

Protocoles & réseaux / Téléinformatique II (2011-2012)

(semestre d'automne)

Revision : 1.59

Marc SCHAEFER

évolution et extension du cours de Romain VOUMARD

8 septembre 2011

haute école  ingénierie
neuchâtel berne jura www.he-arc.ch

ISIC

<http://www.isic-arc.ch/>

Préface

La téléinformatique est aujourd'hui à la frontière des télécommunications, du multimédia et de l'informatique. Internet, conçu au départ uniquement pour les applications informatiques a pendant longtemps été transporté sur des technologies et réseaux WAN construits et conçus pour la téléphonie : aujourd'hui, comme un clin d'oeil de l'histoire, on transporte la téléphonie et d'autres flux multimédia sur Internet, ce qui n'est pas sans poser de nombreux problèmes.

La connaissance des standards d'aujourd'hui permet d'assurer l'interopérabilité des solutions développées. La maîtrise des concepts fondamentaux assure la fiabilité, la sécurité et la performance de ces solutions et permet de prendre conscience des limites de ces technologies, tout en jetant un regard sur leur avenir.

Ce document fait un tour d'horizon des sujets traités au premier semestre. La plupart sont approfondis par des présentations *ex-cathedra* et lors du travail personnel de l'étudiant (exercices, laboratoires, présentations et approfondissement).

Sommaire

Sommaire	iii
1 Théorie de l'information	1
2 Le traitement des erreurs de transmission	11
3 Protocoles sûrs (protocoles à fenêtre)	19
4 Le dernier kilomètre (the last mile)	27
5 Transmission numérique	33
6 Transmission sans fil	45
7 Sécurité des réseaux	51
Références et bibliographie	57
Index des concepts	59
Table des matières	65

Chapitre 1

Théorie de l'information

Sommaire

1.1	L'information	1
1.2	Le codage	2
1.3	Théorie de l'information	3
1.3.1	Types de sources	3
1.3.2	Conversion analogique/digitale	3
1.3.3	Quantité d'information	4
1.3.4	Entropie	5
1.3.5	Quantité de décision et redondance	5
1.4	Les limites de canaux de transmission	6
1.4.1	Etats électriques	6
1.4.2	Bande passante d'un canal parfait	6
1.4.3	Bande passante d'un canal physique (réel)	7
1.4.4	Rapport signal sur bruit	7
1.5	La compression sans perte	7
1.5.1	Méthodes	7
1.5.2	Problèmes	8
1.5.3	En pratique	8
1.5.4	Dynamic Huffman	8

1.1 L'information

L'information, telle qu'elle est traitée par les ordinateurs, n'est en général pas codée de façon optimale pour la transmission. Même si le débit des lignes (en particulier à grande distance) augmente sans cesse, il n'atteindra jamais celui disponible à l'intérieur d'un ordinateur. De plus, les besoins liés aux nouvelles applications multimédia augmentent sans cesse : les applications mobiles ne disposent pas encore de toute la performance nécessaire. Enfin, certains réseaux d'accès facturent au volume (GPRS p.ex.). Une représentation plus optimale de l'information lors de la transmission – voire du stockage – est toujours intéressante.

Par exemple un texte codé en ASCII, comme celui-ci¹, contient une grande **redondance** ce qui signifie qu'il est possible de définir un autre **codage** de la même information, mais qui utilise beaucoup moins de place (de bits). Comme la lettre e apparaît très souvent en français on

1. L'original de ce document L^AT_EX est en ISO-8859-1 (Latin-1). Parler d'ASCII est ici un abus de langage.

pourrait définir un code plus court pour le e que pour le k qui lui n'apparaît que très rarement. Le code Morse est une application de ce principe.

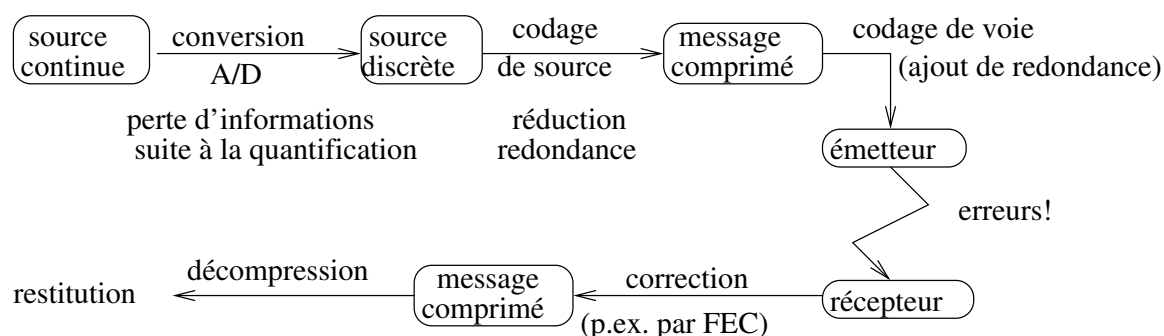
D'autre part on observe un **taux d'erreur** bien plus grand sur les lignes de télécommunication que sur le bus d'un PC. Il faut donc *préparer* les données pour les transmettre dans de bonnes conditions. On peut facilement constater que le code ASCII ne permet pas de détecter des erreurs de transmission car toutes les combinaisons de valeurs des 7 bits du code sont des valeurs admissibles. Pour détecter – voire corriger – des erreurs de transmission, il faut mettre en place des algorithmes visant à augmenter la redondance de l'information (voir chapitre 2 en page 11).

1.2 Le codage

Coder, c'est représenter un **alphabet** donné (ASCII 7 bit, UNICODE, états d'un automate) sous forme informatique : sous forme de bits, qui pourront être transmis par la **couche physique**.

La forme du codage est très importante : elle détermine l'efficacité (et donc la performance) ainsi que la résistance aux erreurs de transmission.

On distingue le codage de source et le codage de voie (ou de canal) :



Le codage de source a pour but de *réduire la redondance*. C'est ici que l'on trouve les algorithmes de **compression** et de décompression. La compression peut se faire **avec perte** (compressions audio et vidéo) ou **sans perte** d'information (compression de données selon Huffman, Lempel-Ziv, ...). Le codage de source est abordé dans la section 1.5 en page 7.

Il est évident que des données compressées sont beaucoup plus sensibles aux erreurs de transmission : une seule erreur sur un bit peut empêcher la compréhension de toutes les données depuis le point de corruption, en particulier vu que les mots codes sont probablement de taille variable.

Le codage de voie a pour but de protéger les données contre les perturbations. On réintroduit un peu de redondance dans le but de détecter (et éventuellement) de corriger les erreurs de transmission créées par les perturbations (bits de parité, CRC, ...).

Le codage de voie est vital dans toutes les applications de téléinformatique et est traité dans le chapitre 2 en page 11.

1.3 Théorie de l'information

Claude SHANNON, des laboratoires BELL² décrit en 1948 les bases de la théorie de l'information en communication électronique [10].

1.3.1 Types de sources

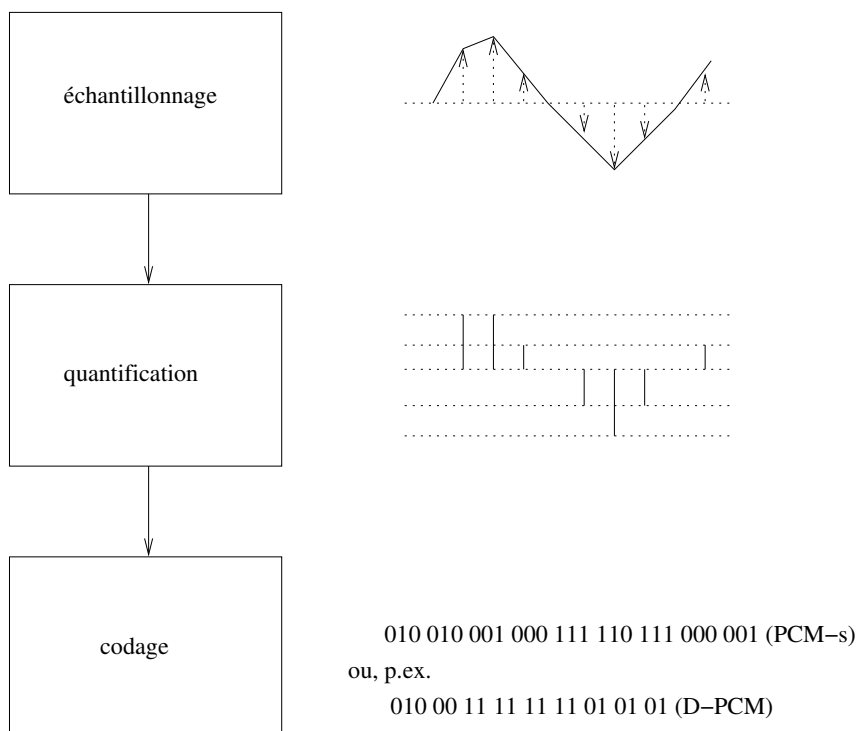
On appelle *source d'information discrète* un système capable de générer un flux d'information selon une loi statistique donnée. Une source discrète possède un alphabet *fini*. Elle ne peut générer qu'un nombre fini de symboles (chiffres, lettres, ...). Si la probabilité l'apparition d'un symbole est **indépendante** des symboles apparus jusque là, on parle de source *sans mémoire*. Si au contraire on a une dépendance, on parle alors de source de *Markov*.

Les langues naturelles (français, anglais, ...) forment une source de Markov : la probabilité d'apparition d'un e après un a est beaucoup plus faible que celle d'un n après un a. Donc si un a est apparu, la probabilité d'apparition d'un n augmente rapport à celle du e.

Nous traiterons en règle générale le cas sans mémoire : la probabilité d'apparition d'un symbole ne dépendant que d'une probabilité absolue (ou d'une répartition dans un tampon de données fini par exemple) et non pas du passé (des symboles précédemment émis par la source). Notamment les algorithmes de compression Huffman et Shannon-Fano ne tiennent pas compte du contexte (au contraire p.ex. de Lempel-Ziv).

1.3.2 Conversion analogique/digitale

La plupart des sources naturelles sont analogiques (on dit aussi continue, par opposition aux sources discrètes). La conversion d'une source analogique à une source discrète ou numérique se fait par la conversion A/D :



2. Connus pour l'invention du transistor et la création d'UNIX, aujourd'hui faisant partie du groupe LUCENT.

La première phase est l'**échantillonnage**, qui prélève $2f_{max}$ échantillons/seconde de l'information analogique reçue : par exemple, la téléphonie analogique occupant le spectre de 300 à 3400 Hz, **G.711** définit $2f_{max}$ à 8000 échantillons par seconde, soit un échantillon toutes les 125 μs . Cette cadence est suffisante, selon le théorème de **Nyquist**, pour échantillonner tout signal compris dans une bande de fréquence $0 < f < 4$ kHz, condition que l'on assure usuellement par un **filtre passe-bas**.

La deuxième phase est la **quantification** : G.711 définit une quantification à 256 niveaux (8 bits), ce choix conduit au débit bien connu de 64 kbits/s des canaux B de l'ISDN, notamment. Graphiquement, on peut la représenter par un escalier qui approche les valeurs échantillonnées. Dans le cas de G.711A et de G.711 μ , le pas de quantification n'est pas constant³ : les plus faibles amplitudes ont plus de «détails».

La troisième et dernière phase de **codage** consiste à choisir une représentation binaire adéquate (le code). Cette représentation peut par exemple offrir un avantage de compression via un codage différentiel-adaptatif (**ADPCM**).

La modulation PCM (modulation par impulsions et codage ; Pulse Code Modulation). est notamment utilisée dans les **CD audio**. Ses dérivés (p.ex. **codec** G.711A/ μ) sont utilisés dans l'**ISDN** ou la **voix-sur-IP**.

Notons que si le théorème d'échantillonnage de Nyquist est respecté, seule l'étape de quantification perd de l'information – cette perte unique est un choix initial conscient de résolution. Une fois numérisée, l'information sera régénérée numériquement, sans perte.

1.3.3 Quantité d'information

Chaque **symbole** porte une certaine **quantité d'information** qui dépend de sa **probabilité d'apparition** (moyenne statistique). Les symboles rares portent une plus grande quantité d'information que les symboles fréquents. La quantité d'information correspond intuitivement à la grandeur de la *surprise* causée par l'apparition d'un symbole.

Si la probabilité d'apparition du symbole i est notée P_i , alors la quantité d'information du symbole i vaut :

$$H_i = -\log_2[P_i] \quad (1.1)$$

L'unité de la quantité d'information est le **shannon** (Sh). Un shannon vaut un bit.

La quantité d'information de symboles émis conjointement est égale à la somme des quantités d'information des symboles :

$$H_{ijk} = H_i + H_j + H_k \quad (1.2)$$

(sous-entendu : pas de dépendances entre symboles, sources sans mémoire !)

3. Il est logarithmique dans une zone bien précise, la fonction est légèrement différente entre les versions A et μ , nécessitant un transcodage.

1.3.4 Entropie

L'entropie correspond à l'aptitude d'une source discrète à produire de l'information. L'entropie est la quantité moyenne d'information calculée sur l'ensemble des symboles de la source, ce qui dans le cas d'une source **sans mémoire** se calcule comme :

$$H = \sum_i P_i H_i = - \sum_i P_i \log_2[P_i] \quad (1.3)$$

Cette équation est simplement une moyenne (ou une **espérance**). On peut la rapprocher de l'équation

$$l_m = \sum_i P_i l_i \quad (1.4)$$

qui donne la longueur moyenne des symboles émis par une source, connaissant la longueur en bits de chaque symbole et la probabilité d'apparition.

L'entropie donne en $\frac{Sh}{\text{symbole}}$ le nombre de bits minimal moyen pour représenter un symbole de la source. Elle est maximale lorsque tous les symboles de la source sont équiprobables.

Une source binaire ne produisant que des 1 a une entropie nulle : elle ne produit aucune information. Une source binaire qui produit à chances égales des 1 et des 0 a une entropie (maximale) de 1 (ce qui correspond au codage trivial binaire de 1 bit par symbole dans ce cas, vu qu'il y a 2 symboles ou états – voir l'équation 1.5).

1.3.5 Quantité de décision et redondance

La quantité de décision est la longueur moyenne d'un symbole codé d'une certaine manière.

La quantité de décision triviale (sans tenir compte de la répartition des symboles) dépend uniquement du nombre de symboles de la source :

$$D = \log_2(n) \quad (1.5)$$

où n est le nombre de symboles de la source.

Elle est au minimum égale à l'entropie et ce, dans le cas où les symboles sont équiprobables. Un codage plus optimal voit D s'approcher de H , sans forcément l'atteindre.

La redondance est donnée par la différence entre la quantité de décision et l'entropie :

$$R = D - H \quad (1.6)$$

C'est cette redondance que l'on essaie d'éliminer par un **codage** plus efficace (p.ex. via une **compression entropique** des données).

1.4 Les limites de canaux de transmission

1.4.1 Etats électriques

L'information à transmettre doit être codée d'une façon adaptée⁴ au support de transmission utilisé. Pour la transmission par un conducteur électrique on peut envisager plusieurs types de codages :

- par l'utilisation de deux tensions, l'une représentant la valeur binaire 0 et l'autre représentant la valeur binaire 1
- par l'utilisation de quatre tensions, chaque tension représentant une paire de bits (-3V=00, -1V=01, +1V=10, +3V=11)
- par l'utilisation d'un nombre de tensions égal à une puissance de 2, ce qui permet de transmettre un nombre de bit égal à la puissance de 2 par état du canal
- par l'utilisation de la modulation d'une porteuse (en fréquence, en amplitude ou en phase) à la place des tensions utilisées ci-dessus (voir même en combinant plusieurs types de modulations afin de créer un grand nombre d'états électriques différents du canal)
- ...

Pour la transmission sur une fibre optique on pourra représenter la valeur binaire 1 par la présence de lumière et la valeur 0 par l'absence de lumière (**OOK**, On Off Keying). C'est un cas particulier.

Le débit de données obtenu est donné en bits par seconde (bps). Si on fait usage de quatre tensions et donc qu'on transmet 2 bits par état (tension) du canal, le débit en bps est le double de la fréquence de changement d'état du canal (ou rapidité de modulation) puisqu'on transmet 2 bits par état.

Cette fréquence de changement d'état du canal est donnée en **Baud** (Bd). Certains modems utilisent jusqu'à 16384 états du canal par combinaison de modulations d'amplitude et de phase et transmettent ainsi 14 bits par état. On a donc un débit de 33600 bps pour une rapidité de modulation de 2400 Baud.

La relation entre **débit binaire** (**débit de décision**, \dot{D}) et fréquence de changement d'état du canal (**débit de moment**, \dot{M}) est la suivante :

$$\dot{D} = \log_2(m) \dot{M} \quad (1.7)$$

où m est le nombre d'états du canal par baud (par symbole).

1.4.2 Bande passante d'un canal parfait

Tous les canaux de transmission ont une **bande passante limitée**. Ils se comportent en général comme un **filtre passe-bande**. Il n'est pas possible d'augmenter arbitrairement la rapidité de modulation. Un canal parfait (sans perturbations, mais limité en fréquence) a donc une bande passante limitée et transmet tous les signaux sans perturbation dans cette bande. Le canal est exempt de bruit.

4. Les critères de choix sont, par exemple : tension moyenne nulle, pas de basses fréquences, pas de hautes fréquences, etc.

Dans ce cas la vitesse de transmission maximale C en bps vaut (Nyquist) :

$$C = 2B \log_2(m) \quad (1.8)$$

B est la largeur de bande du canal en HERTZ (Hz)

m est le nombre d'états du canal

Cette formule donne l'illusion qu'on peut atteindre des vitesses arbitrairement hautes en augmentant le nombre des états du canal. Mais les canaux ne sont jamais parfaits.

1.4.3 Bande passante d'un canal physique (réel)

Une ligne normale présente du bruit (température, interférences, diaphonie). La vitesse maximale sur un canal présentant du bruit est donnée par (SHANNON-HARTLEY) :

$$C = B \log_2\left(1 + \frac{S}{N}\right) \quad (1.9)$$

C : vitesse maximale en bps

S : puissance du signal (Watt)

N : puissance du bruit (Watt)

B : largeur de bande du canal (Hz)

1.4.4 Rapport signal sur bruit

Le rapport entre les puissances du signal et du bruit (**SNR**, Signal to Noise ratio) est souvent indiqué en **décibel** (dB) :

$$\eta = SNR_{dB} = 10 \log_{10} \frac{S}{N} \quad (1.10)$$

S : puissance du signal P_{signal} (Watt)

N : puissance du bruit P_{bruit} (Watt)

1.5 La compression sans perte

1.5.1 Méthodes

De manière à diminuer la redondance (de symbole, p.ex. parce que leur répartition est connue généralement ou localement pour le fichier ou le tampon d'entrées/sorties considéré, ou encore estimée au fur et à mesure), on peut compresser sans perte de différentes manières :

- en considérant que les données ne changent que très lentement (p.ex. périphérique de mesure, différences entre deux images fixes, etc) : **compression différentielle** (ou basée sur des deltas).
- en considérant que certains symboles peuvent se répéter (**Run Length Encoding, RLE**), p.ex. pour le fax.
- en considérant la répartition statistique locale ou globale de chaque symbole pris indépendamment (source sans mémoire) : compression entropique classique (Huffman, Shannon-Fano, etc)
- en considérant la répartition de sous-chaînes ou de sous-textes, en utilisant le fait que les symboles ne sont pas véritablement indépendants entre eux (source à mémoire), p.ex. Lempel-Ziv.

