Protocoles & réseaux – 2024-2025

révision 2.26

Marc SCHAEFER

évolution et extension du cours de Romain VOUMARD

28 août 2024

http://www.he-arc.ch/ingenierie

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).

Remerciements

En plus des auteurs mentionnés sur la page de couverture, je tiens à remercier François Goetz et Alexis Domjan pour certains éléments de physique du chapitre transmission sans fil, et Kolawolé Atchadé pour quelques éléments de probabilité du chapitre des rendements des protocoles sûrs.

Sommaire

Chapitre 1

Offre de télécommunications

Sommaire

1.1 Introduction

1.1.1 Objectifs de ce chapitre

Le but de ce chapitre est de présenter certains aspects spécifiques de l'offre de télécommunication et de sa durabilité (fiabilité, efficacité énergétique, dégroupage, neutralité du réseau) et enfin son évolution et le futur des réseaux, telle qu'elle peut être utile à une entreprise pour se connecter à Internet ou pour transporter des données entre sites, que cela soient des données informatiques, multimédia ou techniques.

En support de ce chapitre, les chapitres 5 (Le dernier kilomètre) et 6 (Hiérarchie des systèmes numériques) seront utilisés pour décrire plus en détail les technologies et concepts nécessaires de l'offre de télécommunications. Les aspects de sécurité réseau sont traités dans les chapitres 8 et 9.

1.1.2 Classification des technologies

On peut classifier les technologies de télécommunications en trois catégories :

1. celles qui ne nécessitent pas d'opérateur car elles peuvent être déployées par l'utilisateur lui-même : xDSL sur liaison cuivre ou Ethernet sur fibre optique, que l'on possède le média ou qu'on le loue à un opérateur ; sans-fil avec par exemple WiFi, Bluetooth, Zigbee Z-Wave, LoRaWAN, FSO...

2. celles qui nécessitent un réseau d'opérateur spécifique, mais pas Internet : services de télécommunications classiques voix ou messages (GSM, SMS), lignes louées numériques, VPN divers par exemple basés MPLS...
3. celles qui nécessitent Internet et donc un accès via un ou plusieurs opérateurs : accès Internet, VPN inter-sites ou cloud

Dans les deux premiers cas, la qualité de service est triviale ou assurée par l'opérateur. Dans le dernier cas, la qualité de service n'est possible que dans certains cas et lorsqu'on ne passe pas à travers plusieurs réseaux différents (sauf encore rare interconnexion **NGN IMS**, voir section 1.3 en page 5).

1.2 Durabilité

1.2.1 Assurer la fiabilité

1.2.1.1 Motivation

Dans un contexte où les réseaux sont essentiels au fonctionnement de l'économie et de la société, la fiabilité ne se limite pas aux aspects techniques de traitement des erreurs (voir chapitre 3) ni à la mise en place de protocoles fiables et performants (voir chapitre 4), ou à la qualité de service (voir section 6.3.6) mais comprend également les risques de **blackout** – la panne totale ou partielle, que cela soit de l'alimentation électrique ou de l'accès à Internet.

1.2.1.2 Besoin

En effet, si les opérateurs télécoms et les centres de données (**datacenter**s) ont déjà des plans de haute fiabilité, de continuité et de reprise après désastre, rares sont les entreprises qui avant l'année 2022 avaient penser à se préparer à un tel blackout.

Un aspect récent est la demande des pouvoirs publics d'identifier les activités essentielles et d'économiser l'énergie.

1.2.1.3 Architecture générale

L'objectif est, dans la mesure du possible, d'identifier, puis d'éliminer les points de panne (**single point of failure**), dans le cas où l'entreprise a suffisamment d'information techniques de détail : sinon, elle peut transférer, de cas en cas ou globalement, ce risque en demandant des garanties, par contrat de service (**SLA**, *Service Level Agreement*) à ses fournisseurs et sous-traitants.

Même si la problématique du **blackout** est complexe, on peut toutefois identifier, en ce qui concerne les aspects informatiques, trois *zones* où des mesures doivent être prises ou des garanties obtenues :

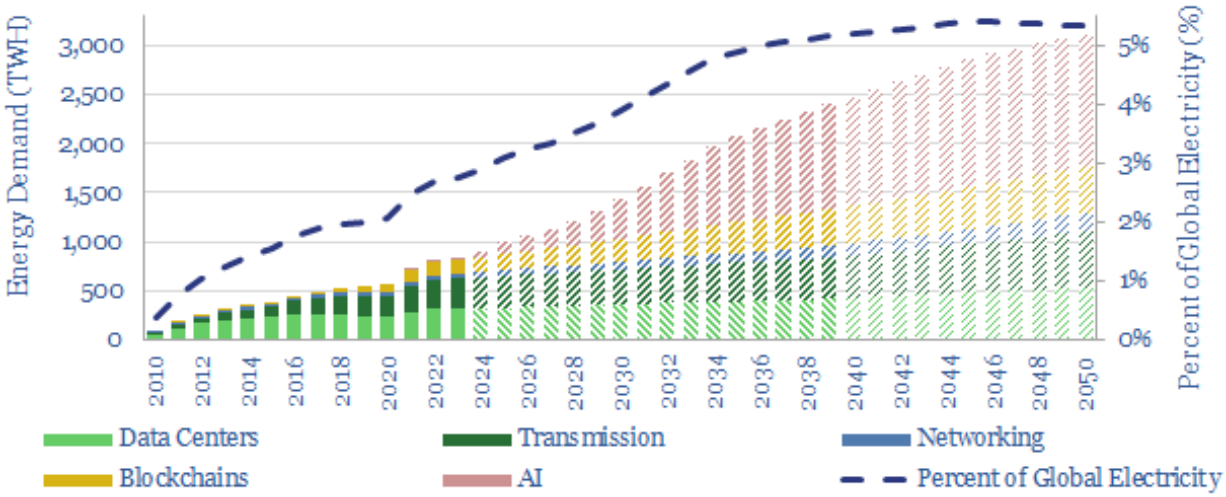
l'entreprise elle-même terminaux, serveurs, équipements réseau, raccordement Internet (réseau d'accès), énergie : ici, on privilégiera les équipements sur batterie (laptop), la téléalimentation (téléphones, équipements réseau), les alimentations permanentes (**UPS**), en définissant un réseau minimal électrique, informatique et technique qui puisse survivre à des pannes (**redondance**, en tenant compte de l'ensemble du réseau (routeurs, switches...) mais aussi l'auto-consommation et les mesures d'économie d'énergie

le domaine de l'opérateur télécom technologie(s) du du réseau d'accès, du **backhaul** qui relie l'entreprise au réseau core de l'opérateur, et bien sûr du réseau core (voir 6.3.2 en page 79) : en particulier pour le réseau d'accès, il est rare que la continuité soit garantie en cas de coupures de courant, et certains systèmes protégés ne vont pas résister à des pannes d'une certaine durée ou en cas de répétition : certaines technologies [1] offrent intrinsèquement de meilleures garanties.

le cloud qu'il soit privé, public ou sous forme de services tiers : la détermination de la localisation et des garanties offertes est essentielle

1.2.2 Energie et ressources

1.2.2.1 Consommation d'énergie Internet



Source : https://thundersaidenergy.com/2023/04/20/what-is-the-energy-consumption-of-the-internet/

FIGURE 1.1 – Consommation d'énergie liée à Internet
(ne tient pas compte des terminaux – périphériques de l'utilisateur – qui par leur multipicité ont un impact important, aussi en énergie grise)

L'utilisation d'applications modernes communicantes via Internet consomme de l'énergie – et donc la plupart du temps produit des gaz à effet de serre, dans une certaine proportion – que l'on peut répartir comme suit, par proportion descendante de la consommation [2]

- les périphériques de l'utilisateur final
- les **datacenter**s (avec des données et applications génériques, **blockchain** et d'**IA**)
- les réseaux d'accès (voir section suivante)
- le réseau Internet lui-même

Même si en 2023, l'énergie totale – hors périphériques de l'utilisateur final – représentait moins de 2% du total de la consommation, des nouvelles applications (éventuellement la blockchain, mais surtout l'IA), un doublement est possible d'ici à 2030.

