal à bande passante limitée sujet à du bruit.

Loi (Shannon)**.** Un canal de transmission bruité de largeur de bande B a pour capacité :

$$C = B\log_2\left(1 + \frac{P_S}{P_N}\right) \qquad \text{(bit/s)}$$

où P_S/P_N représente le rapport signal-à-bruit du canal.

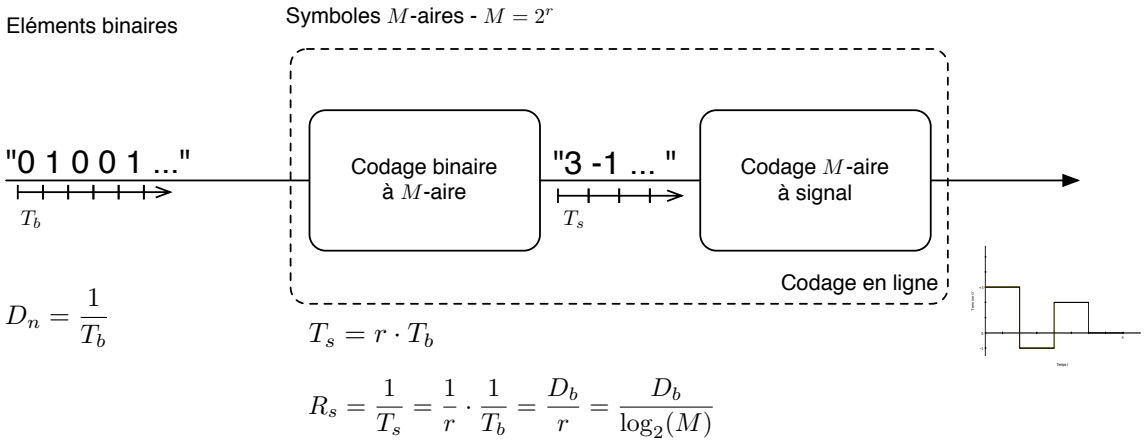


FIGURE 1.4: Codage d'éléments binaires

Loi (Nyquist)**.** La rapidité de modulation R_S d'un canal de largeur de bande B doit vérifier :

$$R_S \leqslant 2B \qquad \text{(symb/s)}$$

Le débit binaire D_b transmis sur un canal de largeur de bande B doit donc vérifier :

$$D_b \leqslant 2B\log_2 M \qquad \text{(bit/s)}$$

Chapitre 2

Techniques de transmission et de protection contre les erreurs

2.1 Message, signal et modes de transmission

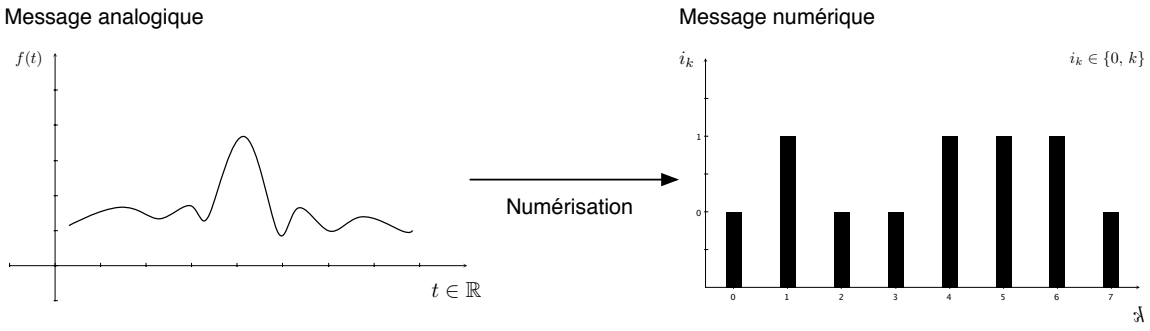


Figure 2.1: Schéma de la numérisation

Définition (Message)**.** Un message est constitué par les éléments d'information / données que l'usager soumet.

- Message analogique : données se présentent sous la forme d'une fonction $f(t)$ continue.
- Message numérique : les données se présentent sous la forme d'i,e suite $\{i_k\}_{k\in\mathbb{N}}$ d'éléments d'information pouvant prendre une valeur parmi un ensemble fini de valeurs discrètes – alphabet.

Définition (Signal)**.** Un signal est la représentation physique du message à transmettre. C'est soit une onde électrique ou électromagnétique.

- Signal analogique : représente un message analogique.
- Signal numérique : signal résultant de la mise en forme d'un message numérique.

Définition (Transmission)**.** La transmission est l'opération qui consiste à transporter le signal d'une machine vers une autre sur un support donné. On distingue la transmission analogique de la transmission numérique, suivant la nature du signal transmis.

- Transmission en bande de base : canal de type passe-bas (ie. de bande passante $[0,\ f_2]$). Il n'y a pas de modulation.
- Transmission sur fréquence porteuse : canal de type passe-bande (ie. de bande passante $[f_1,\ f_2]$). Le signal est alors modulé.

Exemple. Un amplificateur est utilisé pour une transmission analogique. Il sert à amplifier (booster) le signal.

Un répéteur est utilisé en transmission numérique. Il sert à :

- Décoder numériquement le signal (ie. retrouver les 0 et 1 transmis initialement).
- Reformer le signal sans bruit et amplifié.

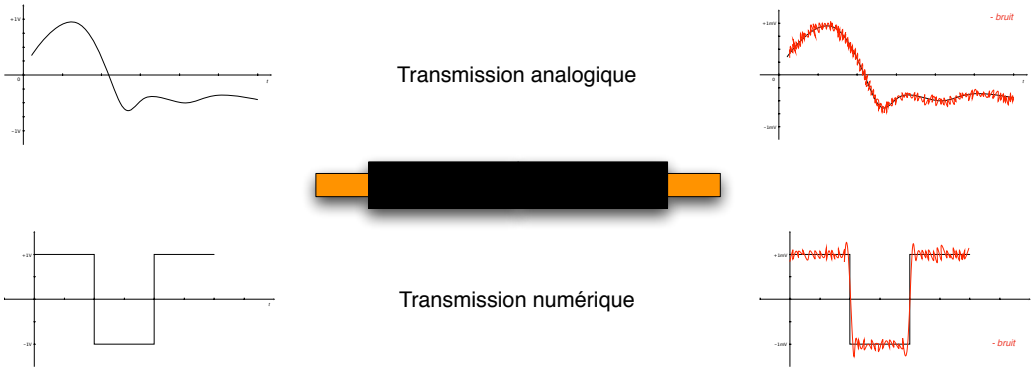


FIGURE 2.2: Transmission d'un signal analogique ou numérique

2.2 La numérisation

Définition (Numérisation)**.** La numérisation est l'opération qui consiste à transformer un message analogique en un message numérique qui pourra ensuite être traité numériquement.

Les trois étapes de la numérisation sont :
- **L'échantillonnage** : consiste à transformer un signal à temps continu en un signal à temps discret par un prélèvement périodique (de période T_e), à la fréquence d'échantillonnage, de la valeur du signal analogique ;
- **La quantification** représente un échantillon par une valeur numérique appartenant à une échelle de quantification. L'erreur de quantification est d'autant plus importante que le nombre de niveaux de quantification est faible et que le pas de quantification est grand ;
- **Le codage** consiste à remplacer la suite des valeurs quantifiées des échantillons par une suite binaire. S'il y a niveaux de quantification, chaque échantillon est codé sur n bits.

Le théorème d'échantillonnage de Shannon exprime le nombre minimum d'échantillons par seconde nécessaire pour reconstituer un signal à partir de ses échantillons.

Théorème (Shannon)**.** L'échantillonnage d'un signal de fréquence de spectre maximum $f_{\max}$ est sans perte si :

$$f_e > 2 \cdot f_{\max}$$

Pas d'échantillonnage (période) :

$$T_e = \frac{1}{f_e}$$

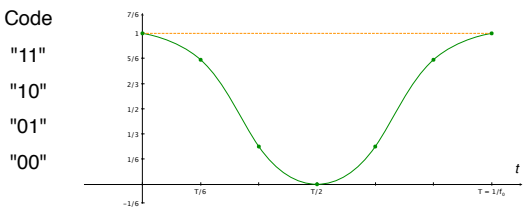


FIGURE 2.3: Quantification d'un signal analogique

2.3 Codage en ligne et transmission en bande de base

Principe (Codage en ligne)**.** Le codage en ligne consiste à associer une représentation physique au message numérique, avant d'effectuer une transmission numérique en bande de base (BdB) sur un support de longueur limitée.

Cette opération est parfois improprement appelée modulation en bande de base.

Le codage en ligne se décompose en 2 opérations :
- La première transforme la suite d'éléments binaires $\{d_k\}_{k\in\mathbb{N}}$ correspondant au message numérique à émettre en une suite de symboles.
  C'est l'opération de codage binaire M-aire.
- La seconde opération consiste à associer une forme d'onde à chaque symbole ainsi produit.
  C'est l'opération de codage M-aire à signal.

Définition (Débit binaire)**.** Débit binaire (D_b) : nombre maximum d'éléments binaires transmis par seconde.

T_b étant la durée de transmission d'un élément binaire, on a :

$$D_b = \frac{1}{T_b} \qquad \text{bit/s}$$

Définition (Rapidité de modulation)**.** Rapidité de modulation (R_s) : vitesse à laquelle les symboles se succèdent.

T étant la durée de transmission d'un symbole (et donc la durée d'un élément de signal), on a :

$$R_s = \frac{1}{T_S} \qquad \text{symb/s (ou bauds)}$$

Définition (Valence)**.** Valence (M) : cardinal de l'alphabet des symboles.

r étant le nombre de bits codés par symbole, on a :

$$M = 2^r \Leftrightarrow r = \log_2(M) \qquad \text{bits}$$

Remarque. La rapidité de modulation R_s se relie au débit binaire D_b par la relation suivante :

$$R_s = \frac{D_b}{r} = \frac{D_b}{\log_2(M)} \qquad \text{symb/s (ou bauds)}$$

ou encore

$$D_b = R_s.r = R_s.\log_2(M) \qquad \text{bit/s}$$

2.4 Différents code en ligne

2.4.1 Code NRZ (Non Return to Zero)

Principe.
- $d_k = 0$, le signal vaut -V ;
- $d_k = 1$, le signal vaut V.

Avantages :
- La détection de la présence ou non du signal ;
- $M = 2$: bonne résistance aux bruits.

Inconvénients :
- Spectre de puissance concentré autour de la fréquence nulle, coupée par de nombreux supports ;
- Présence d'un courant continu lors d'une suite de 0 ou de 1, gênant la synchronisation entre émetteur et récepteur.

2.4.2 Code Manchester (ou biphase)

Principe.
– $d_k = 0$, le signal est un front *montant* au milieu de l'intervalle T_S ;
– $d_k = 1$, le signal est un front *descendant* au milieu de l'intervalle T_S.

Avantages :
– Spectre ne contenant pas la fréquence nulle ;
– Non passage par zéro, rendant possible par le récepteur la détection d'un signal ;
– Au moins une transition par intervalle.
Inconvénient :
– Spectre de puissance concentré autour de la fréquence nulle, coupée par de nombreux supports.

2.4.3 Code Manchester différentiel

Principe.
– $d_k = 0$, le signal force une transition (front montant ou descendant) au début de l'intervalle T_S et une autre au milieu de l'intervalle T_S ;
– $d_k = 1$, le signal est un front *descendant* au milieu de l'intervalle T.

Avantages et inconvénients identiques au code Manchester.

Avantage par rapport au code Manchester :
– Une transition code ou non l'information, donc la polarité des fils n'a pas besoin d'être repérée.

2.4.4 Code bipolaire simple

Principe.
– $d_k = 0$, le signal vaut 0 ;
– $d_k = 1$, le signal vaut alternativement +V ou -V.

Convention d'initialisation : amplitude de +V.

Avantage :
– Spectre limité.
Inconvénients :
– Plus sensible au bruit que les codages à 2 niveaux ;
– Problèmes d'horloge avec les suites de 0.

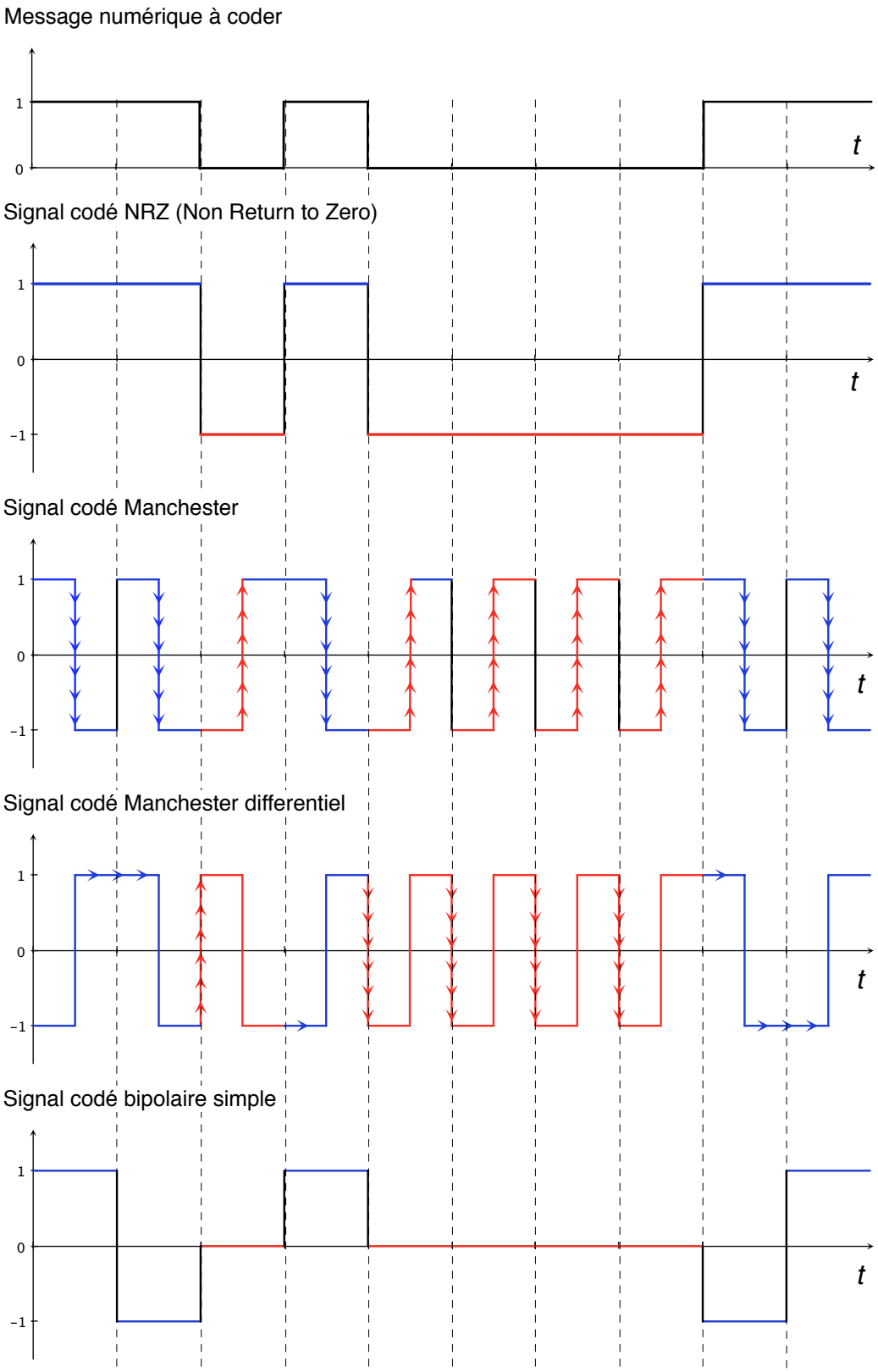


FIGURE 2.4: Les différents types de codage d'un signal

2.5 Modulation et transmission sur fréquences porteuses

Lorsque le canal est de type passe-bande, comme dans le cas de la transmission hertzienne, il est indispensable de transposer le spectre du signal autour d'une fréquence porteuse située au centre de la bande passante du canal. On réalise ainsi une transmission sur fréquence porteuse.

Principe (Modulation)**.** La technique de modulation permet de décaler le spectre du signal, naturellement situé autour de la fréquence nulle (signal en bande de base), autour d'une fréquence porteuse f_0.
Elle est réalisée en modifiant (modulant) une des caractéristiques (amplitude, phase, fréquence instantanée) d'une porteuse sinusoïdale de fréquence f_0 en se servant du signal porteur de l'information à transmettre, appelé signal modulant. Lorsque le signal modulant est analogique on parle de modulation analogique.

Les modulations consistent à modifier l'amplitude, la phase ou la fréquence d'une porteuse sinusoïdale de fréquence f_0 en fonction de la valeur du symbole à transmettre sur chaque intervalle de durée T_S. Il existe donc trois grands types de modulation :
- **la modulation par déplacement d'amplitude** (MDA, ASK *Amplitude Shift Keying*) qui associe à chaque symbole à coder une amplitude différente ;
- **la modulation par déplacement de phase** (MDP, PSK *Phase Shift Keying*) qui associe à chaque symbole à coder une phase différente ;
- **la modulation par déplacement de fréquence** (MDF, FSK *Frequency Shift Keying*) qui associe à chaque symbole à coder une fréquence différente.

Il existe aussi **la modulation par déplacement d'amplitude** (MDAP) et de phase ainsi que **la modulation d'amplitude en quadrature** (MAQ).

Le chiffre associé au sigle d'une modulation correspond au nombre total d'états possibles du signal modulé. Chaque état est associé à un symbole, le nombre d'états est donc égal à la valence M.

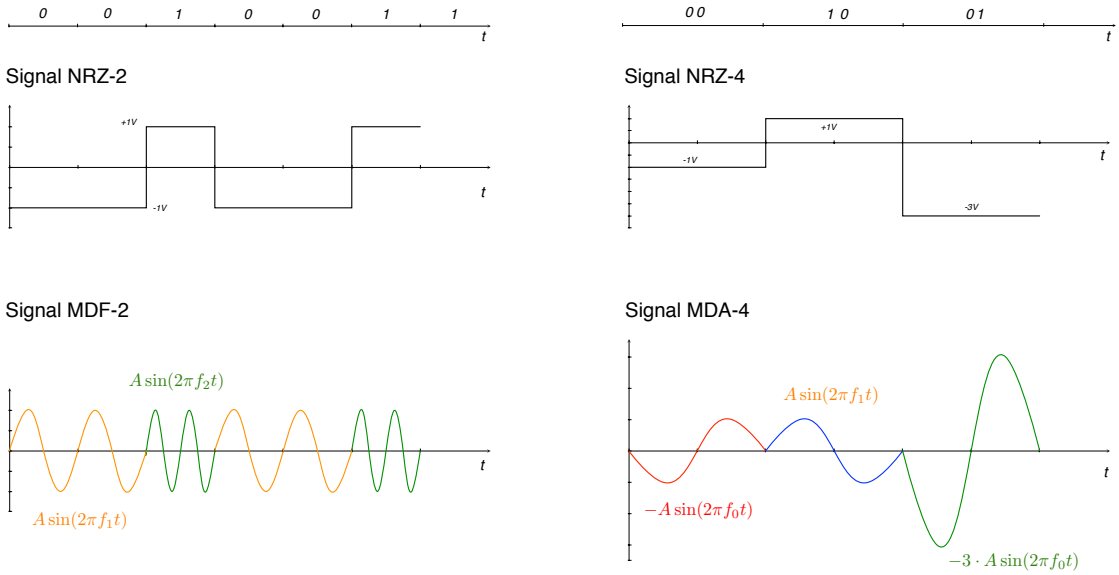


FIGURE 2.5: Signal transmis sur fréquence porteuse

2.6 Contrôle d'erreur

Lors de la transmission des données, un signal peut subir diverses déformations et être altéré par du bruit. A la réception, lors du décodage, une erreur d'estimation peut alors se produire : l'état du signal représentant le bit « 1 » est interprété comme s'il s'agissait d'un bit « 0 », et vice versa.

Plus le débit est élevé plus le risque que le bruit affecte plusieurs bits est grand.

Définition (Taux d'erreur binaire)**.** Le taux d'erreur binaire (TEB) d'une transmission est défini comme le nombre moyen de bits transmis en erreur divisé par le nombre total de bits transmis.

Exemple. Quelques taux d'erreur binaire :
– $TEB_{\text{ligne téléphonique}} = 10^{-5}$
– $TEB_{\text{câble coaxial}} = 10^{-7}$ à 10^{-8}
– $TEB_{\text{fibre optique}} = 10^{-10}$ à 10^{-12}

Les techniques employées, pour effectuer un contrôle d'erreurs, reposent sur l'utilisation de *codes détecteurs* ou de *codes correcteurs* d'erreurs qui enrichissent la suite de bits à envoyer, en lui ajoutant de la redondance. Le récepteur se sert de cette information ajoutée pour déterminer si une erreur s'est produite et éventuellement pour la corriger.

Définition (Mot)**.** Pour m *bits de données*, on ajoutera r *bits de redondance* pour permettre la détection et éventuellement la correction d'erreurs. On transmettra au total $n = m + r$ bits.
– Mot **légal** : les bits de contrôle correspondent aux bits de données ;
– Mot **illégal** sinon. La réception d'un mot de code illégal permet de détecter la présence d'une erreur.

On a alors :
– Nombre de mots de code possibles : 2^n ;
– Nombre de mots de code légaux possibles : 2^m ;
– Nombre de mots de code illégaux possibles : $2^n - 2^m$.

2.6.1 Codes de détection d'erreurs basés sur la parité

Prenons un mot de 7 bits, on lui ajoute un bit de contrôle de valeur telle que le nombre total de bits à 1 du code du caractère est pair. C'est ce qu'on appelle un bit de parité ou **VRC (Vertical Redundancy Checking)**.
Pouvoir de détection d'erreur : 1 par mot.