1.5.2 Problèmes

Les techniques différentielles ne sont pas toujours applicables et peuvent propager des erreurs (une resynchronisation consistant en l'abandon régulier du système différentiel peut être recommandée), les techniques basées sur la répartition statistique nécessitent l'échange de tables de compression : des données difficiles à compresser peuvent alors produire un accroissement de la taille nécessaire !

1.5.3 En pratique

Des combinaisons de ces techniques sont souvent employées : p.ex. deltas, permutation de sous-chaînes, suivie de compression de sous-chaînes, et enfin une compression entropique : le fax utilise p.ex. une compression RLE (des bits noirs et blancs) et ensuite une **entropique** classique.

Dans certains cas, pour éviter l'échange de tables de compression, on peut simplement indiquer un type de table standard (p.ex. fréquence Huffman de la langue française) codé sur peu de bits une fois les données à transmettre reconnues par l'émetteur.

1.5.4 Dynamic Huffman

Un autre manière pour éviter de transmettre des tables de compression est d'utiliser un algorithme dynamique (qui crée la table de compression au fur et à mesure).

La compression Huffman dynamique permet de s'affranchir de l'échange de tables de compression entre l'émetteur et le récepteur. On construit, à la fois chez l'émetteur et le récepteur, un arbre de compression qui évolue en fonction des symboles reçus.

On commence l'arbre avec une racine, et un noeud vide spécial (feuille) noté e_0 , formant la branche bit 0 sous la racine de l'arbre.

Le principe est, pour chaque symbole à envoyer, de vérifier si l'arbre le contient déjà.

Si non, on crée une nouvelle feuille, notée c_1 , avec c le symbole concerné et 1 désignant le nombre d'apparition du symbole jusqu'ici. Cette feuille est insérée dans l'arbre à l'endroit où se trouvait la feuille spéciale e_0 , formant alors avec elle un sous-arbre, la feuille spéciale occupant la branche bit 0, et le nouveau symbole la branche bit 1. On émet alors le mot code actuel (lu depuis le sommet de l'arbre) de e_0 , suivi du symbole codé en clair.

Si oui, on envoie simplement le mot code actuel du symbole (lu depuis le sommet de l'arbre), et l'on augmente le nombre d'apparition du symbole.

Dans les deux cas, on assure que les feuilles et noeuds de l'arbre soient correctement répartis à l'aide de l'algorithme suivant :

1. créer une liste des symboles et des poids des feuilles et noeuds de l'arbre (pour les noeuds : somme des poids des noeuds et feuilles situés en-dessous), en parcourant de gauche à droite puis de haut en bas, en partant de e_0
2. vérifier si l'ordre des poids est correct, sinon permuter les noeuds mal classés.

Dynamic Huffman (comme toute table de compression basée sur un arbre) garantit que l'interprétation des codes est univoque. De plus, l'algorithme ci-dessus garantit la synchronicité de l'émetteur et du récepteur.

On retrouve notamment Dynamic Huffman dans **MNP-5**, un protocole de compression des modems. Le **V.42bis**, un peu plus efficace, implémente lui l'algorithme **LZW** (Lempel-Ziv-Welch) qui est également un algorithme dynamique basé sur les sous-chaînes déjà transmises, et donc compresse mieux des données réelles de sources à mémoire.

Chapitre 2

Le traitement des erreurs de transmission

Sommaire

2.1 Protection contre les erreurs de transmission	11
2.2 Distance de Hamming et conditions de détection et correction . . .	12
2.2.1 Poids et distance de Hamming	12
2.2.2 Conditions sur la détection et la correction d'erreur	12
2.3 Détection d'erreur	14
2.3.1 Parité	14
2.3.2 CRC	14
2.4 Correction d'erreur	16
2.4.1 Code de Hamming	16
2.4.2 Codes de correction d'erreur	17

La **couche physique** n'est jamais totalement fiable¹. Il peut toujours arriver qu'un ou plusieurs bits (**rafale d'erreurs**) soient modifiés au cours de la transmission. Le but de ce chapitre est de présenter des méthodes permettant de détecter, voire de corriger, des erreurs de transmission.

2.1 Protection contre les erreurs de transmission

On peut se prémunir de deux façons contre les **erreurs de transmission** :

- on ajoute aux informations une **redondance** permettant de reconstruire l'information originale correcte (**Forward Error Correction, FEC ; code correcteur**). La redondance ne permet la **correction** que d'un nombre limité d'erreurs. Elle augmente les coûts de communication.
- on ajoute aux informations une redondance permettant de détecter la plupart² les erreurs de transmission mais sans pouvoir les corriger directement. La redondance nécessaire est beaucoup plus petite dans ce cas. En cas d'erreur une **retransmission** est demandée (**couche liaison**).

Ces redondances sont basées sur un code et des propriétés mathématiques.

1. La disposition des bits, le choix de la modulation et une éventuelle correction d'erreur en couche 1 (p.ex. convolution, treillis, turbo-code) peuvent largement améliorer la fiabilité de la couche 1.

2. L'algorithme de correction d'erreur est choisi de manière à détecter les plus probables – voir p.ex. pour les CRCs la section 2.3.2.2.

La variante FEC est généralement complétée par une détection des erreurs résiduelles.

Des erreurs de transmission peuvent aussi se produire ailleurs que sur les lignes de transmission classiques (par exemple si la mémoire d'un ordinateur ou routeur n'est pas fiable). Ceci est beaucoup plus rare que les erreurs de la **couche physique** et peut être détecté ou corrigé par des méthodes similaires (mémoire **ECC**, **parité** sur bus, **CRC**, ...).

Les codes traités ci-dessous sont tous des **codes bloc** : l'information codée est divisible en blocs indépendants de n bits correspondant à des blocs d'entrée de k bits. Le bloc peut être court (8 bits dans le cas de la parité horizontale) ou grand (un message complet dans le cas d'un CRC).

2.2 Distance de Hamming et conditions de détection et correction

2.2.1 Poids et distance de Hamming

La détection ou la correction des erreurs est toujours basée sur des bits de contrôle qui s'ajoutent aux bits de données (redondance) pour former des mots de code.

Etant donné deux mots de code, il est important de connaître le nombre de bits sur lesquels ils diffèrent. Cette distance, définie par HAMMING [11], peut être obtenue en calculant le poids de Hamming (le nombre de 1 figurant dans un mot de code) du résultat d'un **XOR** (ou exclusif \oplus) entre les deux mots de code, par exemple :

$$\begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \oplus \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \text{ d'où une distance de 2.}$$

La connaissance de l'algorithme de calcul des bits de contrôle permet d'obtenir la liste de tous les mots de code possibles : il s'agit de l'ensemble des mots-codes valides. Sur cet ensemble, on peut calculer la distance de Hamming *minimale* entre tous les mots de code valides, pris deux-à-deux.

2.2.2 Conditions sur la détection et la correction d'erreur

2.2.2.1 Intuitivement

L'ensemble des mots de code valides doit être intuitivement disjoint de l'ensemble des mots résultants d'une perturbation unique si l'on veut pouvoir *détecter* une erreur : s'il n'était pas disjoint, on ne pourrait distinguer un mot code qui a subi une perturbation d'un mot code valide (non perturbé).

En d'autres termes, si pour passer de tout mot de code valide à un autre mot de code valide, il faut 2 perturbations au minimum (une perturbation ne donnant pas un code valide car les ensembles sont disjoints), on peut alors **détecter une erreur**. Une autre façon d'exprimer cela

est d'assurer que la distance de Hamming sur l'ensemble des mots-codes valides soit 2 (au minimum).

Pour **corriger une erreur**, il faut en plus de la condition précédente que toute erreur simple e sur un mot-code valide X produisant un mot-code X_e garantisse à la fois le fait que X_e ne soit pas un mot-code valide (détection), mais en plus, qu'aucun autre mot-code valide Y ne produise cet X_e en présence de toute erreur simple e_2 , pour pouvoir le distinguer et donc corriger l'erreur.

Intuitivement, on peut imaginer la situation suivante :

$$X \rightarrow X_e \leftrightarrow Y_{e_2} \leftarrow Y$$

donc que pour passer d'un mot valide à un autre mot valide, il faut au moins trois perturbations simples (représentées ci-dessus par des flèches simples ou doubles). Graphiquement, on peut imaginer que chaque mot-code valide est entouré d'un nuage de mots-codes invalides, qui diffèrent du mot-code valide chacun d'un bit. L'intersection entre tous les nuages entourant des mots-codes valides, pris deux à deux, est vide. On peut alors corriger une erreur simple. Une autre façon d'exprimer cela est d'assurer que la distance de Hamming sur l'ensemble des mots-codes valides soit au moins 3.

2.2.2.2 Formellement

Mathématiquement, on peut exprimer ces quelques concepts comme suit : soit un message m (longueur k bits), l'ensemble de tous les messages possibles \mathbf{M} , la redondance introduite c (comportant r bits), un des mots-codes possibles n et l'ensemble de tous les valides \mathbf{N} , on a :

- $m \in \mathbf{M}$, $n \in \mathbf{N}$
- $n = mx^r \oplus c$ (code binaire exprimé polynomialement, séparable en message et redondance)
- soit $n \in \mathbf{N}$, soit $n_e = n \oplus e$, avec $P_{Hamming}(e) \leq 1$ (poids de Hamming d'au plus 1 : au plus une erreur de transmission)
 - si $\forall e, n_e \notin \mathbf{N}$ (une erreur sur un mot-code valide ne donne pas un mot-code valide), alors on peut détecter une erreur de transmission.
 - si en plus de la condition ci-dessus sur n_e on a de plus $\forall y \in \mathbf{N}, y \neq n, \forall e, P_{Hamming}(e) \leq 1$ et $(y \oplus e) \neq n_e$ (aucune perturbation unique d'un autre mot-code valide ne donne ce mot-code invalide), alors on peut corriger une erreur de transmission.

2.2.2.3 Conditions généralisées

On peut généraliser cette réflexion non seulement à la détection d'un nombre d'erreurs mais aussi à la correction d'erreurs, en utilisant la propriété des ensembles de codes disjoints.

On calcule la distance mutuelle minimale de Hamming pour le code ainsi défini et l'on utilise les règles simples suivantes :

- $D_{Hamming}(\mathbf{N}) \geq (E + 1)$ alors on peut *détecter* E erreurs simples
- $D_{Hamming}(\mathbf{N}) \geq (2E + 1)$ alors on peut *corriger* E erreurs simples

2.3 Détection d'erreur

2.3.1 Parité

L'exemple classique du bit de parité illustre bien le mécanisme de détection d'erreurs. Partant de 128 mots de données de 7 bits, on obtient un code de 128 mots de 8 bits dont la distance minimale vaut 2 grâce au bit de contrôle qui garantit une parité paire (ou impaire). Chaque erreur simple peut donc être détectée par un tel code :

```
0000000 0
0000001 1
0000010 1
0000011 0
0000100 1
...
1111111 1
```

Toutes les erreurs simples sur le premier mot (0000000 1, 0000001 0, 0000010 0,...) conduisent naturellement à des mots interdits. Il en va de même pour chaque mot de ce code.

Il est par contre trivial de montrer que ce code ne peut pas détecter deux erreurs, qui se compensent (à fortiori tout nombre pair d'erreurs). La parité double (horizontale et verticale) permet d'améliorer largement les performances, en ajoutant une correction d'erreur dans certains cas.

La parité était utilisée surtout en mode interactif (terminaux textes avec écho des touches tapées). Le protocole SPI (SCSI Parallel Interface) utilisait également une parité : les versions modernes utilisent plutôt un CRC. Les mémoires des ordinateurs utilisaient également, jusqu'au début des années 90, la parité pour détecter les erreurs – aujourd'hui, si cette détection existe, elle est implémentée par **ECC** (pouvant de plus corriger des erreurs).

2.3.2 CRC

2.3.2.1 Introduction

Les choses se présentent différemment lorsqu'il s'agit d'être efficace avec des trames de grande longueur (par exemple 36000 bits pour FDDI), et pour lesquels des **erreurs en rafales** sont probables.

La technique couramment utilisée s'appelle CRC (**Cyclic Redundancy Check**) et se base sur l'arithmétique polynomiale modulo 2. L'idée principale repose sur un polynôme générateur dont les propriétés seront évoquées plus loin et qui doit être connu de l'émetteur comme du récepteur.

Un algorithme de CRC est aussi une fonction de **hâchage** non sûre³. Les CRCs sont construits sur des **champs de Galois** qui sont des espaces vectoriels de polynômes [19]. Toute perturbation détectée par le générateur est hors de l'espace ainsi généré.

3. Les fonctions de hâchage utilisées en cryptographie sont par exemple : SHA-x, MD5, etc. Elles ont comme propriété garantie qu'il est très difficile, à partir du message original d'exhiber un message modifié qui aurait le même hâchage. S'il est possible d'exhiber un tel message, on parle de collisions. Il a été montré que la fonction MD5 est sujette à des collisions, en particulier si l'attaquant a toute liberté de choix du message original. Il est recommandé d'utiliser aujourd'hui plutôt SHA-x ou une combinaison de plusieurs fonctions de hâchage pour les applications cryptographiques.

Soient donc une séquence de m bits à contrôler formant le polynôme $M(x)$ et un **polynôme générateur** $G(x)$ de **degré** r (avec $m \gg r$; noter que si $G(x)$ est de degré r cela signifie qu'il s'écrit avec $r + 1$ bits) :

1. $M(x) * x^r$ (on ajoute r zéros après le LSB du bloc, *shift left*)
2. $\frac{M(x)*x^r}{G(x)}$ (on obtient un reste $R(x)$ comprenant au plus r bits, car sinon la division par $G(x)$ n'est pas complète)
3. $(M(x) * x^r) - R(x) = T(x)$ (polynôme, complété du reste, qui sera transmis et qui est divisible par $G(x)$ modulo 2)

Après ces opérations effectuées par l'émetteur, le récepteur n'a plus qu'à vérifier la divisibilité de la trame reçue par le polynôme générateur pour savoir s'il s'est produit des erreurs détectables ou pas.

2.3.2.2 Erreurs détectées

En cas d'erreurs, le polynôme reçu peut s'écrire sous la forme $T(x) + E(x)$, où $E(x)$ est en fait le polynôme d'erreur qui marque les bits erronés. On constate que son reste de division par $G(x)$ est égal au reste de division de $E(x)$ par $G(x)$. Il est donc évident que seules les erreurs qui se traduisent par des polynômes facteurs de $G(x)$ ne sont pas détectées⁴. C'est cette remarque qui est à la base du choix du polynôme $G(x)$, qui est donc capital.

2.3.2.2.1 Erreur simple $E(x) = x^i$ si $G(x)$ comprend au moins 2 termes, alors il n'est pas diviseur de $E(x)$.

2.3.2.2.2 Erreur double isolée $E(x) = x^i + x^j = x^j(x^{i-j} + 1)$: si $G(x)$ n'est pas divisible par x^i , ni par $(x^k + 1)$ pour tout $1 < k < (i - j)$ alors il n'est pas diviseur de $E(x)$.

Exemple : $x^{15} + x^{14} + 1$ n'est divisible par $(x^k + 1)$ pour aucun $k < 15$

2.3.2.2.3 Paquet d'erreurs de longueur k paquet d'erreurs $E(x) = x^i(x^{k-1} + \dots + 1)$: si $G(x)$ contient le terme 1 et si $(k - 1) < r$ alors $k \leq r$ (degré de G) alors il n'est pas diviseur de $E(x)$.

Remarque : la probabilité qu'une trame contenant un paquet d'erreurs de longueur $r + 1$ soit considérée comme valide vaut $\frac{1}{2^{r-1}}$.

2.3.2.2.4 Erreurs en nombre impair $E(x)$ n'est pas divisible par $(x + 1)$: si $G(x)$ est divisible par $(x + 1)$, alors il n'est pas diviseur de $E(x)$.

Exemple : $x^{16} + x^{12} + x^5 + 1$ (CRC-CCITT)

Toutes les normes de réseaux locaux (802.x et FDDI) font appel au polynôme **AUTODIN-II (CRC-32)** : $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$, dont les propriétés de détection d'erreurs sont véritablement impressionnantes !

4. Elles font partie de l'espace généré.

2.3.2.3 Applications informatiques et électroniques

Le calcul d'un CRC est souvent effectué en matériel par les circuits d'émission des cartes réseau. Il est parfois également intéressant de pouvoir implémenter un algorithme informatique de calcul de CRC. La méthode intuitive (qui se base sur le principe de la division par soustractions successives coûteuses et nécessite le stockage du message entier) peut être simplifiée par l'observation de propriétés du **calcul par tranches**⁵ d'un CRC.

2.4 Correction d'erreur

Pour illustrer le mécanisme de correction d'erreurs, on peut partir d'un code de longueur 10 dont la distance minimale vaut 5 et qui permet donc de corriger jusqu'à 2 erreurs ($e_{corr} = \frac{d_h-1}{2} = \frac{5-1}{2} = 2$).

0000000000	code le symbole a
0000011111	code le symbole b
1111100000	code le symbole c
1111111111	code le symbole d

Or, sans redondance, il suffirait de 2 bits pour coder ces 4 symboles. L'efficacité du code est donc de $\frac{\log_2 4}{10} = 20\%$ et donc la redondance vaut 80%.

Supposons que l'on reçoit 0000010101. Il y a deux interprétations possibles : soit il s'agit du symbole a (affecté de 3 erreurs de transmission), soit du symbole b, affecté de 2 erreurs de transmission.

Comment décider ? On pourrait affirmer que la probabilité qu'il n'y ait que 2 erreurs est plus grande que celle qu'il y ait 3 erreurs (ce qui pourrait ne pas être vrai en cas d'erreurs non indépendantes). On pourrait aussi combiner cette correction d'erreur à un CRC qui permettrait de déterminer si la correction est juste. Mais sans précautions particulières, la méthode n'est donc pas absolue !

Dans la pratique, la difficulté consiste à trouver des codes comprenant un nombre élevé de mots ayant une longueur et une distance minimale données. On parle de codes optimaux lorsque le nombre de mots atteint le maximum théorique. Pour une longueur de 8 bits et une distance minimale de 3, le code suivant est par exemple optimal avec 20 mots :

00000000, 11010000, 01101000, 00110100, 00011010, 00001101, 10000110, 01000011, 10100001, 10101010, 01010101, 11100100, 01110010, 00111001, 10011100, 01001110, 00100111, 10010011, 11001001, 11111111

Il est frappant de constater qu'il n'y a aucune certitude pour des codes de longueur supérieure à 9 avec une distance minimale de 3.

2.4.1 Code de Hamming

Le code de Hamming a pour caractéristiques de pouvoir corriger une erreur (1-correcteur, distance de Hamming $D_h = 3$). De plus, il est **optimal** : il n'existe pas de code 1-correcteur dont le rendement $U = \frac{k}{n}$, soit le rapport entre le nombre de bits utiles sur le nombre de bits totaux (y compris la redondance) serait meilleur.

5. Voir laboratoire !

On note un code de Hamming comme suit : $H(n, k)$, avec $r = n - k$ (bits de redondance ou de ici parité), avec la propriété que $2^r \geq n + 1$.

Son principe est de transposer le message dans une suite de bits, en évitant les puissances de 2. On fait ensuite la somme modulo 2 des *ordres* des bits dont la valeur est 1, puis on compense par les puissances de 2 (dont les ordres forment des suites de bits dont un seul est à zéro).