1. par exemple la fibre optique pure en topologie point à point **PTP FTTH**, le multiplexage optique passif (**GPON**) **PTMP** ou même l'**xDSL** classique non hybride (voir section 5.3 en page 59)

2. ICTjournal numéro de février 2023, page 35 concernant l'IT en général ; étude spécifique 2021 sur l'impact carbone du streaming https://www.carbontrust.com/our-work-and-impact/guides-reports-and-tools/carbon-impact-of-video-streaming

1.2.2.2 Efficacité énergétique

1.2.2.2.1 Dernier kilomètre (réseaux d'accès) La meilleure efficacité est obtenue avec les technologies de fibres optiques, qui ne nécessitent pas de composants actifs au sein du dernier kilomètre, y compris en tenant compte de l'énergie grise (construction). Cela comprend les technologies point-à-point (PTP : FTTH pur) et point-à-multipoints (PTMP, comme le multiplexage GPON par exemple) – voir section 5.3 en page 59.

Toutefois, "la technologie PTP FTTH offre les les plus grands débits et peut donc transmettre le plus d'information avec le moins de consommation d'énergie" [3]. C'est la technologie désormais retenue par le législateur en Suisse, surtout pour ses avantages de dégroupabilité (voir ci-après) et de débits futurs. Néanmoins, à vitesse fixée et atteignable par lui, le GPON, est la technologie énergétiquement la plus efficace, en particulier en ce qui concerne l'énergie grise – il nécessite moins de travaux de génie civil.

Les technologies hybrides (xDSL, CATV) consomment plus d'énergie, et la palme revient au sans-fil : même si la 5G est plus économe que les générations précédentes (meilleure efficacité spectrale) elle reste comparativement bien moins économique – il ne faudrait surtout pas remplacer les nombreux projets de déploiement de fibre optique par de la 5G !

1.2.2.2.2 Datacenter On peut définir l'efficacité énergétique comme

$$PUE = \frac{E_{totale}}{E_{utile}} \tag{1.1}$$

avec E_{totale} toute l'énergie consommée (y compris refroidissement, traitement de l'air...) et E_{utile} uniquement l'énergie utile consommée par les équipements informatiques, du stockage et du réseau.

L'objectif est de se rapprocher le plus possible de 1, avec des avancées importantes lors de la construction de datacenter récents.

Évidemment, cela ne donne pas d'indication sur l'efficacité des équipements utiles eux-mêmes, mais uniquement sur la qualité énergétique de l'infrastructure du **datacenter**.

1.2.3 Dégroupage

Le dégroupage est un concept légal [4] qui permet à des opérateurs tiers d'utiliser tout ou partie de l'infrastructure d'un autre opérateur (souvent un opérateur historique en position de dominance), de manière à mutualiser les coûts et à réduire les conditions financières (investissements notamment) nécessaires pour pénétrer le marché.

Il existe différents envergures de dégroupages :

complet l'opérateur tiers utilise uniquement ses propres infrastructures

quasi-complet l'opérateur tiers exploite son propre réseau d'amenée sur ses propres infrastructures, mais dépose des équipements de raccordement au dernier kilomètre dans le central d'un autre opérateur en le dédommageant (énergie, forfait de rack...) et lui loue la ligne vers l'abonné

3. https://europacable.eu/wp-content/uploads/2022/07/Europacable-Whitepaper-on-Energy-Efficiency-of-Fiber-networks-05-July-2022.pdf
4. qui a suivi au concept précédent de monopole d'un opérateur unique historique, en Suisse Swisscom

partiel l'opérateur tiers loue la ligne d'abonné, les équipements de multiplexage du central et la liaison de données à l'autre opérateur – le réseau d'amenée est donc en partie virtuel et peut être à grande distance [5]

En Suisse, la loi ne prévoit l'obligation de dégroupage quasi-complet ou partiel que pour l'infrastructure cuivre (xDSL), voire pour le FTTH PTP : au contraire, les technologies hybrides comme le FTTS ou le CATV ne prévoient pas d'obligation (voir section 5.3 en page 59). Le dégroupage complet est toujours possible mais nécessite des investissements (en doublon) très importants, irréalisables à court terme, et donc rend impossible l'entrée rapide d'un opérateur concurrentiel avec une offre à tous les clients potentiels.

Après avoir plutôt réalisé des infrastructures hybrides (**FTTx**) ou multiplexées (**GPON PTMP**), l'opérateur historique Swisscom, qui reste en sitation de dominance et disposant du contrat de service universel, a, suite à une décision du Tribunal fédéral [6] enfin décidé en 2022 de déployer plutôt des solutions **FTTH** point-à-point (**PTP**) à l'avenir.

1.2.4 Neutralité du réseau

On entend par **neutralité du réseau** le principe qui garantit l'égalité de traitement de tous les flux de données, par exemple sur Internet.

Ce principe s'oppose à la volonté de certains opérateurs de garantir le financement des infrastructures, très coûteuses, en segmentant le marché. Par exemple, si Netflix peut passer par le réseau Swisscom sans rien payer, cela peut gêner Swisscom, qui pourrait avoir envie de dégrader la qualité de service de cette offre concurrente par rapport à sa propre offre TV, par exemple à fin de forcer Netflix à signer un contrat de rétrocession.

Chaque pays peut mettre en place une législation plus ou moins stricte pour garantir le niveau souhaité de libre concurrence et de libre choix des utilisateurs tout en garantissant les investissements dans les infrastructures.

En Suisse et depuis 2021, l'article 12 de la loi sur les télécommunications [7] fixe le cadre juridique fixe un cadre qui n'autorise la différenciation que dans des buts légaux, de sécurité, de demande explicite d'un client ou pour lutter contre des congestions (en garantissant un traitement similaire de types de données similaires). L'optimisation de services propres est autorisée si elle ne remplace ni ne dégrade des services Internet. Une information au client et du public est dans tous les cas nécessaire.

En pratique, l'exemple de dégradation mentionnée ci-dessus ne serait plus légale. Ce qui n'empêche pas les opérateurs d'essayer de se refinancer en recapturant la clientèle avec un Netflix sous contrat dans leur assortiment (box Internet) ou en demandant des changements légaux (taxe GAFAM).

1.3 Evolution et futur des réseaux

La plupart des réseaux **core** d'opérateurs ont migré à de la commutation d'IPv6 et IPv4 à des débits très élevés et intégrant, au sein d'un même réseau d'opérateur, la **QoS** (qualité de service), avec MPLS, voir section 6.3.5.2 en page 80.

Ils utilisent des technologies de type **NGN light** pour intégrer les technologies existantes et développer de nouveaux services au sein de chaque opérateur.

5. Sunrise pour xDSL, FTTS et même GPON, utilise le réseau Swisscom avec deux centres (ZH et LS)
6. https://www.letemps.ch/economie/cyber/face-gendarme-concurrence-swisscom-cede-distribution-fibre-optique
7. https://www.fedlex.admin.ch/eli/cc/1997/2187_2187_2187/fr#art_12_e

Toutefois, les réseaux publics, dont la téléphonie classique et les SMS, sont encore interconnectés entre opérateurs avec des technologies obsolètes et peu sûres comme **SS7** : l'espoir est que les interconnexions **NGN IMS** se multiplient pour non seulement sécuriser le réseau public de téléphonie mais également pour permettre la qualité de service et le chiffrement systématique par tronçon à travers Internet (**NGN full**, voir section 6.3.5.3 en page 80).

Chapitre 2

Théorie de l'information

Sommaire

2.1 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-ci [1], contient une grande structure, ou **redondance**, ce qui signifie qu'il serait possible de définir un autre **codage** de la même information, mais qui utiliserait beaucoup moins de place (de bits). Par exemple, comme la lettre e apparaît très souvent en français, on 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 3 en page 23).