Exemple. Soit le mot

1 0 0 1 1 1 1

Le mot obtenu par VRC est alors :

1 0 0 1 1 1 1 1

On utilise aussi la parité bidimensionnelle ou **LRC (Longitudinal Redundancy Checking)**.
Pouvoir de détection d'erreur : 3
Pouvoir de correction d'erreur : 1

Exemple. Soit le code

1 0 0 1 1 1 1
1 0 1 0 0 1 1
1 0 0 0 0 1 1

Le code obtenu par LRC est alors :

1	0	0	1	1	1	1	1
1	0	1	0	0	1	1	0
1	0	0	0	0	1	1	1
1	0	1	1	1	1	1	0

2.6.2 Codes de détection d'erreurs polynomiaux

A un code polynomial C est associé un polynôme générateur (par ex : $g(x) = x^8 + 1$). Le codage est la calcul du mot de code :
- On constitue $M(x)$ le polynôme associé à la suite binaire à transmettre ;
- On multiplie $M(x)$ par x^r, où r est le degré du polynôme générateur $g(x)$;
- On calcule $R(x)$, le reste de la division du polynôme $M(x) * x^r$ par $g(x)$;
- On calcule le mot de code $M(x) * x^r - R(x)$ et on le transmet.

Le décodage est la vérification du mot de code reçu :
- On constitue $M'(x)$ le polynôme associé à la suite binaire reçue ;
- On calcule $R'(x)$, le reste de la division du polynôme de $M'(x)$ par $R'(x)$:
  - Si $R'(x) = 0$; il n'y a pas d'erreur ;
  - Sinon il y a une erreur, on demande alors la retransmission du message.

Chapitre 3

Liaison de données

3.1 Introduction

Lorsque l'on souhaite faire communiquer deux équipements informatiques **ETTD** (Equipement Terminal de Traitement de Données), on commence par mettre en oeuvre un circuit de données, constitué d'un support de transmission et de deux **ETCD** (Equipement de Terminaison de Circuit de Données), qui permet d'émettre et/ou de recevoir des bits en série sur le support physique.

Mais avec :
- Un certain débit ;
- Un certain délai (temps de transmission et de propagation) ;
- Un certain taux d'erreurs,

le circuit de données n'est pas suffisant pour assurer un transport et une interprétation corrects de l'information entre les deux ETTD (aucun moyen de réagir à des anomalies de transmission).

Il est donc nécessaire d'ajouter une interface logique entre la partie de traitement de l'information et la partie de transmission de l'information. Cette interface, appelée procédure de commande ou protocole de communication, est chargée de fiabiliser le transfert de l'information entre deux ETTD.

La liaison de données est donc constituée du circuit de données et du protocole de communication.

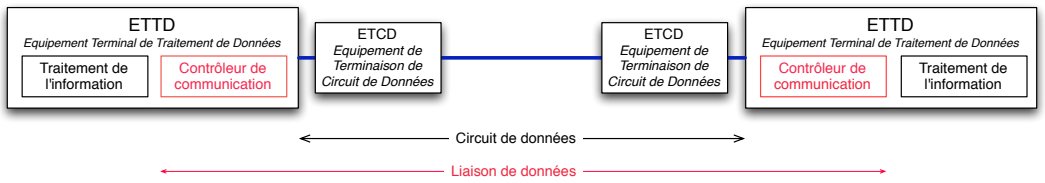


Figure 3.1: Liaison de données

Définition (Fialbilité)**.** Pas d'erreur, pas de perte, pas de déséquencement, pas de duplication.

3.2 Caractéristiques d'une liaison de données

Configuration **point-à-point** ou **multipoint**.

Mode d'exploitation **unidirectionnel**, **bidirectionnel** à l'**alternat** ou **bidirectionnel simultané**.

Mode de gestion L'échange d'information entre deux stations sur une liaison doit être réglé de façon précise afin d'éviter que des stations n'émettent simultanément en se brouillant réciproquement, ou de permettre une configuration de la réception des données transmises. Deux approches sont possibles :
- Approche **centralisée** (ou hiérarchique) : une station primaire joue un rôle particulier dans le protocole pour assurer que la communication entre plusieurs stations secondaires se passe sans problèmes ;

– Approche **symétrique** : toutes les stations ont les « mêmes droits » et les « mêmes devoirs », donc un rôle symétrique vis-à-vis du protocole de communication

3.3 Conception d'un protocole de liaison de données

Une liaison de données est constituée d'un canal physique capable de transmettre des bits en série, sur lequel sont raccordés un certain nombre (2 ou plus) de stations qui doivent pouvoir échanger de l'information. Cette information est structurée en trames.

En plus des trames d'information contenant les données, il faut gérer la liaison, pouvoir échanger des trames de contrôle (ou de commande ou de supervision).

Définition (Protocole de liaison de données)**.** Un protocole de liaison de données (ou procédure de contrôle) est un ensemble de règles permettant de gérer la liaison :
– Règle de codage des informations de contrôle ;
– Règle de structuration (séparation de l'information proprement dite de l'information de contrôle) ;
– Règle d'échange (pour préciser les séquences valides d'échanges de trames).
Ce protocole met en oeuvre un certain nombre de mécanismes de communication sur la liaison pour atteindre l'objectif de fiabilité.

3.4 Quelques mécanismes et leurs principes

La délimitation de trames

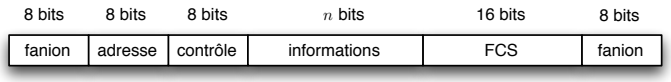


FIGURE 3.2: Une trame HDLC

On va choisir une combinaison particulière (par exemple « 01111110 ») comme délimiteur ou **fanion** (flag).

Problème Les données utilisateurs sont quelconques et peuvent en particulier contenir la combinaison retenue pour le fanion.

Solution En émission, un « 0 » (bit de bourrage) est inséré dés que cinq « 1 » binaire consécutifs apparaissent en dehors des champs fanions. Ces « 0 » sont retirés à la réception.

L'établissement et la libération de la liaison de données

Définition (Etablissement de la liaison)**.** Une fois les stations raccordées physiquement au canal de transmission elles doivent échanger de l'information de contrôle avant de pouvoir échanger les données proprement dites.

Cette information de contrôle permet en particulier aux stations de se reconnaître mutuellement et d s'assurer qu'elles sont prêtes à communiquer.

A l'issue de cette phase, il y a la phase de **transfert de données**.

Définition (Libération de la liaison)**.** Lorsque les stations n'ont plus de données à échanger, elles échangent à nouveau des informations de contrôle pour terminer le transfert de l'information.

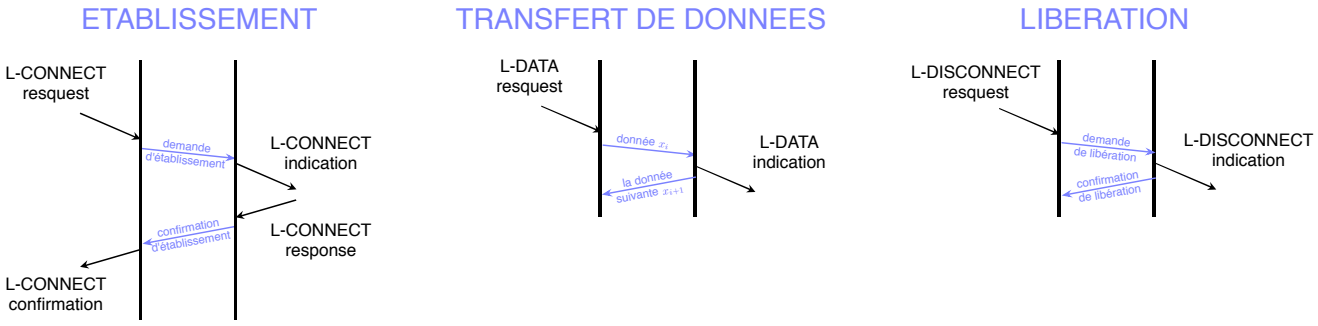


FIGURE 3.3: Transfert d'une trame

Le contrôle de flux

Il s'agit d'asservir l'émetteur de façon à ce qu'il émette à un taux inférieur aux taux d'absorption du récepteur.

Principe (*Send-and-Wait*)**.** Chaque fois qu'il a fini d'envoyer une trame, l'émetteur attend l'autorisation explicite du récepteur pour envoyer la trame suivante.

La détection d'erreur

Problème Le circuit de données n'étant pas parfait, il peut se produire des erreurs de transmission.

Solution Inclure dans la trame un champ de contrôle calculer selon une technique **CRC** (Cycling Redundancy Check) par exemple, qui permettra au récepteur de détecter les erreurs dans la trame et donc de **rejeter** la trame erronée.

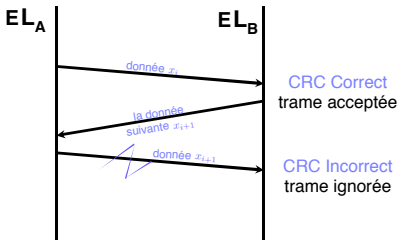


FIGURE 3.4: Détection d'erreur

Les acquittements

Si la trame est en erreur, le récepteur doit demander une retransmission de la trame. Si elle est correcte le récepteur doit autoriser l'émetteur à passer à la trame suivante.

NAK Acquittement négatif pour signaler une erreur de transmission et demander une retransmission.

ACK Acquittement positif, demande la transmission de la trame suivante.

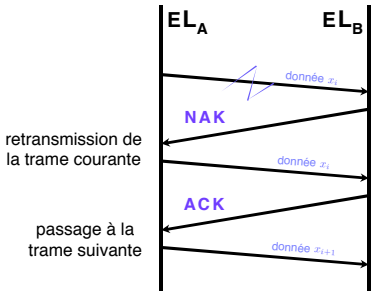


FIGURE 3.5: Acquittement

Le temporisateur de retransmission

Pour éviter que l'émetteur ne reste indéfiniment en attente d'une réponse hypothétique (interblocage) du récepteur, on utilise un mécanisme de temporisation.

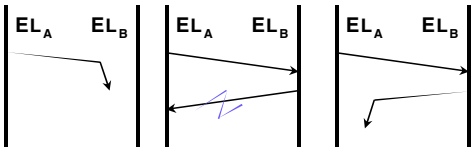


FIGURE 3.6: Problèmes de transmission de données

A l'émission d'une trame, l'émetteur arme un temporisateur dont la durée T est supérieure au délai aller-retour légèrement surestimé d'une trame (**Round-Trip Time**).

Si le temporisateur arrive à échéance avant qu'un acquittement ne soit reçu, la trame est retransmise. Dans le cas contraire, il est désarmé.

Pour éviter un nombre infini de retransmission suite à un accident grave du circuit de données, le nombre de transmission pour une même trame est limité à N. Au bout de N essais infructueux, la liaison est considérée hors service.

Remarque. La retransmission des trames suppose une rétention des trames (on garde une copie des trames non encore transmises).

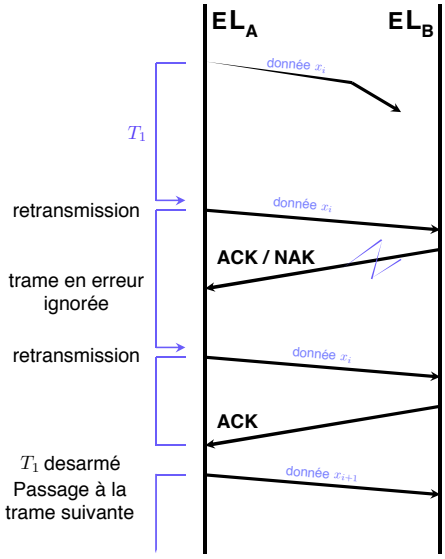


FIGURE 3.7: Temporisateur de retransmission

La numérotation des trames

Solution introduire dans la trame de données un champ véhiculant son numéro de séquence $N(S)$.

Remarque. pour une procédure *Send-and-Wait* une numérotation modulo 2 sur un seul bit suffira puisqu'il ne peut jamais avoir plus de deux trames non encore acquittées.

Il y a aussi une numérotation des acquittements (au moins les positifs). On utilise un numéro $N(R)$ qui correspond au numéro de la prochaine trame attendue par le récepteur.

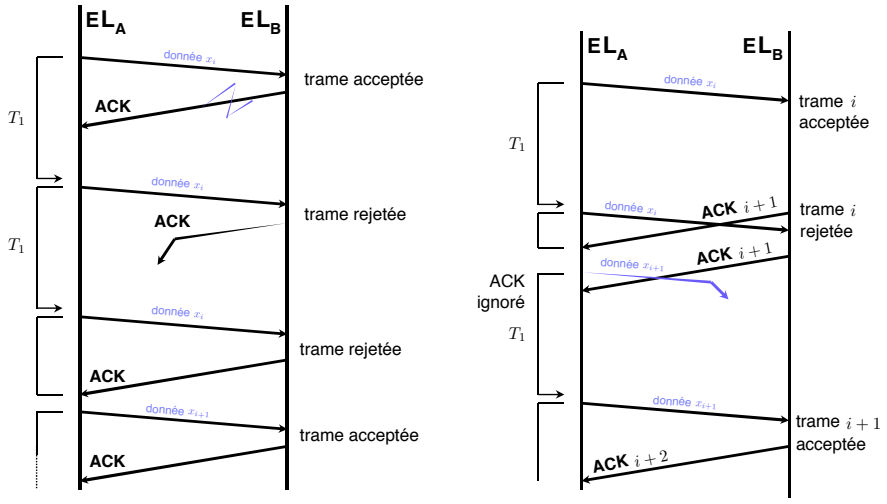


FIGURE 3.8: Numérotation de trames

Le temporisateur de détection d'inactivité

Problème Hormis les erreurs de retransmission et les pertes de trame, il peut y avoir des accidents graves comme la rupture du circuit de données.

Solution Utiliser un temporisateur I qui sera réarmé lors de toute trace d'activité du distant.

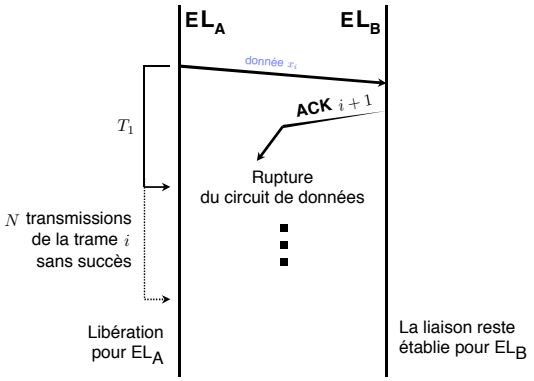


FIGURE 3.9: Inactivité du canal de transmission

La fenêtre d'anticipation

Lorsque le circuit de données présente un temps de propagation important (ex : liaison satellite) une procédure type *Send-and-Wait* ne permet pas d'utiliser d'une manière efficace la bande passante.

Une solution consiste à permettre à l'émetteur d'envoyer consécutivement W trames avant de se bloquer.

Principe (Fenêtre coulissante)**.** L'émetteur est autorisé à envoyer les trames de numéro de séquence $N(S)$ compris entre le numéro r de la prochaine trame attendue (communiqué par le récepteur dans les champ $N(R)$) et $r + W - 1$.

$$r = N(R) \leqslant N(S) \leqslant r + W - 1$$

Remarque. L'utilisation d'une fenêtre d'anticipation peut nécessiter la mise en oeuvre d'un mécanisme de régulation supplémentaire de type « tout ou rien ».

Solution Une trame de contrôle particulière est utilisée pour indiquer que le récepteur est momentanément dans l'incapacité de continuer à recevoir (ex : trame **RNR** dans HDLC – *not ready to receive*).

Principe (Mécanisme de rejet)**.** L'utilisation d'une fenêtre implique que parmi les trames non encore acquittées certaines peuvent ne pas être reçues correctement. Le récepteur doit alors rejeter ces trames.

Solutions
- Le rejet sélectif (**SREJ**) seule la trame en erreur doit être retransmise ;
- Le rejet global (**REJ**) la trame erronée ainsi que toutes les trames qui la suivent sont rejetées et doivent être retransmises.

3.5 Protocole HDLC

3.5.1 Généralités

HDLC *High-level Data Link Control*

Norme ISO
- IS 3309-2 : structure de trames ;
- IS 4335 : éléments de procédures.

Configuration
- Point-à-point ;
- Multipoint.

Exploitation
- Bidirectionnel à l'alternat ;
- Bidirectionnel simultané.

Procédure orientée bit Utilisation de trames de longueur quelconque et constituée de plusieurs champs (données ; information de contrôle ; début/fin : fanions).

Fonctionnement Mode connecté :

Mode NRM (*Normal Response Mode*)
- Liaison point-à-point ou multipoint ;
- Gestion hiérarchique (une station primaire, des stations secondaires) ;
- Exploitation par élection.

Mode ARM (*Asynchronus Response Mode*)
- Liaison point-à-point ou multipoint ;
- Gestion hiérarchique (une station primaire, des stations secondaires) ;
- Exploitation par compétition.

Mode ABM (*Asynchronous Balanced Mode*)
- liaison point-à-point ;
- gestion symétrique (stations identiques).

Remarque. Aujourd'hui, seul le mode ABM est utilisé. Il s'applique sur une liaison point-à-point. Liaison symétrique : chaque station dispose des capacités d'initialisation, de supervision et de reprise (peut envoyer trames de commande ou de réponse).

Fenêtre d'anticipation
- 7 : mode normal ;
- 127 : mode étendu.

3.5.2 Structure de la trame

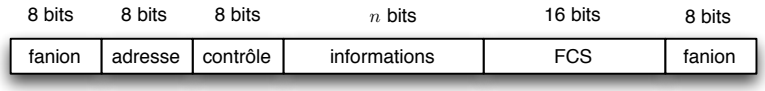


FIGURE 3.10: Une trame HDLC

Fanion (*flag*) `01111110`

- Délimitation de trame : toute trame commence et finit par un fanion ;
- Synchronisation de trame : toutes les stations rattachées à la liaison doivent rechercher en permanence cette séquence ;
- Un même fanion peut servir de fermeture pour une trame et de fanion d'ouverture pour la trame suivant.
- Mécanisme de transparence au fanion par bits de bourrage : en émission un `0` est inséré dès que cinq `1` consécutifs apparaissent en dehors des champs fanion. Ces `0` sont enlevés en réception.

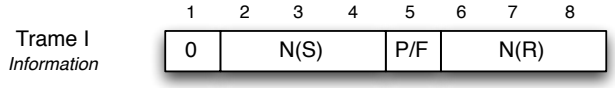
Remarque. Si sept `1` apparaissent n'import où dans une trame, elle est déclarée comme erreur.

Champ adresse Permet d'identifier la trame comme étant une commande ou une réponse :

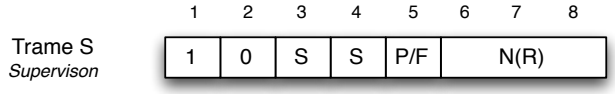
- Trame de commande : l'adresse est celle du destinataire de la trame.
- Trame de réponse : l'adresse est celle de l'émetteur de la trame.

Champ de contrôle Indique le type de la trame :

Trame I (*Information*) Mécanisme de *piggbacking* – acquittement dans les données.

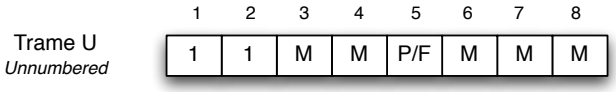


Trame S (*Supervision*) Assurer les fonctions de supervision de base de la liaison.



RR	*Ready to Recieve* $N\,(R)$	`1 0 0 0 P/F N(R)`
REJ	*Reject* $N\,(R)$	`1 0 0 1 P/F N(R)`
SREJ	*Selective Reject* $N\,(R)$	`1 0 1 0 P/F N(R)`
RNR	*Not Ready to Recieve*	`1 0 1 1 P/F N(R)`

Trame U (*Unnumbered*) Ne comporte pas de numéro – établissement et libération de la liaison.



SNRM	*Set NRM*	`1 1 0 0 P 0 0 1`
SARM	*Set ARM*	`1 1 1 1 P 0 0 0`
SABM	*Set ABM*	`1 1 1 1 P 1 0 0`
SNRME	*Set NRM Extended*	`1 1 1 1 P 0 1 1`
SARME	*Set ARM Extended*	`1 1 1 1 P 0 1 0`
SABME	*Set ABM Extended*	`1 1 1 1 P 0 1 1`
UA	*Unnumbered Acknowledge*	`1 1 0 0 F 1 1 0`
DISC	*Disconnect*	`1 1 0 0 F 0 0 1`
CMDR / FRMR	*Command* (ARM, NRN) / *Frame* (ABM) *Reject*	`1 1 1 0 P 0 1 0`
DM	*Disconnect Mode*	`1 1 1 1 F 0 0 0`

Description des champs

- Le champ $N(S)$ indique le numéro de séquence de la trame ;
- Le champ $N(R)$ indique le numéro de la prochaine trame attendue.
- Le champ P/F :
  - P (*Poll*) : demande de réponse immédiate pour les commandes ;
  - F (*Final*) : réponse au bit P ou trame finale (en NRM).
- Jusqu'à 4 trames S différentes ;
- Jusqu'à 32 trames U différentes.

Champ information

- Données de l'utilisateur ;
- Toute trame de longueur inférieure à 6 octets est non valide.

FCS (*Frame Check Sequence*) Calculé sur les champs d'adresse, de commande et d'information à partir du code polynomial V.41 : $x^{16} + x^{12} + x^5 + 1$

3.5.3 Situations d'erreur et anomalies de fonctionnement

Station temporairement saturée

- Indication d'un état d'occupation : émission d'une trame RNR avec un $N(R)$ indiquant le numéro de séquence de la première trame non acceptée.
- Indication du retour à l'état normal : émission d'une trame RR (ou d'une trame I) avec $N(R)$ indiquant le numéro de séquence de la prochaine trame attendue.

Erreurs de transmission

- Rejet des trames dont le FCS indique la présence d'erreurs de transmission ;
- **Tout se passe comme si la station n'avait rien reçu**.

Erreurs de numéro de séquence Le $N(S)$ ne correspond pas au numéro de la prochaine trame attendue : émission d'une trame de rejet :

- REJ : mécanisme de rejet non sélectif : toutes les trames qui suivent la trame erronée sont rejetées ;
- SREJ : mécanisme de rejet sélectif : seule la trame erronée est rejetée ; les autres sont mémorisées.

Chapitre 4

Commutation et multiplexage

4.1 Commutation

Commutation dans sa forme la plus simple : deux équipements reliés directement par un support de transmission en point-à-point.

Définition (Topologie)**.** Manière dont des équipements sont reliés entre eux.
Une topologie est maillée si chaque équipement est relié à l'autre.

Remarque. Si on a un ensemble de N équipements et que chacun est susceptible de communiquer avec n'importe quel autre, une topologie totalement maillée représente

$$C_N^2 = \frac{N(N-1)}{2}$$

liaisons dédiées. Il faudrait alors $N-1$ ports d'entrées/sorties.