Par exemple, avec le code de Hamming $n = 7, k = 4$:

- on peut représenter le positionnement des bits comme suit, avec d_i les données du message de couche supérieure et p_i la redondance générée :

p_1	p_2	d_1	p_3	d_2	d_3	d_4
-------	-------	-------	-------	-------	-------	-------

(on n'a pas besoin de p_4 qui viendrait juste après d_4 car $n + 1 = 8$ et $2^r = 8$)

- il y a effectivement 3 bits de parité ci-dessus, car $7 - 4 = 3$
- le rendement est ici (mauvais) $U = \frac{4}{7} = 57\%$, mais s'améliore avec la taille du code (p.ex. $H(57, 63)$, $U = 91\%$).

On peut alors construire un exemple, en supposant le message 1010 :

p_1	p_2	1	p_3	0	1	0
-------	-------	---	-------	---	---	---

Reste à déterminer les valeurs de p_1 à p_3 :

valeur du bit	ordre	ordre en binaire	valeur retenue
p_1	1	001	
p_2	2	010	
1	3	011	011
p_3	4	100	
0	5	101	
1	6	110	110
0	7	111	
\oplus			101

Comme la somme vaut 101, il faut activer les parités p_1 et p_3 pour compenser. Le mot-code correct est donc 1011010. La *transmission* proprement dite ne se fait pas forcément dans cet ordre-là, on peut aussi utiliser un ordre séparé, représenté sous la forme 1010|101, avec les bits de parité à la fin.

A la réception, on calcule la somme (modulo 2, ou exclusif) des ordres en binaire des bits activés (y compris les parités) et en cas d'absence d'erreur, elle vaut 0. En cas d'une erreur unique, la somme indique le numéro du bit en erreur, que l'on peut facilement corriger. Il est facile de montrer qu'en cas de 2 erreurs, une correction fautive peut survenir : avec plus de 2 erreurs, l'erreur pourrait ne pas être détectée.

2.4.2 Codes de correction d'erreur

Les CRCs sont en général utilisés pour la détection d'erreur : cependant, en choisissant bien le polynôme générateur, il est possible également de corriger jusqu'à $\frac{r}{2}$ erreurs.

En règle générale, les systèmes modernes de correction d'erreurs sont basés sur des **codes convolutifs** (en couche 1, travaillant sur les bits), comme p.ex le **treillis** des modems modernes ; ou des **codes bloc** (**Reed-Solomon**, **Golay**, etc).

Chapitre 3

Protocoles sûrs (protocoles à fenêtre)

Sommaire

3.1 Idle Request (IDLE RQ)	20
3.2 Continuous Request (Continuous RQ)	21
3.2.1 Principes	21
3.2.2 Nombre de numéros de séquence	21
3.2.3 Contrôle de flux	22
3.3 Un exemple : HDLC (résumé)	22
3.4 Rendement des protocoles	23
3.4.1 Rendement intrinsèque	23
3.4.2 Rendement d'Idle Request	24
3.4.3 Continuous request : cas sans retransmissions	24
3.4.4 Ligne réelle	25

Les couches de liaison (2) et de transport (4) ont comme tâche d'assurer une transmission sûre entre deux entités (respectivement sur une liaison physique ou sur un réseau). Dans les deux cas des **erreurs** peuvent se produire :

- le contenu d'un message peut être corrompu (détection par exemple par un **CRC**)
- un message complet peut disparaître suite à une erreur ou une congestion (détection par **numéro de séquence** et **minuterie**)
- l'ordre des messages peut être différent à la réception qu'à l'émission (notamment pour la couche transport)
- un message reçu peut ne pas être destiné au récepteur (erreur de routage p.ex.)

Les protocoles de ces couches doivent aussi assurer que le récepteur n'est pas débordé par le volume des données arrivantes (**contrôle de flux** par le récepteur), voire éventuellement gérer des problèmes de **congestion** du réseau (couche 4) ou de contingences de **qualité de service** pré-négociée (couche 2, voire 4).

Le rôle des protocoles assurant une transmission sûre est de résoudre tous ces problèmes. Dans les explications qui suivent on considère que les données ne sont transmises que dans un sens (émetteur vers récepteur). Le canal de communication entre les deux permet tout de même une communication dans les deux sens, de manière à échanger les messages de contrôle¹.

1. Si un véritable échange bidirectionnel de données de la couche supérieure doit être implémenté, on considérera simplement deux protocoles : un dans chaque direction, voir notamment la section 3.2.1 pour une optimisation.

Le principe général appliqué est que le récepteur confirme par un petit message (ACK, acquittement, quittance) la réception correcte des messages émis par l'émetteur. En cas de messages erroné une **retransmission** du message a lieu.

3.1 Idle Request (IDLE RQ)

La façon la plus simple de résoudre ces problèmes est la suivante :

1. les données à transmettre sont organisées (découpées) en messages (trame, paquet, datagramme, ...)
2. l'émetteur émet un message
3. le message est transmis
4. le récepteur reçoit le message et le contrôle
5. si le message est en ordre le récepteur envoie une confirmation
6. la confirmation est transmise
7. l'émetteur reçoit la confirmation et la contrôle
8. si la confirmation est en ordre l'émetteur envoie le message suivant.

Cette façon de faire s'appelle Idle RQ parce que l'émetteur attend (état Idle) après chaque émission. L'émetteur doit initialiser un timer (une minuterie) à l'émission du message car différents cas peuvent se présenter où il devra spontanément réémettre le message :

- le message peut se perdre complètement
- la confirmation du message peut se perdre

Dans ces deux cas l'émetteur ne recevra pas de confirmation et doit réémettre le message.

Deux façons de réagir à la réception d'un message erroné sont courantes :

- soit le récepteur envoie une confirmation négative (**NACK**) et demande ainsi explicitement la retransmission du message (variante **explicit request**)
- soit il ignore le message et provoque ainsi une retransmission du message lorsque le timer de l'émetteur sera échu (variante **implicit retransmission**)

Dans le cas où une la confirmation d'un message correctement reçu se perd, le message est retransmis par l'émetteur (minuterie, timer). Un timer est donc bien sûr nécessaire dans tous les cas !

Le récepteur reçoit alors deux fois le même message. Il doit pouvoir reconnaître ce cas et confirmer à nouveau la réception mais en ignorant le message lui-même (surtout ne pas l'envoyer à la couche supérieure!). Pour permettre cette reconnaissance il faut bien évidemment une identification des messages, p.ex. sous forme d'une numérotation dans les entêtes de la couche.

On ne peut se baser simplement sur le contenu de couche supérieure (**payload**), qui peut très bien être identique pour deux messages successifs. En fait, dans le cas IDLE REQUEST, deux identifiants différents suffisent car le doute du récepteur ne porte que sur deux messages. Reçoit-il une répétition du dernier message ou un nouveau message ?

On peut utiliser un bit qui est alternativement mis et effacé pour chaque nouveau message. Si le bit a la même valeur que le message précédent, alors il s'agit d'une répétition (par perte du message ou de la confirmation). Sinon c'est un nouveau message.

Idle RQ est simple à réaliser et peut fonctionner sur une liaison **half-duplex** (à l'**alternat**). Le rendement est par contre très bas si les temps de transmission ou/et les temps de traitement

sont longs par rapport à la durée d'émission d'un paquet. Ces conditions sont souvent réalisées, soit parce que le réseau impose des délais importants (Internet!), soit parce que les lignes sont longues (WAN : liaisons internationales ou par satellite). La capacité de la ligne est alors mal utilisée parce que l'émetteur doit attendre après chaque message.

Des protocoles historiques comme **X-Modem** ou Kermit, utilisés sur des lignes séries et des modems, ou encore le **TFTP**, un protocole IP souvent utilisé pour le téléchargement de firmware pour les équipements réseau, utilisent la méthode IDLE REQUEST (dans un des deux modes implicit retransmission ou explicit request).

3.2 Continuous Request (Continuous RQ)

3.2.1 Principes

Pour éviter que l'émetteur ne doive attendre l'arrivée d'une confirmation pour envoyer le message suivant, ce qui, comme on l'a vu, peut causer un délai prohibitif, on peut lui permettre d'envoyer plusieurs messages sans attendre de confirmation. Comme l'émetteur ne sait pas si ses messages arrivent, il doit les conserver afin de pouvoir, si nécessaire, les retransmettre. Comme la mémoire est en général limitée, on limite également le nombre de messages qui peuvent être émis mais pas encore confirmés. Cette limite est appelée taille de la **fenêtre**².

Les demandes de retransmission peuvent également être implicites (pas de confirmation négative) ou explicites (**NACK**). Deux comportements de l'émetteur sont possibles lorsqu'une retransmission est nécessaire :

- seul le message erroné est retransmis. Les messages suivants, qui ont déjà été envoyés ne sont pas retransmis (variante **selective repeat**)
- le message erroné et tous les messages suivants sont retransmis (variante **go-back-n**)

La variante go-back-n est généralement employée car elle est plus simple et ne nécessite pas de tampon chez le récepteur. Si un message est faux il peut ignorer tout ce qui suit, puisque tout sera répété depuis le point de l'erreur. Cette variante est bien sûr moins efficace sur des lignes de mauvaise qualité puisque davantage de données sont répétées après une erreur.

Dans le cas d'un échange bidirectionnel de données de la couche supérieure, on parle de primaires/secondaires combinés : la plupart des protocoles à fenêtre proposent alors une option de **piggy-backing**, soit la combinaison de confirmation d'une direction avec les données de l'autre direction, de manière à augmenter l'efficacité.

Citons notamment **Z-Modem** et **UUCP** comme exemples de protocoles à fenêtre historiques.

3.2.2 Nombre de numéros de séquence

Le tableau ci-dessous résume le nombre de numéros de séquence nécessaires pour chacun des protocoles, en supposant une taille de fenêtre k :

protocole	nombre
Idle Request	2
Continuous Request, variante Go-Back N	$k + 1$
Continuous Request, variante Selective repeat	$2k$

2. Une taille $k = 1$ est le cas dégénéré Idle Request

On peut montrer la raison de ces nombres de numéros de séquence sur un contre-exemple, utilisant une confirmation qui se perd. Dans le cas Idle request, il faut pouvoir différencier entre une confirmation (ACK) qui se perd et le cas où un message s'est réellement perdu : en effet, dans le cas d'une confirmation perdue, l'état du primaire (message courant) et l'état du secondaire (message suivant) sont incompatibles. Deux numéros de séquence alternés permettent de corriger ce problème.

En étendant l'exemple au cas Continuous Request, Go-Back N, on doit pouvoir différencier entre chacun des messages de la fenêtre de k messages, ainsi qu'entre le premier message de la fenêtre courante et le premier message suivant la fenêtre : $k + 1$ numéros de séquence sont donc nécessaires.

Enfin, la variante Selective Repeat nous demande de pouvoir répéter n'importe lequel des messages de la fenêtre, et lui seulement. Il faut donc un nombre d'identifiants égal à 2 fois la dimension de la fenêtre.

Une autre façon de montrer les conditions sur le numéro de séquence est de considérer la mémoire nécessaire du côté primaire et secondaire.

3.2.3 Contrôle de flux

Notons enfin que si le secondaire désire éviter une surcharge de ses tampons de réception, il peut le faire en ne confirmant pas de manière efficace, ou en confirmant tout en demandant à ne pas poursuivre l'envoi (c'est ce que fait le message de supervision RNR, Receiver Not Ready, comme nous le verrons dans HDLC). Cette dernière méthode évite les retransmissions inutiles.

Dans **TCP**, le secondaire peut à chaque confirmation modifier la taille de la fenêtre autorisée (**advertised window**), voire carrément la fermer. Notons que la plupart des protocoles à fenêtre, à part TCP, numérotent les messages plutôt que les octets.

3.3 Un exemple : HDLC (résumé)

Pas traité
en détail
en 2010-
2011

HDLC (High-Level Data Link Control) constitue la base d'une famille de protocoles de la couche de Liaison (couche 2). On peut parler de méta-protocole : la norme est très large et différents sous-ensembles ont été définis pour des usages particuliers (**LAPM** pour les modems, **LAPD** pour le canal D de ISDN, **LAPB** pour X.25, LLC pour les réseaux locaux, etc).

HDLC offre les possibilités suivantes :

- liaisons point à point ou multipoints
- configuration symétrique (balancée, ABM) ou non (ARM)
- scrutation (polling) et/ou envoi asynchrone de données
- variantes go-back-n ou selective repeat
- longueur d'adresse variable
- nombre de numéros de séquence 8 ou 128 (mode étendu)³
- variantes avec ou sans connexion
- dimension variable des trames
- transport transparent
- détection des erreurs grâce à un **CRC** (**FCS** = *frame check sequence*)

Les trames ont le format général suivant :

3. En mode normal, cela signifie que la taille de fenêtre maximum est de 4 en Selective repeat et 7 en go-back-N, voir section 3.2.2 en page 21.

Flag	Adresse	Contrôle	Données	FCS	Flag
01111110					01111110
8 bits	8 ou multiple de 8	8 ou 16	variable	16 ou 32	8

Les **fanions** (*flags*, drapeaux) permettent la détection des débuts et fins de trame : ils ne doivent évidemment pas figurer dans la zone de données, c'est pourquoi la norme prévoit l'insertion par l'émetteur d'un bit de transparence (0) après chaque séquence de cinq 1 consécutifs (uniquement pour les bits situés *entre* les fanions de début et de fin). Le récepteur n'a plus qu'à tenir compte de cette convention (**bit stuffing**) pour obtenir des données correctes.

Le champ *adresse* contient l'adresse du destinataire. Le champ *contrôle* contient toutes les informations nécessaires à la gestion du protocole (type de trame, numérotation des trames, etc). Le champ *données* contient les données à transférer. Il n'est présent que dans certains types de trames. Le FCS est un CRC.

Les trames sont divisées en trois types principaux :

I-Frames information frames : trames de transport des données

S-Frames supervisory frames : trames de contrôle numérotées

U-Frames unnumbered frames : trames de contrôle non-numérotées

Le champ *contrôle* contient de façon très compacte le groupe et le genre exact de la trame, 0, 1 ou 2 numéros de trame et 1 bit de contrôle (bit de scrutation poll/final, décrit plus avant dans ce texte).

Un I-Frame peut confirmer la réception d'un autre I-Frame transmis dans la direction opposée (**piggy-backing**). La fenêtre n'a pas forcément la grandeur maximale de 8. Elle peut être limitée par exemple à 2. Même dans ce cas la numérotation se fait modulo 8.

Le champ N(S) est le numéro de séquence, le bit Poll/Final signifie Scrutation/Fin : il permet à un bus maître/esclave d'autoriser un esclave particulier à prendre le contrôle du bus pour répondre à une requête du maître. L'esclave rend le contrôle du bus après sa dernière réponse avec le bit Final activé. Le champ N(R) contient par convention le numéro de la trame que l'on s'attend à recevoir (et pas celui de la dernière trame reçue correctement !). On confirme donc jusqu'à N(R) - 1.

3.4 Rendement des protocoles

3.4.1 Rendement intrinsèque

Le **rendement intrinsèque** mesure l'efficacité d'encodage du protocole, notamment du rapport entre charge utile (**payload**) et longueur totale des messages (y compris les entêtes).

Il se définit comme :

$$U_{intr} = \frac{l_{utile}}{l_{totale}} \quad (3.1)$$

Il n'est pas influencé par les caractéristiques physiques des lignes (débit, délai), mais uniquement par des choix de conception du protocole (notamment la longueur des messages).

3.4.2 Rendement d'Idle Request

La durée de la transmission complète d'un message dans la variante Idle RQ se compose des temps suivants :

1. durée de l'émission du message ($T_{ix} = \frac{l_{message}}{D}$, longueur du message/vitesse de transmission, $\frac{bit}{s} = s$)
2. temps de propagation du message (T_p) soit la longueur de la ligne/vitesse de propagation pour une ligne donnée : dans le cas d'un réseau c'est la somme des délais dans les nœuds et du transit dans les lignes)
3. temps de traitement du message chez le récepteur (détection d'erreur, génération de la confirmation) (T_{tr}), supposé négligeable
4. durée de l'émission de la confirmation (longueur de la confirmation/vitesse de transmission), supposé négligeable (T_{ack})
5. temps de propagation de la confirmation (comme au point 2)
6. temps de traitement de la confirmation (T_{tc}), supposé négligeable

Pour simplifier les calculs on admet que les temps 3, 4 et 6 sont très petits et peuvent être négligés. Seul le temps 1 est du temps utile. Les temps 2 et 5 sont des temps d'attente pour l'émetteur.

Le rendement U est donné par le rapport entre le temps utile et le temps total :

$$U = \frac{T_{utile}}{T_{total}} \quad (3.2)$$

et comme $T_{total} = T_{ix} + T_p + T_{tr} + T_{ack} + T_p + T_{tc}$ ce qui peut s'estimer à $T_{ix} + 2T_p$, on a :

$$U = \frac{T_{ix}}{T_{ix} + 2T_p} = \frac{1}{1 + 2a} \quad (3.3)$$

avec

$$a = \frac{T_p}{T_{ix}} \quad (3.4)$$

On représentera le rapport $\frac{T_p}{T_{ix}}$ par la lettre a dans les équations qui suivront. Ce rapport a représente le nombre de messages pouvant être transmis, au mieux, pendant que le premier message effectue un aller vers sa destination (délai T_p). Il est donc évident que la valeur $1 + 2a$ est le nombre maximum de messages qui auraient pu être transmis si l'on avait utilisé au mieux l'aller-retour durant un cycle d'Idle Request.

3.4.3 Continuous request : cas sans retransmissions

Dans le cas d'une ligne parfaite (sans erreurs de transmission ni donc de retransmissions), l'on doit considérer les cas suivants pour Continuous request :

- soit la fenêtre est assez grande pour que l'émetteur puisse émettre en permanence. C'est le cas si la confirmation du premier message arrive avant l'épuisement de la fenêtre. Dans ce cas le rendement est parfait et vaut 1.
- soit la fenêtre est trop petite et l'émetteur doit tout de même attendre. Dans ce cas le rendement vaut :

$$U = \frac{k}{1 + 2a} \quad (3.5)$$

La fenêtre est suffisamment grande si : $k \geq 1 + 2a$

3.4.4 Ligne réelle

Les formules ci-dessus sont valables pour des lignes parfaites où aucune retransmission n'est nécessaire. Pour tenir compte des retransmissions il faut calculer la probabilité P_f d'une ou plusieurs⁴ erreurs de transmission dans un paquet de grandeur N sur une ligne ayant un taux d'erreur par bit P .

La probabilité qu'un message soit en erreur :

$$P_f = 1 - (1 - P)^N \quad (3.6)$$

De plus, on peut définir l'espérance de transmission (le nombre de transmission moyen) comme :

$$E = \lim_{j \rightarrow \infty} \sum_{i=1}^j i (1 - P_f) P_f^{i-1} = \frac{1}{1 - P_f} \quad (3.7)$$

on en déduit que

$$U_{retransmission} = \frac{U}{E} \quad (3.8)$$

Les rendements sont alors les suivants :

Idle RQ :

$$U = \frac{1 - P_f}{1 + 2a} \quad (3.9)$$

En Continuous RQ, il faut distinguer les cas **selective repeat** et **go-back-N**. Ces deux cas ne retransmettent effectivement pas les mêmes messages !