2.2 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) :

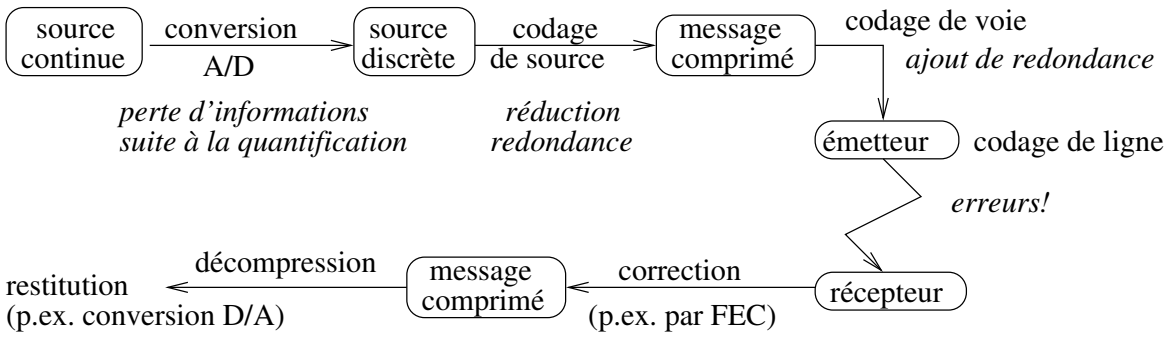


FIGURE 2.1 – Codage de source et de voie

Le codage de source a pour but de *réduire la redondance*. C'est ici que l'on trouve les algorithmes (**codec**s) de **compression** [2] et de décompression. La compression peut se faire **avec perte** (compressions audio et vidéo, voir section 2.6 en page 20) ou **sans perte** d'information (compression de données selon Huffmann, Lempel-Ziv, voir section 2.5 en page 18).

Il est évident que des données compressées sont beaucoup plus sensibles aux erreurs de transmission : une seule erreur sur un bit peut potentiellement se reporter sur un nombre important de symboles suivant l'erreur (décalage de l'interprétation).

1. l'original de ce document LaTeX est en ISO-8859-1 (Latin-1). Parler d'ASCII est ici un abus de langage.
2. la compression de données a pour but de réduire la taille des données – ne pas confondre avec la *Dynamic Range Compression*, qui en audio permet de modifier la dynamique du signal.

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

Le codage de voie est vital dans toutes les applications de téléinformatique et est traité dans le chapitre 3 en page 23.

2.3 Théorie de l'information

Claude SHANNON, des laboratoires BELL [3] décrit en 1948 les bases de la théorie de l'information en communication électronique [7].

2.3.1 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, ...).

Pour une source donnée, 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 inter-symboles, on l'appellera une source avec mémoire.

Les langues naturelles (français, anglais, ...) sont des sources *avec* mémoire, modélisables sous forme de source de **Markov** : par exemple, la probabilité d'apparition d'un `n` après un `t` est beaucoup plus faible que celle d'un `a` après un `t`. Donc si un `t` est apparu, la probabilité d'apparition d'un `a` augmente rapport à celle du `n` : il s'agit de probabilités conditionnelles.

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, qui lui est adapté aux sources *avec* mémoire, comme par exemple les langues naturelles).

2.3.2 Conversion analogique/digitale

2.3.2.1 Principe

La plupart des sources naturelles sont analogiques (on dit aussi continues, par opposition aux sources discrètes). La conversion d'une source analogique à une source discrète ou numérique se fait par la conversion A/D (voir figure 2.2 en page 10).

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, le **codec 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 **Shannon-Nyquist** [4], 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**.

3. connus pour l'invention du transistor et la création d'UNIX.

4. il faut échantillonner au double de la plage de fréquence à représenter : il faut déduire f_{min} de f_{max} si la borne inférieure de la plage n'est pas voisine de zéro, comme si l'on décalait le spectre en bande de base

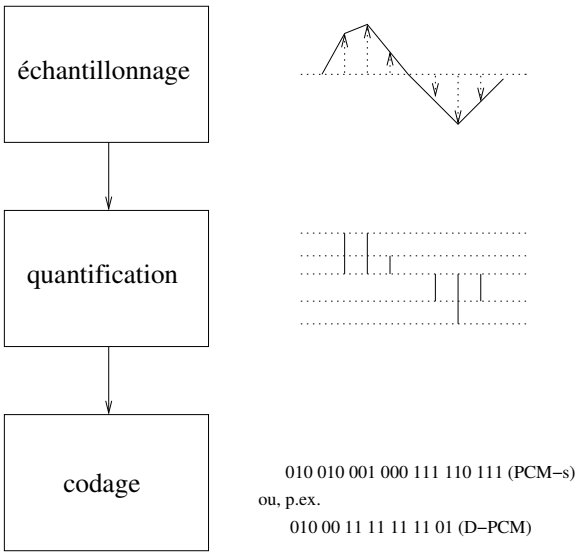


FIGURE 2.2 – Conversion analogique/digitale

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. Graphiquement, on peut la représenter par un escalier qui approche les valeurs échantillonnées. Dans le cas de G.711A et de G.711μ, le pas de quantitification n'est pas constant [5] : les plus faibles amplitudes sont codées sur plus de bits, donc offrent plus de détails.

La troisième et dernière phase, le **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**).

Le codec 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 Shannon-Nyquist est respecté, seule l'étape de quantification perd de l'information – cette perte unique est un choix initial conscient de résolution (**bruit de quantification**). Une fois numérisée, l'information sera régénérée numériquement, sans perte. L'étape de codage peut toutefois introduire une compression avec perte.

2.3.3 Avantages de la numérisation

Parmi les avantages de la numérisation, on peut citer une meilleure résistance aux erreurs de transmission : la **régénération numérique** permet de reconstituer toute l'information numérisée, tant que le bruit ne provoque pas la lecture d'un état discret différent de celui qui a été émis. Les erreurs résiduelles liées à une erreur de lecture d'état peuvent être détectées ou corrigées par l'ajout de redondance numérique. Au contraire d'une simple *amplification analogique* qui amplifie signal *et* erreur et qui ne peut jamais retrouver la qualité originale du signal.

En corollaire et jusqu'à un certain niveau de bruit, un avantage supplémentaire est de pouvoir limiter la perte de qualité de bout en bout à celle – calculable – du processus de conversion

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

A/D : le bruit de quantification de l'étape 2 (et l'éventuelle compression avec perte le cas échéant à l'étape 3). Une autre façon de le dire est que la perte en dB, au sein d'un système de transmission à régénération numérique, est nulle. Ou que la correction d'erreur peut être vue comme un gain de rapport signal sur bruit.

De plus, des études[6] montrent que la numérisation peut amener à des économies d'énergie et de spectre de fréquence. Enfin, la numérisation permet la mise en place de nouveaux services intégrés.

2.3.4 Quantité de décision et débit binaire

Si l'on a n éléments à différencier (par exemple une source discrète comportant n symboles distincts), l'envergure de la décision à prendre à chaque symbole, nommée **quantité de décision**, soit le nombre de bits nécessaires pour coder ces n symboles, vaut, trivialement :

$$D_{triviale} = \lceil \log_2(n) \rceil \tag{2.1}$$

Cette quantité de décision dite *triviale* (qui ne tient pas compte de la répartition des symboles) dépend uniquement du nombre de symboles de la source n.

Autrement dit, si l'on a D bits, on peut compter 2^D symboles différents. Bien sûr, si D n'est pas entier, il faut l'arrondir à l'entier supérieur.

Si le codage n'est pas fait sur un nombre constant de bits (symboles de différentes longueurs tenant compte de la répartition de ceux-ci – nous verrons que c'est essentiel pour la compression), alors on peut définir un $D = l_m$ qui représente la longueur moyenne d'un symbole codé d'une certaine manière (pondération des probabilités d'apparition et des longueurs de chaque symbole, voir l'équation 2.5).

Le **débit binaire**, exprimé en bits par seconde ou **bps**, peut être vu comme la dérivé de D par rapport au temps t et se symbolise par $\dot{D}$ (D point). On l'appelle souvent le débit de décision. Lorsqu'on parle du débit binaire maximum d'un canal, on note souvent ce débit C (capacité d'un canal).

2.3.5 Quantité d'information

Chaque **symbole** d'une source 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.

Nous avons déjà vu que la quantité de décision permettant d'identifier 1 symbole parmi n, si la répartition de ces symbole est équiprobable, vaut $D = \log_2(n)$. On peut se demander intuitivement combien de bits un symbole plus ou moins surprenant porte : on a l'impression que plus un symbole est surprenant, plus il porte d'information et plus on peut le coder sur beaucoup de bits (car il est rare). A contrario, s'il est très courant, il est avantageux – pour minimiser la longueur moyenne – de le coder sur moins de bits (il porte moins d'information).