Les équipements doivent alors être attachés à un réseau de communication (déployé le plus souvent par un opérateur) comprenant :
– Noeud d'accès : relie chaque équipement (ou station) ;
– Frontières visibles du réseau : ensemble des noeuds d'accès ;
– Noeud de communication (commutateurs) : ensemble de noeud interconnectés par des liaisons.
Le réseau doit être fortement connexe : il faut disposer de plus d'un chemin entre toute paire de commutateurs.

La topologie du réseau peut être complètement / partiellement maillée car il est souhaitable (fiabilité / robustesse) de disposer de plus d'un chemin entre toute paire de noeuds d'accès (redondance).

Définition (Commutation)**.** Fonction réalisée par des noeuds du réseau. Elle consiste à aiguiller au niveau de chaque commutateur (noeud) une communication provenant d'un canal en entrée vers un canal en sortie.

Deux familles de techniques de commutation :
– Commutation de circuits – réseau téléphonique.
– Commutation d'unités de données – réseaux informatiques :
  – Commutation de messages ;
  – Commutation de paquets ;
  – Commutation de cellules.

4.1.1 Commutation de circuit

Définition (Commutation de circuits)**.** Itinéraire physique permanent pour chaque canal de communication. Circuit n'appartenant qu'au deux entités qui communiquent :
– Le circuit doit être établi avant que les informations ne transitent ;
– Les ressources sont réservées jusqu'à l'interruption de la communication (et donc du circuit).

Avantages

- Une fois établi, le circuit va offrir un délai constant de transfert de l'information (important pour les applications en temps réel) ;
- Pas de risque de congestion du réseau.

Inconvénients

- Les ressources sont dédiées au circuit qu'il y ait ou non des échanges.
  ⇒ gachis de ressources ;
- Par manque temporaire de ressource une demande d'établissement de circuit peut être rejetée ;
- Le délai d'établissement du circuit peut être handicapant pour un certain type d'applications (avec des échanges brefs).

Temps de transfert d'un message sur N noeuds :

$$T_C = t_e + \frac{L}{D_b} + N \times t_p$$

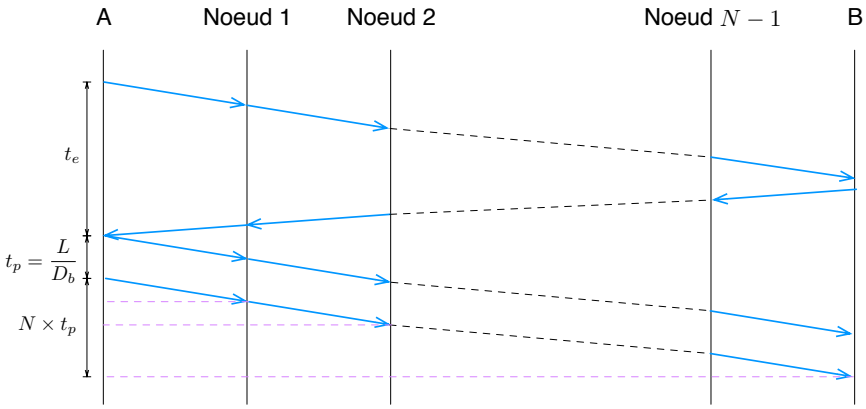


FIGURE 4.1: Commutation de circuit

4.1.2 Commutation de messages

Définition (Message)**.** Suite d'information formant un tout pour l'expéditeur et le destinataire.

Messages envoyés indépendamment les uns des autres.
Commutateur : *store and forward* – reçoit, stocke le message dans une file d'attente et le vérifie avant de le renvoyer vers le port de sortie approprié.

Avantage Permet d'utiliser les ressources du réseau uniquement lorsque c'est nécessaire.
⇒ Echanges variables, sporadiques ou par rafale.

Inconvénients

- Le message n'étant pas de taille bornée, il faudra des sources importantes de stockage ;
- Temps de transfert importants à cause de la technique *store and forward* ;
- Mécanisme de contrôle nécessaire pour éviter les congestions ;
- Du fait de leur taille parfois importante les messages on une forte probabilité de contenir des erreurs. En cas d'erreur le message entier doit être retransmis.

Temps de transfert d'un message sur N noeuds :

$$T_M = \frac{L + E_m}{D_b} + t_p + \left(t_r + \frac{L + E_m}{D_b} + t_p\right)(N - 1)$$

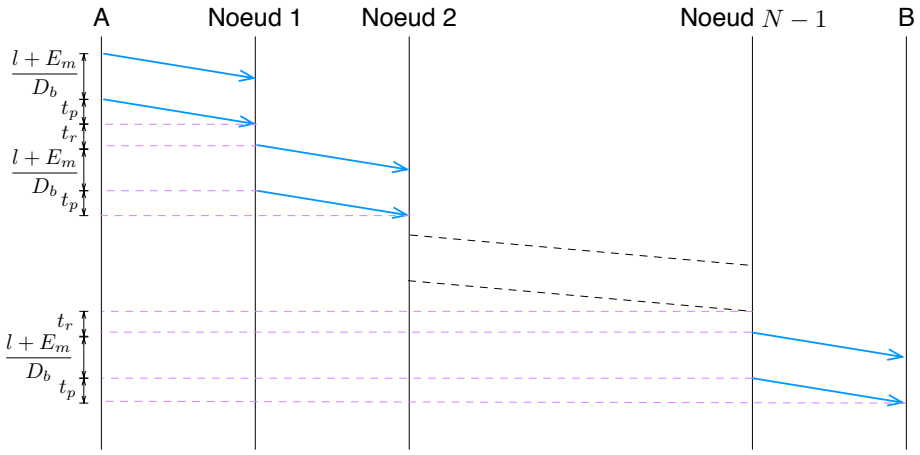


FIGURE 4.2: Commutation de messages

4.1.3 Commutation de paquets

Définition (Paquet)**.** Message de taille limitée (bornée à 1500, 2000 octets).

Le message à transmettre est découpé en paquets de données. Un paquet contient :
- Une portion du message découpé ;
- Origine, destination du message ;
- Indication sur le réassemblage du message.

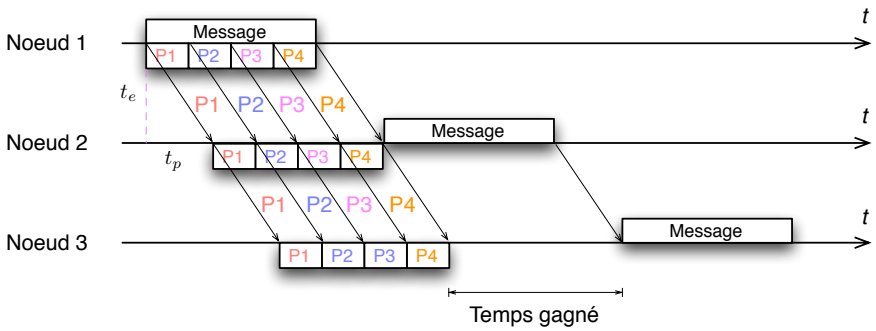


FIGURE 4.3: La commutation de message vs. la commutation de paquets

Mode connecté - circuit virtuel (CV) eg. X.25
- Etablissement du CV – réservé par le billet des voies logiques ;
- Les paquets vont alors suivre le même chemin dans le réseau.
- Les paquets arrivent dans le bon ordre.

Temps de transfert d'un paquet en mode CV sur N noeuds :

$$T_{CV} = \begin{cases} t_e + \left(\frac{l+E_p}{D_b} + t_p\right) N + \left(\frac{l+E_p}{D_b}\right)(q-1) & \text{si } r = 0 \\ t_e + \left(\frac{l+E_p}{D_b} + t_p\right) N + \left(\frac{l+E_p}{D_b}\right)(q-1) + \frac{r+E_r}{D_b} & \text{si } r \neq 0 \end{cases}$$

avec $L = l \times q + r$ et $0 \leqslant r < l$

Mode non connecté - datagramme eg. IP

– Les paquets sont indépendants les uns des autres. Ils ne vont pas suivre le même chemin dans le réseau.
⇒ Déséquencement

– La station réceptrice devra alors rassembler les paquets dans le bon ordre pour reconstituer le message.

Temps de transfert d'un paquet en mode datagramme sur N noeuds :

$$T_D = \begin{cases} \left(\frac{l+E_p}{D_b} + t_p\right) + \left(t_r + \frac{l+E_p}{D_b} + t_p\right)(N-1) + \left(\frac{l+E_p}{D_b}\right)(q-1) & \text{si } r = 0 \\ \left(\frac{l+E_p}{D_b} + t_p\right) + \left(t_r + \frac{l+E_p}{D_b} + t_p\right)(N-1) + \left(\frac{l+E_p}{D_b}\right)(q-1) + \frac{r+E_r}{D_b} & \text{si } r \neq 0 \end{cases}$$

avec $L = l \times q + r$ et $0 \leqslant r < l$

Avantages

- Permet de résoudre efficacement les erreurs de transmission, seul le paquet erroné (et non le message entier) va être retransmis ;
- Diminution du temps de transfert en augmentant le parallélisme ;
- Cela permet de multiplexer temporellement sur une même liaison les paquets appartenants à plusieurs messages.

Inconvénients

- Un dé-séquancement peut être introduit en mode non connecté ;
- Risque de congestion ;
- Délai d'acheminement/transfert variable.

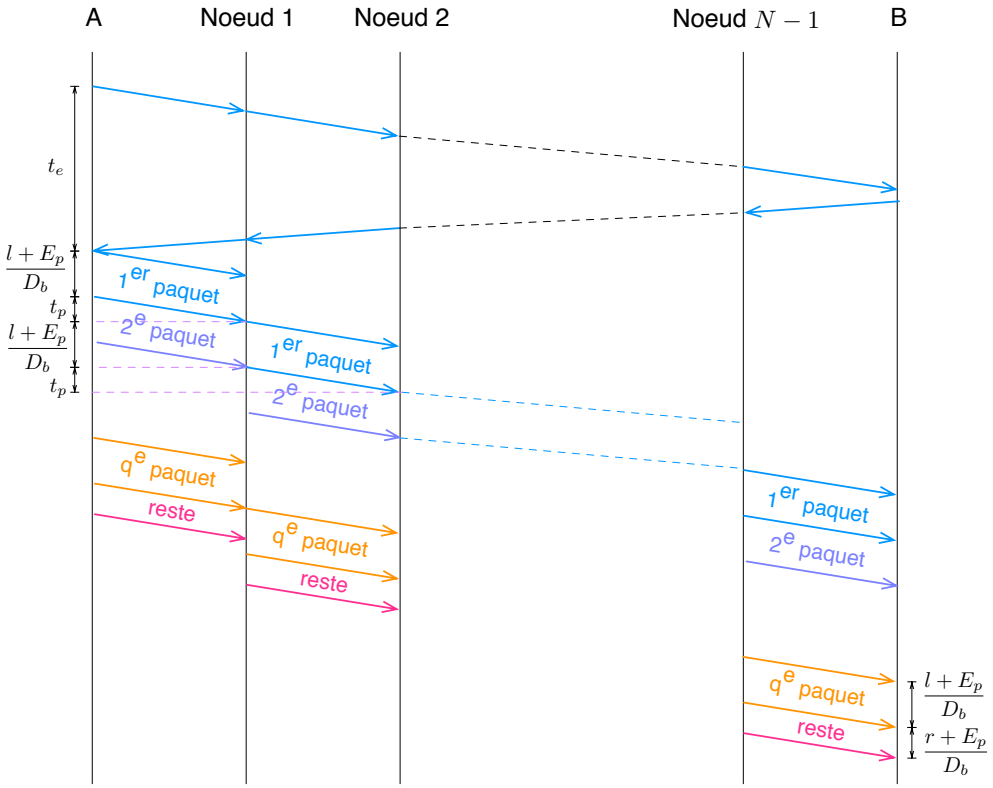


FIGURE 4.4: Commutation de paquets (mode CV)

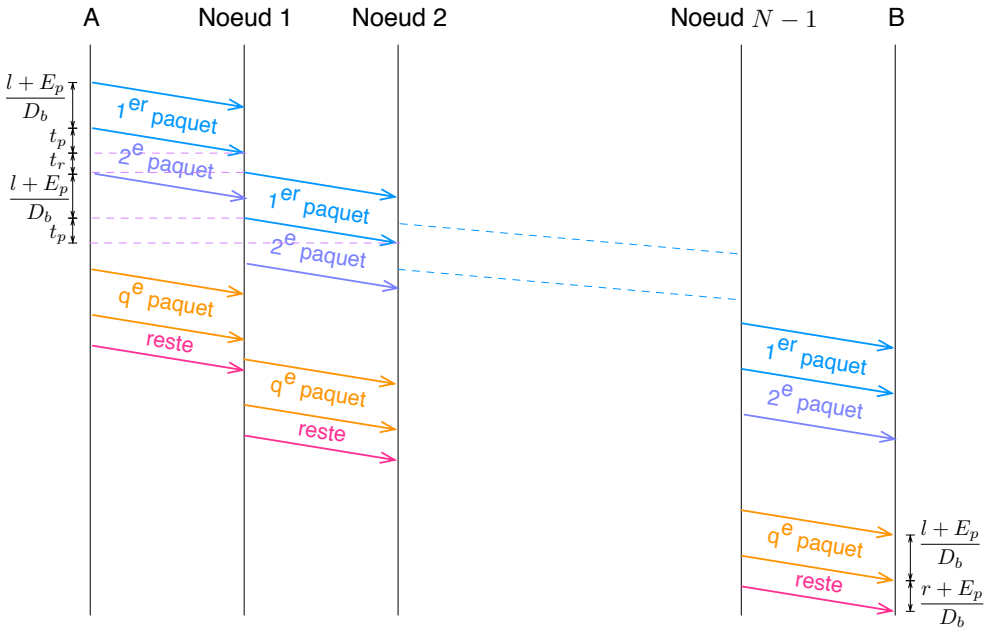


FIGURE 4.5: Commutation de paquets (mode datagramme)

4.1.4 Commutation de cellules

Définition (Cellule)**.** Une cellule a pour caractéristiques :
– Une taille fixe ;
– Une petite taille (eg. cellule ATM : 53 octets)

Avantages – taille fixe
- Augmentation de la capacité des noeuds ;
- Meilleures performances (utilisation de circuit électronique préprogrammé) ;
- Gestion mémoire des commutateurs plus simple.

Inconvénient – taille fixe Mauvaise utilisation de la bande passante : le cadrage des cellules va nécessiter des octets de bourrage.

Avantages – petite taille
- Réduction du temps de constitution des paquets/unités de données (de cellules) ;
- Réduction du délai/temps d'acheminements grâce au parallélisme ;
- Réduction du nombre de perte et de la taille de tampon des noeuds ;
- Meilleurs entrelacement des messages ;
- Gigue(écart type du temps de transfert) faible (environ 100 μs).

Inconvénients – petite taille
- Diminution de l'efficacité de transmission car chaque cellule comporte un entête $\Rightarrow$ *overhead* (débit) important ;
- Augmentation de traitements dans les noeuds de commutation.

4.2 Multiplexage statique

Le coût d'installation et de maintenance d'un lien entre deux éléments de commutation est le même pour des artères ayant une large bande passante que pour des liens de faible bande passante.

D'où l'idée du multiplexage permettant de regrouper plusieurs communications simultanées (des conversations dans le cas d'un réseau téléphonique) sur un même lien physique.

Les matériels qui réalisent cette juxtaposition ont pour rôle de regrouper les informations de plusieurs circuits de données sur un seul circuit, appelé **circuit composite**.

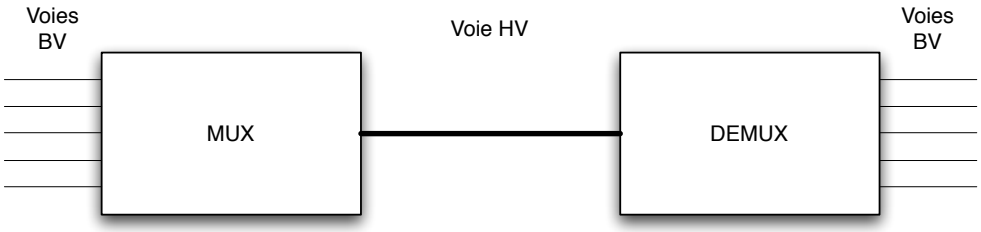


FIGURE 4.6: Principe du multiplexage

Définition (Multiplexage)**.** Le multiplexage statique consiste à partager, par une méthode invariable dans le temps, le débit binaire D_b d'une voie haute vitesse (HV) entre plusieurs voies basse vitesse (BV) ou canaux.

Remarque. La somme des débits D_i des voies BV ne peut excéder le débit D_b de la voie HV. Le multiplexeur va combiner plusieurs voies BV en un seul train de données sur la ligne HV.

Il existe deux modes de multiplexage :
– Le multiplexage fréquentiel ;
– le multiplexage temporel.
Ces deux modes de multiplexage consistent à découper la voie HV en différents canaux, chacun étant associé à une voie BV.

Définition (Signalisation)**.** En plus des données, un circuit composite doit pouvoir transmettre d'autres informations relatives à chaque circuit multiplexé. Ces informations sont appelées **signalisations**.

Deux méthodes sont possibles pour la transmission des signalisations :
– La **signalisation dans la bande** : la signalisation est transmise sur les différents canaux, à la place des données ;
– La **signalisation hors bande** : la signalisation est transmise sur un canal séparé, appelé canal sémaphore.

Remarque. Il est à noter que même si le multiplexage fréquentiel n'est plus beaucoup utilisé dans les réseaux filaires, ceux-ci utilisant de plus en plus la transmission numérique, il reste une technique très employée dans les réseaux sans fils (GSM). Il existe par ailleurs un mode de multiplexage très proche du multiplexage fréquentiel, le multiplexage en longueur d'onde. Celui-ci s'applique sur une fibre optique et consiste à partager la plage de longueurs d'ondes de la fibre en sous-bandes de longueur d'ondes disjointes.

4.2.1 Multiplexage fréquentiel (FDM)

Le multiplexage fréquentiel (Multiplexage par Répartition en Fréquence ou *Frequency Division Multiplexing*) est principalement utilisé dans les systèmes analogiques (réseau téléphonique).

Il consiste à partager la bande de fréquences disponible en un certain nombre de canaux (ou sous-bandes) plus étroits et à affecter en permanence chacun de ces canaux à une communication exclusive.

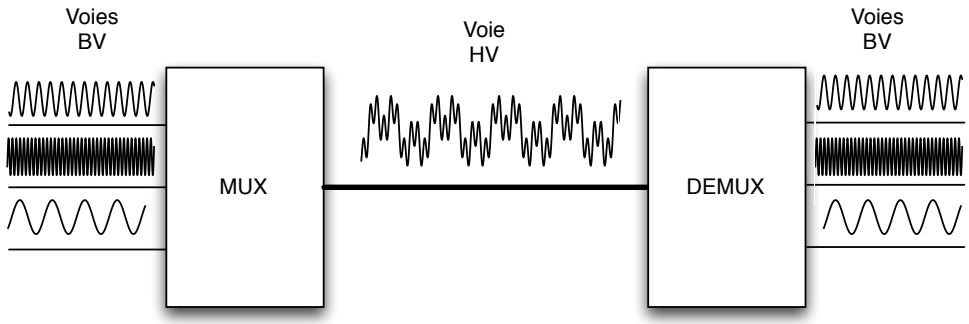


FIGURE 4.7: Principe du multiplexage fréquentiel

4.2.2 Multiplexage temporel (TDM)

Le multiplexage temporel (Multiplexage par Répartition dans le Temps ou *Time Division Multiplexing*) est mieux adapté aux signaux numériques.

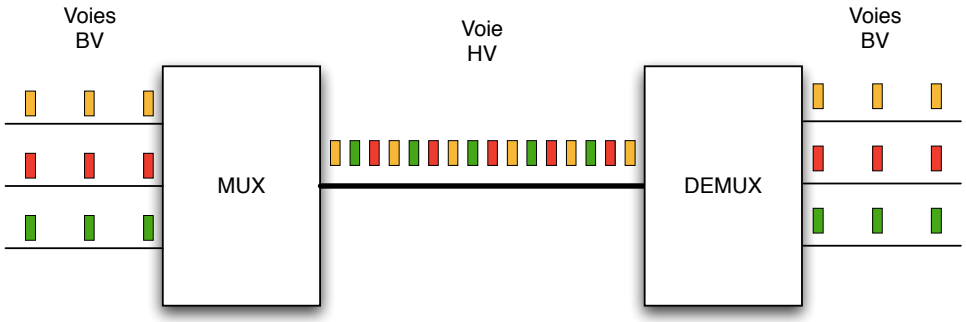


FIGURE 4.8: Principe du multiplexage temporel

Le partage des ressources se fait dans le temps. La totalité de la capacité du canal composite est allouée à un canal de communication pendant une tranche de temps fixe à intervalles réguliers.

Définition (Intervalle de temps (IT))**.** Le multiplexeur manipule des **intervalles de temps** (IT) ou *time-slot* contenant des prélèvements d'unités de données de chaque canal. Ces IT sont regroupés en une suite bornée nommée **trame multiplexée** ou **multiplex**. Cette structure de trame est répétée avec une certaine fréquence.

Remarque. Un IT est réservé à chaque canal de communication qui a la même position à l'intérieur de 2 multiplex quelconques.

Définition (Verrouillage de trame)**.** Le premier IT de chaque trame n'est affecté à aucun canal : il transmet une combinaison particulière appelée **verrouillage de trame**, qui sert à reconnaître le début d'une trame et à maintenir la synchronisation entre les deux multiplexeurs.