4. dès que le paquet contient une erreur il est considéré comme invalide.

Continuous RQ, selective repeat :

$K < 1 + 2a$:

$$U = \frac{k(1 - P_f)}{1 + 2a} \quad (3.10)$$

$K \geq 1 + 2a$:

$$U = 1 - P_f \quad (3.11)$$

(correspond en fait à la probabilité d'aucune erreur)

Continuous RQ, go-back-N :

$K < 1 + 2a$:

$$U = \frac{k(1 - P_f)}{(1 + 2a)(1 + P_f(k - 1))} \quad (3.12)$$

$K \geq 1 + 2a$:

$$U = \frac{1 - P_f}{1 + P_f(k - 1)} \quad (3.13)$$

Remarquons que le rendement en cas go-back-N est simplement celui du selective repeat, divisé par $(1 + P_f(k - 1))$, ce qui baisse le rendement d'un facteur lié au mauvais cas où on retransmet pour rien les $k - 1$ paquets de la fenêtre alors que seul le premier était erroné.

Une analyse de ces formules montre qu'il existe souvent un optimum dans la dimension des messages : avec des messages trop petits la fenêtre ne suffit pas et on a de l'attente chez l'émetteur. Avec des messages trop grands, la probabilité d'erreur dans les messages devient trop grande et beaucoup de messages doivent être répétés, ce qui diminue le rendement.

Chapitre 4

Le dernier kilomètre (the last mile)

Sommaire

4.1	PME et usagers résidentiels	27
4.2	Entreprises	28
4.3	Réseaux d'accès	29
4.3.1	xDSL	29
4.3.2	Câble TV	30
4.3.3	Internet par réseau électrique	31
4.3.4	Boucle locale sans fils (wireless local loop)	31

Autant les réseaux locaux (Ethernet, FDDI) que les réseaux publics à grande distance **WAN** (*Wide Area Network* : SDH¹, ATM², MPLS³ ...) permettent des vitesses de transmission élevées. Pour que les usagers résidentiels ou entreprises bénéficient d'un **accès rapide au WAN** (en particulier à Internet), il faut trouver une solution bon marché et performante pour relier les clients aux réseaux [2].

Le **dernier kilomètre** est le maillon stratégique, qui était originellement en main des monopoles étatiques, et est aujourd'hui soumis à la concurrence et aux offres multiples (dans la plupart des pays), utilisant toutes les technologies possibles, dans la plupart des pays.

4.1 PME et usagers résidentiels

En particulier pour les petites entreprises (PME) et les usagers résidentiels et à court terme, il *était* longtemps impensable, pour des raisons financières, de créer un nouveau raccordement pour chaque immeuble ou appartement. C'est pourquoi les solutions classiques font appel à l'**infrastructure existante** : les lignes téléphoniques, le câble TV, l'alimentation électrique, et la transmission hertziennne (sans-fil) sont possibles, avec chacune ses avantages et ses inconvénients.

Une évolution des dernières années, le **triple play**, consiste à rassembler les offres télévision, téléphone et Internet dans un seul produit, permettant (jusqu'alors uniquement pour les réseaux de câble TV (CATV)) de s'affranchir des monopoles étatiques (Swisscom) pour l'accès aux réseaux WAN. Depuis 2009, les opérateurs concurrents peuvent désormais installer leurs équipements dans les centraux Swisscom et rendront donc le triple play plus concurrentiel : la pla-

1. Synchronous Digital Hierarchy : voir section 5.3.3 en page 39

2. Asynchronous Transfer Mode, voir section 5.4 en page 40

3. Multi Protocol Label Switching : le réseau *core* actuel chez les opérateurs.

teforme dominante d'accès abonné (ADSL⁴ classique sur ligne fixe de Swisscom, en perte de vitesse par rapport au VDSL offrant le triple-play) sera de plus en plus concurrencée par ce **dégroupage**. Il faut constater cependant que rare sont les fournisseurs qui se sont lancés sur ce marché pour le moment (VTX et Sunrise, principalement). De plus, le téléphone reste souvent analogique : les offres de **voix-sur-IP** sont encore limitées.

Au niveau technique, l'émergence de plateformes transportant à la fois la voix, la vidéo et les données informatiques a poussé les opérateurs à standardiser leur infrastructure réseau centrale (**core**) sur **MPLS** (qui est, en bref, de la commutation haute performance d'IP incorporant la qualité de service [12]).

Or, aujourd'hui, le déploiement des fibres optiques jusqu'au client final est en marche, du moins dans les grandes villes en Suisse et dans certains projets pilotes cantonaux p.ex. à Fribourg. Ces technologies, comme **FTTH** (*Fiber To The Home*) ou **FTTB** (*Fiber To The Building*, avec boîtiers de distribution VDSL ou d'autres technologies), relie l'abonné final aux réseaux de très haute performance, nécessaires pour des applications avancées de télévision haute définition à la demande et de nouvelles applications interactives-

Malgré toutes ces évolutions, une constante reste : l'**asymétrie** des liaisons montantes et descendantes : le client Internet est vu plus comme un consommateur qu'un fournisseur de contenu. A l'inverse, des communautés coopératives d'échange symétrique pourraient vivre à travers la mise en place de réseaux décentralisés de type **mesh**⁵, notamment basés sur le sans-fil (p.ex. 802.11), ne visant pas seulement l'accès à Internet bon marché.

4.2 Entreprises

Les entreprises ont un choix plus important de connexions WAN, dans un contexte plus large de l'accès à Internet seul. Les différentes principales avec les accès privées étant l'omniprésence d'offres **symétriques**, proposant un débit identique pour la voie montante et la voie descendante, rendues nécessaires par l'installation de serveurs ou la connexion avec des succursales proches ou éloignées, la mise à disposition de plages d'adresses fixes et la qualité des liaisons (voire une gestion/surveillance intégrée par le fournisseur).

- réseau classique ISDN de base ou primaire vers le réseau commuté (paire symétrique, utilisable également en **HDSL**)
- liaisons ADSL ou VDSL (cuivre) classiques vers Internet
- liaisons **SDSL** (cuivre) vers Internet ou vers une succursale proche
- fibre optique et combinaison de technologies : FTTH, FTTB, **Gbit Ethernet**, pour un réseau Ethernet émulé (**VLAN**), du trafic IP, ou une connexion directe **MPLS**.

A cela s'ajoute la problématique de la connexion inter-succursales, qui peut s'implémenter soit sur la base d'une des méthodes d'accès ci-dessus (p.ex. via un **VPN**⁶ sur Internet ou MPLS[14], ou plus classiquement par des groupes fermés d'utilisateurs (**CUG**)), soit par liaisons directes réelles (ligne louée, p.ex. en SDSL⁷ ou Gbit Ethernet, éventuellement exploitées en MPLS) ou virtuelles (**circuit virtuel** avec **SLA**⁸ géré par un opérateur et traversant son réseau, implémenté aujourd'hui le plus souvent en SDSL (cuivre) ou Gbit Ethernet (**fibre optique** ou connexion directe) vers les équipements de l'opérateur, puis MPLS).

4. Asymmetric Digital Subscriber Line, voir section 4.3.1 en page 29.

5. réseau maillé, voir par exemple le téléphone coopératif <http://www.craslab.org/bricophone/?page=FAQfr> ou le projet SFNet à Genève.

6. Virtual Private Network : réseau privé virtuel

7. Symetric Digital Subscriber Line

8. Service Level Agreement : contrat de qualité de service (débit offert, délais, etc)

Lors de connexions à faible distance, par exemple entre deux bâtiments, la pose d'une fibre enterrée ou l'usage du sans-fil (**faisceaux hertziens** ou **lasers**) se justifient pleinement face à des locations de lignes devant passer par un central.

Les opérateurs de télécommunications proposent aussi, dans le cas des mobiles (**Road Warrior**), ayant besoin d'utiliser les services de l'entreprise, des offres de VPN globaux accessibles dans le monde entier.

Enfin, l'arrivée de la **voix-sur-IP**, en interne mais également pour communiquer avec les succursales voire pour toutes les communications externes, a modifié complètement la vision du réseau (qualité de service, **haute fiabilité**, etc) et des accès au WAN.

4.3 Réseaux d'accès

4.3.1 xDSL

La téléphonie a été le moyen le plus simple de se connecter à Internet : son principal défaut étant son faible débit, lié à la bande passante téléphonique (3.1 kHz). En effet, les technologies les plus modernes de modulations dans les modems analogiques ne permettent qu'une vitesse asymétrique de 56kbps/33kbps (V.90)⁹. Le raccordement de base ISDN (**BRI**), quant à lui, même s'il offre jusqu'à 128 kbps symétrique grâce à une bande passante un peu plus élevée et deux canaux de téléphonie est une solution coûteuse et qui n'a été finalement déployée, pour l'usager final, qu'en Europe, et est en large repli chez les usagers résidentiels.

La **paire torsadée** qui relie les résidences aux centraux téléphoniques a une capacité de transmission bien plus élevée que l'usage qui en est fait par la téléphonie analogique ou numérique (RNIS/ISDN).

Toute la série des normes xDSL fonctionnent sur le même principe : on utilise le haut de la bande passante de la paire torsadée pour transférer des données, en parallèle avec la téléphonie qui utilise le bas de cette bande passante.

Ceci nécessite à chaque bout de la ligne un **filtre** séparant les deux plages de fréquence. L'usager connecte sa téléphonie et son réseau de données sur les deux entrées respectives de son filtre. La sortie réseau de données (xDSL) mène typiquement à un routeur IP contenant également le modem et l'équipement de tramage ; ou alors à un modem sur Ethernet (**PPP-over-Ethernet**¹⁰) ou encore via **USB**.

Du côté du central d'un opérateur A, la sortie du filtre correspondant aux basses fréquences va sur le réseau téléphonique. La sortie haute fréquence mène à Internet par l'intermédiaire d'un équipement couche 1 et 2 (**DSLAM**, *Digital Subscriber Line Access Multiplexer*), connecté à un réseau WAN couche 2 (**ATM**) de l'opérateur B, commutant les trames multiplexées jusqu'à un démultiplexeur chez l'opérateur C (**BRAS**, *Broadband Remote Access Server*) qui s'occupe des sessions **PPP**, de l'authentification (p.ex. **RADIUS**) et des décomptes de trafic éventuels.

Dans le système suisse jusqu'en 2008, A et B sont Swisscom, et C est l'opérateur xDSL concerné (Sunrise, green, etc). L'abonné paie la maintenance de la paire symétrique vers le central et des équipements de téléphonie de ce dernier via une taxe de base (25.50 en analogique, 43 CHF en ISDN) à l'opérateur A (Swisscom). L'opérateur C quant à lui paie une redevance – dont le montant a été maintes fois contesté légalement – pour la mise à disposition du port du DSLAM et pour le transport de données ATM (la qualité étant négociable suivant l'**over-booking** désiré

9. Frisant la limite théorique, grâce au fait qu'une partie de la liaison doit être numérique, en ISDN, du côté serveur.

10. ici Ethernet n'est utilisé que comme transport de trames **PPPoE**, et non pour IP directement

par C). La qualité d'une liaison ADSL n'est donc pas uniquement facteur de la connexion entre le fournisseur C et Internet, mais également du prix facturé pour le transport de données par B. Vu l'importance des redevances, il reste très peu de marge de manoeuvre à l'opérateur C pour rentabiliser son offre (il doit alors agir sur les coûts de fonctionnement : marketing, administration, support technique ou sur la qualité de l'offre). Dès 2009, les opérateurs peuvent déposer leurs équipements dans les centraux et donc se substituer à B, voire à A, ce qui devrait totalement modifier, du moins dans les zones bien desservies, les conditions d'accès¹¹

En ce qui concerne les plages de fréquences définies, les premières implémentations n'étaient prévues que pour les lignes analogiques. Un standard différent (plage de fréquence légèrement décalée vers le haut) est donc nécessaire pour ISDN, ce qui complexifie et renchérit inutilement, rendant l'ISDN en lui-même encore moins intéressant du point de vue commercial.

Les différentes variantes de xDSL (ADSL, **HDSL**, VDSL, etc) correspondent à des débits et modulations différents. La variante prévue pour le grand public est l'ADSL (le VDSL pour les applications triple-play). Elle est bien adaptée à l'accès usager Internet car les débits sont asymétriques (grand débit descendant, petit débit montant). La variante symétrique est le SDSL (voire le VDSL dans certaines de ses implémentations).

Il faut noter qu'ATM est utilisé en ADSL dans la couche 2 pour le transport des données¹², en général via la couche **AAL5**¹³. Un réseau ATM est implémenté entre le point de terminaison du central (DSLAM) et le fournisseur de connectivité IP via son BRAS. Des trames PPP-over-ATM sont encapsulées dans des trames AAL5. Entre le DSLAM et l'abonné, xDSL est employé en couche 1, et soit PPP-over-ATM, soit PPP-over-Ethernet sont utilisés en couche 2.

4.3.2 Câble TV

Le câble TV a été conçu pour le transport en mode simplex (unidirectionnel) et en diffusion (1 vers N) des canaux de la **télévision analogique**. Les premières variantes réalisées prévoyaient l'utilisation du câble TV pour le flux descendant et l'usage du téléphone (modem) pour le flux ascendant. C'est d'ailleurs la méthode toujours utilisée dans certaines offres d'accès Internet satellitaires, le problème ici étant la puissance d'émission chez l'abonné et le coût des transponders supplémentaires.

Dès le milieu des années 90, les exploitants des réseaux TV ont adapté leurs réseaux pour permettre un flux bidirectionnel, en remplaçant notamment les amplificateurs de quartier, voire d'immeuble. L'adaptation du réseau pour permettre l'échange bidirectionnel nécessite également de tirer de la fibre optique jusqu'au distributeur de quartier. Les usagers du quartier se partagent la bande passante disponible, contrairement à ADSL où chaque usager dispose de son propre canal jusqu'au DSLAM. Mais c'est le ratio d'occupation des canaux (**over-booking**) et le dimensionnement de la liaison fournisseur/Internet, qui comme dans le cas de la liaison DSLAM-BRAS, sont réellement déterminants sur le débit véritablement accessible.

Un réseau TV usuel offre une bande passante de 600 MHz, ce qui correspond à une soixantaine de canaux TV analogiques de 10 MHz de bande passante. Le transport des données utilise un ou plusieurs canaux TV non utilisés. En cas de besoin, des canaux supplémentaires peuvent être utilisés, en concurrence avec la fonction première du réseau, les chaînes de télévision, et en particulier l'offre numérique payante.

Les opérateurs proposent en général un bouquet principal de chaînes de base (p.ex. 40 chaînes

11. Rien n'oblige d'ailleurs le fournisseur dégroupé d'utiliser les technologies en voie d'obsolescence et coûteuses comme ATM dans son réseau.

12. en VDSL, la plateforme Swisscom n'utilise plus l'ATM, mais du GBit Ethernet puis le réseau MPLS core.

13. ATM Adaptation Layer 5 : aussi appelée **SEAL**, pour *Simple and Efficient Adaptation Layer*.

analogiques, donc 40 canaux), puis un bouquet complémentaire gratuit nécessitant un décodeur (un peu moins une dizaine de chaînes TV numérisées tiennent dans un seul canal TV), et enfin des chaînes thématiques payantes. A la longue, le seul moyen d'allouer des canaux TV supplémentaires pour l'accès à Internet sera le passage au numérique de plus en plus de chaînes TV. Il est également important de constater que le passage réel à la télévision haute définition ne pourra se faire qu'en sacrifiant entièrement la télévision analogique classique.

Les technologies les plus modernes (p.ex. en 2008, à Urbatel/Lausanne) permettent des débits jusqu'à 20 MBit/s descendants (et 2 MBit/s montants) par abonné via des techniques de modulation avancées à près de 70 MBps par canal TV.

4.3.3 Internet par réseau électrique

Il est également possible de transporter des données sur les lignes d'alimentation électrique. Cela fait très longtemps que les fournisseurs d'électricité utilisent des transmissions de données à faible débit en direction de l'abonné, par exemple pour l'enclenchement et le déclenchement d'équipements à certains moments. Ce procédé est très intéressant car on trouve des prises électriques dans toutes les pièces des habitations, de plus certains pays ou régions disposent de réseaux électriques mais pas de réseaux de données, l'accès Internet par ce biais pourrait alors être intéressant. Il existe également des modems spéciaux, à usage interne, qui permettent d'émuler un réseau Ethernet sur une phase 220V du réseau électrique interne d'un logement, voire d'un bâtiment.

Les signaux servant au transport des données ne traversant pas les transformateurs (fréquence trop haute) il est nécessaire des les extraire avant le transformateur de quartier et il faut que ce dernier ait un accès rapide à Internet. En Europe, le nombre d'abonnés sur un transformateur est relativement faible et le débit proposé est probablement insuffisant pour que les essais régionaux p.ex. proposés par les Forces Motrices Fribourgeoises (Groupe e) deviennent une véritable alternative aux autres réseaux (CATV, ADSL, sans-fil) disponibles déjà assez largement en Suisse. Un dernier problème de cette technologie est la plage de fréquence utilisée qui correspond à des fréquences radio et donc les risques d'inter-brouillages inhérents, les câbles électriques agissant comme antennes.

4.3.4 Boucle locale sans fils (wireless local loop)

Diverses technologies issues soit des réseaux locaux (WiFi 802.11), soit de la téléphonie (**GSM**, **HSCSD**, **GPRS** et déjà **UMTS**), ou encore développées uniquement à cette fin, permettent de se connecter aux réseaux à grande distance. Suivant la densité d'antennes et le nombre de mobiles, les vitesses varient fortement, et en particulier les réseaux WiFi souffrent de **brouillage** et d'inter-brouillage (utilisation d'une bande non réservée) (voir section 6.2.5 en page 47).

La dernière technologie, le **WiMAX**, utilisant une bande de fréquence réservée et donc soumis à redevance OFCOM, est actuellement en production dans quelques régions de Suisse.

Chapitre 5

Transmission numérique

Sommaire

5.1	Transmission numérique de la voix	33
5.2	ISDN/RNIS : Le réseau numérique à intégration de services (résumé)	34
5.2.1	Introduction	34
5.2.2	DSS1 I.430 (BRI) : couche physique	34
5.2.3	DSS1 I.441 (Q.921, LAPD) : couche liaison	36
5.2.4	DSS1 I.451 (Q.931, PLP) : couche réseau	36
5.2.5	Données du canal B	37
5.3	Hiérarchies numériques	37
5.3.1	Introduction	37
5.3.2	Hiérarchique numérique plésiochrone (PDH)	38
5.3.3	Hiérarchique numérique synchrone (SDH)	39
5.4	ATM : Asynchronous Transfer Mode	40
5.4.1	Modèle en couche d'ATM	40
5.4.2	ATM LANE : émulation LAN	42
5.4.3	Permanent Virtual Connection	42
5.4.4	Conclusion	43

5.1 Transmission numérique de la voix

La numérisation de la voix est un des premiers problèmes qui s'est posé lors de la transmission digitale. Ce fait explique le lien assez étroit entre les premières méthodes de transmission (p.ex. **PDH**) ainsi que certains choix des implémentations plus récentes (**SDH**, **ATM**).

La transformation analogique/digitale vue à la section 1.3.2 en page 3 était effectuée originellement par un dispositif spécialisé. Aujourd'hui, on nomme **codec** un algorithme de codage/compression d'une source audio. Le plus connu – et utilisé dans ISDN comme codec téléphonique, ainsi que dans la voix-sur-IP lorsque la bande passante disponible et fiabilité sont suffisantes – est le codec **G.711**¹.