On peut quantifier intuitivement le nombre de bits idéal porté par un symbole : si l'on considère le **nombre d'occurences** d'un symbole, on peut calculer le nombre de bits portés par ce

6. https://www.bbc.co.uk/rd/blog/2020-10-sustainability-radio-audio-energy-streaming-broadcast

symbole (appelé ici la **quantité d'information**) : par exemple, supposons que l'on établisse la statistique suivante : le symbole X_i apparaît 16 fois sur 64 en moyenne. Cela signifie que pour identifier X_i nous n'avons besoin non pas de 6 bits (1 symbole sur 64) mais bien de 2 bits (1 cas sur 4 est un X_i).

Plus formellement, si la probabilité d'apparition du symbole i est notée P_i, alors la quantité d'information du symbole X_i vaut :

$$H_i = -\log_2 P_i \tag{2.2}$$

Comme on définit P_i comme étant la fréquence d'apparition du symbole X_i, autrement dit le nombre d'occurence N_i du symbole X_i divisé par le nombre total d'occurences N_{total} de tous les symboles, H_i vaut bien $-\log_2 \frac{N_i}{N_{total}}$ ou encore, par propriété du logarithme, $\log_2 \frac{N_{total}}{N_i}$. On retrouve donc l'exemple intuitif ci-dessus avec $\log_2 \frac{64}{16} = 2$ bits.

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 :

$$H_{ijk} = H_i + H_j + H_k \tag{2.3}$$

(sous-entendu : pas de dépendances entre symboles, **source sans mémoire** ! sinon cette formule n'est pas valable !)

2.3.6 Entropie

L'entropie correspond à l'aptitude d'une source à produire de l'information. L'entropie est comprise entre zéro[7] et au maximum la quantité de décision (voir l'équation 2.1).

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 :

$$H = \sum_i P_i \; H_i = -\sum_i P_i \; \log_2 P_i \tag{2.4}$$

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

$$l_m = \sum_i P_i \; l_i \tag{2.5}$$

qui donne la longueur moyenne des symboles émis par une source, une fois codés concrètement (l_i étant la longueur en bits du code du symbole i et P_i sa probabilité d'apparition).

7. source ne produisant pas d'information, par exemple une source binaire ne sortant que des 1

L'entropie donne en $\frac{Sh}{symbole}$ 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 et vaut alors la quantité de décision (voir l'équation 2.1).

Par exemple, une source de 64 symboles équiprobables produira une entropie de $\log_2 64 = 6$. Et une source binaire qui produit à chances égales des 1 et des 0 a une entropie de 1.

2.3.7 Redondance

Comment peut-on évaluer si la représentation choisie d'une source (son code) est efficace ou non ? L'entropie représente en fait la *borne minimale* de taille de code en-dessous de laquelle il n'est pas possible de descendre, dans cette source, car il n'y a plus de structure redondante.

La redondance est donc le potentiel de compression de la source dans un code donné : elle est définie par la différence entre la quantité de décision de ce code et l'entropie de la source :

$$R = D - H \tag{2.6}$$

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

Par exemple, si l'on prend une source avec 64 symboles, le codage binaire trivial est $D = \log_2 64 = 6$. Supposons que la répartition des symboles est équiprobable, alors l'entropie est maximale et vaut $H = D$ et donc $R = 0$: il n'est pas possible de compresser. Si, au contraire, la répartition des symboles n'est pas équiprobable, H sera inférieur à D et donc R sera positif. Il sera possible de compresser avec un nouveau code plus efficace, par exemple **Huffman**, et donc de déterminer la longueur moyenne d'un symbole codé plus efficacement $l_{Huffman}$ (selon l'équation 2.5) et donc calculer une nouvelle redondance résiduelle $R_{Huffman}$ (avec l_m remplaçant D). Un codage plus optimal voit R diminuer, sans forcément atteindre zéro.

Huffman est d'ailleurs un algorithme de compression optimal pour les source sans mémoire, sous conditions [8]

Dans les langues naturelles (sources *avec mémoire*, dont l'entropie ne peut *pas* se calculer avec la formule 2.4 en page 12, vu la dépendance inter-symboles), on a constaté que les langues énonçant les syllabes le plus rapidement sont aussi celles dont la quantité d'information de chaque syllable est la plus faible. Cela signifie que le japonais, avec 8 syllabes/s, contient plus de redondance (structure soutenant la compréhension) que le thaï avec 5 syllabes/s. De plus, si l'on accélère de moitié un enregistrement d'un texte en langue naturelle, le cerveau a de la peine à suivre : il est possible de montrer que la limite est la capacité de traitement de l'information par le cerveau (de l'ordre de 39 $\frac{bit}{s}$), lié à un optimum évolutif entre énergie nécessaire et optimum d'informations pour survivre [9].

8. les probabilités doivent être voisines de puissances de 2 et le nombre de symboles une puissance de 2 : sinon, le codage arithmétique qui généralise Huffman peut être meilleur, car il permet de coder chaque symbole sur un nombre non entier de bits (voir http://pageperso.lif.univ-mrs.fr/~andreea.dragut/enseignementCLAA/CoursTD5_Fin.pdf), mais il est plus lent, complexe, et breveté.

9. https://advances.sciencemag.org/content/5/9/eaaw2594

2.3.8 Application au générateur de nombres aléatoires (RNG)

2.3.8.1 Introduction

Les applications ayant besoin de nombres tirés au hasard sont de deux types :

1. celles ayant besoin de nombres dont la répartition statistique soit la plus irréprochable [10] possible (jeux, tests et simulations p.ex.) : **nombres pseudo-aléatoires**
2. celles qui ont besoin de nombres qui sont non devinables par un attaquant, même en disposant de toutes les valeurs du passé (cryptographie p.ex.) : **nombres aléatoires**

2.3.8.2 Générateur pseudo-aléatoire (PRNG)

La plupart des générateurs aléatoires utilisés dans les logiciels (p.ex. `random(3)` de la bibliothèque C standard, `rand` en PHP ...) produisent en fait des nombres pseudo-aléatoires, dont la génération est basée sur des fonctions déterministes mais dont les sorties sont suffisamment différentes des entrées.

On utilise en général une *graine* qui initialise la séquence pseudo-aléatoire. Cette graine est souvent basée sur des éléments contenant peu d'entropie (p.ex. secondes depuis minuit [11]). A une même graine correspond la même [12] suite de nombres pseudo-aléatoires.

La suite de nombres ainsi produit remplit en général certains critères statistiques, avec quelques biais plus ou moins gênants suivant l'application (périodicité, manque d'uniformité ...). Elle n'est absolument pas appropriée lorsque les nombres produits doivent résister à des attaques [13] !

2.3.8.3 Générateur basé sur une source d'entropie (RNG)

Pour obtenir de meilleurs nombres aléatoires, appropriés aux applications cryptographiques, le mieux est de se baser sur des phénomènes *physiques* : lorsqu'une source physique (circuit spécialisé [14] basé sur des bruits thermiques, des sons, etc) n'est pas disponible ou non suffisante, on peut combiner des sources d'entropie provenant d'événements jugés non déterministes : délai entre les touches du clavier, déplacement de la souris, délais entre paquets réseaux, entre interruptions, etc.

Ces sources ainsi combinées alimentent un **pool d'entropie** (*entropy pool*), dans lequel des consommateurs puisent des nombres aléatoires. L'entropie du pool à un moment donné peut être évaluée. Lorsque l'entropie descend au-dessous d'un seuil, le système peut signaler que les nombres aléatoires ne sont pas de qualité suffisante.

De manière à ne pas réduire trop rapidement l'entropie du pool et à ne pas donner d'indices de son contenu à un éventuel consommateur-attaquant, les nombres aléatoires sont dérivés par hachage du pool, ou, plus récemment, par un **CPRNG** (*Cryptographic Pseudo Random Number Generator*) qui utilise un PRNG tout en produisant des nombres cryptographiquement sûrs.