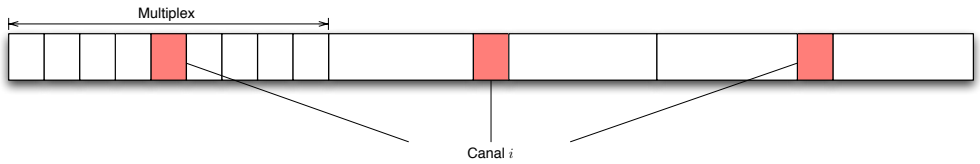
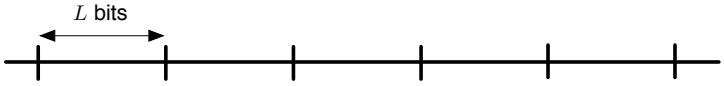


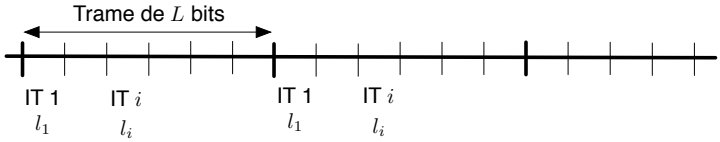
FIGURE 4.9: Un multiplex temporel

Multiplexage par bits / par caractères

On considère un train numérique permanent de débit D_b bit/s. On découpe le train en trames de L bits :



Chaque trame est ensuite découpée en intervalles de temps (IT) :



Dans un mode de **multiplexage par caractères**, chaque IT i a une longueur de l_i bits. Les différents IT peuvent avoir des longueurs différentes, mais les IT ayant la même position (i) à l'intérieur de deux trames quelconques ont la même longueur :

$$L = \sum_i l_i$$

Le **rythme** (ou **cadence**) de répétition (ou d'occurrence) des trames est :

$$R = \frac{D}{L} \qquad \text{trame/s}$$

La succession des IT de numéro i des différentes trames constitue un circuit de données appelé canal N°i. Le débit binaire du canal i sur la voie HV est :

$$d_i = \frac{l_i}{L} D \qquad \text{bit/s}$$

Le principe du **multiplexage par bits** est similaire excepté que la longueur de chaque IT est de 1 bit.

Voies BV asynchrones / synchrones

Définition (Voies BV asynchrones)**.** Lorsque les voies BV sont **asynchrones**, la transmission sur la voie BV n'est pas synchronisée : les horloges de l'émetteur et du récepteur (le multiplexeur) sont indépendantes.

Remarque. L'unité de données sur une voie BV asynchrone est le caractère.

La délimitation des caractères sur la voie BV doit donc être assurée par des bits supplémentaires : chaque caractère commence par un bit de *Start* et se termine par un (ou plusieurs) bit(s) de *Stop*.

Sur la voie HV, seule l'information utile (sans les bits *Start* et *Stop*) est transmise.

Définition (Voies BV synchrones)**.** Lorsque les voies BV sont **synchrones**, la transmission sur la voie BV est synchronisée entre l'émetteur de la voie et le récepteur (le multiplexeur) à l'aide d'une horloge commune (obtenue par la transmission d'un signal de temps sur la ligne).

Remarque. Cette synchronisation permet de ne transmettre sur une voie BV que l'information utile, sans bit supplémentaire.

Définition (Efficacité d'un multiplexeur)**.** L'efficacité e d'un multiplexeur est donnée par :

$$e = \frac{\sum_i C_i N_i}{D_b}$$

– C_i est la rapidité de transfert en car/s de la voie BV i (Basse Vitesse i) ;
– N_i est le nombre de bits utiles par caractère (sans bit *Start* ni *Stop*) ;
– D_b est le débit en bit/s de la voie HV (Haute Vitesse).
L'efficacité e s'interprète comme le rapport du débit utile (nombre de bits de données à transmettre sur la voie HV par unité de temps et provenant des voies BV), sur le débit maximum de la voie HV.

4.3 Normes de multiplexage dans les artères de commutation

Pour utiliser un mode de multiplexage temporel dans les réseaux téléphoniques, il faut convertir les signaux analogiques en signaux numériques. Cette numérisation s'effectue dans les commutateurs locaux et requiert des codecs (codeur/décodeurs). Il existe différentes techniques de numérisation : *Pulse Code Modulation*, *Differential Pulse Code Modulation*, *Delta Modulation*, etc.

La Modulation par Impulsion et Codage, *MIC* (PCM - *Pulse Code Modulation*), échantillonne le signal à une fréquence de 8000 Hz (ce qui correspond à un échantillon toutes les 125 μs) et code chaque échantillon sur 8 bits. Le débit engendré est donc de 64 kbit/s.

Originellement il fût impossible d'obtenir une norme internationale de multiplexage, aussi plusieurs standards coexistent.

Canal E1 (Europe) Le canal E1 est largement répandu en dehors des US et du Japon, et a été normalisé par l'IUT sous la norme G.732. Ce standard repose sur une numérisation par modulation MIC, et utilise des IT de 8 bits et une signalisation hors bande.

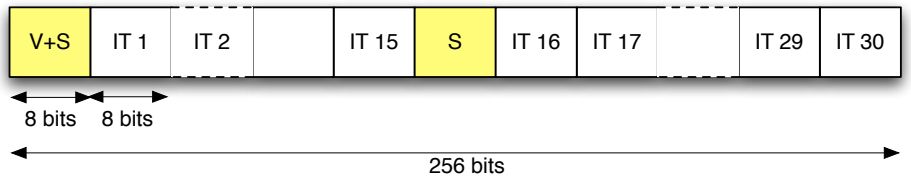


FIGURE 4.10: Le multiplex E1

Les trames sont émises toutes les 125 μs et contiennent chacune 32 IT de 8 bits (numérotées de 0 à 31), dont 30 IT pour les données et 2 IT pour le verrouillage et la signalisation (les IT N° 0 et N° 16). L'IT N° 0 est alternativement consacré au verrouillage (trames impaires) et à la signalisation (trames paires) tandis que l'IT N° 16 est entièrement consacré à la signalisation. Le multiplexage est temporel, seul mode possible pour les messages numériques.

Les trames sont émises toutes les 125 μs, ce qui correspond à une fréquence de 8000 Hz. On a L'efficacité du multiplexage est égal à 30/32.

Chapitre 5

Les réseaux

5.1 Rôle d'un réseau

Définition (Réseau)**.** Le réseau est un temporisateur : son rôle est d'acheminer des données de leur source à leur destination. Les données qui pénètrent dans le réseau en provenance d'une station sont routées vers leur destinataire en étant commutées de nœud en nœud. Certains nœuds ne sont reliés qu'à d'autres nœuds du réseau : leur seule tâche est alors la commutation interne. D'autres (les nœuds d'accès) sont également reliés à des stations : en plus de la fonction de commutation, ils doivent pouvoir accepter des données en provenance des utilisateurs du réseau et, réciproquement, leur délivrer des données.

Du point de vue de deux stations communicantes, le réseau peut donc être considéré comme une généralisation d'une liaison point-à-point. Mais les fonctionnalités assurées par une liaison de données ne sont pas suffisantes pour permettre l'acheminement de données entre un ETTD source et un ETTD destinataire, trois fonctions sont absentes :

L'adressage Pour délivrer un paquets à un destinataire il faut connaître sont adresse ce qui implique de définir un mécanisme d'adressage.

Le routage Consiste à déterminer la route que le paquet devra emprunté dans le réseau. Cela pose 3 questions :
- Quels sont les critères de choix pour trouver la bonne route ;
- Quel algorithme appliquer pour calculer les routes (eg. Dijkstra) ;
- Comment échanger les informations nécessaire au calcule des routes ? ce qui va ce faire par le protocole de routage.

Le contrôle de congestion

On ajoute donc des fonctionnalités qui n'étaient pas disponible au niveau liaison.

Deux approches sont possibles.

5.2 Service en mode connecté (circuit virtuel)

Analogie : le téléphone.

La communication se divise en trois phases :

Etablissement de la connexion virtuel Identification des entités en communication, négociation tripartite (les deux ETTD et le réseau), détermination de la route qui seras emprunté par tous les paquets de la connexion (cela ce fait par un paquet d'appel qui comporte les adresse de l'appelant et de l'appelé qui trace la route), réservation des ressources de communication ;

Transfert de données Avec un volume d'information de contrôle minimisé ;

Libération des ressources de communication

Ce service est considéré comme **fiable** : les paquets qui appartiennent à une connections sont référencé et il est possible de faire un contrôle de flux, un contrôle d'erreur et un contrôle de séquence. C'est l'approche que fournissait l'opérateur France Telecom^{TM}.

5.3 Service en mode non connecté (datagrame)

Analogie : la poste.

Les paquets sont indépendants les un des autres il n'y a pas de phase d'établissement ni libération de la connexion.

Transfert de données direct :les datagrames sont indépendant les un des autres, chaque paquets est considéré par le réseau comme une entité propre, chaque paquets peut éventuellement choisir un trajets différents des autres et donc des délais de transferts différents et donc une réception dé-séquencé. Chaque paquet doit contenir toute l'information de contrôle, notamment d'adressage (ce qui prend de la place) nécessaire à son acheminement.

- Ce service est considéré comme **non fiable**, souvent qualifié comme *best effort*.
- Le séquencement des données n'est pas garanti.
- Le contrôle de flux est très difficile, il peut y avoir des erreurs des pertes et duplication de paquets sont possible.

Conclusion. Le réseau est peu fiable, c'est à l'usager d'effectuer tous les contrôles. Le réseau fourni un service minimum (*best effort*). C'est l'approche Internet.

5.4 Interface X.25

5.4.1 Généralités

La recommandation X.25 a été adopté par l'ITU-T en 1976. Protocole proposé à l'origine par les PTT françaises, les PTT britanniques, le canadien Trans Canada Telephone System et l'américain Telenet Communication Corps, il a été implanté sur les réseaux publics des ces quatre compagnies (Transpac, EPSS, Datapac et Telenet).

X.25 offre un service de réseau en ***mode connecté*** et spécifie une **interface d'accès** à un réseau à commutation de paquets : il s'agit donc d'une interface locale entre un équipement connecté au réseau et le réseau lui-même, entre un ETTD et un ETCD.

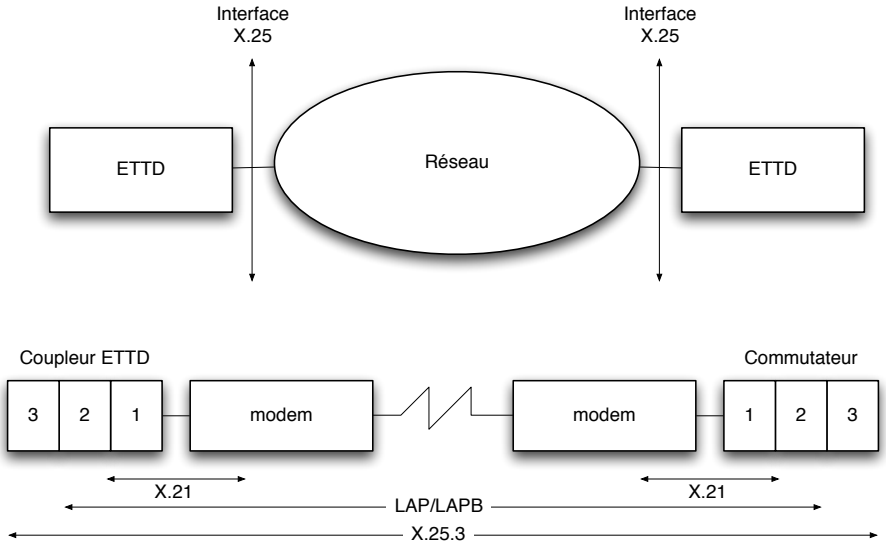


FIGURE 5.1: Interface d'accès X.25

X.25 contient trois niveaux de protocoles :

Un niveau physique : interface conforme à la norme X.21 ou X.21 bis, solution intérimaire qui permet de se connecter à un réseau public numérique à partir d'un équipement utilisant l'interface V.24 (V.24 ayant été conçu à l'origine pour l'interface avec des réseaux de télécommunications analogiques) ;

Un niveau liaison LAP-B = sous-ensemble de la procédure HDLC (en mode ABM) ;

Un niveau réseau X.25 niveau paquet (PLP *Packet Level Protocol*) gère le transfert de paquets d'une station source vers une station destinataire ; l'unité d'information y est le **paquet** :
- tout paquet est placé dans le champ d'information d'une trame I (**encapsulation**) ;
- la longueur de ce champ limite la longueur du paquet ;
- par convention, il ne peut y avoir plus d'un paquet par trame.

X.25 définit les types de paquets et leur format, mais ne spécifie pas comment certaines informations de contrôle doivent être interprétées (eg. la fenêtre de contrôle de flux). Ces imprécisions ont donné naissance à des réseaux conformes à X.25, mais différents dans leur implantation.

5.4.2 Le niveau réseau PLP : circuit virtuel et voie logique

X.25 supporte un service de réseau en ***mode connecté*** on parle également de ***circuit virtuel*** (CV). Les deux partenaires en communication ont l'illusion d'un canal point-à-point dédié. La connexion correspond à une association bidirectionnelle entre les deux ETTD.

Deux types de circuits virtuels sont possibles :

CVP (Circuit Virtuel Permanent) Un CVP relie en permanence deux (mêmes) abonnés ; l'établissement du CV est fait pour toute la durée de l'abonnement.

CVC (Circuit Virtuel Commuté) Un CVC permet à un abonné d'atteindre tout autre abonné, tout transfert de données doit être précédé d'une phase d'établissement et doit se terminer par une phase de libération du CVC.

Définition (Voie logique)**.** La notion de voie logique (VL) permet la coexistence de tronçons de plusieurs CV sur une même liaison, c'est un moyen de transmission bidirectionnelle simultanée sur une liaison de données. Elle est identifiée par un N° de GVL (groupe de VL) ≤ 15 et un numéro de VL ≤ 255 (signification purement locale à l'interface), donc globalement sur 12 bits. Ces valeurs sont attribuées à l'abonnement dans le cas d'un CVP et lors de l'établissement pour un CVC.

Remarque. Un circuit virtuel est une connexion de bout en bout, alors que le numéro de voie logique est un identifiant local du circuit virtuel, on peut voir un circuit virtuel comme une succession de numéro de voie logique sur chaque liaison emprunté.

$2^{12} = 4096$ circuits virtuels peuvent, au plus, transiter entre deux noeuds.

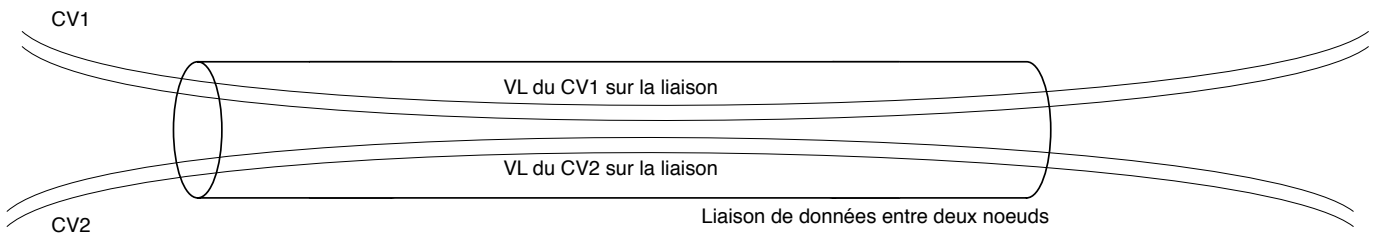


FIGURE 5.2: Voies logiques (VL)

Le multiplexage des VL sur une même liaison se traduit par un entrelacement de paquets appartenant à des CV différents dans les trames qui circulent sur la liaison de données.

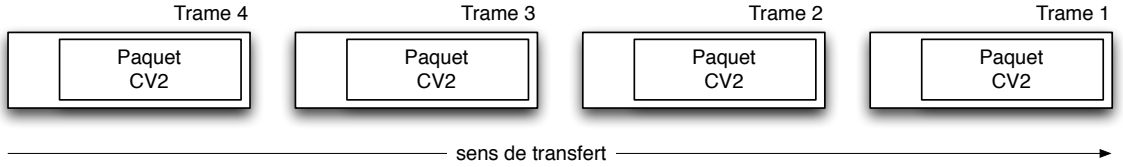


FIGURE 5.3: Multiplexage sur une VL

5.4.3 Le niveau réseau PLP : Les paquets X.25

Chaque paquet transféré à travers l'interface comporte au moins trois octets contenant :
– Une identification générale de format : indique entre autres le modulo utilisé (8 ou 128) ;
– Une identification de voie logique : N° GVL et N° VL ;
– Une identification de type de paquet.

Remarque. Il est à noter que seuls les paquets d'appel et d'appel entrant contiennent nécessairement l'adresse source (ETTD appelant) et l'adresse destination (ETTD appelé).

5.4.4 Le niveau réseau PLP : Les différentes phases d'un circuit virtuel

Etablissement d'un CVC

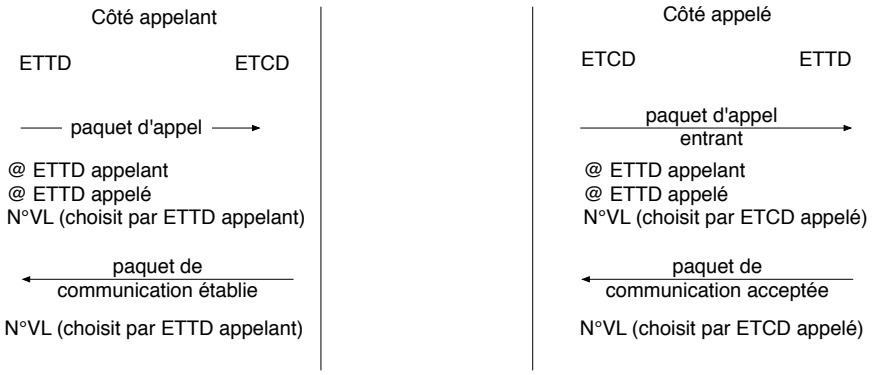


FIGURE 5.4: Etablissement d'une connexion circuit virtuel permanent

En cas de refus de l'ETTD appelé, celui-ci envoie un paquet de demande de libération à l'ETCD appelé qui provoquera l'envoi par l'ETCD appelant d'un paquet d'indication de libération.

En cas de collision d'appel sur une interface, lorsque l'ETTD et l'ETCD transmettent en même temps un paquet d'appel et un paquet d'appel entrant indiquant le même numéro de voie logique, l'ETCD traite le paquet d'appel et ne tient pas compte du paquet d'appel entrant, toutefois, l'ETTD distant reçoit une indication de libération. Afin de minimiser les risques de collision d'appels, et par convention, l'ETTD choisit comme numéro de voie logique (N° GVL + N° VL) le plus grand numéro disponible, alors que l'ETCD choisira le plus petit possible.

Transfert de données

Les paquets utilisés pendant cette phase sont :

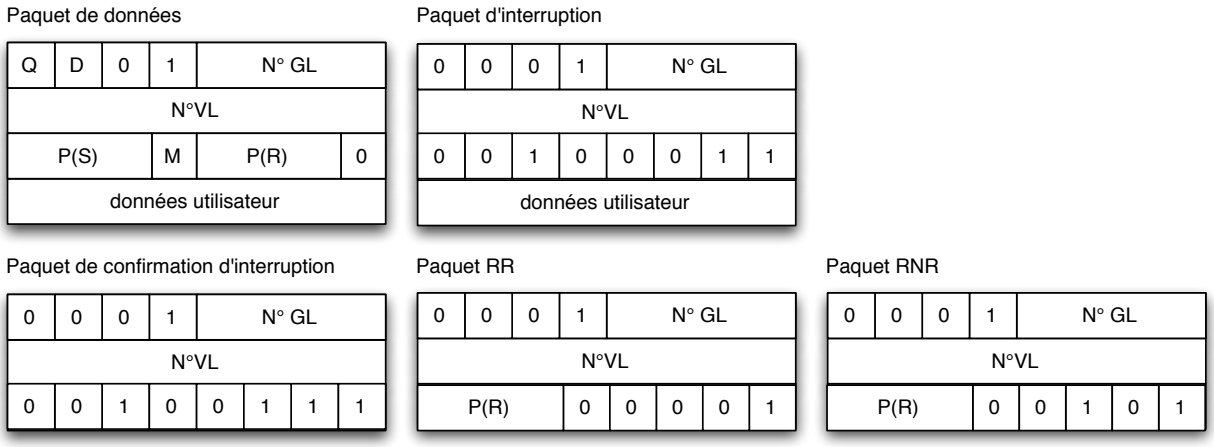


FIGURE 5.5: Paquets X.25

Dans le paquet de données :
- Q est le bit de qualification des données. Dans certains cas l'usager peut souhaiter différencier deux types d'information dans les paquets de données, par exemple les données de l'usager et l'information de commande ;
- D est le bit de portée de l'acquittement :
  - si $D = 0$, l'acquittement se fait par l'ETCD local ;
  - si $D = 1$, l'acquittement est de bout en bout (il est alors généré par l'ETTD distant).
- 01 indique une numérotation modulo 8 (10 pour un modulo 128) ;
- $P(S)$ est le numéro de séquence du paquet ;
- M est la marque "données à suivre". Ce bit est utilisé par un ETCD ou un ETTD s'il souhaite indiquer une séquence comportant plus d'un paquet, le dernier paquet de la séquence a le bit M à 0, les autres le bit M à 1 ;
- $P(R)$ est le numéro de séquence en réception (*piggybacking*) ;
- La longueur maximale normalisée du champ de données est de 128 octets.

Le paquet d'interruption est un paquet comportant entre 1 et 32 octets de données, non numéroté, non soumis au contrôle de flux, et qui doit être acquitté par un paquet de confirmation d'interruption. Il sert à transporter des informations urgentes. A tout moment un ETTD ne peut avoir plus d'un paquet d'interruption en attente d'acquittement.

Contrôle de flux

Le contrôle de flux au niveau des paquets utilise, comme au niveau de la liaison, un mécanisme de fenêtrage. A l'interface ETTD/ETCD, une fenêtre est définie pour chaque sens de transmission, dont la taille W normalisée est de 2 (cette taille peut être négociée lors de la phase d'établissement, il s'agit cependant d'un service optionnel facturé à l'usager). De façon similaire à HDLC, un paquet de données doit, pour pouvoir être émis, avoir son $P(S)$ à l'intérieur de la fenêtre :

$$\text{dernier } P(R) \text{ reçu} \leqslant P(S) \leqslant \text{dernier } P(R) \text{ reçu} + W - 1$$

Remarque. La fenêtre de niveau liaison indique le nombre de trame I qu'un émetteur peut émettre sans

acquittement, alors que la fenêtre de niveau réseau donne le nombre de paquets qui peuvent être envoyés sans acquittement.

Les deux fenêtres sont gérées par des couches différentes, toutefois la fenêtre de niveau liaison doit être la plus grande possible dans le cas contraire il y aurait toujours blocage au niveau liaison avant qu'il n'y ait blocage sur le circuit virtuel.