1. Appelé aussi **PCM/A**. Il existe aussi la version US μ comme vu précédemment.

5.2 ISDN/RNIS : Le réseau numérique à intégration de services (résumé)

5.2.1 Introduction

Pas traité
en détail
en 2010-
2011

Le Réseau Numérique à Intégration de Services **RNIS** (**ISDN** : Integrated Services Digital Network) représente l'ultime phase de la numérisation du réseau téléphonique public [7] hors technologie voix-sur-IP. Forts de la technique de numérisation de la voix, les fournisseurs ont pour but d'offrir à chaque abonné l'accès à plusieurs canaux binaires de 64 kbps (56 kbps aux Etats-Unis).

L'accès de base (2B+D ou **BRI** : Basic Rate Interface) offre ainsi 2 canaux binaires à 64 kbps et 1 canal de signalisation à 16 kbps alors que l'accès primaire (30B+D ou **PRI** : Primary Rate Interface) comprend 30 canaux binaires à 64 kbps, 1 canal de signalisation à 64 kbps, et 1 canal de synchronisation et de contrôle de la couche physique.

Les services supportés sur ces canaux binaires (B) vont de la téléphonie classique (voix ou 3.1 kHz brut) à la téléphonie à **large bande**² en passant par la télécopie numérique (groupe 4), le transfert de données (X.75 ou mode brut) ou encore la visiophonie, ...

RNIS permet de connecter plusieurs appareils sur la ligne et d'adresser ces différents appareils individuellement. Du point de vue de la numérotation, un abonné peut disposer de plusieurs numéros (**MSN**, **Multiple Subscriber Number**) qu'il peut attribuer aux différents appareils (téléphone privé, téléphone du bureau, fax, ...). Par exemple, Swisscom propose 3 MSN au raccordement BRI (**Swisscom MultiLine ISDN**), extensible à 5, 7 ou 10 MSN. Le raccordement BRI **DDI** (Direct Dial-In, proposé par Swisscom sous le nom **Business ISDN**), offre des plages de 10 numéros contigus.

Le **canal D** est principalement un canal de signalisation : à la différence des canaux B qui sont des circuits commutés (couche 1), usuellement facturés à la minute, le canal D travaille en mode de commutation de paquets, mettant en oeuvre les couches OSI 1 à 3 entre l'usager et le central de raccordement. Après l'établissement d'une communication, l'abonné peut par exemple être informé d'un appel entrant (canal D) sans que la liaison ne soit interrompue. Une application très répandue du canal D est la vérification de cartes de crédit.

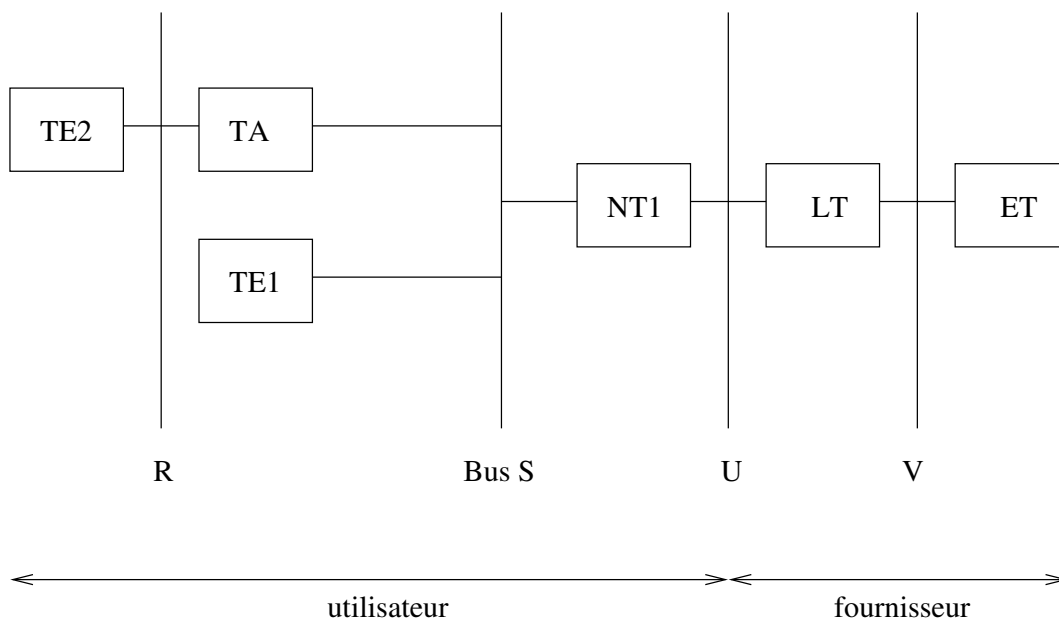
5.2.2 DSS1 I.430 (BRI) : couche physique

5.2.2.1 Topologie et interfaces U/S/T

Du point de vue topologique, on relie en point à point (**interface U**) avec les câbles à paires torsadées existants la terminaison de ligne (LT : Line Termination) du central à un équipement de terminaison de réseau (NT : Network Termination) chez l'abonné, qui donne ensuite naissance soit à une interface T (pour connecter un central d'abonné) soit une interface S (pour directement connecter des équipements), voire éventuellement dans ce dernier cas des équipements analogiques.

2. La téléphonie à large bande (environ 7 kHz) est également disponible en voix-sur-IP, grâce, notamment, au codec **G.722**.

Cas sans central privé



R, S, T, U et V sont des interfaces (S est surtout un bus point à multipoint). TE2, TE1, TA, NT2, NT1, LT et ET sont des unités fonctionnelles.

5.2.2.2 Bus S

Ce NT est le point de départ d'un bus (**interface S**) formé de deux paires torsadées blindées bouclées chacune par une terminaison ohmique de 100 Ohms. La transmission en mode full-duplex est obtenue par annulation d'écho sur l'interface U à 2 fils et par séparation émission-réception sur l'interface S à 4 fils.

Le bus S autorise le raccordement d'au plus 8 équipements terminaux (TE1 : Terminal Equipment 1) par l'intermédiaire de câbles terminés sur connecteurs à 8 broches normalisés ISO 8877 (RJ45) et dont la longueur ne doit pas dépasser 10 mètres. Comme il n'y a que 2 canaux binaires, seuls 2 équipements sur les 8 potentiellement raccordés peuvent être actifs simultanément (le trafic en mode paquet sur le canal D reste cependant possible).

5.2.2.3 Anciens équipements analogiques

Les anciens ETTD (TE2) avec interface V.24 et V.35 peuvent être raccordés (interface R) par l'intermédiaire d'un adaptateur de terminal (TA, souvent intégré à l'équipement NT2ab).

5.2.2.4 Variante centrale d'abonné

Notons que si un central d'abonné est utilisé, le bus S se retrouve alors du côté interne, et entre le NT1 et le NT2 un bus T est utilisé. Techniquement, ce mode se nomme *point-à-point* (le mode bus S étant aussi nommé point-à-multipoint).

Notons que la configuration Swisscom standard dans ce cas (le mode **DDI**, Direct Dial Interface) permet alors d'utiliser des plages de multiples de 10 numéros contigus.

5.2.3 DSS1 I.441 (Q.921, LAPD) : couche liaison

Dans les réseaux publics comme dans les réseaux locaux, les protocoles de la couche liaison de données sont tous issus du protocole **SDLC** (Synchronous Data Link Control) développé par IBM dans le cadre de son architecture SNA. L'ANSI le modifia pour en faire ADCCP (Advanced Data Communication Control Procedure), l'ISO en fit **HDLC** (High-level Data Link Control), tandis que le CCITT adopta à son tour HDLC avec quelques modifications pour en faire **LAP** (Link Access Procedure). LAP fut ensuite intégré à la norme sur l'interface d'accès aux réseaux X.25, avant d'être encore modifié en **LAPB** (Link Access Procedure Balanced) pour devenir plus compatible avec la nouvelle version HDLC.

Désormais, une version **LAPD** est généralement associée à la norme sur l'interface d'accès au réseau RNIS BRI (I.430). Dans le cas des réseaux locaux, l'IEEE a défini avec sa norme **802.2** la demi-couche **LLC** (Logical Link Control) très ressemblante au protocole LAPB. Cette demi-couche LLC est en fait encapsulée au début du champ d'information contenu dans les différents formats de trames prévus par les demi-couches inférieures MAC (voir plus loin).

Le format général d'une trame de type LAPD est le suivant :

Fanion 01111110	DSAP 1	SSAP 1	Contrôle 1 (2)	[Information] (n)	FCS 2	Fanion 01111110
--------------------	-----------	-----------	-------------------	----------------------	----------	--------------------

Le **bit stuffing**, voir section 3.3 en page 23, est utilisé pour éviter que les fanions ne se retrouvent au sein de la trame.

Le champ de contrôle détermine en fait quatre types de trames : **I**, **UI**, **S** et **U** qui signifient Information, Unnumbered Information, Supervisory et Unnumbered. Les commandes de type Information permettent d'acquiescer les trames grâce à la structure suivante (taille en bits) :

0	N(S)	P/F	N(R)
1	3	1	3

Le champ N(S) est le numéro de séquence de la trame LLC, le bit Poll/Final signifie Scrutation/Fin, alors que le champ N(R) contient par convention le numéro de la trame que l'on s'attend à recevoir (et pas celui de la dernière trame reçue correctement !).

5.2.4 DSS1 I.451 (Q.931, PLP) : couche réseau

Ces normes spécifient la signalisation à proprement parler. Elles définissent les messages qui sont utilisés pour l'établissement, l'entretien et la libération d'une connexion RNIS. Elles constituent la couche 3 de ce type de réseau.

On notera que le protocole voix-sur-IP **H.323** utilise également Q.931 dans l'établissement de connexions.

Ces protocoles sont appliqués à l'interface utilisateur du réseau. A l'intérieur de réseau RNIS d'autres protocoles de signalisation sont utilisés.

Le côté appelant indique quel service il souhaite. Les appareils connectés au bus S du côté appelé reçoivent un message indiquant le numéro appelé et le service souhaité (fax, téléphonie, ...). Ces appareils décident s'ils sont capables de traiter ce genre d'appel. Si oui ils sonnent (téléphone) ou acceptent directement la connexion (fax, données). La décision est basée sur le **MSN** et sur le **type de service** demandé (voix, transfert de données).

5.2.5 Données du canal B

Les données transférées dépendent du type de téléservice (p.ex. téléphonie loi A, fax ISDN, etc) et du type de service de support (p.ex. données à 64 kbit/s, voix avec ou sans suppression d'écho, etc).

Une **couche d'adaptation** peut être négociée dans les modes de données numériques. Des fonctionnalités spéciales (p.ex. le conceptbundling de deux canaux pour doubler la vitesse disponibles en utilisant deux communications ISDN séparées) sont également possibles.

Ces fonctionnalités sont négociées à l'appel (**SETUP**, couche 3 du canal D : Q.931) et implémentées en couche 2 des canaux B (LAPB).

Exemples de couches d'adaptation :

Protocole	description
V.110	émulation d'une ou plusieurs liaisons séries asynchrones (de 50 à 38.4 kbit/s) sur un canal B.
V.120	émulation d'une ou plusieurs liaisons séries asynchrones (de 50 à 19.2 kbit/s) sur un canal B.
X.75	flot de données en mode caractère (émulé asynchrone), avec correction d'erreur (basé sur LAPB/HDLC, en mode confirmé, trames numérotées et avec un support potentiel d'adresser divers services (SAP). Cette possibilité de multiplexage est rarement utilisée).
HDLC (raw)	flot brut de données à 64bit/s en mode synchrone, utilisable p.ex. pour mettre en oeuvre de l'HDLC, ou du (sync) PPP, ou encore du MP-PPP

De ce qui précède, on déduit que seul le mode brut permet une implémentation efficace d'un protocole basé sur **HDLC**. Le mode **X.75** convient bien à tout type d'application en mode caractère ayant besoin d'un protocole sûr et les émulations V.110 et V.120 conviennent aux vieux équipements (terminaux, ports série, etc). Notons que seule l'implémentation d'un protocole sûr de couche 2 (p.ex. dérivé d'HDLC) de *bout en bout* de la liaison (pas juste entre les deux TE1 en oubliant les équipements connectés aux TE1) garantit un protocole sûr.

Le protocole MP-PPP, lui, n'est pas défini dans ISDN. Il s'agit d'un protocole Internet qui implémente une communication multilien **MP-PPP** (PPP Multilink Protocol, RFC-1990) via l'utilisation conjointe – facturée – de deux canaux B de support (**bearer**) ISDN pour une connexion à Internet à 128 kbit/s.

5.3 Hiérarchies numériques

5.3.1 Introduction

Une hiérarchie numérique est un ensemble de protocoles permettant de transmettre de manière numérique des informations (voix, vidéo, données) à différents débits. Un des concepts centraux est la manière dont on gère, à la façon d'un réseau de distribution électrique, les lignes à haut débit et les lignes à débit plus faible, et comment l'on multiplexe ces différents niveaux, en

fonction des différents affluents : le problème de la synchronisation (fréquence et phase) est central.

Historiquement, la hiérarchie numérique plésiochrone (PDH) s'est développée en fonction des besoins des réseaux de téléphonie analogique (de la communication aux liaisons inter-continetales en passant par les interconnexions de centraux). Elle permet de gérer des affluents de phases, voire de fréquences légèrement différentes. Son inconvénient principal est la difficulté de démultiplexer un sous-affluent.

Une solution plus moderne est la hiérarchie numérique synchrone (SDH) qui, par son concept de conteneurs glissants référencés résoud de manière très évolutive et ouverte la plupart des problèmes de PDH.

L'intégration des deux technologies est possibles au sein de SDH.

5.3.2 Hiérarchie numérique plésiochrone (PDH)

Le CCITT (ITU-T) a défini une hiérarchie (**G.702**) de 5 (voire 6) ordres de multiplexage numérique, originellement pour la téléphonie, mais qui peut être utilisée également pour le transfert de données.

Cette hiérarchie digitale plésiochrone (du grec *plésio* : voisin, proche ; Plesiochronous Digital Hierarchy) propose un système permettant de transporter des *affluents* de fréquences proches.

Originellement, le but de PDH était de transporter la multitude des communications ISDN entre centraux (p.ex.) par multiplexage, puis au niveau supérieur de transporter ces affluents ensemble. Le problème est qu'il n'était pas possible de garantir des fréquences identiques : même avec des variations très faibles de fréquence, des dérives de phase se produisent alors. PDH doit donc compenser cela par l'insertion de trames vides ou la suppression de trames.

Le problème principal de PDH est le démultiplexage d'un affluent de données : comme la position relative des affluents peut varier (parce que leur phase ou fréquence peut varier dans chaque niveau, sans qu'il ne soit possible de le déterminer facilement), un démultiplexage complet est nécessaire pour atteindre chacun des niveaux de la hiérarchie.

Niveau	nom	vitesse [$\frac{Mbit}{s}$]
0	E0	1 canal à $64 \frac{kBit}{s}$
1	E1	30 canaux sur 2.048 MBit/s
2	E2	120 canaux sur 8.448 MBit/s
3	E3	480 canaux sur 34.368 MBit/s
4	E4	1920 canaux sur 97.73 MBit/s
5	E5	7860 canaux sur 564.736 MBit/s

Les chiffres ci-dessus sont valables pour l'Europe. En Amérique du Nord, les groupements sont différents (24 PCM forment une **T1**).

On voit que le nombre de canaux quadruple à chaque fois alors que le débit progresse un peu plus vite dès le niveau E2. Ceci est nécessaire car les horloges des affluents combinés à chaque niveau peuvent ne pas être parfaitement identiques. La réserve de débit (appelé **surdébit**) ainsi prévue permet de compenser des différences de phase, mais aussi, jusqu'à un certain point, de légères variations de fréquence.

PDH est en fin de vie, remplacé notamment par **SDH**. On retrouve encore les deux niveaux les plus bas de la hiérarchie PDH dans la liaison de base (**BRI**, deux canaux B en E0 + 1 D) et primaire ISDN (**PRI**, 30 canaux B + 1 D + signalisation, E1 en Europe) (recommandation

G.702 pour les débits; **G.703** pour l'interface; **G.704** pour une variante à débit utile de 31 x 64 kBit/s). Notons qu'**HDSL** peut utiliser une organisation similaire au niveau E1.

5.3.3 Hiérarchique numérique synchrone (SDH)

SDH (Synchronous Digital Hierarchy) est dérivé d'une technologie étatsunienne appelée **SONET**. Cette technologie prévoit d'empaqueter les flux dans des *conteneurs* dont la position dans les trames peut fluctuer (variations de phase, voire légères différences de fréquence). Ceci résout le problème de la synchronisation des horloges des différents affluents. Comme des **pointeurs** de la trame indiquent la position des conteneurs, il est aisé d'extraire un flux, sans tout démultiplexer, à chaque niveau de la hiérarchie.

A l'origine, le réseau SONET aux Etats-Unis (1985) fut utilisé pour transmettre des données digitales sur fibres optiques en remplaçant **PDH**, tout en gardant la compatibilité avec ce système pour les niveaux inférieurs de la hiérarchie. Par exemple, dans un module de transport de base **STM-1** on peut trouver des affluents plésiochrones (non synchrones) empaquetés chacun dans un conteneur de taille suffisante pour absorber des déphasages et glissements. Des pointeurs permettent d'indiquer le début des données effectives.

Le module de transport **STM-1** (*Synchronous Transport Module*) est standardisé à 155,520 Mbit/s. Les niveaux supérieurs sont obtenus par multiplexage par *entrelacement* de STM-1. Par exemple, le **STM-4** est constitué de 4 trames STM-1 expédiées selon la suite 0 1 2 3 0 1 2 3. Une quadritrame dure alors la même durée qu'une trame STM-1 (125 μ s).

Niveau / nom	vitesse [$\frac{Mbit}{s}$]	vitesse utile [$\frac{Mbit}{s}$]
STM-1	155.52	150.34
STM-4	622.08	601.34
STM-16	2488.32	2405.37
STM-64	9953.28	9621.50
STM-256	env. 40 $\frac{Gbit}{s}$	

Les données sont entremêlées d'informations de gestion : 2430 bytes transmis peuvent être vus comme 9 lignes de 270 colonnes dont les 9 premières colonnes (sur les 9 lignes) sont des informations de gestion. Les 261 x 9 autres bytes sont utilisables pour des données p.ex. des données structurées (p.ex. des conteneurs, eux-mêmes contenant des données de gestion et des données utiles).

L'avis **G.709** du CCITT définit par exemple une structure de sous-multiplexage qui permet de retrouver les débits PDH (**G.702**) dans les canaux SDH, ce qui permet de se passer de PDH entièrement, sauf peut-être pour la connexion abonné : les seuls niveaux de la hiérarchie PDH qui sont encore déployés en production sont le 2 x B (E0) (ISDN BRI sans canal D) et le **E1** (ISDN **PRI**, 30 x B (E0) + 1 D (E0)). Il est vraisemblable qu'à moyen terme, la voix-sur-IP remplace les niveaux E0 et E1 de PDH.

Enfin, si SDH peut transporter **ATM**, c'est également le cas de **MPLS** : Swisscom prévoit par exemple de fusionner d'ici 2008 ses réseaux de données ATM et ses réseaux SDH dans un grand réseau unique à base de technologie MPLS/ATM.