---

10. absence de *biais* statistiques
11. donc moins de 17 bits d'entropie, voir quasi rien si l'on connaît l'heure approximative de génération
12. ce qui peut être souhaité dans certains processus de test p.ex.
13. une analyse et des recommandations génériques et spécifiques à des APIs particulières : https://peteroupc.github.io/random.html
14. par exemple http://onerng.info/ ou certains circuits de cartes-mères ou embarqués

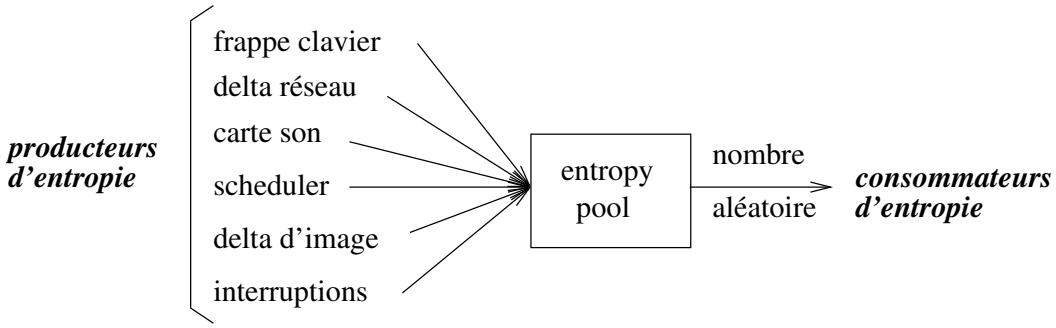


FIGURE 2.3 – Générateur aléatoire entropique

2.3.8.4 Exemple de générateur aléatoire entropique : /dev/random et /dev/urandom sous Linux

Linux implémente un pool d'entropie, alimenté par des événements systèmes comme vu ci-dessus. Deux périphériques caractères, `/dev/urandom` et `/dev/random` permettent d'obtenir des octets aléatoires de qualité cryptographique.

Le fonctionnement de ce système a changé récemment, en utilisant notamment un **CPRNG** basé sur ChaCha20, mais en quelques mots, `/dev/random`, qui bloquait auparavant lorsque le pool d'entropie n'était pas suffisamment alimenté, se comporte exactement comme `/dev/urandom` dès lors que le CPRNG a été initialisé [15].

2.4 Les limites de canaux de transmission

2.4.1 Etats électriques

L'information à transmettre doit être codée d'une façon adaptée [16] 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)
- etc.

Et pour une fibre optique, le plus souvent c'est la présence (1) ou l'absence (0) de lumière (**OOK**, On Off Keying) qui encode l'information.

15. http://ermanarslan.blogspot.com/2020/05/entropy-linux-kernel-csprngs-devurandom.html

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

2.4.2 Débit de décision et débit de moment

Le débit de décision (débit binaire) 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 (débit de moment, ou rapidité de modulation) puisqu'on transmet 2 bits par état.

Ce débit de moment est donné 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 :

$$\dot{D} = \log_2(m)\ \dot{M} \tag{2.7}$$

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

2.4.3 Relation entre débit de moment et bande passante

Intuitivement, $\dot{M}$, le débit de moment, dépend de la bande passante du canal B : en effet, la vitesse de montée d'un signal dépend de la bande passante à disposition. Nyquist a montré que dans le cas d'un canal vu comme filtre passe-bas idéal garantissant l'absence d'interférence entre moments successifs du signal on a :

$$\dot{M}_{max} = 2B \tag{2.8}$$

$\dot{M}_{max}$ est la borne supérieure de la vitesse de modulation en Baud (bd)
B : largeur de bande[17] du canal (Hz)

Ce résultat est un corollaire du théorème de l'**échantillonnage** (**Shannon-Nyquist**) : en effet, ce théorème indique que la première étape de la conversion A/D (échantillonnage) est sans perte si l'on échantillonne à une fréquence double de la fréquence maximum du signal[18].

Il s'agit toutefois d'une valeur maximum, difficile à atteindre en pratique. On peut s'en approcher si le canal a de bonnes caractéristiques au départ, ou si l'on utilise des systèmes d'équaliseurs actifs corrigeant ses caractéristiques. En pratique, on prend souvent $\dot{M} = 1.25B$ [4]. L'ADSL classique, travaillant dans un milieu difficile, prend environ $\dot{M} = B$.

2.4.4 Débit de décision 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

17. nous utilisons ici bande passante et largeur de bande de manière interchangeable
18. ou plus exactement, à deux fois la bande passante B du signal considéré, définit comme la différence entre la fréquence la plus haute et la fréquence la plus base, voir section 2.3.2.1 en page 9

passante limitée et transmet tous les signaux sans perturbation dans cette bande : on dit que le canal est exempt de bruit.

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

$$\dot{D}_{max} = C = \dot{M} \ \log_2(m) \tag{2.9}$$

$\dot{M}$ est la vitesse de modulation en Baud (bd)
m est le nombre d'états du canal

Comme limite supérieure, on peut utiliser l'équation

$$C = 2B \quad \log_2(m) \tag{2.10}$$

dont le débit n'est pas atteignable en pratique.

Ces deux formules donnent l'illusion qu'on peut atteindre des vitesses arbitrairement hautes en augmentant le nombre d'états possibles m du canal. Toutefois, plus les états sont rapprochés, plus le risque d'erreur augmente, en présence de bruit.

2.4.5 Débit de décision d'un canal physique (réel, bruité)

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

$$\dot{D}_{max} = C = B \ \log_2(1 + \frac{S}{N}) \tag{2.11}$$

C : vitesse maximale en bps
S : puissance du signal (Watt)
N : puissance du bruit (Watt)
B : largeur de bande du canal (Hz)

2.4.6 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) :

$$\eta = SNR_{dB} = 10 \ \log_{10} \frac{S}{N} \tag{2.12}$$

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

Par exemple, une installation d'Internet par câble téléréseau (CATV) nécessite un SNR de 40 dB. Il faut donc que la puissance du signal en Watts soit $10^4 = 10'000$ fois plus grande que celle du bruit.

2.5 La compression sans perte

2.5.1 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 la répartition statistique locale ou globale de chaque symbole pris indépendamment : compression entropique classique (**Huffman**, Shannon-Fano, etc)
- 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 ou les images.
- en considérant la répartition de sous-chaînes ou de sous-textes, p.ex. **Lempel-Ziv**.

Des méthodes listées ci-dessus, la seule qui n'utilise pas de dépendance entre symboles est la première (suppression uniquement de la redondance intra-symbole, basée sur la statistique des symboles pris indépendamment, donc pour une source sans mémoire) : on l'appelle souvent la compression entropique pure. Les autres techniques présupposent des sources avec mémoire (voir section 2.3.1 en page 9), où l'on supprimera la redondance inter-symboles.

2.5.1.1 Un exemple d'algorithme sans mémoire : Huffman

Avec Huffman, on va supprimer la redondance intra-symbole (liée à la répartition des symboles de la source sans mémoire). L'algorithme (voir figure 2.4 en page 19) consiste à sélectionner les symboles avec les deux plus faibles occurences, par exemple à l'aide d'un tri ou d'une queue de priorité, attribuer respectivement, par convention, les codes 0 et 1 au plus faible et au deuxième plus faible, puis de créer un nouveau symbole composite (additionnant les occurences des symboles ainsi combinés), et de continuer l'algorithme jusqu'à obtenir un symbole-racine unique composé de tous les symboles[19]. A partir de là, on peut lire le code de chaque symbole depuis la racine (dans l'ordre inverse des attributions, soit en descendant l'arbre virtuellement construit de cette racine).

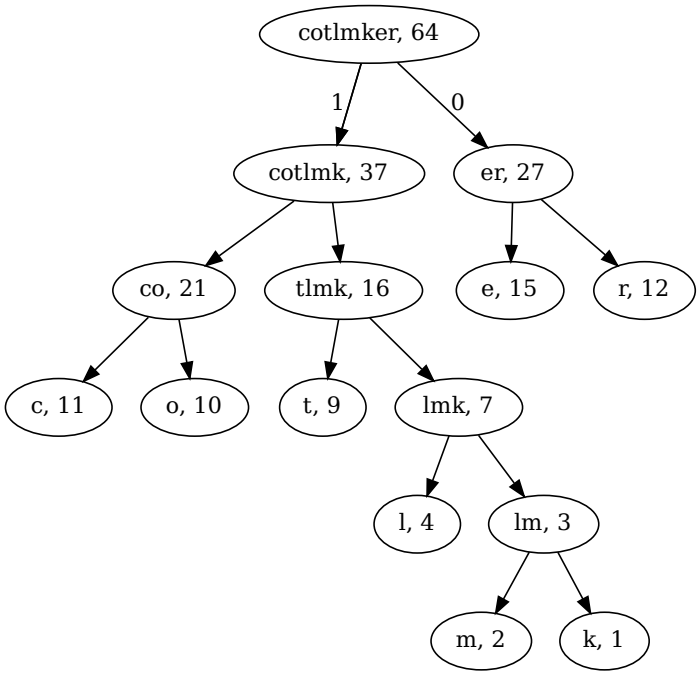
On envoie ensuite la table de compression (symbole ⇒ code), par exemple encodée de manière efficace via le codage canonique de Huffman.