Libération de CVC

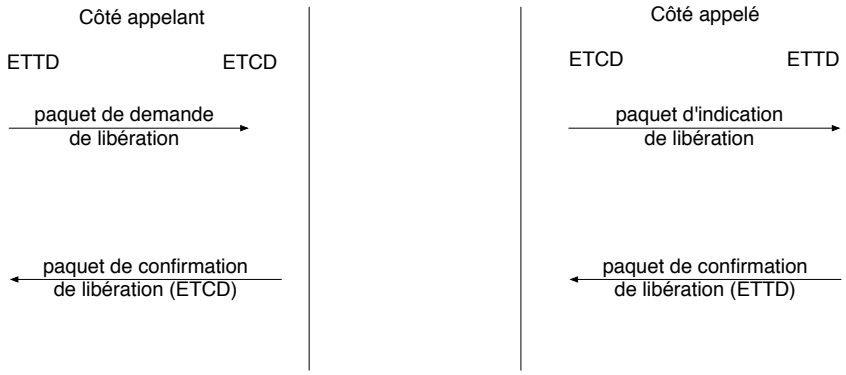


FIGURE 5.6: Libération d'une connexion circuit virtuel permanent

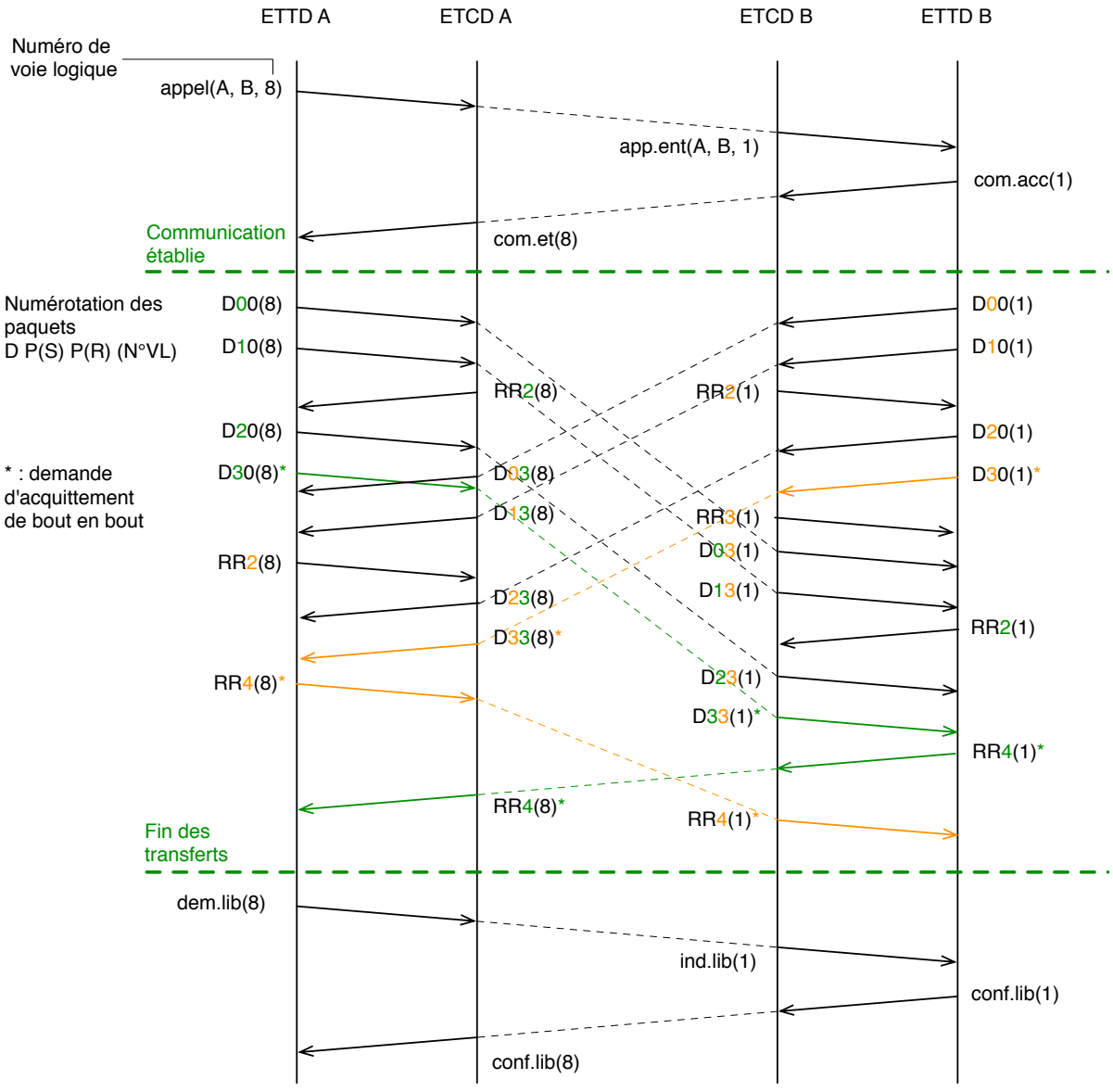


FIGURE 5.7: Echange de paquets entre ETTD et ETCD

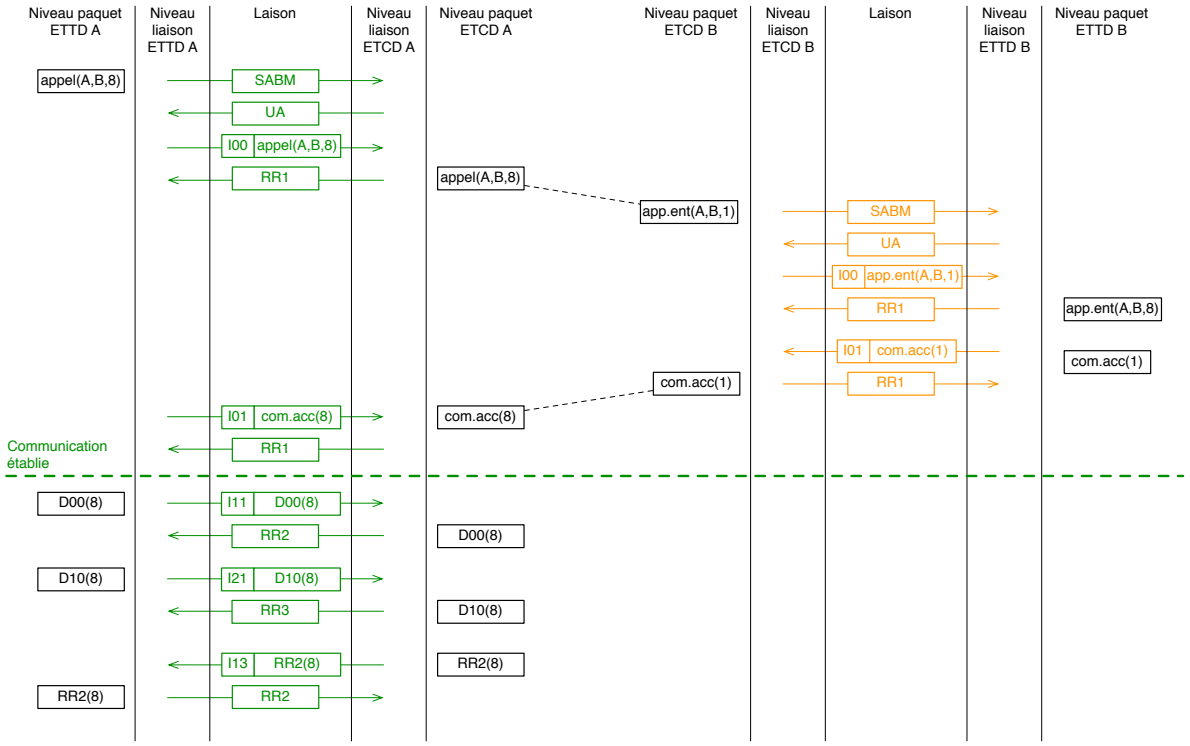


FIGURE 5.8: Echange de paquets entre ETTD et ETCD

Chapitre 6

Routage

6.1 Généralités

Définition (Routage)**.** Le routage est le processus qui consiste, dans un réseau, ou au travers de différents réseaux, à **trouver un chemin entre une source et une destination**. C'est l'une des fonctionnalités principales de la couche réseau. On appelle ***routeur*** un équipement relié à au moins deux réseaux et dont le rôle est de réémettre des paquets venus d'une de ses interfaces vers une autre.

Schématiquement, on peut représenter l'Internet comme une hiérarchie de routeurs.

Définition (Système autonome)**.** On appelle ***système autonome*** (*Autonomous System* ou AS) un ensemble de routeurs et de réseaux contrôlés par une même autorité administrative : on peut donc voir l'Internet comme un ensemble d'AS. Deux types de routage :
- Routage intra-domaine (IGP – *Interior Gateway Protocol*) : routeurs situés à l'intérieur d'un même AS ;
- Routage inter-domaine (EGP – *Exterior Gateway Protocol*) : routeurs qui permettent de relier entre eux différents AS.

Le routage (intra-domaine) est réalisé grâce aux **protocoles de routage** qui maintiennent des **tables de routage** dans les routeurs (et commutateurs) du réseau.

Une table de routage comporte au moins deux colonnes :
- destination (ou pour le réseau de destination) ;
- adresse de l'élément réseau correspondant au "saut" suivant (*next hop*) sur le "meilleur" chemin vers la destination souhaitée.

Lorsqu'un paquet arrive sur un routeur (ou lorsqu'un paquet d'appel arrive sur un commutateur), le routeur (le commutateur) consulte sa table de routage pour décider du prochain saut pour ce paquet.

Le moment de prise de décision du routage (consultation de la table de routage) dépend du mode d'acheminement :

Acheminement par voie logique
- décision prise une seule fois lors du passage du paquet d'établissement ;
- tous les autres paquets de la connexion suivent la même route, après consultation de la table de translation.

Par datagramme
- une décision de routage est prise pour chaque paquet ;
- la route peut être identique ou non pour deux paquets ayant le même destinataire.

Tout protocole de routage doit communiquer des informations sur la topologie globale du réseau à chaque routeur afin que celui-ci puisse prendre une décision locale de routage. Or, cette information globale est difficile à collecter, sujette à des modifications fréquentes et de plus, volumineuse.

Un "bon" protocole de routage visera donc :
- Une minimisation des messages de contrôle échangés
- Une minimisation de l'espace des tables de routage :

  - minimiser le coût des routeurs ;
  - minimiser le trafic d'information de routage (puisque les routeurs s'échangent leurs tables de routage pour garantir une vue consistante de la topologie du réseau).
- La robustesse : éviter les trous noirs, les boucles et les oscillations ;
- L'utilisation du chemin "optimal" qui n'est pas forcément le plus court : il peut s'agir du chemin au délai le plus court, du chemin le plus sécurisé, du chemin le moins cher, ou tout simplement du chemin utilisant le moins de sauts.

Le routage est par essence, un problème de la **théorie des graphes** : il s'agit de trouver le chemin du coût minimum entre deux nœuds quelconques, sachant que le coût d'un chemin est la somme des coûts des liens qui le composent.

Il existe plusieurs classes de techniques de routage :

Le routage isolé Cette classe regroupe des techniques qui ne nécessitent aucun échange d'information entre les nœuds et ne construisent pas de table de routage. (eg. le *le routage par inondation* dans lequel chaque nœud retransmet toujours tous les paquets sur toutes ces interfaces, sauf l'interface entrante).

Le routage centralisé On dispose d'un centre de contrôle de routage qui reçoit périodiquement des informations décrivant l'encombrement du réseau. Il en déduit les tables de routage de chaque routeur et les leur expédie.

Le routage distribué Chaque routeur échange périodiquement des informations avec ses voisins et recalcule sa table de routage. Deux types d'algorithmes de routage distribué sont largement utilisés :
  - **Le routage à vecteurs de distance** (eg. *Routing Information Protocol* ou RIP) ;
  - **Le routage à états des liens** (eg. *Open Shortest Path First* ou OSPF).

Les algorithmes à vecteurs de distance et à états des liens partagent des caractéristiques communes :
- ils supposent que chaque routeur connaît l'adresse de chacun de ses voisins, ainsi que le "coût" pour l'atteindre ;
- ils permettent à chaque routeur de déterminer l'information de routage globale (ie. le prochain nœud pour atteindre chaque destination possible sur la route la plus courte, en échangeant de l'information de routage seulement avec ses voisins).

6.2 Algorithme à vecteurs de distance

6.2.1 Principe

L'algorithme de base est dû à **Bellman-Ford**. Chaque nœud est supposé connaître la "distance" (ou le «coût») qui le sépare de chacun de ses voisins (une liaison hors service a un coût infini). Périodiquement, chaque nœud envoie à chacun de ses voisins la liste des distances estimées vers chaque nœud du réseau : c'est le ***vecteur de distance***. Il reçoit parfois une liste similaire de chacun de ses voisins.

Lorsque le nœud i reçoit le vecteur de distance V_j de son nœud voisin j, $V_j(k)$ lui donne la distance (estimée par j) pour aller de j à k, et il sait donc qu'il peut atteindre k via j, avec un coût de $V_j(k)$ augmenté du coût de sa liaison à j. en poursuivant ce calcul pour chacun de ses voisins, i peut déterminer l'estimation qui lui semble la meilleure pour atteindre chaque destination, et inscrire cette estimation ainsi que la liaison correspondante dans sa table de routage.

Plusieurs solutions permettent de pallier le problème du comptage à l'infini :

Le vecteur de chemin Chaque entrée du vecteur de distance contient le chemin complet associé à la valeur. C'est une technique utilisée dans BGP, mais qui présente l'inconvénient de générer des

tables volumineuses. (Il est à noter que BGP, *Border Gateway Protocol*, est un protocole de routage inter-AS) ;

L'horizon partagé (*split horizon*) Un routeur ne communique jamais le coût vers une destination à son voisin N, si N est le prochain nœud vers cette destination ;

L'horizon partagé avec antidote (*split horizon with poisonous reverse*) Un routeur communique toujours un coût infini vers une destination à son voisin N, dés l'instant où N est le prochain nœud vers cette destination. Cela se traduit par une modification mineure du protocole de vecteurs de distance : au lieu de diffuser le même vecteur sur toutes leurs liaisons, les nœuds devront en composer des versions différentes, pour tenir compte des destinations qui sont atteintes via chacune de ces liaisons. C'est la technique utilisée dans le RIP.

Remarque. L'horizon partagé est efficace pour éviter le comptage à l'infini lorsqu'on a une boucle entre 2 routeurs (efficace dans le sens où elle accélère la convergence), mais elle ne l'est pas lorsqu'on a 3 routeurs ou plus dans la boucle.

Remarque. La plupart des algorithme prévoient d'envoyer les vecteurs de distances de façon périodique. Cependant, il est souhaitable de signaler aussitôt que possible des pannes, les vecteurs de distance reportant une panne de noeud ou de lien sont immédiatement envoyés (donc de façon asynchrone).

6.3 Algorithme de routage à effets de liens

6.3.1 Principe

Dans cette approche, un routeur communique à tout autre nœud du réseau sa distance avec ses voisins. En effet, l'idée est de distribuer les topologies et le coût de chaque lien à tous les routeurs ? chaque routeur peut alors calculer de façon autonome, son chemin optimum pour chaque destination. Si les routeurs voient les mêmes coûts pour tous les liens et s'ils utilisent le même algorithme, les routes obtenues sont garanties sans boucle. Le problème est donc de distribuer la connaissance de la topologie à chaque routeur pour qu'il puisse ensuite calculer ses plus courts chemins.

Chaque routeur doit ainsi :

1. *Découvrir ses voisins*

   Un routeur en cours d'initialisation envoie sur chacune de ses lignes en sortie un paquet `Hello`, les routeurs aux extrémités répondront à ce paquet en retournant des informations de routage (ID, adresse IP, etc).

2. *Mesurer le coût vers chacun des voisins*

   Si le coût est basé sur le temps d'acheminement, le routeur peut en obtenir une bonne estimation en envoyant un paquet `Echo` estampillé et en prenant la moitié du temps aller-retour.

3. *Construire un paquet spécial*

   Le routeur crée un ensemble de LSP (*Link-State Packet*) décrivant ses liens. Un LSP contient l'ID du routeur, l'ID du voisin et le coût du lien vers ce voisin.

4. *Envoyer ce paquet à tous les autres routeurs du réseau*

   Ceci, est fait par inondation contrôlée. L'idée est lorsqu'un routeur reçoit un nouveau LSP, il en garde une copie dans sa base de données de LSP et il le retransmet sur chacune de ses voies de sorties (sauf celle sur laquelle le LSP lui est parvenu).

5. *Calculer le plus court chemin vers tous les autres routeurs*

   L'**algorithme de Dijkstra** calcule le chemin le plus court entre un sommet source s et tous les autres sommets d'un graphe $G = (S, A)$ valué par une fonction de pondération $\omega \colon A \to \mathbb{R}$.

   On définit :

– E : le sous-ensemble des sommets déjà évalués (pour lesquels les chemins les plus courts sont connus) ;
– R : le sous-ensemble des sommets restants ($R = S \backslash E$) ;

Pour chaque sommet $v \in S$, on maintient deux attributs :
– $d[v]$: l'estimation du coût d'un plus court chemin de s à v ;
– pred$[v]$: le prédécesseur de v sur le chemin estimé.

6.4 Routage et commutation

Un commutateur est un équipement permettant d'interconnecter des liaisons de façon à former un réseau. Cet équipement, possédant plusieurs ports en entrée et plusieurs ports en sortie, a pou rôle de transférer un paquet qu'il reçoit sur un port d'entrée vers un port de sortie. La question qui se pose est alors de savoir comment le (bon) port de sortie est déterminé. La réponse se trouve dans l'en-tête d'un paquet, deux approches sont possibles :

Commutation en mode non connecté (datagramme) Le paquet est réacheminé sur la base de son adresse de destination ;

Commutation en mode connecté (circuit virtuel) Le paquet est réacheminé sur la base de son identificateur de circuit virtuel.

Chapitre 7

Réseaux Locaux, Techniques d'accès

7.1 Topologies

Un réseau est un système qui relie entre eux des postes de travail. La manière de relier les stations définit la ***topologie physique*** (encore appelée ***plan de câblage***). Elle est à distinguer de la ***topologie logique*** (encore appelée ***topologie d'accès***), qui décrit la façon dont circule "logiquement" l'information. La topologie est celle prise en compte par la ***méthode d'accès***.

Il existe plusieurs types de topologies (physiques). Le choix de l'une ou de l'autre sera influencé par la vitesse à laquelle on souhaite travailler, par la disposition des lieux, par le type de câble que l'on veut utiliser ou qui est déjà installé et par le coût.

Toutes les architectures réseaux dérivent de trois topologies fondamentales :

Bus : les ordinateurs sont connectés les uns à la suite des autres le long d'un seul câble (segment). Le bus est une topologie passive : les ordinateurs qui y sont connectés (et qui n'ont pas de données à transmettre) ne font qu'"écouter" les données qui circulent sur le réseau. Le retrait des informations aux deux extrémités est implicite et s'effectue grâce aux terminateurs en fin de bus qui évitent les problèmes de réflexion du signal arrivant à l'extrémité du support ;

Etoile : les ordinateurs sont connectés à des segments de câble qui partent tous d'un même point, le *concentrateur* ou *contrôleur central*. Celui-ci gère l'accès au support ;

Anneau : les ordinateurs sont connectés à un câble qui forme une boucle. Chaque ordinateur fait office de répéteur en amplifiant le signal qui lui parvient afin de l'envoyer à l'ordinateur suivant.

Ces trois topologies de base sont simples en elles-mêmes. Toutefois, les topologies utilisées dans la pratique combinent souvent les caractéristiques de plusieurs d'entre elles, et peuvent donc s'avérer plus complexes.

Ces topologies ont des propriétés identiques de diffusion et de partage du support :
- toutes les stations peuvent lire les données qui transitent sur le support ;
- toutes les stations peuvent écrire sur le support.

7.1.1 Bus

Avantages – facile à installer ;
- retrait implicite des informations (grâce aux terminateurs en fin de bus qui évitent les problèmes de réflexion du signal arrivant à l'extrémité du support), contrairement à l'anneau, où c'est à l'émetteur de retirer sa propre information lorsqu'elle a fait un tour complet ;
- topologie passive. si un ordinateur tombe en panne, cela n'a pas d'incidence sur le reste du réseau.

Inconvénients – risques de collisions (ou contention d'accès) si deux signaux sont émis simultanément, se superposent et deviennent incompréhensibles (cacophonie), d'où un temps d'accès indéterministe ;
- les unités d'accès sont passives, le signal n'est donc pas régénéré. La taille du segment est donc limitée (à cause de l'atténuation). La taille peut être augmentée en ajoutant des répéteurs entre les segments ;

– lien multipoint non adapté à la fibre optique (support unidirectionnel) et complexe à maintenir ;
– en cas de rupture du câble coaxial commun, le réseau sera dit hors service car il y aura alors "rebond" du signal aux 2 nouvelles extrémités créées (sauf si l'on ajoute deux bouchons de terminaisons).

7.1.2 Etoile

Avantages – conflits d'accès réglés en central ;
– technologie simple et éprouvée : topologie classique des réseaux téléphoniques avec le PABX (*Private Auto Branch eXchange*)= contrôleur qui relie toutes les machines ;
– les pannes d'une machine sont gérées simplement en déconnectant la branche de la machine en cause ;
– liens point à point : on peut utiliser la fibre optique.

Inconvénients – problèmes de fiabilité du nœud central (en cas de panne aucune transmission n'est possible) ;
– la taille du réseau (le nombre de stations) dépend de la puissance du contrôleur central ;
– si le réseau est grand, cette topologie exige davantage de câblage.

7.1.3 Anneau

Avantages
– pas de taille limite au réseau (chaque station est connectée à l'anneau - lien en boucle unidirectionnel - par un répéteur) ;
– liens point à point adaptés à la fibre optique et simples à maintenir ;
– temps de transfert borné.

Inconvénients
– retrait des informations nécessaire ;
– problèmes de fiabilité si l'anneau se coupe (ce problème peut être pallié en doublant l'anneau) ;
– les connecteurs sont complexes et coûteux ;
– la gestion des pannes est difficiles : dans la mesure où le signal passe par tous les ordinateurs, une panne d'un ordinateur peut avoir une incidence sur l'ensemble du réseau.