5.4 ATM : Asynchronous Transfer Mode

Pas traité
en détail
en 2010-
2011

La question sous-jacente est celle de l'intérêt du mariage d'un réseau de paquets de données asynchrones³ et d'un réseau de circuits de données synchrones. L'émergence des applications multimédias rend ce genre de réseau indispensable.

Une solution possible est **ATM** : Asynchronous Transfer Mode [8]. Ce réseau conçu dès le départ pour de grandes performances et une qualité de service négociable propose en fait une solution hybride de petits paquets (53 octets, dont 48⁴ bytes de données et 5 d'entête), nommées *cellules* appartenant à un canal virtuel lui-même contenu dans un faisceau virtuel de données.

Partant d'un flux synchrone de cellules, une technique de multiplexage permet de distinguer les cellules libres (bourrage) des différentes cellules d'informations organisées ainsi comme des flux informationnels asynchrones (le nombre de cellules intercalées n'est pas forcément constant). Si cela s'avère nécessaire, le synchronisme est rétabli dans les multiplexeurs grâce à un buffer d'égalisation capable de compenser la variation du délai des cellules (**CDV** : Cell Delay Variation). C'est cette originalité qui a valu son nom à cette technique de transmission.

5.4.1 Modèle en couche d'ATM

Le réseau ATM se situe comme ses concurrents aux couches 1 et 2 du modèle OSI :

2. Liaison	AAL (Adaptation Layer)
	ATM (MAC Layer)
1. Physique	TC (Transmission Convergence)
	PM (Physical Medium)

5.4.1.1 ATM : couche physique

La couche physique d'ATM (demi-couche inférieure PM) peut être implémentée de différentes manières :

- dans un affluent de la hiérarchie numérique plésiochrone (**PDH**) : 2.048 (1.544) et 34.368 (44.736) Mbps
- encapsulé dans un conteneur d'un niveau de la hiérarchie numérique synchrone (**SDH**) : 51.84, 155.52 et 622.08 Mbps
- dans un niveau numérique propre (ATM cell stream, sur paire torsadée **UTP**) : 25.6 et 155.52 Mbps
- via la transmission sur fibre optique **FDDI** à 100 Mbps (mode **TAXI**, voir <http://www.protocols.com/pbook/taxi.htm>) : 100 MBit/s (TAXI)

Pour les interfaces publiques PDH et SDH, on utilise les supports traditionnels à savoir respectivement la paire coaxiale et la fibre optique monomodale (SMF/15 km). Par contre, on recourt plutôt à la paire torsadée (UTP/100 m) ou la fibre optique multimodale (MMF/2 km) pour les interfaces privées ATM, FDDI et SDH.

3. ici *asynchrone* signifie que les paquets peuvent être envoyés à n'importe quel moment : ils n'occupent pas un slot fixe comme en **TDM** (Time Division Multiplexing ; ISDN PRI p.ex.).

4. le choix de 48 bytes est un compromis : les réseaux de données désiraient 64, les réseaux de voix 32 :-)

5.4.1.2 Couche liaison : sous-couche MAC

La demi-couche ATM qui contrôle l'accès au média est basée sur le format de cellule suivant (avec une variante selon le type d'interface : utilisateur ou réseau) :

UNI (User/Network Interface) :

GFC (flux 4	VPI (fais- ceau) 8	VCI (canal virtuel) 16	PT 3	CLP (prio) 1	HEC (erreurs) 8	PAYLOAD (AAL...) (info) 384 bits (48 bytes)
-------------------	-----------------------------	---------------------------------	---------	--------------------	-----------------------	---

NNI (Network/Network Interface) :

VPI (faisceau) 12	VCI (canal virtuel) 16	PT 3	CLP (prio) 1	HEC (erreurs) 8	PAYLOAD (AAL...) (info) 384 bits (48 bytes)
-------------------------	---------------------------------	---------	--------------------	-----------------------	---

Si les aspects UNI (User/Network Interface) et NNI (Network/Network Interface) semblent en bonne voie de normalisation, il faut bien reconnaître que la capacité utile (voir **rendement intrinsèque** à la section 3.4.1 en page 23) des cellules varie de 48 à 44 octets selon le niveau **AAL** utilisé (voir ci-après). Le débit utile se voit ainsi réduit de 9.5% (AAL5) à 17% (AAL3/4), ce qui est énorme par rapport à 0.6% pour FDDI. Pour les stations de travail multimédia, les facteurs d'interopérabilité et de coût de l'accès au réseau interviennent fortement, si bien qu'ATM a vite été remplacé par d'autres solutions plus économiques.

5.4.1.3 Couche liaison : sous-couche AAL

La demi-couche d'adaptation ATM est codée au début et à la fin du champ Payload, à l'instar du niveau LLC qui figure au début du champ d'information des trames 802.x ou FDDI. Formellement, cette demi-couche AAL est elle-même subdivisée en une sous-couche de Convergence (**CS**) et une sous-couche de Segmentation/Réassemblage (**SAR**). Cinq niveaux AAL ont été définis pour répondre aux besoins des différents services véhiculés par ATM :

AAL1 est dédiée aux services nécessitant un débit constant (**CBR**) comme la transmission de voix ou de vidéo :

SN (4)	SNP (4)	PDU-SAR (376 bits)
-----------	------------	-----------------------

Le champ Sequence Number (SN) est protégé par **CRC** ($x^3 + x + 1$) dans le champ SNP.

AAL2 aux services supportant un débit variable (**VBR**) avec des contraintes temporelles élevées (synchronisme) comme la transmission vidéo compressée **MPEG** :

SN (4)	IT (4?)	PDU-SAR (360 bits)	LI (6)	CRC (10)
-----------	------------	-----------------------	-----------	-------------

Le champ Information Type (IT) indique le début, la suite ou la fin d'un message, tandis que le champ Length Indicator (LI) indique combien d'octets de PDU-CS sont inclus. Le champ CRC est basé sur le polynôme générateur $x^{10} + x^9 + x^5 + x^4 + x + 1$.

AAL3/4 aux services **VBR** moins critiques comme les transmissions de données (service en mode message ou continu, exploitation garantie ou non) :

ST (2)	SN (4)	MID (10)	PDU-SAR (352 bits)	LI (6)	CRC (10)
-----------	-----------	-------------	-----------------------	-----------	-------------

Le champ Segment Type (ST) ressemble au champ IT (BOM, COM, EOM, SSM), alors que le champ Multiplex Indicator (MID) identifie toutes les PDU-SAR d'une même PDU-CS. Dans AAL3, le premier bit du MID est utilisé pour coder une priorité.

AAL5 aux services pouvant se contenter du débit disponible restant (**ABR**) comme ceux des réseaux locaux à haute vitesse ou le transport de trames IP/PPP/**MPEG** (ADSL, CATV) :

PDU-SAR (384 bits)

Notons qu'AAL5 est une encapsulation multi-cellule, dans laquelle les informations de contrôle se trouvent dans la dernière cellule.

La sous-couche **SAR** utilise dans ce cas le champ Payload Type (PT) qui figure dans l'entête de la cellule ATM (ex : 001=Fin SAR-SDU sans congestion, 011=Idem avec congestion).

Pour cette dernière catégorie, le but est tout de même de garantir un accès équitable à la bande passante disponible tout en garantissant une perte minimale de cellules. Il s'agit en outre d'éviter le risque majeur d'une congestion du réseau en instaurant une boucle de **contrôle de flux** qui doit être capable d'alerter les stations émettrices avant la catastrophe. Pour ce faire, le forum ATM a débattu à propos de deux approches radicalement opposées : l'approche basée sur le débit (supportable par le réseau) qui convient particulièrement bien aux liaisons WAN et l'approche basée sur le crédit (accordé par le réseau) qui est spécialement efficace pour les environnements LAN et permet le développement d'interfaces peu coûteuses. Une approche mixte a même été envisagée, mais il a été finalement décidé (09-95) que l'approche unique basée sur le débit (QFC : Quantum Flow Control) devait régler la question du contrôle de flux dans les environnements WAN et LAN.

5.4.2 ATM LANE : émulation LAN

Il subsiste le problème non moins épineux de **l'émulation LAN** dont le but est de rendre les couches ATM complètement transparentes pour les stations du réseau, de façon à ce que les applications habituelles fonctionnent aussi bien que dans leur environnement réseau classique 802.x ou FDDI. Il ne s'agit en fait de rien d'autre que de reconstituer par une émulation les formats de trames de ces différents LAN, d'où l'abréviation usuelle LANE.

Cependant, la tâche n'est pas vraiment facile si l'on considère que l'approche point à point des canaux ATM ne se prête pas du tout à la diffusion des broadcasts si souvent utilisés par les LAN's. La solution proposée pour les réseaux 802.3 et 802.5 se base sur trois éléments :

- LAN Emulation Client (LEC) : partie émulation installée sur chaque client
- LAN Emul. Server (LES) : machine assurant la conversion des adresses ATM en adresses MAC
- LANE Config Serv (LECS) : machine (évt. la même) permettant la configuration !
- Broadcast and Unknown Server (BUS) : machine simulant l'envoi de broadcasts.

5.4.3 Permanent Virtual Connection

ATM peut prévoir l'établissement permanent d'une connexion, par opposition à l'ouverture de celle-ci à la demande des couches supérieures. Un exemple de cela est l'ADSL. On indique par exemple 8 et 35 comme respectivement les VPI et VCI, et le réseau ATM transporte ces données en AAL/5 (MPEG).

5.4.4 Conclusion

ATM est un protocole complexe qui offre des fonctionnalités avancées de qualité de service. Il permet l'intégration informatique / télécommunications. Il forme une base générique utilisable dans d'autres infrastructures comme CATV, Internet par réseau électrique et l'ADSL.

Cependant, à la fois les réseaux locaux (LAN) switchés ont augmenté leur performance et la possibilité d'ajouter de la qualité de service à coût bien plus faible, mais aussi IP a rendu possible le transport de données multimédia grâce à la définition de protocoles de réservation de bande passante, à l'augmentation du débit des lignes (diminuant notamment les délais et les variations de délais ou de phase (**jitter**, **gigue**) mais aussi à la compression.

Beaucoup des idées d'ATM ont été reprises dans la définition de **MPLS**, un réseau générique à commutation de paquet de couche 2, utilisé dans la plupart des implémentations actuelles de réseaux **core** des opérateurs pour implémenter des circuits virtuels à qualité de service.

Chapitre 6

Transmission sans fil

Sommaire

6.1 Technologies	45
6.2 Calcul de liaison pour des faisceaux hertziens courts	46
6.2.1 Facteurs limitatifs	46
6.2.2 Niveaux et puissance	46
6.2.3 Affaiblissement	46
6.2.4 Bilan de liaison	47
6.2.5 Rapport signal sur bruit	47
6.2.6 Limitation de la puissance rayonnée	47
6.3 Affaiblissements	47
6.3.1 Affaiblissement linéique	47
6.3.2 Affaiblissement en espace libre	48
6.3.3 Autres affaiblissements	48
6.4 Ellipsoïde de Fresnel	49
6.4.1 Exemple de calcul	50

Les réseaux sans fils sont de plus en plus utilisés, à la fois comme prolongation du réseau local (LAN), pour des liaisons points à points en visibilité directe, mais aussi dans la boucle locale sans fil (**WLL**, voir section 4.3.4 en page 31).

La norme la plus présente actuellement est **802.11** (notamment dans ses normes g et n). Ses caractéristiques (bande de fréquence à usage multiples, plages de fréquences se chevauchant, sécurité totalement insuffisante en clair ou mode **WEP**) peuvent poser des problèmes sérieux dans un environnement ouvert.

Cette section traitera plus précisément du dimensionnement de **liaisons points à points** plus éloignées qu'en utilisation classique, tout en donnant quelques pistes pour les réseaux classiques (intra-muros ou intra-campus) [15].

6.1 Technologies

La transmission sans fil utilise diverses méthodes [4] :

- satellites géostationnaires (télécoms classiques) ou à orbite basse (GPS, nouveaux systèmes de télécom) ; problème principal : coût, délais (en particulier en géostationnaire), maintenance et puissance

- **faisceaux hertziens** terrestres : p.ex. Chasseral-Dôle ou entre deux relais **GSM** non reliés en fixe ; problème principal sur de longues distances : courbure terrestre
- réseaux sans fils (**WLAN**, **WLL**, etc) : infra-rouge, laser, ondes radio

Nous allons nous borner à étudier le cas de la propagation des ondes lors de faisceaux hertziens terrestres à courte distance, pour lesquels, notamment, la courbure terrestre peut être négligée.

6.2 Calcul de liaison pour des faisceaux hertziens courts

6.2.1 Facteurs limitatifs

On doit tenir compte des facteurs suivants :

- puissance d'émission et sensibilité de réception
- **rapport signal sur bruit** nécessaire en réception
- **gain** des antennes (dans un secteur)
- perte dans les câbles
- **affaiblissement** en espace libre
- réfraction dans l'atmosphère (notamment conditions météorologiques)
- diffraction sur des obstacles proches (Fresnel)
- réflexions partielles sur des obstacles

6.2.2 Niveaux et puissance

On définit souvent un niveau absolu comme rapport à une puissance standard, p.ex. 1 mW, en **dBm** :

$$L_x = 10 \log_{10} \frac{P_x}{P_{ref}} \quad (6.1)$$

On considère notamment le niveau d'émission (la puissance d'émission) et le niveau de réception (ou la sensibilité de réception). Noter que plus la sensibilité de réception est faible, meilleure l'adaptabilité à de grands affaiblissements.

6.2.3 Affaiblissement

On définit l'affaiblissement A sur une liaison en fonction du rapport logarithmique de la puissance émise sur la puissance reçue, en décibels (dB) :

$$A = 10 \log_{10} \frac{P_E}{P_R} = L_{emission} - L_{reception} \quad (6.2)$$

6.2.4 Bilan de liaison

L'utilisation des **dB** permet d'effectuer des calculs de liaisons (affaiblissement total) : l'affaiblissement total sur une liaison est la somme des affaiblissements¹ :

$$A = A_{cables} + A_{antennes} + A_{espace\text{libre}} + A_{obstacles} \quad (6.3)$$

D'où la formule de bilan de liaison, qui vérifie si la liaison est possible :

$$L_{emission} - A \geq L_{sensibilite\text{reception}} \quad (6.4)$$

6.2.5 Rapport signal sur bruit

Dans le cas d'un réseau 802.11, le **bruit** est dû aux autres équipements partageant les fréquences (autres réseaux sans fil, autres technologies de la bande des 2.4 GHz). Rappelons que la plage de fréquence des 2.4GHz n'est pas soumise à autorisation² et que de plus, les plages 802.11 se recouvrent (sauf les 1, 6 et 11).

Le **rapport signal sur bruit**, parfois noté SNR, est utilisé pour évaluer la qualité du signal (voir section 1.4.4 en page 7).

6.2.6 Limitation de la puissance rayonnée

On pourrait penser qu'il suffit d'augmenter la puissance d'émission – ou le **gain** des antennes – pour compenser tout affaiblissement : des contraintes légales, en particulier en 802.11 (qui est utilisé dans des bandes de fréquences partagées) interdisent une émission à plus de 100 mW (**EIRP** 20 dBm). Il faut d'ailleurs considérer non pas la densité moyenne de puissance à 360 degrés, mais bien la puissance équivalente d'une **antenne isotropique**³ (émission uniforme dans toutes les directions, sans gain dans un secteur).

En habitation, la limite de rayonnement **ORNI**⁴ s'applique. Le champ électrique ne doit pas dépasser 6 Volts par mètre.

6.3 Affaiblissements

6.3.1 Affaiblissement linéique

Un câble d'antenne provoque un affaiblissement linéique (proportionnel à la distance et dépendant de la fréquence). On peut le définir en dB comme suit :

-
1. $A_{antennes}$ vaut en général $-G_{antennes}$, car il s'agit souvent d'un gain, plus qu'un affaiblissement.
 2. C'est pour éviter les interférences que la puissance **EIRP** est limitée à 100 mW.
 3. L'unité **dB_i**, pour dB isotropique, permet de faire la différence entre une puissance dans un secteur ou une puissance isotropique.
 4. Ordonnance sur la protection contre le rayonnement non ionisant.

$$A_{lin} = \alpha d \quad (6.5)$$

où α dépend du type de câble d'antenne coaxial : de $1 \frac{dB}{m}$ à $0.22 \frac{dB}{m}$ suivant l'adéquation et la qualité du câble.

Par exemple, un câble d'antenne de 3m de mauvaise qualité provoquera un affaiblissement de 3 dB (soit 50% du signal).

6.3.2 Affaiblissement en espace libre

L'affaiblissement en espace libre (il s'agit donc d'une borne minimale) selon FRIIS [21] est :

$$A_{espace\ libre} = 20 \log_{10} \frac{4\pi d}{\lambda} = 20 \log_{10} \frac{4\pi d f}{c} \quad (6.6)$$

où :

c : vitesse de la lumière

d : distance en mètres

f : fréquence en Hz (ou λ : longueur d'onde)

(le 20 = $2 * 10$ découle de l'atténuation proportionnelle au carré de la distance)

6.3.2.1 Application

On peut remarquer que l'affaiblissement peut également s'écrire, par approximation et utilisation d'unités différentes :

$$A_{espace\ libre} = 32.5 + 20 \log_{10} f_{MHz} + 20 \log_{10} d_{km} \quad (6.7)$$

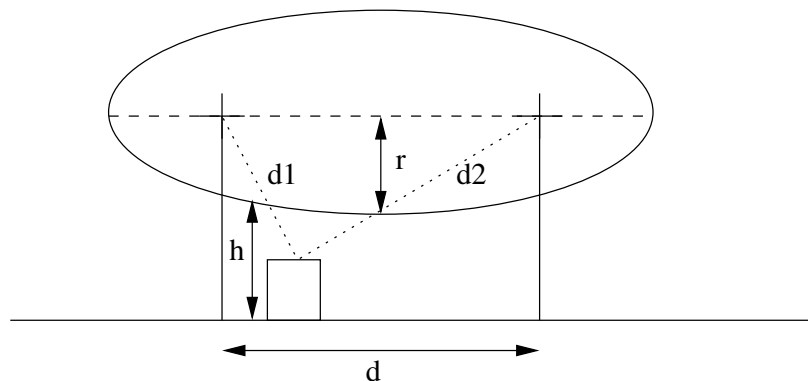
On constate donc comme attendu que chaque doublement de la distance d_{km} augmente l'affaiblissement d'environ 6 dB (division par 4 de la puissance du signal).

6.3.3 Autres affaiblissements

Quelques exemples (en négligeant l'impact de la fréquence et en supposant que l'obstacle est assez éloigné) [16].

milieu	affaiblissement par m
forêt	0.4 dB
Mur en plâtre	3 dB
Mur en verre, armature métallique	6 dB
Béton de scories (non armé)	4 dB
Fenêtre	3 dB
Porte en métal	6 dB
Porte en métal, mur en brique	12.4 dB

6.4 Ellipsoïde de Fresnel



L'ellipsoïde⁵ de Fresnel est défini par les deux antennes (comme foyers). Il permet d'évaluer les perturbations par diffraction. Les diffractions créent des émissions secondaires qui parviennent aussi au destinataire. Si les deux trajets sont en opposition de phase, le signal va être amoindri. Le rayon (demi-axe vertical) du premier ellipsoïde de Fresnel est donné par

$$r = \frac{1}{2} \sqrt{d\lambda} = \frac{1}{2} \sqrt{d \frac{c}{f}} \quad (6.8)$$

où :

d : distance en mètres entre les deux foyers (antennes)

f : fréquence en Hz (ou λ : longueur d'onde)

Cette première formule nous permet de déterminer la hauteur de l'ellipsoïde de Fresnel. En la reportant sur un schéma avec les antennes, on peut déterminer si un obstacle est dans le rectangle entourant l'ellipsoïde, ce qui est déjà une bonne approximation. On considère que 60% de l'ellipsoïde doit être libre pour une bonne transmission. Il faut éviter de toucher des plans d'eau.