2.5.1.2 Dynamic Huffman

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

La compression Huffman dynamique (aussi appelée **Adaptative Huffman coding**) permet de s'affranchir de l'échange de tables de compression entre l'émetteur et le récepteur. On construit,

19. démonstrateur : https://cmps-people.ok.ubc.ca/ylucet/DS/Huffman.html



symbole	code
e	0 1
r	0 0
c	1 1 1
o	1 1 0
t	1 0 1
l	1 0 0 1
m	1 0 0 0 1
k	1 0 0 0 0

FIGURE 2.4 – Exemple d'arbre de compression de Huffman et code obtenu

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 dont seules les feuilles sont codantes) garantit que l'interprétation des codes est univoque [20]. De plus, l'algorithme ci-dessus garantit la synchronicité de l'émetteur et du récepteur.

20. on parle de **code instantané** lorsqu'aucun code n'est le même que le début d'un autre code

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.

2.5.2 En pratique

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 !

On combine souvent plusieurs techniques : par exemple un prétraitement sous forme d'une permutation de sous-chaînes (block-sorting) améliorant la localité de l'information, suivi d'une compression adaptée aux sources avec mémoire supprimant la redondance inter-symboles (compression de sous-chaîne Lempel-Ziv, différentielle ou RLE), puis enfin une compression *entropique* classique, sans mémoire, par exemple Huffman.

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.

Aujourd'hui, on utilise souvent d'autres algorithmes de compression entropique, comme le **codage arithmétique**, par exemple dans la compression d'images **JPEG2000** utilisée pour la diffusion cinéma, ou des optimisations d'algorithmes visant à un taux de compression proche du codage arithmétique mais en utilisant des machines à états finis de manière efficace : par exemple **zstd** [21].

Le codage arithmétique peut d'ailleurs être modifié pour faire usage de probabilités conditionnelles adaptées au fur et à mesure : des probabilités locales dépendantes d'un modèle de Markov [22] permettront alors d'en faire un algorithme non seulement adaptatif mais optimal [23] également pour les sources avec mémoire.

2.6 La compression avec perte

2.6.1 Introduction

On entend par compression *avec* perte un procédé de compression qui, à la décompression, ne restitue pas l'ensemble des informations du départ : il mène à une perte *irréversible*. L'application principale de la compression avec perte est le stockage et la transmission de données multimédia destinées à restitution humaine (fichiers sons, images et vidéos). La compression obtenue est beaucoup plus importante que celle obtenue par une compression sans perte. Elle s'applique bien sûr sur des sources avec mémoire.

21. Lempel-Ziv 77 + FSE, avec encore du Huffman pour certains types de données : FSE est une évolution optimisée par tables d'une machine à état fini, encodant le résultat dans un entier (tANS). Lorsque les probabilités sont des puissances de 2, c'est la même chose que de code de Huffman. zstd s'approche de l'optimalité concernant le taux de compression obtenu face à la vitesse de compression

22. Markov encode le passé complètement dans l'état actuel au sein du modèle

23. http://www.ee.ui.ac.id/wasp/wp-content/uploads/2011/09/4.-Huffman-and-Arithmetic-Coding.pdf

La compréhension de la manière dont les sens perçoivent les informations est indispensable pour ne supprimer que les éléments difficilement différentiables. Il est aussi important de comprendre que le processus de compression/décompression ajoute du bruit qui peut nuire à la restitution (*artefacts*).

2.6.2 Classes d'algorithmes

2.6.2.1 Algorithmes prédictifs

L'idée est de définir un signal par rapport au passé, sous forme d'un codage de variation du signal (sur peu de bits), en minimisant les erreurs de prédiction ainsi obtenues : par exemple, le codec audio DPCM (*Differential PCM*) encode les différences sur moins de bits que le signal original, au prix de pertes d'information dès que les variations sont suffisamment grandes.

Le **FLAC** étend cette idée en ajoutant de la correction d'erreur permettant de corriger les erreurs produites (au final, il s'agit donc de compression *sans* perte).

2.6.2.2 Algorithmes transformatifs

Ces algorithmes sont basés sur des constatations liés au fonctionnement des sens humains. Par exemple, lors d'un mouvement rapide, la résolution graphique peut être faible. L'oeil ne peut pas distinguer plus qu'un certain nombre de couleurs. L'oreille humaine comporte des schémas de compréhension (modèle psychoacoustique de perception relative des fréquences) et est un filtre passe-bande.

2.6.2.3 Autres algorithmes

D'autres algorithmes existent, par exemple basés sur la similitude d'une image ou de parties d'image à une représentation fractale.

2.6.3 Un exemple : la compression vidéo MPEG/MJPEG

2.6.3.1 Introduction

Le format, ou codec, d'image fixe **JPEG** (*Joint Picture Expert Group*) combine des techniques de compression avec perte (réduction de qualité imperceptible *intra-image*), suivie d'une compression sans perte des éléments produits.

Une première approche pour encoder des images mobiles est de simplement empiler des JPEG à la suite (**MJPEG**, *Motion JPEG*, utilisé par exemple dans le format caméra DV). Le résultat utilise encore beaucoup de place, mais a l'avantage que chaque image est accessible séparément (utile pour l'édition vidéo non linéaire).

L'autre approche est de supprimer, en plus, les redondances *inter-images*[24] : on transmet alors régulièrement des images de type I (indépendantes ou complètes : semblable au MJPEG), et plus souvent de type P (avec compensation de mouvement par rapport à des images décodées dans le passé), ou de type B (prédiction par rapport au passé et/ou à l'avenir, uniquement sur

24. p.ex. avec des transformations géométriques sur des blocs d'une image précédente, prédiction et delta par rapport à des images passées et futures

des images I ou P). Le taux d'image I est un compromis entre efficacité de la compression et temps de récupération en cas d'erreur de transmission non corrigée par ailleurs.

Cette dernière approche, celle de la famille des normes **MPEG**, est particulièrement adaptée à la restitution humaine, et permet de grands taux de compression, d'autant plus que les informations de mouvement superflues pour l'humain peuvent être réduites (perdues).

2.6.3.2 Etapes simplifiées du MPEG

Pas traité en détail cette année

1. sous-échantillonnage du nombre de couleurs : cette diminution se fait tout d'abord en passant de l'espace RGB (niveaux de rouge, vert et bleu) à l'espace de chrominance/luminance, ce qui permet de baisser la résolution en couleur indépendamment de la luminance. Il s'agit d'une perte de qualité, qui ne devrait pas être perceptible par l'oeil humain ;
2. dans certaines applications (vidéophonie), on peut effectuer un sous-échantillonnage temporel (réduction d'images par seconde) et/ou sous-échantillonnage spatial : en fonction des besoin (p.ex. vidéophonie), redimensionnement de l'image ;
3. création de macro-blocs (p.ex. 16x16 pixels) qui seront traités dans la suite de l'algorithme
4. envoi conditionnel : en fonction des images précédentes (dans les macro-blocs), on prédit les macro-blocs futurs : si la prédiction correspond à la réalité dans une certaine approximation, il n'est pas nécessaire d'envoyer les données.
5. DCT [25] : à partir de l'image, on forme des blocs (p.ex. 8 x 8) : sous cette forme il est difficile de compresser : en conséquence, on transforme du domaine temporel en fréquentiel et à partir de cette nouvelle représentation on cherche les coefficients nécessaires pour exprimer ces blocs depuis des images-types ; de plus, si l'image est relativement uniforme, beaucoup de coefficients sont petits et peuvent être négligés (perte d'information dans les détails) ; puis, comme l'oeil humain est plus sensible aux basses fréquences, on considère alors en premier les coefficients des fréquences basses : suivant le tableau des images types DCT cela correspond à faire un zig-zag du coin supérieur gauche au coin inférieur droite. Le résultat d'un bloc de 8 x 8 est codé dans un vecteur de 1 x 64 dont les premiers coefficients concernent les basses fréquences ; ce vecteur a de nombreux zéros et donc est candidat pour une compression **RLE** (voir 2.5.1 en page 18), puis une compression Huffman ;
6. on termine par une compensation de mouvement qui permet de décrire des déplacements de macro-blocs à l'intérieur de l'image.