Le partage du support est arbitré par des mécanismes appelés ***méthodes d'accès*** ou ***protocoles d'accès*** ou encore ***politiques d'accès***. Il existe deux grandes classes de méthodes d'accès :
– Les méthodes d'accès statiques ;
– Les méthodes d'accès dynamiques ; parmi ces dernières, on fait la distinction entre :
  – Les méthodes à allocation déterministes ;
  – Les méthodes à allocation aléatoire.

7.2 Politiques d'accès statiques

Les méthodes d'accès statiques consistent à partager de façon statique (donc sans évolution dans le temps) les ressources de transmission (abusivement appelées "bande passante") entre plusieurs communicateurs. Il existe deux façons simples d'effectuer ce partage :

AMRF (Accès multiple à répartition en fréquence) : la bande passante du support physique est découpée en sous-bandes dont chacune est affectée à un seul communicateur ;

AMRT (Accès multiple à répartition dans le temps) : Le temps est découpé en tranches (appelés *Intervalle de Temps* ou IT), que l'on affect successivement aux différents communicateurs.

Avantages

- ces techniques sont simples et efficaces si le nombre de stations est fixe ;
- elles sont équitables entre les stations et permettent un accès régulier au support ;
- elles permettent d'implémenter simplement des mécanismes de priorité.

Inconvénients

- si une station n'a rien à émettre il y a un gâchis de bande passante ;
- pour l'AMRT : il est nécessaire de "synchroniser" les stations. Cette synchronisation doit être mise en place par une station primaire qui doit gérer les ajouts ou les retraits de stations sur le réseau. Il se pose donc des problèmes de fiabilité lorsque la station primaire est en panne.
- pour l'AMRF : le découpage en sous-bandes introduit des inter-bandes (afin d'éviter au maximum les interférences), d'où un gâchis de la bande passante. Chaque station a besoin d'autant de démodulateurs qu'il a de sous-bandes afin de pouvoir recevoir de tous les émetteurs.

Remarque. Ces techniques semblent peu adaptées aux réseaux locaux en effet :
- elles sont peu flexibles dès l'instant où le nombre de stations varie ;
- si toutes les stations n'émettent pas il y a un gâchis de bande passante qui limite les stations qui ont envie d'émettre.

Il est à noter cependant que les techniques de multiplexage sont largement utilisées dans les réseaux longue distance (WAN).

7.3 Politiques d'accès dynamiques à allocation déterministes

Ces techniques d'allocation permettent de n'allouer des ressources (de la "bande passante") qu'aux utilisateurs qui en ont réellement besoin. La difficulté provient du manque de connaissance des besoins utilisateurs à tout instant. Cela nécessite donc la mise en place d'une "intelligence", celle-ci pouvant être centralisée ou répartie entre les utilisateurs.

7.3.1 Politiques d'accès à allocation sélective ou *polling* (solution centralisée)

Cette technique d'allocation d'une ressource unique (le canal) entre plusieurs compétiteurs consiste à effectuer une consultation des divers compétiteurs en les invitant à émettre à tour de rôle. Une ***station centrale*** (encore appelée ***site maître***), qui peut être soit une machine soit un concentrateur de terminaux„ gère l'ensemble du système en interrogeant séquentiellement chaque station, afin de voir si elle a des trames à émettre. En cas de réponse positive, la station transmet sa trame au site maître qui, ultérieurement, interrogera la station destinataire pour savoir si elle est prête à recevoir, et lui enverra alors les données en cas de réponse positive. On distingue deux variantes :

Roll-call polling (topologie logique en étoile) : la station primaire interroge successivement chacune des stations secondaires en lui envoyant une ***trame de poll***. La station interrogée lui répond par une trame d'acquittement négatif si elle n'a rien à lui envoyer ou par une trame de données dans le cas contraire ;

Hub polling (topologie logique en bus) : la station primaire démarre un cycle en envoyant une trame de poll à la station secondaire "la plus éloignée" sur le bus. Si cette dernière a des données à émettre, elle les lui envoie puis elle envoie une trame de poll à la station secondaire "suivante" sur la liaison. Dans le cas contraire, elle envoie immédiatement la trame de poll à la station secondaire suivante. La "dernière" station secondaire envoie une trame de poll au primaire qui démarre un nouveau cycle.

Avantages

- la technique du polling est simple à mettre en œuvre ;

- il est facile d'instaurer des priorités ;
- le site maître peut facilement redonner la main au récepteur si la transmission requiert des acquittements.

Inconvénients

- c'est une méthode très "lourde" à gérer : le polling engendre un grand nombre de messages de contrôle (message de poll) et implique donc un *overhead* (sur-débit) important ;
- comme dans toute solution centralisée, il y a le problème d'une panne et/ou du goulet d'étranglement au niveau du site maître.

7.3.2 Politique d'accès à allocation de Jeton (solution répartie)

Le principe consiste à faire circuler sur le réseau, une trame spéciale appelée "**Jeton**". Seule la station qui possède le jeton, à un moment donné, est autorisée à émettre. Il existe plusieurs manières d'implanter cette technique. On peut différencier les techniques à ***jeton non adressé*** et les techniques à ***jeton adressé***.

Anneau à jeton ou politique du jeton non adressé (*Token Ring*)

Cette méthode d'accès, est employée dans des **topologies physiques en anneau**. Un **jeton** circule sur l'anneau, et celui-ci, selon son état, *libre* ou *occupé*, donne ou ne donne pas le droit d'émettre à la station qui le détient.

Une station voulant émettre va devoir attendre le jeton. Lorsqu'une station reçoit le jeton et que celui-ci est "libre", elle change l'état du jeton puis attache au jeton le message ainsi que l'adresse du destinataire du message et sa propre adresse. Lorsqu'une station "voit" passer un jeton occupé, elle ne peut pas émettre, mais elle "consulte" l'adresse de destination et l'adresse d'émission du message. Si le message lui est destiné, elle met à jour le champ d'acquittement de la trame. Si l'adresse source correspond à sa propre adresse (le jeton a donc fait un tour complet), la station retire l'information associée à la trame, et réémet un jeton à l'état libre. Il existe plusieurs variantes quant au retour du jeton à l'état libre. Par exemple, la station ayant émis un message peut réémettre un jeton libre après que sa trame lui soit intégralement revenue ou dés que sa trame commence à lui revenir.

La méthode du jeton adressé permet de garantir l'absence de collisions. En effet, à un instant donné, une seule station est susceptible d'émettre sur l'anneau : celle qui possède le jeton.

Bus à jeton ou politique du jeton adressé (*Token Bus*)

Cette méthode d'accès est employée dans des **topologies physiques en bus**. Un **anneau logique** est crée ordonnant les stations de façon cyclique. chaque station connaît son successeur sur l'anneau logique. Lorsque l'anneau logique est initialisé, les stations sont insérées au sein de l'anneau dans l'ordre de leur adresse, de la plus haute à la plus basse. Le transfert du jeton d'une station à l'autre s'effectue selon la même séquence. A un instant donné, seule la machine possédant le jeton a le droit d'émettre. Si elle n'a rien à émettre, elle envoie le jeton explicitement à une autre station (c'est pourquoi le jeton est dit "adressé"). Dans le cas contraire, elle peut transmettre pendant une durée de temps maximum déterminée, contrôlée par un compteur de temps de transmission, à la suite de quoi elle doit passer le jeton à la station suivante sur l'anneau logique.

7.4 Politiques d'accès dynamiques à allocation aléatoire

7.4.1 Le protocole Aloha pur : accès aléatoire sans référence temporelle

Le principe du protocole **ALOHA pur** est simple : dès qu'une station a besoin d'envoyer une information, elle l'émet, sans aucune précaution particulière. Cependant, si deux ordinateurs émettent une trame en même temps, les signaux se superposent : il y a **collision**. Le signal émis est incompréhensible, il faut alors le **réémettre**. Les collisions sont détectées par les stations émettrices en examinant le niveau électrique ou la largeur des impulsions des signaux reçus (lors de l'écoute) et en les comparant à ceux des signaux transmis : si le signal reçu est différent du signal transmis, la station émettrice en déduit que les signaux ont été perturbés par une collision. Afin de déterminer les performances de cette méthode d'accès, on va supposer que les trames sont de longueur fixe et on néglige le temps de propagation sur le support. Si toutes les trames sont de longueur l, il faut donc une durée constante $t = \frac{l}{D}$ (où D est le débit de la liaison) pour transmettre une trame. Si une machine commence à émettre une trame à un instant t_0, cette trame sera correctement émise à l'instant $t_0 + t$ (et donc correctement reçue par les autres stations dès l'instant où l'on néglige le temps de propagation) si aucune autre machine ne commence à émettre pendant l'intervalle de temps $[t_0 - t, t_0 + t]$. Cette "période de vulnérabilité" est de longueur $2t$.

Une amélioration du protocole ALOHA pur consiste à définir des intervalles de temps répétitifs (les slots) et à n'autoriser les stations à émettre qu'en début de chaque intervalle. Ce protocole connu sous le nom de protocole **ALOHA discrétisé** permet de réduire la période de vulnérabilité de moitié (par rapport à ALOHA pur, celle-ci passant de $2t$ à t et donc d'augmenter l'efficacité de l'accès au support. (On montre que $S = Ge^{-G}$ et que l'on peut alors atteindre une utilisation de 37 %.) En revanche, il nécessite une synchronisation entre les différentes stations qui y sont connectées.

7.4.2 La méthode d'accès CSMA/CD (Ethernet)

CSMA/CD (Carrier Sense Multiple Access with Collision Detection) peut se traduire en français par « protocole d'accès multiple avec surveillance de porteuse (signal circulant sur le canal) et détection de collision ». Avec cette méthode d'accès, toute machine est autorisée à émettre sur la ligne à n'importe quel moment et sans notion de priorité entre les machines. Deux règles simples gouvernent l'accès au support :

- chaque machine « écoute » ce qui se passe sur le support et vérifie qu'il n'y a aucune communication sur la ligne avant d'émettre ;
- lorsqu'une machine détecte une collision, elle cesse d'émettre ses données.

Lors d'une collision, les deux machines impliquées interrompent leur communication et attendent un délai aléatoire avant de réémettre. La valeur de ce délai est déterminé par l'algorithme du "retrait exponentiel" (exponential backoff) :

- après la première collision, une machine attend un temps aléatoire égal soit à 0 soit à 1 ($= 2^1 - 1$) *unité de temps* (appelé généralement *intervalle de temps élémentaire* ou *slot time*) ;
- après la seconde collision, une machine attend un temps aléatoire compris entre 0 et $3 = 2^2 - 1$ *unités de temps* ;
- après i collisions, une machine attend un temps aléatoire compris entre 0 et $2^i - 1$ *unités de temps* ; Il existe généralement une limite sur le nombre maximum d'unités de temps ($1023 = 2^10 - 1$ pour les réseaux Ethernet) ainsi que sur le nombre maximum de tentatives de retransmission après collision (16 dans les réseaux Ethernet).

La technique d'accès CSMA/CD est la plus répandue. Elle est utilisée dans les réseaux locaux de type Ethernet (norme IEEE 802.3).

Une collision peut se produire lorsque le contenu d'au moins deux trames se superposent et qu'elles deviennent corrompues. Pour être certain de détecter une collision, on prend ϵ = temps de propagation entre les machines les plus éloignées possible du réseau. Pour pouvoir détecter une collision, il faut que le temps de transmission d'une trame soit au moins égal à deux fois le temps de propagation sur le médium :

$$t_{trans} \geq 2t_{prop}$$

Notons que pour un câble coaxial de 1 km, le temps de propagation τ est de l'ordre de 5 μs (pour une vitesse de propagation de 200 000 km/s).

En réalité, après avoir détecté la collision, une station stoppe son émission et émet un signal de brouillage pour renforcer la collision et avertir la station de destination de la trame (qui n'est généralement pas impliquée dans la collision) que la trame qu'elle est train de recevoir est invalide et qu'elle doit l'ignorer.

En pratique, on ajoute une marge de sécurité :

$$t_{trans} \geq 2t_{prop} + \text{Marge de sécurité}$$

Définition (Intervalle de temps élémentaire (*Slot Time*))**.** Délai maximum qui peut s'écouler avant qu'une collision soit détectée ou encore délai après lequel une station est certaine d'avoir réussi sa transmission. Elle est égale à deux fois le temps de propagation d'un signal sur le support (plus une marge de sécurité). C'est l'unité de temps du protocole. Elle sert pour la détermination du délai aléatoire avant retransmission. Dans les réseaux Ethernet le Slot Time est de de 51.2μs.

Définition (Séquence de brouillage (*jam sequence*))**.** Si pendant qu'il émet, un ETTD détecte une collision, il continue d'émettre pendant un certain temps dit "séquence de brouillage" puis cesse d'émettre, ceci afin de "renforcer" la collision. S'il cessait d'émettre immédiatement, il y aurait un risque que le ou les ETTD concernés par la collision n'aient pas le temps de détecter celle-ci. Comme indiqué plus haut, la séquence de brouillage sert également à informer la station de destination de la trame (qui n'est généralement pas impliquée dans la collision) que la trame qu'elle est train de recevoir est invalide et qu'elle doit l'ignorer. Il est à noter que la durée de la séquence de brouillage n'est pas normalisée dans Ethernet (32 bits de brouillage à "1" sont souvent utilisés, ce qui correspond à un temps de brouillage de 3.2 μs).

Définition (Délai inter-trame (*intertrame gap*))**.** Le délai inter-trame est un silence "obligatoire" entre 2 trames successives les obligeant à ne pas se suivre de trop près. Le délai inter-trame est normalisé dans Ethernet à 9.6 μs. En pratique, c'est le temps minimum que doit attendre une station qui détecte un silence sur la ligne avant d'émettre une trame.

La trame Ethernet est structurée de la façon suivante :

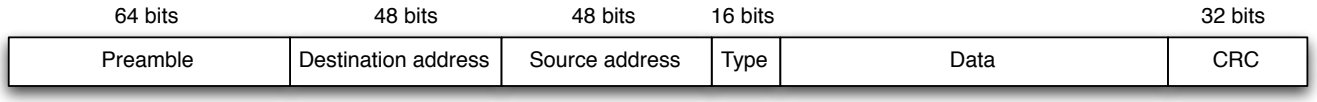


FIGURE 7.1: Trame Ethernet

Préambule détermine le début d'une trame : "10101010...10101011" ;

Adresse destination détermine la destination de la trame ;

Type définit le type de contenu de la trame ; à titre d'exemple, si le champ type est à 0800 (en hexadécimal) le champ de données contient un paquet IP. Ainsi, il est possible de déterminer quel protocole de niveau supérieur va utiliser le paquet encapsulé dans le champ de données (Data) de la trame ;

Données sont les données brutes de la trame à passer au protocole déterminé par le champ *type* ;
CRC est le *checksum* (contrôle de parité) de la trame permettant d'assurer son intégrité.

Chapitre 8

IP : Internet Protocol

8.1 Le protocole IP

8.1.1 Le datagramme IP

L'en-tête IP est alignée sur des mots de 32 bits. Sa longueur est donc multiple de 4 octets. Par défaut, sans option, l'en-tête IP fait 20 octets de long :

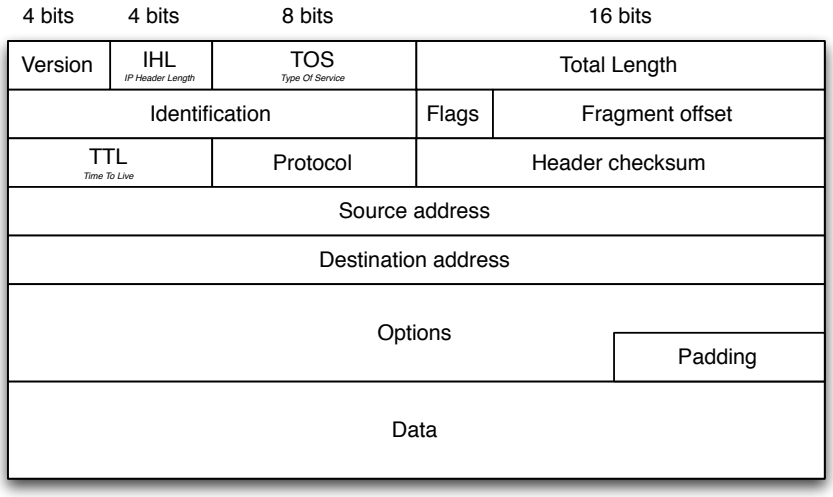


Figure 8.1: Datagramme IP

Version indique le format de l'en-tête. Ce champ sert à l'identification de la version courante du protocole. La version aujourd'hui utilisée porte le n°4 ;

IHL ***(IP Header Length)*** est la longueur de l'en-tête IP exprimée en mots de 32 bits (5 au minimum, 15 au maximum) ;

TOS ***(Type Of Service)*** définit le type de service à appliquer au paquet en fonction de certains paramètres comme le délai de transit, la sécurité. Codé sur 8 bits, il comprend les champs suivants :

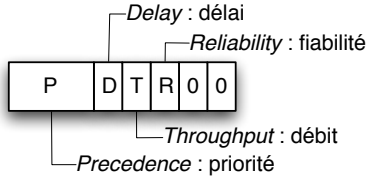


Figure 8.2: TOS

Total Length est la longueur totale du datagramme, en-tête et données inclus, exprimée en octets. La longueur d'un datagramme IP est donc limité à 65 535 octets (en pratique il est rare qu'un datagramme IP fasse plus de 1500 octets) ;

Identification est une valeur fournie par l'émetteur aidant au réassemblage des différents fragments du datagramme. Le seul usage de ce champs est donc de permettre à une entité réceptrice de reconnaître les datagrammes qui appartiennent à un même datagramme initial et qui doivent donc faire l'objet d'un réassemblage ;

Flags est utilisé par la fragmentation. Il est composé de deux indicateurs : **DF** *(Don't Fragment)* pour interdire la fragmentation et de **MF** *(More Fragment)* pour signaler des fragments à suivre :

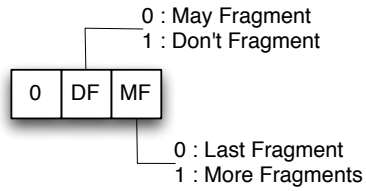


FIGURE 8.3: Flags

Fragment Offset indique sur 13 bits la position relative du fragment dans le datagramme initial, le déplacement étant donné en unités de 64 bits (seuls un datagramme complet ou un premier fragment de datagramme peuvent avoir ce champ à 0) ;

TTL *(Time To Live)* représente une indication de la date limite supérieure du temps de vie d'un datagramme. Cette valeur est comprise entre 0 et 255 et est initialisée par l'émetteur du datagramme, elle est décrémentée tout au long de la route suivie par chaque passerelle traversée. Tout intermédiaire ou destinataire qui détecte le passage de ce champ à la valeur 0 est supposé écarter le datagramme et renvoyer un datagramme ICMP *Time exceeded* à l'émetteur contenant notamment l'en-tête du datagramme détruit. Cette procédure est destinée à éviter les boucles ou les cheminements trop anormaux permettant ainsi de garantir une durée de vie maximale à un datagramme ;

Protocol indique le protocole (de niveau supérieur) utilisé pour le champ de données du datagramme ;

Header Checksum est une zone de contrôle d'erreur portant uniquement sur l'en-tête du datagramme ;

Source Adress est l'adresse IP de la source du datagramme ;

Destination Adress est l'adresse IP de destination du datagramme ;

Options sert à des fonctions utiles dans certaines situations (estampillage temporel, sécurité, routage particulier, etc.). Le champ est donc de longueur variable. Il est constitué d'une succession d'options élémentaires, également de longueurs variables. Les options sont codées sur le principe ***TLV*** *(Type, Longueur, Valeur)*. La longueur indique la taille complète de l'option en octets. L'option *Record Route* a la structure suivante :

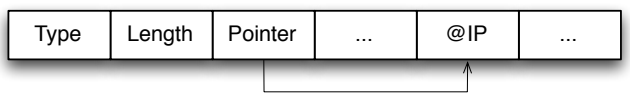


FIGURE 8.4: Structure de l'option Record Route

Padding permet d'aligner l'en-tête sur 32 bits ;

8.1.2 Le paquet ARP (*Address Resolution Protocol*) / RARP (*Reverse ARP*)

Le protocole ARP permet à une machine d'obtenir l'adresse Ethernet (physique) d'une autre machine, connaissant son adresse IP (logique). Le protocole RARP fait l'inverse. Un paquet ARP (ou RARP) est structuré de la façon suivante :

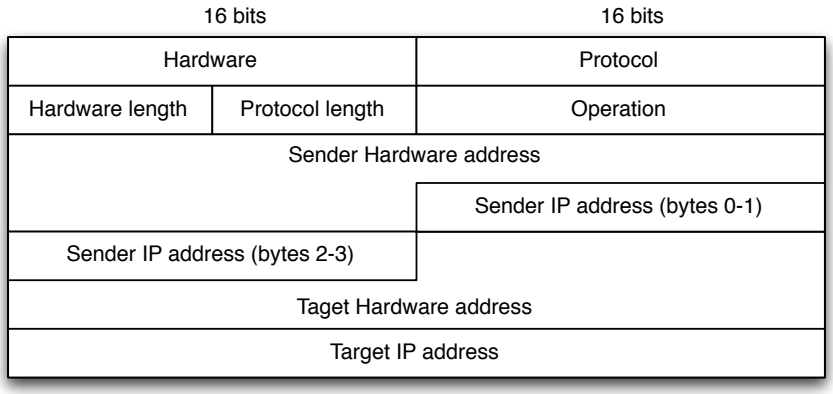


FIGURE 8.5: Paquet ARP

Hardware définit le type d'interface pour laquelle l'émetteur cherche une réponse (eg.`00 01` pour une interface Ethernet) ;

Protocol définit le type de protocole pour lequel une requête a été émise (eg. `08 00` pour une adresse logique IP) ;

Hlen définit la taille de l'adresse physique en octets ;

Plen définit la taille de l'adresse au niveau protocolaire (IP) ;

Operation décrit le type d'opération à effectuer par le récepteur (eg. `00 01` pour une requête ARP ("`00 03`" pour une requête RARP), "`00 02`" pour une réponse ARP (`00 04` pour une réponse RARP)) ;

Sender HA définit l'adresse physique (Ethernet) de l'émetteur ;

Sender IA définit l'adresse de niveau protocolaire (IP) demandé de l'émetteur ;

Target HA définit l'adresse physique (Ethernet) du récepteur ;

Target IA définit l'adresse de niveau protocolaire (IP) demandé du récepteur ;

8.1.3 Le message ICMP (*Internet Control Message Protocol*)

Le protocole ICMP est utilisé lorsqu'un imprévu se produit ou pour tester Internet. Les messages ICMP sont encapsulés dans des datagrammes IP. Ils ont tous en commun le même format pour le premier mot de 32 bits :

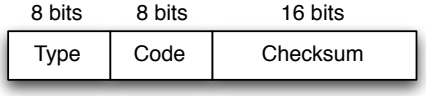


FIGURE 8.6: ICMP

FIGURE 8.7: Message ICMP encapsulé dans un datagramme IP

A titre d'exemple, `Echo` et `Echo Reply` sont utilisés pour vérifier l'état d'activité d'une machine. Une machine source envoie alors un message "Echo" à la machine destinataire dont elle veut vérifier l'activité. Celle-ci doit alors lui répondre par un message "Echo Reply".