En considérant que f est en GHz, et d en km on peut obtenir la formule simplifiée et arrondie suivante (en mètres) :

$$r = 17.32 \sqrt{\frac{d_{km}}{4f_{GHz}}} \quad (6.9)$$

5. penser à la 3^e dimension : ellipse de révolution, patate.

Pour de plus longues liaisons (distance bien plus grande que la hauteur de l'obstacle), la hauteur de l'obstacle par rapport à celle des antennes est le facteur déterminant. On peut calculer la hauteur au sol maximum en mètre d'un obstacle situé à un point donné entre les deux antennes de manière à ce que les ondes soient transmises (diffraction au sommet de l'obstacle). Ici, la hauteur des antennes est explicitement intégrée dans le calcul des deux distances.

$$h = 17.32 \sqrt{\frac{d_1 d_2}{f(d_1 + d_2)}} \quad (6.10)$$

où :

d_1 : distance en kilomètres du sommet de l'obstacle à l'antenne 1

d_2 : distance en kilomètres du sommet de l'obstacle à l'antenne 2

f : fréquence en GHz

6.4.1 Exemple de calcul

Pour une distance entre deux antennes de 6 km à 2.412 GHz (canal 1 802.11), on obtient $r = 14.66$ m. Un immeuble de 14m est situé à 2 km d'une des antennes⁶. On obtient $h = 12.88$, l'ellipsoïde de Fresnel n'est pas assez dégagé !

NB : pour de grandes distances, la courbure terrestre doit être prise en compte : $y = \frac{d^2}{8R^c}$, avec $R^c = 8500$ km⁷. Dans l'exemple ci-dessus, il faudrait relever les antennes d'un demi-mètre.

6. on calculera ici $d_2 = d - d_1$ en négligeant h , comme la distance est grande

7. rayon terrestre compensé : la formule exacte valable jusqu'à la demi-circonférence serait $y = R_t(1 - \cos \frac{d\pi}{C_{terre}})$, avec $R_t = 6378$ km, d la distance en km et $C_{terre} = 40075$ km ; or la décroissance de l'indice de réfraction avec l'altitude a pour conséquence de courber le faisceau hertzien en direction du sol : en utilisant un rayon terrestre compensé de 8500 km on compense cet effet (aux latitudes moyennes).

Chapitre 7

Sécurité des réseaux

Sommaire

7.1 La problématique	51
7.1.1 Les menaces	51
7.1.2 Les moyens de mitigation des menaces	52
7.2 Sécurisation d'un réseau d'entreprise	52
7.2.1 Moyens permettant d'améliorer la sécurité du périmètre	52
7.2.2 Surveillance	55
7.2.3 Réseaux privés virtuels (VPN)	55

Le but de ce chapitre est de donner quelques informations de base sur la sécurisation d'un réseau, en particulier de la sécurité du périmètre d'un réseau d'entreprise, en complétant et appliquant les notions du cours sur les **VLAN**, les firewalls et proxies et les VPNs.

Il ne remplace pas un cours général sur la sécurité informatique.

7.1 La problématique

La sécurité des réseaux est un sujet vaste qui est malheureusement souvent négligé. On conçoit et met en place encore aujourd'hui des réseaux et protocoles qui transmettent toutes les informations en clair et dont les bornes ne sont pas clairement définies. Citons pour exemple la prolifération des réseaux sans fils (WiFi/WLAN/802.11) non ou mal (WEP) sécurisés.

7.1.1 Les menaces

On peut identifier rapidement les menaces comme suit :

- sécurité des personnes, biens et locaux (surveillance, veille)
- sécurité du système informatique interne
 - sécurité des informations
 - disponibilité
 - confidentialité
 - intégrité
 - sécurité externe
- sécurité dans la communication (**B2B**, **B2C**)

Quelques exemples de menaces :

Menace	impact
Serveurs en panne franche	disponibilité
Corruption silencieuse de données	intégrité
Erreur de manipulation	intégrité, disponibilité, confidentialité
Feu	disponibilité
Code malicieux (trojan)	disponibilité, confidentialité, intégrité
Sabotage	disponibilité, intégrité
Surveillance passive (keylogger)	confidentialité
Logiciel mal conçu	disponibilité, confidentialité, intégrité
Format de données ou de stockage obsolète	disponibilité

7.1.2 Les moyens de mitigation des menaces

Divers outils – tant techniques que non techniques – sont à disposition pour évaluer, préparer, détecter et réagir aux menaces. Cela n'est pas l'objet de ce chapitre. Vous trouverez plus d'information notamment dans [25].

7.2 Sécurisation d'un réseau d'entreprise

Il faut tout d'abord déterminer ce que l'on veut réellement sécuriser. L'approche souvent prise est que l'on protège le réseau interne des attaques provenant de l'extérieur (coté Internet du routeur/firewall), les attaques provenant forcément de personnes mal intentionnées ayant pour but de détruire l'entreprise.

Cette approche est trop caricaturale : elle est nécessaire mais non suffisante.

En effet, les problèmes de sécurité proviennent en règle générale de l'intérieur du réseau (80% selon diverses études parfois contestées, voir [28]). De plus, d'autres problèmes touchant à la sécurité (intégrité des données stockées, confidentialité) que les attaques directes sont à considérer. Les attaques indirectes globales sont de plus en plus fréquentes. Enfin, des attaques non directement liées à l'informatique ne sont pas à négliger dans une politique globale de sécurité.

Un firewall est donc nécessaire, mais ne remplace pas une politique de sécurité générale : formation du personnel, mise à jour des logiciels sur les machines, limitation des services au strict minimum, droits d'accès, audit et journaux (logs), sauvegardes, pour n'en citer que quelques aspects.

Un contre-exemple est facile à trouver sans faire appel à la volonté de nuire : lorsque des employés travaillent chez eux avec leur ordinateur portable puis viennent se connecter en entreprise, derrière le firewall, avec tous leurs spywares (espions), virii et autres problèmes. Dans ce cas, deux solutions sont applicables : soit éviter que cela arrive en intégrant ces ordinateurs dans une centralisation de la sécurité (accès par VPN et proxy/firewall via l'entreprise même en déplacement, voir section 7.2.1.5), ou installer un système de détection d'intrusion et de réaction aux intrusions dans l'entreprise via un réseau de quarantaine (voir section 7.2.3.1) permettant de confiner les problèmes très rapidement.

7.2.1 Moyens permettant d'améliorer la sécurité du périmètre

Quelques principes de base permettent d'améliorer la sécurité d'un réseau de manière très importante. Ces principes sont exposés dans les sections suivantes, et font appel à des dispositifs

logiciels d'isolation, parfois embarqués, comme le **pare-feu** (de l'anglais **firewall**), le **proxy** (ou relais applicatif), ou organisationnels (limitation des applications, structuration du réseau).

7.2.1.1 Dispositifs logiciels ou embarqués

Les deux dispositifs logiciels (sur les postes de travail ou serveurs) ou embarqués (sur des équipements réseau dédiés, comme des routeurs) permettant une isolation à divers degrés sont :

le firewall dispositif ayant pour but de filtrer le trafic, en bloquant les datagrammes interdits par des règles administratives (adresses, ports, ...), qu'elles soient statiques (préconfigurées : firewall sans état) ou dynamiques (dépendantes du contexte, du passé) ; généralement actif en couche 3, voire supérieures : firewall avec état. De la réécriture d'adresse (NAT/PAT) est souvent associée pour des raisons administratives.

le proxy logiciel étant au centre de deux flux de données, un provenant du client, et un destiné au serveur. Dans le cas d'un proxy-application Web, deux connexions TCP sont établies. Il n'y a plus d'échange direct en dehors de la couche 7 (voire 6 en cas de chiffrement) entre les deux partenaires. Le filtrage peut plus facilement se faire sur la base du contenu (anti-virus ou contrôle parental). Un cache peut être associé pour la performance. Un contrôle d'accès est envisageable (utilisateur et mot de passe ou identification distribuée).

7.2.1.2 Limitation des applications

Suivant les applications à supporter, la configuration et le type de firewall ou de proxy seront à adapter. Le problème est surtout dans le support de protocoles complexes comme FTP, H.323, ICQ ou SIP, protocoles qui ont besoin de connexion inverses (entrant dans le réseau interne !) en fonction d'informations indiquées par le client.

Le firewall devient alors plus complexe : il doit inspecter les protocoles de couches 7 pour déterminer les ports à ouvrir dynamiquement.

Une faille de sécurité qui permet alors de contrôler ce que le client du réseau interne demande comme ouverture via l'abus d'un programme tournant sur le client peut alors avoir des conséquences catastrophiques pour la sécurité [29].

En particulier si le réseau interne est en NAT ou PAT, le firewall doit même *modifier* certaines données de protocole couche 7 en plus des données couche 4. En effet, les données couches 7 peuvent contenir des informations comme : *j'attends une connexion sur le port 6512 de l'adresse IP 192.168.1.42* et doivent être corrigées (NAT/PAT), en plus d'être prises en compte pour l'ouverture de ports supplémentaires.

En conséquence, moins d'applications seront à supporter, plus la sécurité sera augmentée. Idéalement, il ne restera plus qu'un protocole traversant le firewall et/ou le proxy : le protocole HTTP du WWW (un problème de sécurité à lui tout seul, que seul le filtrage de contenu au niveau d'un proxy ou d'un firewall très évolué pourrait être limité).

7.2.1.3 Séparation des fonctionnalités

La séparation de fonctionnalités peut être une arme intéressante pour décourager ou au moins retarder des attaquants. Plusieurs firewalls en séquence, effectuant des tâches différentes (p.ex. accès au DMZ ou au réseau interne), pourraient le permettre, dans la mesure où la gestion administrative supplémentaire n'est pas, en elle-même un frein à la sécurité : il vaut mieux un seul point de sécurité bien géré que plusieurs mal surveillés.

On peut aussi, en cas de charge importante, vouloir séparer les fonctions d'analyse des diagrammes et de modification sur deux routeurs/firewall/**NAT-PAT** différents. Cependant, dans la mesure où un chiffrement/**VPN IPsec** est utilisé, par exemple, il faut que toutes les opérations (en particulier un éventuel NAT/PAT) soit faites sur la même machine.

7.2.1.4 Couper la connexion jusqu'à la couche 7 : le proxy

Avec un firewall, il y a toujours échange direct, jusqu'à la couche 3 (réseau) directement entre les machines du réseau interne et celles du réseau derrière le firewall. Cela signifie que certains problèmes liés à la pile TCP/IP de la machine à protéger pourraient être utilisés par un attaquant éventuel.

La plupart des firewalls incorporent des méthode statique et dynamique qui permettent d'éviter certaines de ces attaques (IDS/IRS). De nouvelles sont découvertes chaque jour et la mise à jour du firewall est donc critique. Par exemple, certains drapeaux de l'entête peuvent être supprimés, ce qui peut créer de nouveaux problèmes (p.ex. mauvais support des nouvelles options TCP comme l'ECN¹).

Pour séparer plus complètement le réseau externe du réseau interne, on peut abandonner l'idée d'un routage par le firewall et n'autoriser des connexions vers l'extérieur qu'au travers d'un proxy-application (couche 7). Toutes les requêtes sont alors envoyées au proxy-application qui effectue la requête pour le client. La connexion directe en couche 3 entre le client et le serveur n'est plus nécessaire. Seules des attaques liées au protocole couche 7 sont encore possibles.

On peut configurer le firewall de manière à rediriger toutes les requêtes vers l'extérieur sur le proxy, de manière transparente (proxy transparent). Cela évite de procéder à de fastidieux changements de configuration sur les postes clients.

De plus, le proxy peut disposer d'un cache de manière à accélérer les requêtes ainsi que d'un filtre intelligent (sur le contenu, en plus des adresses ; p.ex. un anti-virus).

7.2.1.4.1 Inconvénients Un proxy ne supporte pas forcément tous les protocoles : en particulier les protocoles complexes comme FTP, ICQ, H.323, SIP ou d'autres ne sont pas forcément supportés. De nouveaux protocoles ou des applications spéciales non orientées WWW peuvent également poser problème.

Un proxy-application typique HTTP ou TCP (SOCKS) ne supportera en aucun cas le trafic UDP.

7.2.1.5 Réseau d'entreprise typique

En règle générale, on distinguera les sous-réseaux suivants :

nom	description
réseau interne	machines clientes, éventuellement serveurs de fichiers ou d'impression
réseau externe	réseau de connexion à Internet
réseau DMZ	réseau des serveurs accédés à la fois de l'intérieur et de l'extérieur de l'entreprise : serveur de courrier électronique, serveur de VPN, serveur WWW extérieur (y compris éventuel <i>extranet</i>).

1. Enhanced Congestion Notification

(on parle souvent d'intranet pour le réseau interne d'entreprise et d'extranet pour les services fournis à des tiers, p.ex. B2B ou B2C)

Le principe d'une zone démilitarisée (**DMZ**) est de contenir dans un réseau séparé, assuré avec des règles de firewall strictes, les risques extérieurs perçus.

Suivant la taille de l'entreprise et la présence de filiales, le réseau interne peut très bien former un grand internet sous la forme de liaisons VPN, connecté au réseau Internet public global par des firewalls dans chaque succursale, ou uniquement au siège. Le choix de la topologie dépendra de nombreux facteurs comme le nombre de points d'entrées, la présence de clients mobiles, la centralisation de la sécurité, la qualité des liaisons VPN vers le siège, la redondance, etc.

7.2.2 Surveillance

7.2.2.1 Surveillance des logs et alarmes

Chaque machine connectée à un réseau – et en particulier les routeurs et firewalls – peuvent participer à un effort d'audit / journal qui peut permettre, par analyse préventive ou après un problème, de prévenir ou de réagir de manière appropriée à une attaque. Certains systèmes peuvent prévenir automatiquement pour certains types d'attaques, voire même filtrer complètement les adresses fautives. Ces outils sont cependant à manier avec précaution de manière à ne pas causer d'attaque **DoS** (Denial of Service), en particulier en saturant les responsables de fausses alarmes.

7.2.2.2 Détection d'intrusion active

L'idée de la détection d'intrusion active est d'analyser les logs ainsi que tout le trafic à la recherche de signatures connues d'attaques. Cela s'apparente à la détection de virus. Seules les stratégies d'attaques déjà connues sont détectées. Des faux positifs sont possibles. Citons comme outil p.ex. `snort`.

Des améliorations dynamiques sont possibles : la mise à jour automatique des listes de signatures ainsi qu'éventuellement l'exploitation des données normales du réseau (analyse comportementale passée et actuelle du réseau) et la variance sur l'activité observée [26].

7.2.2.3 Détection d'intrusion passive (Honey Pot)

L'idée ici est de mettre en place une machine volontairement vulnérable à des attaques et qui permet de s'en rendre compte lorsqu'elle est attaquée ou piratée. Cette machine est disposée dans une partie normalement inaccessible ou protégée du réseau. Si elle se fait attaquer c'est que les dispositifs ont été violés (Honey Pot, voir [27]).

Les machines virtuelles sont une des manières d'implémenter les Honey Pots, surveillés dans ce cas p.ex. de la machine hôte : une autre méthode est d'utiliser des logiciels conçus dans le but de simuler des services.

7.2.3 Réseaux privés virtuels (VPN)

D'une manière peut-être arbitraire, nous relierons ici quelques méthodes permettant de constituer un réseau séparé d'un autre (avec ou sans chiffrement et à diverses couches).

D'autres méthodes plus anciennes pour créer des réseaux privés étaient :

- les lignes louées dédiées
- les groupes fermés d'usagers (CUG, *closed user groups*), notamment en X.25 et ISDN
- les circuits virtuels (Frame Relay, ATM)

Aujourd'hui, les méthodes suivantes sont utilisées :

- **VLAN** Ethernet
- tunnels IP
- les réseaux d'accès et d'entreprise globaux, gérés par des opérateurs (technologies : accès modem commuté, VPN ou **MPLS**)

7.2.3.1 VLAN Ethernet

Les VLANs Ethernet couche 2 permettent d'isoler le trafic d'un sous-réseau tout en utilisant le même équipement. Cela peut donner, sous certaines conditions, d'excellents résultats : p.ex. en isolant un réseau de téléphonie sur IP du réseau général (pour des questions de sécurité et/ou de qualité de service).

Un exemple d'implémentation combinée à de la détection d'intrusion (voir section 7.2.2.2) est celui du *réseau de quarantaine* de l'EPFL[30] : l'idée est de transférer les machines infectées sur un VLAN spécifique n'ayant pas accès direct aux autres machines du réseau et accès Internet limité à certains services, uniquement via proxy (ce qui permet de réparer la machine sans en mettre en danger d'autres – et d'inciter l'utilisateur à le faire). Cette solution forme un véritable système IRS (*Intrusion Response System*).

7.2.3.2 Tunnels IP

Le principe des tunnels est d'utiliser un canal non sûr (p.ex. Internet) et d'y envoyer des paquets d'un réseau spécifique (couche 2 ou 3, voire 7) de manière à créer un réseau privé virtuel. A l'aide de technologies de chiffrement, on peut assurer la confidentialité, l'intégrité et l'authenticité des données échangées.

Quelques exemples :

TLS/SSL Transport Layer Security / Secure Socket Layer est une couche qui est ajoutée au-dessus de la couche 4² et qui permet d'assurer une certaine sécurité dans les échanges de données HTTP par chiffrement et certificats.

L2TP Cette classe de protocoles permet d'encapsuler des PDU de couche 2 dans le tunnel, de manière chiffrée ou non. C'est utilisé notamment dans l'ADSL pour séparer l'équipement terminal de l'équipement de routage, ou dans certains VPN comme p.ex. OpenVPN en mode couche 2, Cisco VPN, Microsoft PPTP ou d'autres encore.

IPsec Standard intéopérable de chiffrement et/ou authentification couche 3 pour IPv4/IPv6.