25. discrete cosinus transformation

Chapitre 3

Le traitement des erreurs de transmission

Sommaire

La **couche physique** n'est jamais totalement fiable [1]. 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 le *codage de voie* : des méthodes permettant de détecter, voire de corriger, des erreurs de transmission.

Ces méthodes sont le plus souvent implémentées en couche **liaison** (2) ou **transport** (4). Les réseaux IP actuellement déployés offrent en général de la détection d'erreur en couche 2 (p.ex. le CRC d'Ethernet) et de la détection et correction d'erreur par retransmission, uniquement si nécessaire à l'application [2], en couche 4 (TCP). Dans certains cas, de la correction d'erreur sans retransmission peut être utilisée en couche 2 (p.ex. Reed-Solomon en xDSL).

1. la disposition des bits, le codage des états p.ex. en **code de Gray**, le choix de la modulation et une éventuelle correction d'erreur en couche 1 (p.ex. convolution, treillis, turbocode, voir section 3.4.2 en page 34) peuvent largement diminuer les erreurs de couche 1 que la couche 2 ou 4 devra traiter.

2. **802.11** (WiFi) est une exception car la norme de base propose des acquittements par défaut obligatoires en couche 2 – que l'on peut désactiver au besoin dans les variantes multimédia de la norme (**WMM** no-ack).

3.1 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 plupart [3] 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 nécessaire pour corriger l'information.

Ces redondances sont basées sur des codes et des propriétés mathématiques.

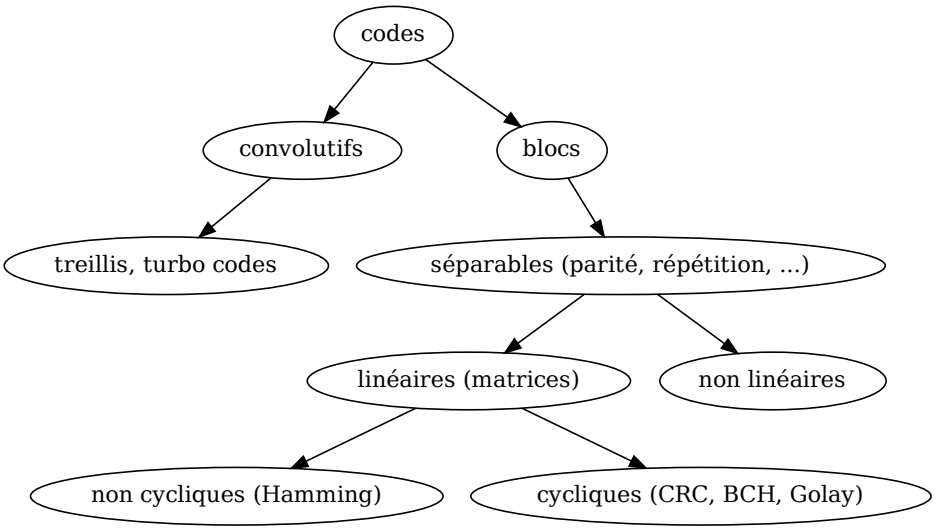


FIGURE 3.1 – Typologie hiérarchique des codes correcteurs ou détecteurs

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 de type **code bloc** : l'information codée est divisible en blocs indépendants de n bits correspondants 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).

3. l'algorithme de détection d'erreur est choisi de manière à détecter les plus probables – voir p.ex. pour les CRCs la section 3.3.2.3.

3.2 Distance de Hamming et conditions de détection et correction

3.2.1 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 [8], 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 :

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

FIGURE 3.2 – Distance de Hamming

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*.

3.2.2 Conditions sur la détection et la correction d'erreur

3.2.2.1 Intuitivement

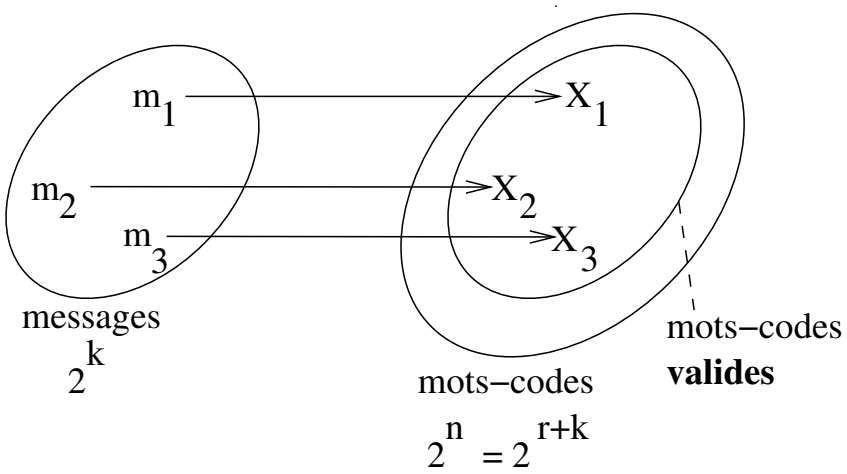


FIGURE 3.3 – Ensemble des 2^k messages de k bits, 2^n mots-codes de n bits et 2^k mots-codes valides de n bits, avec redondance de r bits

Si le passage de l'ensemble des messages possibles à l'ensemble des mots-codes est une injection, par contre, l'ensemble des messages possibles forme une bijection avec l'ensemble des mots-codes *valides* : chaque message possible ne correspond qu'à un mot-code valide, et chaque mot-code valide ne correspond qu'à un seul message.

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).

$$X \rightarrow X_e \text{ (différent de tout Y valide !)}$$

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). Graphiquement, c'est comme si tous les mots-codes valides étaient entourés d'un nuage de points produits par une erreur unique quelconque, dont aucun point n'est un mot-code valide.

Pour **corriger une erreur**, cela ne suffit pas : non seulement toute erreur simple e sur un mot-code valide X produisant un mot-code X_e doit assurer 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.

3.2.2.2 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.

3.2.2.3 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 (deux à deux) minimale de Hamming pour le code ainsi défini sur l'ensemble des mots-codes valides $\mathbf{N}$ et l'on utilise les règles simples suivantes :

— $D_{H_{min}}(\mathbf{N}) \geq (E_d + 1)$ alors on peut *détecter* E_d erreurs simples
— $D_{H_{min}}(\mathbf{N}) \geq (2E_c + 1)$ alors on peut *corriger* E_c erreurs simples

Notons qu'il n'est pas toujours possible de calculer la distance minimale de Hamming en particulier quand l'espace est très grand, mais on peut estimer sa borne minimale.

3.3 Détection d'erreur

3.3.1 Parité

L'exemple classique du bit de parité[4] illustre bien le mécanisme de détection d'erreurs. Par exemple en 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) – voir figure 3.4 en page 27.

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

FIGURE 3.4 – Calcul de toutes les parités paires possibles avec 7 bits de données

En effet, 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. Chaque erreur simple peut donc être détectée par un tel code : sa distance de Hamming est donc au moins de 2.

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).

Il est possible d'arranger intelligement des parités de manière à *corriger* certaines erreurs : voir par exemple la parité croisée ou le code de Hamming (voir section 3.4.1 en page 33).

4. utilisé classiquement par exemple en mode interactif sur des lignes série, sur l'interface SPI (*SCSI Parallel Interface*) ou dans les mémoires de certaines ordinateurs – les protocoles de stockage modernes comme Fibre Channel, SATA ou Firewire utilisent plutôt un CRC permettant de détecter des rafales d'erreur, et les mémoires des ordinateurs de l'**ECC** permettant en plus de corriger des erreurs ; toutefois le RAID5 (et le RAID4) utilisent des parités de bloc – voir section 3.5.4 en page 36.

3.3.2 CRC

3.3.2.1 Introduction

Les choses se présentent différemment lorsqu'il s'agit d'être efficace avec des trames de grande longueur, 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 **hachage** non sûre[5]. Les CRCs sont construits sur des **champs de Galois** qui sont des espaces vectoriels de polynômes [15]. Les mots-codes valides sont construits comme multiples d'un vecteur générateur. Toute perturbation *détectée* par le générateur n'est plus un multiple du générateur.