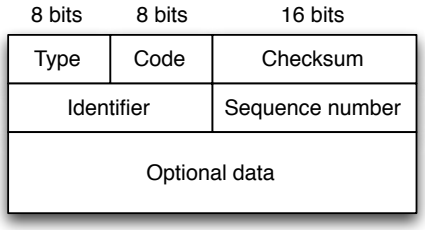


FIGURE 8.8: Echo

Principe de l'encapsulation

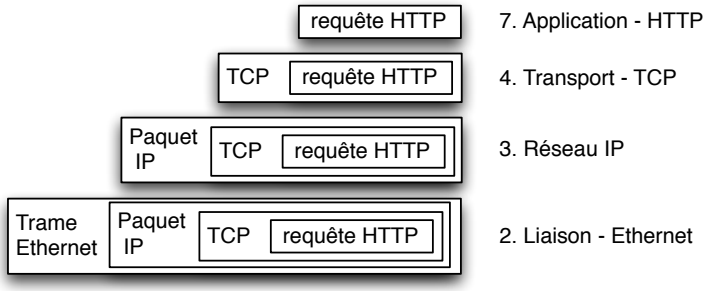


FIGURE 8.9: Encapsulation

8.2 Adressage de réseau et routage IP

8.2.1 Classes d'adresse

L'espace d'adressage IP est structuré. Une adresse IP est codée sur 4 octets et est constituée de deux parties : une partie *réseau* et une partie *hôte* :

Adresse IP = `<network>.<host>`

Initialement, trois classes d'adresses ont été définie s :

Classe A La partie `<network>` est codée sur 1 octet (dont le bit de poids fort est égal à `0`) et la partie `<host>` est codée sur 3 octets ;

Classe B La partie `<network>` est codée sur 2 octets (dont les deux bits de poids fort sont égaux à `10`) et la partie `<host>` est codée sur 2 octets ;

Classe C La partie `<network>` est codée sur 3 octets (dont les trois bits de poids fort sont égaux à `110`) et la partie `<host>` est codée sur 1 octet ;

Un ***masque de réseau*** (encore appelé masque "primaire") est associé à chacune de ces classes :
– Le masque d'un réseau de classe A est `255.0.0.0` (`/8`) ;
– Le masque d'un réseau de classe B est `255.255.0.0` (`/16`) ;
– Le masque d'un réseau de classe C est `255.255.255.0` (`/24`).
La valeur (optionnelle) spécifiée après le "`/`" indique la longueur du préfixe réseau.

En théorie, on dispose de $2^7 = 128$ réseaux de classes A (quelques très grands réseaux), de $2^{14} = 16\,384$ réseaux de classe B (des réseaux intermédiaires) et de $2^{21} = 2\,097\,152$ réseaux de classes C (des petits réseaux).

En pratique, certaines plages d'adresses sont réservées. Par exemple, il n'y a que 126 réseaux de classe A possibles, car un identifiant réseau "tout à 0" signifie "*ce réseau*" et une adresse IP commençant par 127 est une adresse de rebouclage (non routable).

Au maximum, un réseau de classe A peut adresser $2^{24} = 16\,777\,216$ machines, un réseau de classe B peut adresser $2^{16} = 65\,536$ machines et un réseau de classe C peut en adresser $2^8 = 256$. En pratique, les adresses pour lesquelles la partie "hôte" est "tout à 0" ou "tout à 1" sont réservées (au réseau pour la première et au *broadcast* pour la seconde).

8.2.2 Sous-réseaux

Tout réseau peut être décomposé en un certain nombre de ***sous-réseaux*** partageant le même préfixe réseau. L'adresse IP possède alors la forme suivante :

Adresse IP = `<netowork><subnetwork><host>`

Contrairement aux identifiants réseau dont la longueur est prédéterminée par la classe de l'adresse, les préfixes de sous-réseaux peuvent être de longueur variable. La longueur d'un préfixe est choisie selon le nombre de sous-réseaux constituant un site et le nombre d'hôtes par sous-réseaux. Il est donc nécessaire de spécifier pour toute adresse IP affectée, la longueur de son préfixe. C'est la vocation des ***masques de sous-réseau***. Le masque associé à une adresse IP est un mot de 32 bits où les seuls bits positionnés à 1 sont ceux associés au préfixe de cette adresse.

Ce procédé rend obsolète la structuration des adresses IP en 3 classes A, B et C. En effet, une adresse réseau appartenant à l'une des 3 classes peut à présent adresser plusieurs réseaux physiques. Les adresses réseaux consécutives et dont la longueur du préfixe le plus long partagé est différente de celle de l'identifiant réseau imposée par sa classe, peuvent être agrégées en une adresse réseau dont la classe spécifierait un identifiant réseau de la longueur de ce préfixe.

Attribuer des adresses réseau dont la longueur de l'identifiant réseau est variable permet une structuration hiérarchique non plus sur 2 niveaux, mais sur plusieurs niveaux de l'espace d'adressage, indispensable pour résoudre le problème de l'équipement des adresses IP allouables et celui de la taille croissante des tables de routage.

8.2.3 Routage IP

Dans un réseau à commutation par paquets, le ***routage*** est le traitement qui consiste à choisir le chemin sur lequel transmettre un paquet en fonction de son adresse destination et des informations contenues dans les tables de routage. Le ***protocole de routage*** permet, quant à lui, de construire dynamiquement les tables de routage. Une ***table de routage*** (ou FIB, *Forwarding Information Base*) est une structure complexe qui contient les informations nécessaires pour atteindre toute adresse IP valide.

Un Internet est composé de plusieurs réseaux interconnectés par des équipements appelés ***gateways*** (***passerelles*** ou ***routeurs***). Chaque *gateway* est directement connecté à au moins deux réseaux physiques et assure le transfert des paquets (relaye les paquets) d'un réseau à un autre. Un ***hôte*** est directement connecté à au moins un réseau physique mais ne relaye jamais de paquet. Ainsi, contrairement au *gateways*, il rejette systématiquement les paquets qu'il reçoit et dont il n'est pas le destinataire. En théorie, un hôte n'exécute donc pas de protocole de routage et ses tables de routage sont généralement construites manuellement par l'administrateur réseau.

Il existe deux façons de faire du routage dans IP :

Le routage direct Concerne la transmission d'un paquet IP entre deux machines connectées au même réseau physique (pas de *gateway* impliquée). La source encapsule le paquet dans une trame dont l'adresse de destination est l'adresse MAC (Ethernet) de la destination ;

Le routage indirect Intervient lorsque la destination n'est pas connectée au même réseau physique que la source. La transmission du paquet est alors effectuée de proche en proche (*hop by hop*) ; le routage IP fournit l'adresse du prochain routeur sur le chemin vers la destination souhaitée (*gateway* la plus proche de la destination). Chaque paquet est alors encapsulé dans une trame dont l'adresse MAC de destination est celle de la *gateway* empruntée.

Une table de routage IP est constituée d'au moins quatre colonnes :

Destination	Mask	Gateway	Interface
Destination que permet de joindre cette entrée. Il peut s'agir d'une adresse complète d'hôte ou d'une adresse réseau ; `0.0.0.0` correspond à la route par défaut.	Spécifie, si la destination s un (sous-)réseau, le masque de ce (sous-)réseau. Si la destination est un hôte, elle contient `255.255.255.255`. Si l'entrée est celle de la route par défaut, elle contient `0.0.0.0`.	Indique l'adresse du prochain routeur. Si l'adresse de destination est celle d'un hôte ou d'un réseau directement accessible par une interface locale, apparaît, selon les systèmes, `0.0.0.0` ou un astérisque.	Indique l'interface (la carte Ethernet) sur laquelle le paquet doit être transmis pour suivre la route considérée.

L'algorithme, simplifié, suivant permet à un routeur de déterminer, lorsqu'il reçoit un paquet IP contenant l'adresse IP "DestAdr", sur quelle interface il doit le relayer. Les entrées de la table de routage sont notées :

- `(H, 255.255.255.255, G, I)` si l'adresse `H` est celle d'un hôte ;
- `(R, M, G, I)` si l'adresse `R` est celle d'un réseau.

Algorithme 1: Algorithme de routage

début
  si ∃ *entrée* `(H, 255.255.255.255, G, I)` *telle que* `DestAdr = H` **alors**
    Routage comme spécifié dans la table (direct si `G = *`, indirect sinon) sur l'interface `I`.
  sinon
    si ∃ *entrée* `(R, M, G, I)` *telle que* `(DestAdr AND M)= R` **alors**
      Routage comme spécifié dans la table (direct si `G = *`, indirect sinon) sur l'interface `I`
    sinon
      si ∃ *route par défaut* `(0.0.0.0, 0.0.0.0, G, I)` **alors**
        Routage indirect via `G` sur l'interface `I`
      sinon
        Envoyer une erreur *host unreachable* ou *unreachable network* à l'application émettrice

Il est à noter que si plusieurs entrées de la table de routage conviennent (*match*), l'algorithme choisit celle dont l'adresse possède le plus de bits identiques avec celle du paquet (*best matching*).

Exemple de routage

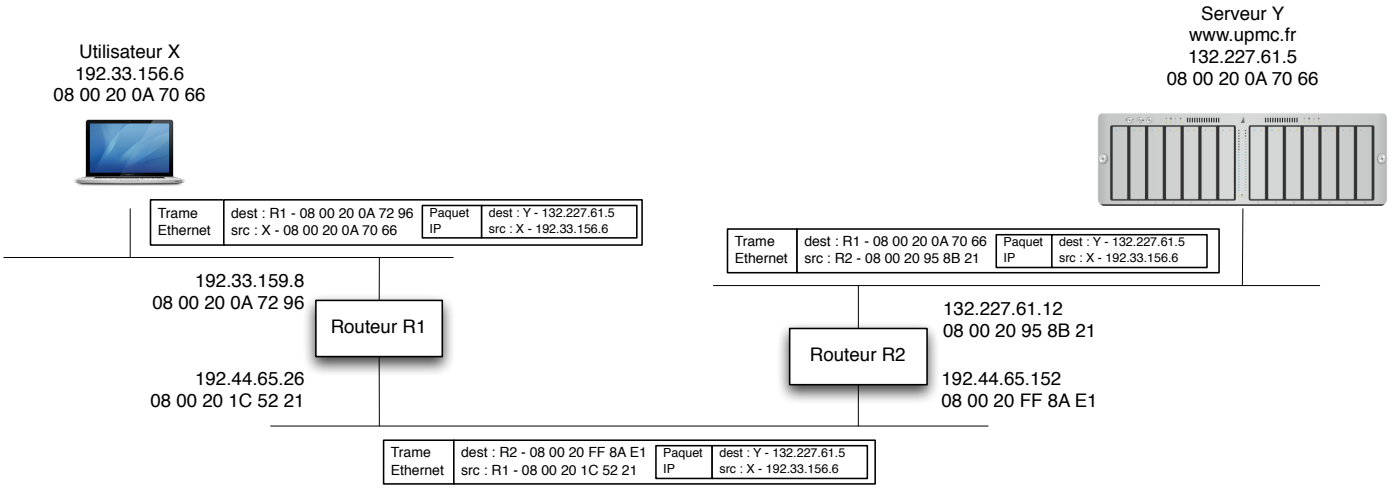


FIGURE 8.10: Routage d'un paquet d'un utilisateur X vers un serveur Y

8.3 Fragmentation

Un message utilisateur de 2048 octets de données et de 20 octets d'en-tête est envoyé d'un hôte A vers un hôte B en empruntant un seul routeur. Le premier sous-réseau traversé utilise un en-tête de trame de 14 octets et une MTU (*Maximum Transfer Unit*) de 1024 octets, alors que le second sous-réseau utilise un en-tête de trame de 8 octets et une MTU de 512 octets.

On considérera que :
- La MTU inclut les en-têtes de trame et les en-têtes IP ;
- Le champ "Fragment Offset" est codé sur 13 bits ; il exprime, en unités de 8 octets, la position relative des données contenues dans le fragment par rapport au datagramme initial ;
- Aucune option IP n'est utilisée ;
- Les sous-réseaux n'introduisent pas de déséquencement.

Chapitre 9

UDP et TCP : Protocoles de transport

9.1 UDP (*User Datagram Protocol*)

9.1.1 Format de message

Figure 9.1: Paquet UDP

9.1.2 Fonctionnalités assurées par UDP

UDP est le protocole de transport (i.e de bout en bout) le plus simple qui soit. Il se contente d'étendre le service de remise de machine à machine (offert par IP) en un service de communication de processus à processus.

A ce titre, il assure essentiellement une fonction de ***(dé)multiplexage*** grâce aux champs `Source Port` et `Destination Port`, qui permettent d'identifier (quoique de façon indirecte) les processus de communication sur les deux machines distantes. On notera que c'est la paire (port, hôte) appelée *socket*, qui constitue la clé de démultiplexage pour UDP.

S'il n'y a pas de contrôle d'erreur à proprement parler, il y a néanmoins une fonction de **détection d'erreur** (optionnelle pour IPv4 mais obligatoire pour IPv6) qui repose sur le champ `Checksum` : elle permet de vérifier que le message UDP a bien été délivré entre les deux bons ports (l'adresse IP `dest` aurait pu être modifiée lors du transit su paquet, ce qui aurait provoqué une remise à un mauvais destinataire). Lorsqu'il est calculé, le *checksum* porte sur la totalité du message UDP augmentée d'un pseudo-header comportant entre autre les @ IP source et destination (en conséquence de quoi UDP repose forcément sur IP).

En théorie les messages UDP peuvent être fragmentés par IP. En pratique, la plupart des applications utilisant UDP limitent la taille de leurs messages à 512 octets, pour éviter toute fragmentation.

9.2 TCP (*Transmission Control Protocol*)

9.2.1 Généralités

TCP est u protocole de transport, donc de bout en bout, offrant un service de remise **fiable** de flux d'octets en **full-duplex** et en **mode connecté**. Il met notamment en œuvre des mécanismes de :
– (Dé)multiplexage ;
– Contrôle d'erreur ;
– Contrôle de flux ;
– Contrôle de congestion.
À la différence des protocoles en mode connecté que nous avons pu voir jusqu'à présent, TCP n'utilise qu'un seul format de segment qui sert aussi bien à établir/libérer une connexion qu'à transférer des données :

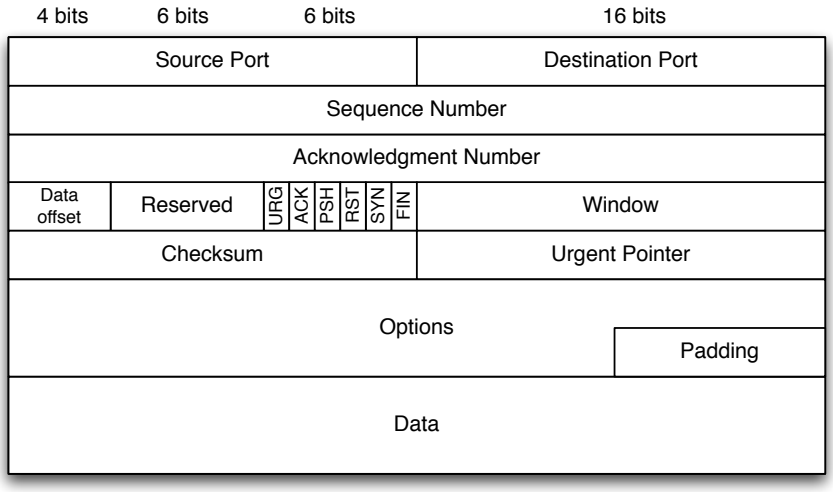


FIGURE 9.2: Paquet TCP

Source Port et Source Destination Identifient les extrémités locales de la connexion. Les numéros jusqu'à 1024 correspondent à des ports réservés (eg. le port 21 correspond à FTP, le port 80 à HTTP) ;

Sequence Number est le numéro de séquence du premier octet de données du segment TCP, si le drapeau SYN est à 1, ce numéro est l'ISN (*Initial Sequence Number*) et le premier octet de données sera numéroté ISN + 1 ;

Acknowledgment Number est le numéro d'acquittement, si le drapeau ACK est à 1, ce numéro contient la valeur du prochain numéro de séquence que l'émetteur est prêt à recevoir ;

Data Offset est la longueur de l'en-tête TCP exprimée en mots de 32 bits ; elle indique donc où les données commencent ;

Reserved n'est pas utilisé et doit être mis à zéro ;

URG drapeau positionné à 1 si le pointeur d'urgence est en cours d'utilisation ;

ACK drapeau positionné à 1 pour indiquer la validité du numéro d'acquittement ;

PSH drapeau positionné à 1 pour indiquer au destinataire de remettre les données à l'application concernée dés leur arrivée ;

RST drapeau positionné à 1 pour réinitialiser une connexion devenue incohérente, pour rejeter un segment altéré ou pour rejeter une tentative d'ouverture de connexion ;

SYN drapeau positionné à 1 lors d'une phase d'établissement de connexion ;

FIN drapeau positionné à 1 lors d'une phase de libération de connexion ;

Window fenêtre d'anticipation de taille variable ; la valeur de ce champ indique au récepteur combien il peut émettre d'octets après l'octet acquitté ;

Checksum champs de contrôle portant sur tout le segment augmenté d'un pseudo en-tête, constitué principalement des adresses IP source et destination ;

Urgent Pointer pointeur indiquant l'emplacement des données urgentes ; utilisé uniquement si le flag `urg` est positionné à 1 ;

Options champs de longueur variable offrant des possibilités non offertes dans l'en-tête de base (eg. Détermination du MSS (*Maximum Segment Size*), la taille maximum d'un segment TCP (en fait la taille maximum du champs de données d'un segment TCP, car MSS n'inclue pas l'en-tête)) ;

Padding champs permettant de cadrer l'en-tête TCP sur des mots de 32 bits ;

9.2.2 Établissement de connexion

L'établissement d'une connexion TCP se fait par échange de trois messages (*three-way handshake*) :

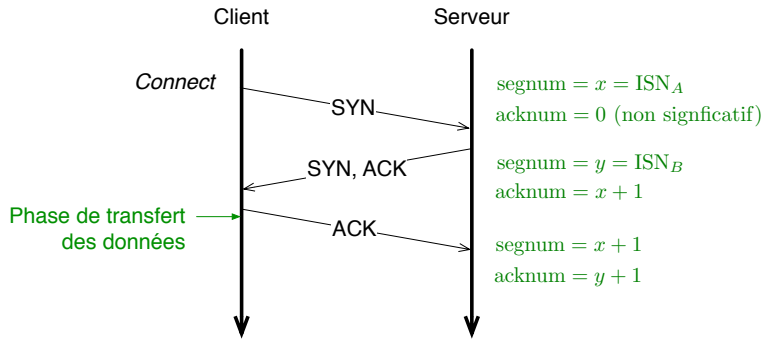


FIGURE 9.3: Etablissement de connexion

Remarque. Avec 2 messages, le sous-réseau pouvant déséquencer les paquets, l'appelant de la connexion reçoit un segment de données pour une connexion qui est en attente de confirmation d'établissement.

Avec 3 messages, l'appelé doit attendre de recevoir un acquittement de sa confirmation (sous la forme de données ou d'acquittement) avant de pouvoir émettre ses propres données. On évite ainsi à l'appelant de se retrouver dans un état mal défini.

Une connexion TCP est identifiée par (port S, hôte S, port D, hôte D). Il est tout à fait possible d'ouvrir une connexion entre les machines A et B, ports X et Y, de la fermer, puis quelque temps après, d'ouvrir une nouvelle connexion, toujours entre les machines A et B et es ports X et Y. On parle dans ce cas d'***incarnations*** différentes d'une même connexion. Le scénario suivant est alors possible :

On y voit une seconde incarnation réutilisant un numéro de séquence trop tôt, c'est-à-dire quand il y a encore une chance qu'un segment appartenant à une incarnation précédente interfère avec l'incarnation courante.

Le numéro de séquence initial (ISN – *Initial Sequence Number*) est choisi de façon à éviter cette situation. À l'origine, il devait être tiré aléatoirement ; en pratique, il est généralement mis à jour grâce à un compteur de 32 bits incrémenté toutes les 4 μs.

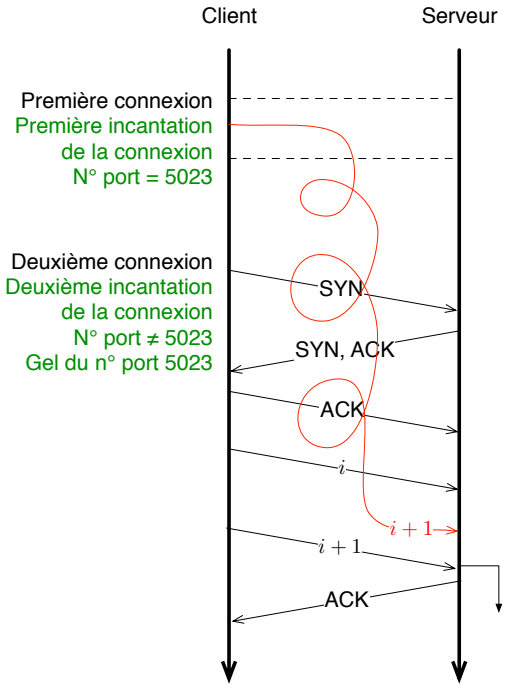


FIGURE 9.4: Problème de numérotation

Remarque. Lorsqu'un hôte A reçoit deux segments SYM en provenance d'un même port hôte B, le second SYN peut être une retransmission du SYN original ou alors une nouvelle requête de connexion (cas de panne suivie d'un redémarrage de B). Si le SYN est une retransmission, alors sa valeur de ISN est la même que dans le premier SYN. Dans le cas contraire, et si les valeurs d'ISN sont générées par une horloge, alors l'ISN du second SYN sera différent de celui du premier SYN.