2. formellement en couche 6

Références et bibliographie

- [1] Claude SERVIN, *Réseaux et télécoms*, 2e édition, Dunod, ISBN 2-10-049148-2
- [2] *Technologies d'accès aux réseaux : xDSL, CATV, PLC, WiMAX, UMTS, satellites, etc*, 3ème édition ; HES-SO Fribourg ; ISBN 2-940156-19-0
- [3] Fred HALSALL, *Data Communications, Computer Networks and Open Systems* (Fourth Edition), Addison-Wesley 1996, ISBN 0-201-42293-x.
- [4] Pierre-Gérard FONTOLLIET, *Traité d'Electricité XVIII : Systèmes de Télécommunications*.
- [5] Andrew S. TANENBAUM, *RESEAUX : Architectures, protocoles, applications*, InterEditions 1990, ISBN 2-7296-0301-8, (Traduction de *Computer Networks*, Prentice-Hall 1989).
- [6] Antoine DELLEY, *Téléinformatique*, Dunod Informatique 1987, ISBN 2-04-016907-5.
- [7] *RNIS*, Ecole d'ingénieurs de Fribourg, 1995
- [8] *ATM : Télécommunications à large bande*, Ecole d'ingénieurs de Fribourg, 1996
- [9] *Information Technology Networks*, ISBN 2-940156-26-3, 1ère édition, HES-SO Fribourg
- [10] Claude E. SHANNON, *A Mathematical Theory of Communication*, Bell System Technical Journal, vol. 27, pp. 379-423 and 623-656, July and October, 1948 (ISBN 0252725484)
- [11] R. W. HAMMING, *Error-detecting and error-correcting codes*, Bell System Technical Journal, vol 27, pp. 147-160, 1950
- [12] Ulysse BLACK, *MPLS and Label Switching Networks*, Prentice Hall, ISBN 0-13-015823-2
- [13] Thomas D. NADEAU, *MPLS Network Management : MIBs, tools and techniques*, Cisco Systems, ISBN 1-55860-751-X
- [14] Ivan PEPELNJAK & Jim GUICHARD, *MPLS and VPN architectures*, Cisco Press, ISBN 1-58705-002-1
- [15] Matthew S. GAST, *802.11 Wireless Networks : the definitive guide*, 2nd Edition, O'Reilly, ISBN-0-596-10052-3
- [16] P. ALI-RANTALA & L. UKKONEN, *Different kinds of walls and their effect on the attenuation of radiowaves indoors*, Antennas and Propagation Society International Symposium, IEEE, Volume 3, 22-27 June 2003 p. 1020-1023
- [17] W. Richard STEVENS, *TCP/IP Illustrated volume 1 : The Protocols*, Addison-Wesley, ISBN 0-201-63346-9
- [18] Christian HUITEMA, *Le routage dans l'Internet*, Eyrolles, 10/1994, 418p, ISBN 2-212-08902-3 (10/1994)
- [19] Alain PELAT, *Signaux numériques, protection contre les erreurs*, Ellipses, 2005, ISBN 2-7298-2367-0
- [20] Shon HARRIS, *All-in-one CISSP exam guide*, 4th Edition, Mc Graw-Hill, ISBN 978-0-07-149786-2
- [21] Harald T. FRIIS, IRE Proceedings, vol. 34, p. 254, 1946.

- [22] Elizabeth D. ZWICKY & Simon COOPER & D. Brent CHAPMAN, *Building Internet Firewalls*, 2nd Edition, O'Reilly, ISBN 1-56592-871-7
- [23] Simson GARFINKEL & Gene SPAFFORD, *Practical UNIX & Internet security*, 2nd Edition, O'Reilly, ISBN 1-56592-148-8
- [24] Marc SCHAEFER, *Wiki du cours*, <http://he-arc.alphanet.ch/>³, 2009.
- [25] Cours postgrade ES sécurité 2003-2004,
<http://cvs.alphanet.ch/cgi-bin/cvsweb/schaefer/public/cours/ESNIG/securite/cours/interne/>
- [26] Les systèmes de détection d'intrusion classiques et comportementaux, voir
http://www.securiteinfo.com/conseils/choix_ids.shtml
- [27] Les HoneyPots (voir aussi Wikipedia.org), <http://www.honeypots.net/>
- [28] Voir <http://www.informit.com/articles/article.asp?p=350391&seqNum=5>
- [29] Injection de commande sur le canal FTP, pouvant amener à des ouvertures de ports non souhaitées (Microsoft Internet Explorer), voir <http://secunia.com/advisories/13404/>
- [30] Réseau de quarantaine (snort IDS, VLAN, proxy squid) <http://dit.epfl.ch/page59201.html>

3. anciennement : <http://www.alphanet.ch/cgi-bin/he-arc/wiki.pl>

Index des concepts

- 802.11, 45
- 802.2, 36
- AAL, 41
- AAL5, 30
- ABR, 42
- AD, 3, 4, 33
- ADPCM, 4
- ADSL, 29
 - BRAS, 29
 - DSLAM, 29
 - filtre, 29
- advertised window, 22
- affaiblissement, 46
- alphabet, 2, 3
 - fini, 3
- alternat, 20
- analogique, 3
- antenne, 47
 - isotropique, 47
- asymétrie, 28
- ATM, 29, 30, 33, 39–42
 - AAL, 41
 - AAL5, 30
 - CS, 41
 - LAN-E, 42
 - BUS, 42
 - LEC, 42
 - LECS, 42
 - LES, 42
 - NNI, 41
 - PVC, 42
 - SAR, 41, 42
 - SEAL, 30
 - UNI, 41
- AUTODIN-II, 15
- avec perte, 2
- B2B, 51
- B2C, 51
- bande passante, 6
- Baud, 6
- bearer, 37
- binaire, 6
- bit stuffing, 23, 36
- bloc, 12
- BRAS, 29
- BRI, 29, 34, 38
- brouillage, 31
- bruit, 47
- bundling, 37
- BUS, 42
- Business ISDN, 34
- calcul par tranches, 16
- canal B, 37
- canal D, 34
- CATV, 30
- CBR, 41
- CD audio, 4
- CDV, 40
- champs de Galois, 14
- codage, 1, 2, 4, 5
 - de source, 2
 - de voie, 2
- code correcteur, 11
- code de, 16
- codec, 4, 33, 34
 - G.711, 4
 - G.722, 34
- codes, 12
 - bloc, 12
- codes bloc, 17
- codes convolutifs, 17
- compression, 2, 5, 8, 9
 - avec perte, 2
 - différentielle, 8
 - Dynamic Huffman, 8
 - entropique, 5, 8
 - LZW, 9
 - MNP-5, 9
 - RLE, 8
 - sans perte, 2
 - V.42bis, 9
- congestion, 19
- conteneurs, 39
- continue, 3

- Continuous RQ, 21, 24
- contrôle de flux, 19, 22, 42
- conversion, 3, 4, 33
 - AD, 3, 33
 - codage, 4
 - échantillonnage, 4
 - quantification, 4
- core, 28, 43
- correction, 11, 13, 16, 17
- couche, 2, 11, 12, 19, 30, 40
 - liaison, 11, 19, 30, 40
 - physique, 2, 11, 12, 19, 30, 40
 - réseau, 19
 - transport, 19
- couche d'adaptation, 37
- CRC, 12, 14–16, 19, 22, 41
 - AUTODIN-II, 15
 - calcul par tranches, 16
 - CRC-32, 15
 - degré, 15
 - polynôme générateur, 15
- CRC-32, 15
- CS, 41
- CUG, 28
- Cyclic Redundancy Check, voir CRC
- débit, 6
 - binaire, 6
 - de décision, 6
 - de moment, 6
- décibel, 7
- dégroupage, 28
- détection, 12–14
- dB, 47
- dBi, 47
- dBm, 46
- DDI, 34, 35
- de décision, 6
- de Markov, 3
- de moment, 6
- de source, 2
- de transmission, 11
- de voie, 2
- degré, 15
- dernier kilomètre, 27
- différentielle, 8
- discrète, 3
- distance de, 12
- DMZ, 55
- DoS, 55
- DSLAM, 29
- Dynamic Huffman, 8
- E1, 38, 39
- ECC, 12, 14
- échantillonnage, 4
- EIRP, 47
- en rafales, 14
- entropie, 5
- entropique, 5, 8
- erreurs, 11–14, 16, 17, 19
 - correction, 11, 13, 16, 17
 - détection, 12–14
 - de transmission, 11
 - en rafales, 14
- espérance, 5
- explicit request, 20
- faisceaux hertziens, 29, 46
- fanions, 23
- FCS, voir CRC
- FDDI, 40
- FEC, 11
- fenêtre, 21
- fibres optiques, 28
- filtre, 4, 6, 29
 - passes-bande, 6
 - passes-bas, 4
- fini, 3
- firewall, 51, 53
- Fresnel, 49
- FTTB, 28
- FTTH, 28
- G.702, 38, 39
- G.703, 39
- G.704, 39
- G.709, 39
- G.711, 4, 33
- G.722, 34
- gain, 46, 47
- GBit Ethernet, 28
- gigue, 43
- go-back-n, 21
- Golay, 17
- GPRS, 31
- GSM, 31, 46
- H.323, 36
- hâchage, 14
- half-duplex, 20
- Hamming, 12, 16
 - code de, 16

- distance de, 12
- optimal, 16
- poids de, 12
- haute fiabilité, 29
- HDLC, 22, 23, 36, 37
 - I, 23
 - S, 23
 - U, 23
 - UI, 23
- HDSL, 28, 30, 39
- hiérarchie, 37–39
 - numérique, 37
 - plésiochrone, 38
 - synchrone, 39
- Honey Pot, 55
- HSCSD, 31
- I, 23, 36
- IDLE RQ, 20
- IDS, 55
- implicit retransmission, 20
- information, 1
- interface S, 35
- interface U, 34
- intrinsèque, 23
- IPsec, 54, 56
- IRS, 56
- ISDN, 4, 29, 34–38
 - bearer, 37
 - BRI, 29, 34, 38
 - canal B, 37
 - canal D, 34
 - DDI, 35
 - interface S, 35
 - interface U, 34
 - MSN, 34, 36
 - PRI, 34, 38
 - SETUP, 37
 - type de service, 36
- isotropique, 47
- jitter, 43
- L2TP, 56
- LAN-E, 42
- LAP, 36
- LAPB, 22, 36
- LAPD, 22, 36
- LAPM, 22
- large bande, 34
- lasers, 29
- LEC, 42
- LECS, 42
- LES, 42
- liaison, 11, 19, 30, 40
- liaisons points à points, 45
- LLC, 36
- logarithmique, 4
- longueur moyenne des symboles, 5
- LZW, 9
- mesh, 28
- minuterie, 19
- MNP-5, 9
- mots-codes, 13
- MP-PPP, 37
- MPEG, 41, 42
- MPLS, 28, 39, 43, 56
- MSN, 34, 36
- MultiLine ISDN, 34
- Multiple Subscriber Number, voir MSN
- NACK, 20, 21
- NAT, 54
- NNI, 41
- numérique, 37–39
- numéro de séquence, 19, 21
- Nyquist, 4
- OOK, 6
- optimal, 16
- ORNI, 47
- over-booking, 29, 30
- paire torsadée, 29
- paquet d'erreurs, voir erreurs en rafales
- pare-feu, 53
- parité, 12, 14
- passer-bande, 6
- passer-bas, 4
- PAT, 54
- payload, 20, 23
- PCM/A, 33
- PDH, 33, 38–40
- physique, 2, 11, 12, 19, 30, 40
- piggy-backing, 21, 23
- plésiochrone, 38, 39
- poids de, 12
- pointeurs, 39
- polling, 23
- polynôme générateur, 15
- power-line, 31
- PPP, 29, 37
- PPP-over-Ethernet, voir PPPoE

- PPPoE, 29
- PRI, 34, 38, 39
- probabilité d'apparition, 4
- protocoles, 19–22, 24
 - sûrs
 - Continuous RQ, 21, 24
 - explicit request, 20
 - fenêtre, 21
 - go-back-n, 21
 - IDLE RQ, 20
 - implicit retransmission, 20
 - numéro de séquence, 21
 - rendement, 24
 - secondaire, 22
 - selective repeat, 21
- proxy, 51, 53
- PVC, 42

- Q.931, 36
- QoS, 19, 28, 41–43
 - ABR, 42
 - CBR, 41
 - SLA, 28
 - VBR, 41
- qualité de service, voir QoS
- quantification, 4
- quantité d'information, 4
- quantité de décision, 5

- réseau, 19
- RADIUS, 29
- rafale d'erreurs, 11
- rapport signal sur bruit, 7, 46, 47
- redondance, 1, 2, 5, 11
- Reed-Solomon, 17
- rendement, 23, 24
 - intrinsèque, 23
- rendement intrinsèque, 41
- retransmission, 11, 20
- RLE, 8
- RNIS, 34
- Road Warrior, 29
- Run Length Encoding, voir RLE

- S, 23, 36
- sûrs, 20–22, 24
- sans mémoire, 3, 5
- sans perte, 2
- SAR, 41, 42
- scrutation, 23
- SDH, 33, 38–40
 - STM-1, 39
 - STM-4, 39
- SDLC, 36
- SDSL, 28
- SEAL, 30
- secondaire, 22
- selective repeat, 21
- SETUP, 37
- SLA, 28
- SNR, voir rapport signal sur bruit
- SONET, 39
- source, 3, 5
 - analogique, 3
 - continue, 3
 - de Markov, 3
 - discrète, 3
 - longueur moyenne des symboles, 5
 - sans mémoire, 3, 5
- SSL, 56
- STM-1, 39
- STM-4, 39
- surdébit, 38
- Swisscom, 34
 - Business ISDN, 34
 - MultiLine ISDN, 34
- symétriques, 28
- symbole, 4
- synchrone, 39

- T1, 38
- télévision analogique, 30
- taux d'erreur, 2
- TAXI, 40
- TCP, 22
 - advertised window, 22
- TDM, 40
- TFTP, 21
- TLS, 56
- Trames, 36
 - HDLC
 - I, 36
 - S, 36
 - U, 36
 - UI, 36
- transport, 19
- treillis, 17
- triple play, 27
- tunnel, 56
- type de service, 36

- U, 23, 36
- UI, 23, 36
- UMTS, 31

UNI, 41
USB, 29
UTP, 40
UUCP, 21

V.42bis, 9
VBR, 41
VLAN, 28, 51, 56
voix-sur-IP, 4, 28, 29
VPN, 28, 51, 54

WAN, 27
WEP, 45
WiMAX, 31
WLAN, 46
WLL, 31, 45, 46

X-Modem, 21
X.75, 37
xDSL, 29
XOR, 12

Z-Modem, 21

Table des matières

Sommaire	iii
1 Théorie de l'information	1
1.1 L'information	1
1.2 Le codage	2
1.3 Théorie de l'information	3
1.3.1 Types de sources	3
1.3.2 Conversion analogique/digitale	3
1.3.3 Quantité d'information	4
1.3.4 Entropie	5
1.3.5 Quantité de décision et redondance	5
1.4 Les limites de canaux de transmission	6
1.4.1 Etats électriques	6
1.4.2 Bande passante d'un canal parfait	6
1.4.3 Bande passante d'un canal physique (réel)	7
1.4.4 Rapport signal sur bruit	7
1.5 La compression sans perte	7
1.5.1 Méthodes	7
1.5.2 Problèmes	8
1.5.3 En pratique	8
1.5.4 Dynamic Huffman	8
2 Le traitement des erreurs de transmission	11
2.1 Protection contre les erreurs de transmission	11
2.2 Distance de Hamming et conditions de détection et correction	12
2.2.1 Poids et distance de Hamming	12
2.2.2 Conditions sur la détection et la correction d'erreur	12
2.2.2.1 Intuitivement	12
2.2.2.2 Formellement	13
2.2.2.3 Conditions généralisées	13
2.3 Détection d'erreur	14
2.3.1 Parité	14
2.3.2 CRC	14
2.3.2.1 Introduction	14
2.3.2.2 Erreurs détectées	15
2.3.2.2.1 Erreur simple	15
2.3.2.2.2 Erreur double isolée	15
2.3.2.2.3 Paquet d'erreurs de longueur k	15
2.3.2.2.4 Erreurs en nombre impair	15
2.3.2.3 Applications informatiques et électroniques	16
2.4 Correction d'erreur	16

2.4.1	Code de Hamming	16
2.4.2	Codes de correction d'erreur	17
3	Protocoles sûrs (protocoles à fenêtre)	19
3.1	Idle Request (IDLE RQ)	20
3.2	Continuous Request (Continuous RQ)	21
3.2.1	Principes	21
3.2.2	Nombre de numéros de séquence	21
3.2.3	Contrôle de flux	22
3.3	Un exemple : HDLC (résumé)	22
3.4	Rendement des protocoles	23
3.4.1	Rendement intrinsèque	23
3.4.2	Rendement d'Idle Request	24
3.4.3	Continuous request : cas sans retransmissions	24
3.4.4	Ligne réelle	25
4	Le dernier kilomètre (the last mile)	27
4.1	PME et usagers résidentiels	27
4.2	Entreprises	28
4.3	Réseaux d'accès	29
4.3.1	xDSL	29
4.3.2	Câble TV	30
4.3.3	Internet par réseau électrique	31
4.3.4	Boucle locale sans fils (wireless local loop)	31
5	Transmission numérique	33
5.1	Transmission numérique de la voix	33
5.2	ISDN/RNIS : Le réseau numérique à intégration de services (résumé)	34
5.2.1	Introduction	34
5.2.2	DSS1 I.430 (BRI) : couche physique	34
5.2.2.1	Topologie et interfaces U/S/T	34
5.2.2.2	Bus S	35
5.2.2.3	Anciens équipements analogiques	35
5.2.2.4	Variante centrale d'abonné	35
5.2.3	DSS1 I.441 (Q.921, LAPD) : couche liaison	36
5.2.4	DSS1 I.451 (Q.931, PLP) : couche réseau	36
5.2.5	Données du canal B	37
5.3	Hiérarchies numériques	37
5.3.1	Introduction	37
5.3.2	Hiérarchique numérique plésiochrone (PDH)	38
5.3.3	Hiérarchique numérique synchrone (SDH)	39
5.4	ATM : Asynchronous Transfer Mode	40
5.4.1	Modèle en couche d'ATM	40
5.4.1.1	ATM : couche physique	40
5.4.1.2	Couche liaison : sous-couche MAC	41
5.4.1.3	Couche liaison : sous-couche AAL	41
5.4.2	ATM LANE : émulation LAN	42
5.4.3	Permanent Virtual Connection	42
5.4.4	Conclusion	43

6	Transmission sans fil	45
6.1	Technologies	45
6.2	Calcul de liaison pour des faisceaux hertziens courts	46
6.2.1	Facteurs limitatifs	46
6.2.2	Niveaux et puissance	46
6.2.3	Affaiblissement	46
6.2.4	Bilan de liaison	47
6.2.5	Rapport signal sur bruit	47
6.2.6	Limitation de la puissance rayonnée	47
6.3	Affaiblissements	47
6.3.1	Affaiblissement linéique	47
6.3.2	Affaiblissement en espace libre	48
6.3.2.1	Application	48
6.3.3	Autres affaiblissements	48
6.4	Ellipsoïde de Fresnel	49
6.4.1	Exemple de calcul	50
7	Sécurité des réseaux	51
7.1	La problématique	51
7.1.1	Les menaces	51
7.1.2	Les moyens de mitigation des menaces	52
7.2	Sécurisation d'un réseau d'entreprise	52
7.2.1	Moyens permettant d'améliorer la sécurité du périmètre	52
7.2.1.1	Dispositifs logiciels ou embarqués	53
7.2.1.2	Limitation des applications	53
7.2.1.3	Séparation des fonctionnalités	53
7.2.1.4	Couper la connexion jusqu'à la couche 7 : le proxy	54
7.2.1.4.1	Inconvénients	54
7.2.1.5	Réseau d'entreprise typique	54
7.2.2	Surveillance	55
7.2.2.1	Surveillance des logs et alarmes	55
7.2.2.2	Détection d'intrusion active	55
7.2.2.3	Détection d'intrusion passive (Honey Pot)	55
7.2.3	Réseaux privés virtuels (VPN)	55
7.2.3.1	VLAN Ethernet	56
7.2.3.2	Tunnels IP	56
	Références et bibliographie	57
	Index des concepts	59
	Table des matières	65

ISBN 978-2-940387-04-5



9 782940 387045