9.2.3 Numérotation des données

L'une des principales propriétés de TCP est la fiabilité. TCP doit résoudre les problèmes de données en erreur, perdues, dupliquées ou délivrées dans le mauvais ordre par le système de communication Internet. Cela est réalisé au moyen d'un numéro de séquence assigné à chaque octet transmis, et d'un numéro d'acquittement (ACK) envoyé par le récepteur TCP. Si l'ACK n'est pas reçu pendant une durée de temporisation fixée (*Timeout*), la donnée doit être retransmise. Au niveau du récepteur, les numéros de séquence sont utilisées pour ordonner correctement les segments qui ont pu être reçus dans u ordre incorrect, et pour éliminer les duplications. Les erreurs sont détectées par un champ de contrôle présent dans chaque segment TCP transmis, et vérifié par le récepteur. Si ce champ a une valeur incorrect, le segment TCP est ignoré.

TCP numérote donc ses octets de données et non pas ses segments. Le numéro de séquence du premier octet de données est transmis avec le segment, dans le champ `SequenceNumber` de l'en-tête TCP (32 bits), et est appelé numéro de séquence du segment.

9.2.4 Acquittement des données

Les segments TCP véhiculent également un numéro d'acquittement (`AckNum`) dans le champ *Acknowledgment Number*, qui représente le numéro de séquence du prochain octet de données attendu dans le flux

de données en sens inverse. Un segment TCP est “acceptable” si son `AckNum` est compris dans la plage :

Plus “grand” `SeqNum` envoyé et non encore acquitté $<$ `AckNum` $\leq$ Prochain `SeqNum` à envoyer

Un segment non acceptable est rejeté et provoque l'envoi d'un acquittement portant les numéros `SeqNum` et `AckNum` courants.

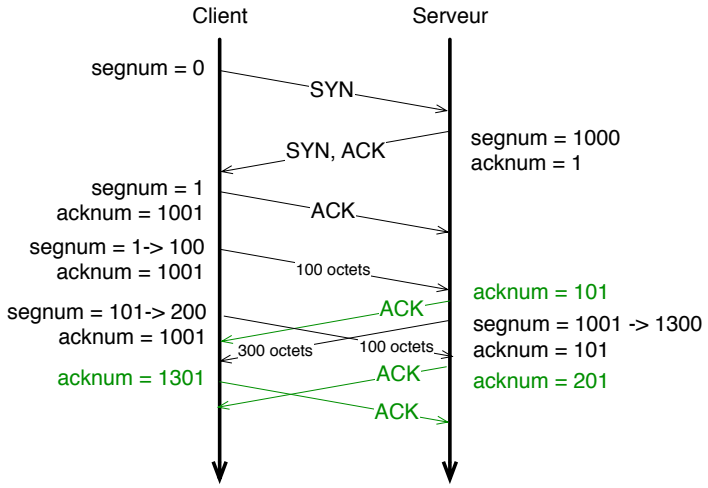


FIGURE 9.5: Acquittement

9.2.5 Contrôle de flux

Pour son contrôle de flux, TCP utilise un mécanisme de fenêtre coulissante de taille variable.

TCP offre un moyen pour le récepteur de contrôler la quantité de données envoyées par l'émetteur. Cela est réalisé en retournant avec chaque acquittement, une “fenêtre” indiquant la plage des numéros de séquence acceptables à partir du dernier segment correctement reçu. Cette fenêtre indique les numéros des octets que l'émetteur a le droit de transmettre avant de recevoir plus d'autorisations du récepteur.

La taille courante de la fenêtre est communiquée dans le champ `Window` du segment.

TCP est orienté octet : la numérotation et la fenêtre portent sur des octets. TCP décide de constituer et d'envoyer un segment lorsque :

1. Dès qu'il y a MSS (*Maximum Segment Size*) octets de données à envoyer MSS = MTU du sous-réseau local – en-tête IP – en-tête TCP (les entités TCP se communiquent leur MSS respectifs au moment de l'établissement (champ options) et c'est la valeur la plus faible qui l'emporte. Par défaut la MSS est de 536 octets (+ 20 en-tête)) ;
2. Lorsque son application lui demande explicitement : TCP supporte une fonction `push` que le processus peut invoquer pour vider le *buffer* des octets en attente d'émission ;
3. Sur expiration d'un temporisateur : pour éviter d'attendre trop longtemps les MSS octets.

9.2.6 Temporisateur de retransmission

Un élément important du contrôle d'erreur est le temporisateur de retransmission.
Il faut mesurer le temps écoulé entre l'envoi d'un octet de données ayant un numéro de séquence donné

et la réception de l'acquittement couvrant ce numéro de séquence. Cette mesure s'appelle **RTT** (*Round Trip Time*). Il faut alors calculer le SRTT (*Smoothed Round Trip Time*) de la façon suivante :

$$SRTT = \alpha \times SRTT + (1 - \alpha) \times RTT$$

et en déduire le temporisateur de retransmission, RTO (*Retransmission TimeOut*) :

$$RTP = \min\left[UBOUND, \max\left[LBOUND, (\beta \times SRTT)\right]\right]$$

où $UBOUND$ et $LBOUND$ sont les bornes supérieure et inférieure sur les *timeout*, α est un facteur de pondération et β est un facteur lié à la variance du délai.

9.2.7 Libération de connexion

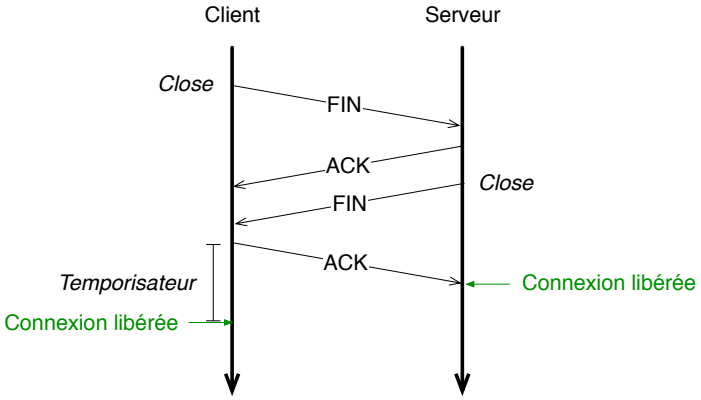


FIGURE 9.6: Libération de connexion

Chapitre 10

Web et HTTP

Réseau d'extrémité :
- Système d'extrémité (ie. machines hôtes) : se situent à la périphérie du réseau et exécutent des programmes applicatifs (eg. email, web...) ;
- Modèle client/serveur : ;
- Modèle pair-à-pair : pas (ou peu) de serveurs (eg. Sykype, BitTorrent, KaZaA...).

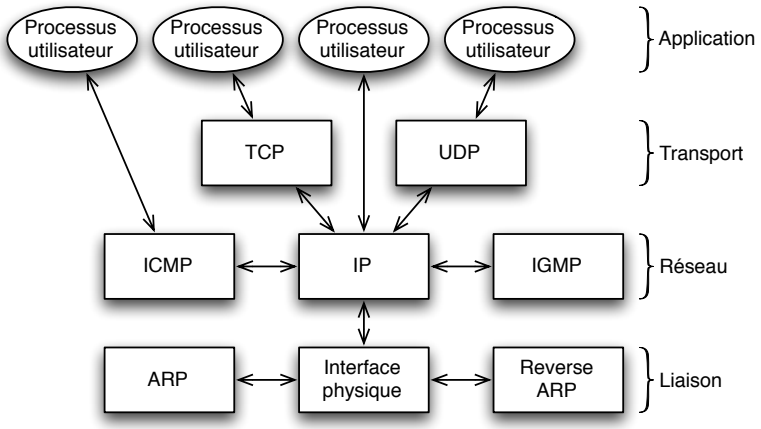


Figure 10.1: Relations entre les protocoles

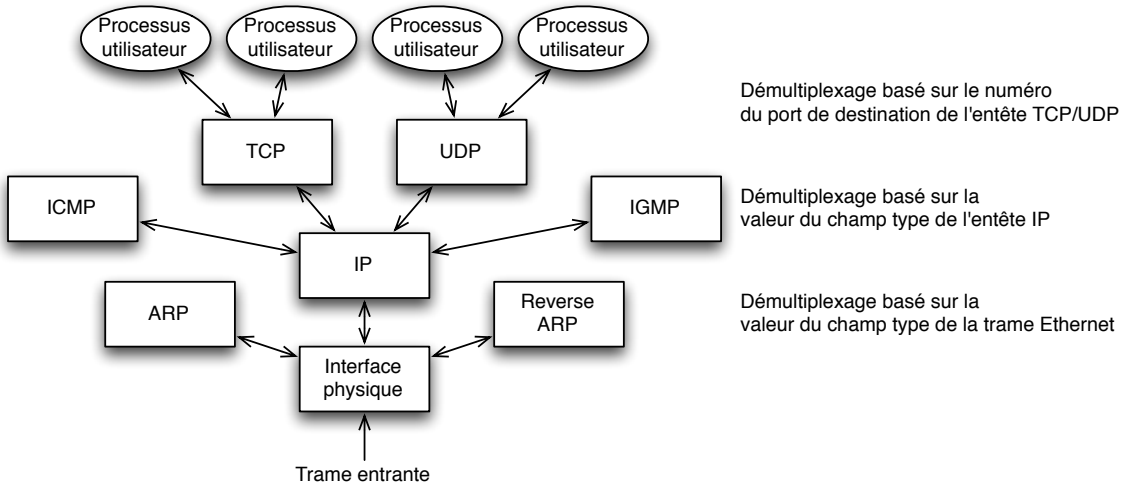


Figure 10.2: Démultiplexage des protocoles

Architecture des applications

Client serveur Des machines clientes demandent et reçoivent un service depuis un serveur *always-on* (eg. navigation/serveur Web).
- Serveur :
  - S'exécute toujours sur un équipement terminal ;
  - Configuré avec une adresse IP permanente ;
  - Ferme de serveurs pour passer le facteur d'échelle.
- Client :
  - Communique avec un serveur ;
  - Eventuellement connecté par intermittence ;
  - Configuré avec une adresse IP qui peut être dynamique ;
  - Ne communique pas directement avec d'autres pairs.

Pair-à-Pair Pas (ou peu) de serveurs (eg. Sykype, BitTorrent, KaZaA...).
- Un pair n'est pas toujours un serveur ;
- Communication directe entre équipements terminaux arbitraires ;
- Les pairs changent d'adresse IP et sont connectés par intermittence.

Communication entre processus

Un processus est un programme qui s'exécute sur une machine hôte.

Sur une même machine hôte, deux processus communiquent entre eux en utilisant un mécanisme de communication **inter-processus** (tubes, segment de mémoire partagée, signaux...) défini au niveau su système d'exploitation.

Les processus s'exécutant sur des machines hôtes différentes communiquent en échangeant des **messages**.
- Processus client : initie la communication ;
- Processus serveur : processus qui boucle, en attente d'être contacté par un processus client.

Remarque. Les applications P2P sont constituées de processus à la fois client **et** serveur.

	TCP	UDP
Description	Orienté connexion : mis en place d'une connexion entre les processus clients et serveur (à l'initiative du client) permettant un transport **fiable** entre les processus émetteur et récepteur.	Transfert de données en mode non connecté non fiable entre processus émetteur et récepteur.
Contrôle de flux	Eviter qu'un émetteur n'engorge un récepteur	Pas fourni
Contrôle de congestion	Régulation du débit d'émission de l'émetteur en fonction de la saturation du réseau	Pas fourni
Garanties	Pas de garanties temporelles ou de bande passante minimum	Respect de contraintes temporelles, bande passante garantie

Applications usuelles

Application	Perte de données	Bande passante	Garantie temporelle	Protocole	Protocole de transport
Transfert de fichiers	Pas de perte	Elastique	Aucune	FTP (RFC 959)	TCP
Email	Pas de perte	Elastique	Aucune	SMTP (RFC 2821)	TCP
Web	Pas de perte	Elastique	Aucune	HTTP (RFC 2616)	TCP
Streaming	Tolérant	Audio : 5kbps-1Mbps ; vidéo : 10kbps-5Mbps	Propriétaire (eg. RealNetworks)	TCP/UDP	
Téléphonie IP	Tolérant	Audio : 5kbps-1Mbps	Propriétaire (eg. Vonage, DialPad)	UDP	

Définition (Socket)**.** Un socket est une interface locale, créée à l'initiative d'une application, contrôlée par le système d'exploitation à travers laquelle les processus applicatifs peuvent envoyer et recevoir des messages à/depuis un autre processus applicatif.

Un socket est donc un point d'accès offert aux applications pour accéder aux protocoles de transport de bout-en-bout (TCP/UDP).

Interface de programmation Socket introduite dans BSD4.1 UNIX en 1981. Elle est utilisable par des applications pour créer, utiliser et libérer des sockets. C'est un paradigme client/serveur. Deux types de services de transport sont accessibles via l'API Socket :
– Non fiable, datagramme ;
– Fiable, orienté flux d'octets.

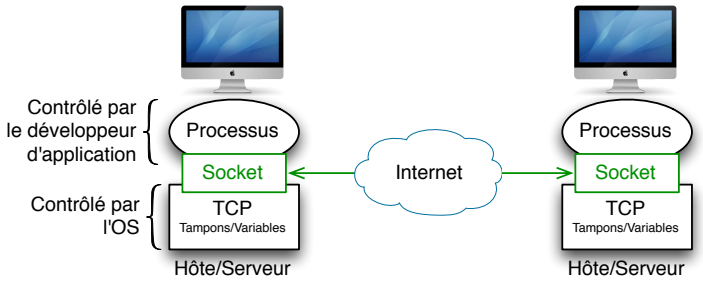


FIGURE 10.3: Démultiplexage des protocoles

10.1 HTTP (HyperText Transfer Protocol)

HTTP est un protocole de la couche application (couche 7). Il implémente deux programmes : un programme client et un programme serveur. Ces programmes s'exécutent sur deux machines distantes et s'échangent des messages HTTP.

HTTP définit le format des messages, et comment le client et le serveur se les échangent.

Une page web est constituée d'objets. Un objet peut être un fichier HTML, une image JPEG, un applet Java, un fichier audio... Ces objets sont adressables par une URL. La page web est constituée d'un document (fichier) de base qui contient des objets référencés.

HTTP utilise TCP de la manière suivante :
- Le client initie une connexion TCP (création d'un socket) vers le serveur HTTP sur le port de destination 80 ;
- Le serveur accepte la connexion TCP émanent du client ;
- Des messages HTTP (du protocole de niveau 7) sont échangés entre le navigateur (client HTTP) et le serveur Web (serveur HTTP) ;
- Fermeture de la connexion TCP .

HTTP est un protocole *sans état* : le serveur ne maintient aucune information concernant les requêtes du client.

Echanges HTTP :

Non persistant Au plus un objet est envoyé ans une connexion TCP. Utilisé par HTTP/1.0.
- Requière 2 RTT par objet référencé (document de base inclus) ;
- Surcharge de l'OS pour chaque connexion TCP ;
- Les navigateurs ouvrent souvent des connexions TCP concurrentes (ie. en parallèle) pour récupérer les objets référencés.

Persistant Plusieurs objets peuvent être envoyés sur une même connexion TCP entre le client et le serveur. Utilisé par HTTP/1.1 par défaut.
- Le serveur laisse la connexion ouverte après l'envoi de la réponse ;
- Les messages HTTP suivants entre le client et le serveur sont envoyés sur la même connexion ;
- Fermeture de la connexion TCP sur un *timeout*.
- Il existe deux types de connexions persistantes :
  - **Sans pipelining** : un client émet une nouvelle requête seulement si la réponse précédente à été reçus. Un RTT pour chaque objet référencé ;
  - **Avec pipelining** : un client envoie une nouvelle requête dès qu'il détecte la présence d'un objet référencé (ie. lors du *parsing* du document de base). Par défaut dans HTTP/1.1.

Types de méthodes :
- Méthode POST : s'applique à la page Web qui propose un formulaire. Les valeurs saisies sont renvoyées au serveur dans le champ *entity body*.
- Méthode GET : les valeurs saisies sont renvoyées dans le champ *url* sous la forme `?key=value&key_1=value_1`.
- Méthode HEAD : demande au serveur de laisser les objets demandés hors de la réponse ;
- Méthode PUT (HTTP/1.1) : permet d'uploader un fichier placé dans le champ *entity body* à l'adresse spécifiée dans le champ *url*.
- Méthode DELETE : supprime sur le serveur le fichier spécifié dans le champ *url*.

Codes de statut HTTP

Code	Description
200	`OK` Requête accédée, l'objet demandé se trouve plus bas dans le message.
301	`Moved permanently` L'objet demandé a été déplacé , son nouvel emplacement est spécifié plus bas dans le message.
400	`Bad Request` La requête n'a pas été comprise par le serveur.
404	`Not Found` Le document demandé n'a pas été trouvé sur le serveur.
505	`HTTP Version Not Supported` Problème de compatibilité des versions HTTP

Chapitre 11

Annexes

11.1 Equipements

Equipement	Couche OSI	Rôle
Amplificateur (*amplifier*)	1	Amplification d'un signal analogique.
Répéteur (*Repeater*)	1	Régénération du signal, dans le cadre d'une transmission numérique (signal numérique ou analogique représentant des données numériques).
Hub	1	Répéteur multiports : répète les informations provenant d'un port vers tous les autres ports raccordables. Il ne limite pas les collisions et n'améliore pas l'utilisation de la bande passante.
Pont (*Bridge*)	2	Relie plusieurs segments d'un réseau local. Filtre les trames en ne laissant passer sur un segment donné que celles qui sont destinées à des équipements situés sur ce segment. Un pont ne génère pas de collisions : il attend le moment propice pour émettre (il peut stocker les trames). Fonctionnement : – apprentissage : le pont construit lui-même sa table d'adresses MAC, au fur et à mesure qu'il reçoit des trames. – algorithme de *Spanning Tree* : pour éviter les boucles. – mode *promiscuous* : mode bloquant dès que l'on émet une trame.
Commutateur (*switch*)	2	Un commutateur de niveau 2 fonctionne selon le même principe qu'un pont, mais il est plus performant grâce à sa matrice de commutation plus rapide.
	3	Utilisé par exemple dans les réseaux X25. Lors de l'établissement du circuit virtuel, il prend des décisions de routage et il garde un état de chaque connexion pour ensuite aiguiller les paquets en fonction de leur numéro de voie logique. Un commutateur de niveau 3 fonctionne dans un environnement homogène, à la différence d'un routeur.
Routeur (*Router*)	3	Prend des décisions de destination ; un routeur possède au moins deux interfaces réseau. Un routeur permet de relier des sous-réseaux différents.
Passerelle (*gateway*)	variable	Terme générique qui permet de désigner des équipements travaillant à différents niveaux : passerelle de niveau 1, 2, 3, etc.

	Mode CV	Mode datagramme
Adressage	Chaque paquet contient une identification de CV	Tous les paquets doivent contenir adresse source et adresse destination ⇒ overhead
Routage	Tous les paquets suivent la même route ⇒ tables de routage + tables de commutation	Chaque paquet est routé indépendamment des autres
Contrôle de congestion	Facile (réservation de ressources)	Difficile

Table 11.1: Commutation Mode CV vs. Mode datagramme

Acheminement	Principe	Avantages	Inconvénients
Voie logique	– mode connecté – les paquets suivent la même route : aucun déséquencement – décision de routage à l'établissement de la connexion ; même route pour tous les paquets – possibilité de préallouer les ressources dans chque nœud traversé par la connexion – possibilité d'utiliser une fenêtre de contrôle de flux sur chaque connexion	– étiquette de petite taille	– vulnérabilité, sensibilité aux pannes d'un nœud u d'une liaison – gestion de la table de translation et place mémoire nécessaire pour la table
Datagramme	– mode non connecté – aucune route n'est établie à l'avance ; décision de routage est prise pour chaque paquet – déséquencement des paquets (routage individuel) – il suffit que la source connaisse l'adresse du destinataire pour pouvoir émettre	– flexibilité ⇒ robustesse face aux défaillances + réactivité face au phénomène de congestion – répartition du trafic (paquets routés indépendamment)	– temps de commutation dans les nœuds plus important (décision de routage)

TABLE 11.2: Routage Mode d'acheminement par voie logique vs. par datagramme

	Algorithme	Avantages	Inconvénients
Etat des liens	Bellman-Ford	– plus stable : chaque routeur est censé connaître la topologie du réseau – permet $\neq$ métriques : chaque LSP pet transporter plusieurs coût – converge plus vite	Aucun
Vecteurs de distance	Dijkstra	– overhead réduit : pas de précaution pour empêcher la corruption des données – moins d'espace mémoire nécessaire	Aucun non plus

TABLE 11.3: Routage à états des liens vs. routage à vecteurs de distance

	Avantages	Inconvénients
Bus	– facile à installer – retrait implicite des informations (terminateur) – topologie passive : la panne d'une machine n'a pas d'incidence sur le réseau ⇒ robustesse	– Risques de collision ⇒ temps d'accès indéterministe – unités d'accès passives ⇒ signal non régénéré (besoin de répéteurs) – non adapté à la fibre optique (support unidirectionnel) – rupture du support ⇒ réseau HS (rebonds du signal aux 2 nouvelles extrémités)
Etoile	– Conflits d'accès générés en central – technologie simple et éprouvée – panne d'une machine ⇒ déconnexion de la branche de la machine en cause – fibre optique utilisable (liaison point à point)	– problèmes de fiabilité du point central – taille du réseau dépend de la puissance du point central – plus le réseau est grand, plus il faut de câblage
Anneau	– pas de taille limite au réseau – fibre optique (liaison point à point) – temps de transfert borné	– Retrait des informations nécessaires – problème de fiabilité si l'anneau se coupe (⇒ doubler l'anneau) – connecteurs complexes et coûteux – topologie active : panne d'une machine ⇒ panne du réseau

Table 11.4: Topologies