Soient donc une séquence de m bits à contrôler formant le polynôme $M(x)$ et un **polynôme générateur** $G(x)$ de **degré** r (avec $m >> r$; 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 $G(x)$)

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.

3.3.2.2 Calcul de CRC intuitif

3.3.2.2.1 Introduction On peut assez facilement calculer le CRC d'un message, en se rappelant qu'une division est en fait une suite de soustractions, en particulier en considérant les simplifications pratiques[6] suivantes :

1. les coefficients des polynômes sont binaires (0 ou 1), on travaillera donc sur des bits en colonne, plutôt que des polynômes
2. comme il n'est pas possible par des opérations linéaires (addition ou soustraction) de changer la puissance des termes, il n'y a pas de reports entre les colonnes, on peut donc appliquer la division modulo 2 *sans retenue*
3. comme il n'y a pas de retenue (pas d'impact entre les colonnes de bits), l'addition est la même chose que la soustraction et est en fait un ou exclusif (XOR)

5. les fonctions de hachage utilisées en cryptographie sont par exemple : SHA-x, MD5, etc. Elles ont notamment comme propriétés qu'il est très difficile de retrouver le message original à partir d'un hachage (sauf cas triviaux) et également difficile, à partir du message original, d'exhiber un message modifié qui aurait le même hachage. 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 hachage pour les applications cryptographiques.

6. utilisées par les codes CRC les plus courant en détection.

NB : La méthode pédagogique présentée ici n'est pas celle utilisée par le matériel réseau ou le logiciel, car elle nécessite de stocker le message entier en mémoire et est très fortement simplifiable : voir la section 3.3.2.4 en page 31 et le calcul par tranches.

3.3.2.2.2 Principes Comme la division classique vue à l'École primaire, nous disposons dividende, diviseur, quotient et reste, avec les valeurs à calculer en italique :

dividende	diviseur
	quotient
...	
...	
reste	

Ici, l'objectif est surtout trouver le reste de la division, qui sera ajouté au message, donnant un mot-code parallèle au générateur, car divisible par lui.

Les différences avec la division classique sont qu'on travaille en binaire, et en modulo 2 sans retenue (XOR).

3.3.2.2.3 Exemple concret Soit le message à protéger 11001010 et le polynôme générateur $x^3 + x + 1$ (en binaire : 1011).

Objectif : trouver la redondance à ajouter au message pour que la division du mot-code (message et redondance ajoutée) par le polynôme générateur donne un syndrome (reste) nul.

On constate déjà que notre générateur fait 4 bits significatifs[7]. Cela signifie qu'un reste de la division fait au plus 3 bits, soit un de moins que le nombre de bits significatifs du générateur[8].

En conséquence, on réserve la place pour 3 bits de reste (inconnu, sous forme de X ci-après, à calculer) après le message, car c'est le mot-code (message puis reste) qui doit être divisible par le générateur, sans reste.

Divisons :

```
11001010XXX : 1011
1011          11101
----
01111
 1011
 ----
 01000
  1011            ici on peut diviser, car degrés polynômes égaux
  ----
  001110
   *1011
    ----
    0101XXX
```

7. il ferait aussi 4 bits significatif s'il était 00001011 – les bits à zéro à gauche du 1er 1 sont ignorés.

8. c'est la même chose que de dire que le polynôme du reste ne peut pas avoir un degré supérieur ou égal au polynôme du générateur : si le degré est égal, on peut encore soustraire (diviser) une fois pour supprimer le terme de plus haut degré.

Il faut continuer jusqu'à ce que le nombre de bits significatifs restants soit inférieur à la taille du générateur. Par propriété du ou exclusif (XOR), les XXX valent zéro lorsqu'ils sont utilisés dans un calcul :

```
0101000 : 1011
 1011     (11101)10
 ----
 000100            ici on ne peut plus continuer: reste 100
```

La condition d'arrêt de la division est quand le reste a un degré plus petit que le générateur (ou un nombre de bits significatifs inférieur).

Par propriété du ou exclusif, ici les XXX valent 100 pour que le syndrome soit nul.

On transmet alors le mot-code (message complété du reste calculé) :

11001010	100
message	reste

3.3.2.2.4 Vérification Le récepteur va vérifier le mot-code entier en le divisant par le même générateur :

```
11001010100 : 1011
1011          11101100
----
 1111
 1011
 ----
 01000
  1011
  ----
  001110
   *1011
    ----
    0101100
     1011
     ----
     000000
```

Comme ici le reste est nul (le syndrome est nul), il n'y a pas d'erreur *détectée*.

3.3.2.3 Erreurs détectées

En cas d'erreur(s), 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 sont des polynômes facteurs de $G(x)$ ne sont pas détectées[9] C'est cette remarque qui est à la base du choix du polynôme $G(x)$, qui est donc capital : le générateur ne doit pas, en lui-même, former une suite probable de bits en erreurs. Mais il y a d'autres critères à remplir, qui sont exposés ci-après.

9. le message erroné se décompose en deux composantes multiples du générateur, la division donne donc un reste nul.

3.3.2.3.1 Erreur simple Ici l'erreur est représentée par un seul bit faux : $E(x) = x^i$; donc si le générateur $G(x)$ comprend au moins 2 termes (2 bits), alors il n'est pas diviseur de $E(x)$ [10].

3.3.2.3.2 Erreur double isolée L'erreur fait deux bits quelconques, mais différents, que l'on peut exprimer comme $E(x) = x^i + x^j$, or ceci peut aussi s'écrire par le multiple de deux termes $x^j(x^{i-j} + 1)$: alors 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 $k < 15$

3.3.2.3.3 Paquet d'erreurs de longueur k Soit des erreurs en rafale $E(x) = x^i(x^{k-1} + ... + 1)$: 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 $\frac{1}{2r-1}$.

3.3.2.3.4 Erreurs en nombre impair S'il y a un nombre impair d'erreur, alors $E(x)$ 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) 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 [11] !

3.3.2.4 Applications logicielles 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** d'un CRC, où l'on calcule le CRC par morceaux par une méthode itérative facilement implémentable en logiciel (avec une table précalculée avec une implémentation selon la méthode intuitive vue précédemment) ou matériel.

3.4 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
```

10. par exemple, la parité simple est le CRC $x + 1$ (CRC-1) et peut détecter une erreur simple
11. mais pas optimales, ce qui a été découvert ensuite, voir https://users.ece.cmu.edu/~koopman/crc/ pour une liste d'algorithmes CRC en fonction des besoins

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

Supposons que l'on reçoive 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 et peut même *aggraver* l'erreur.

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 n'y a aucune certitude pour des codes de longueur supérieure à 9 avec une distance minimale de 3.

3.4.1 Code correcteur de Hamming

Le code de Hamming a pour caractéristiques de pouvoir corriger une erreur (1-correcteur, distance de Hamming $D_{H_{min}} = 3$, donc aussi 2-détecteur). 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.

On note un code de Hamming comme suit : $H(n, k)$, avec $r = n - k$ (bits de redondance ou ici de 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'erreurs détectées, 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 fausse peut survenir, mais que la détection est toujours possible : avec plus de 2 erreurs, l'erreur pourrait ne pas être détectée [12].

12. il est possible d'améliorer la discrimination de ces différents cas en ajoutant une simple parité.

3.4.2 La correction d'erreur en pratique

On constate que la puissance proposée par le code de Hamming ci-dessus est trop faible[13] pour les applications modernes : d'autres codes permettant de corriger plus d'erreurs sont alors nécessaires : par exemple **Reed-Solomon**. Mais même eux peuvent ne pas suffire s'il y a trop d'erreurs : une technique possible – utilisée dans le mode **interleave** du **xDSL** par exemple – est de répartir les bits transmis : une seule rafale localisée qui serait suffisante pour dépasser la puissance du code est alors répartie sur plusieurs blocs corrigés indépendamment, sans dépasser la puissance du code correcteur. L'inconvénient est toutefois le délai créé par cette méthode.

Pour *détecter*[14] les éventuelles erreurs résiduelles après correction d'erreur, un CRC peut être utilisé.

En règle générale, les systèmes modernes de correction d'erreurs sont de type **convolutif** (en couche 1, travaillant sur les bits), comme p.ex le **treillis** des modems modernes ou les **turbo-codes** du LTE ou du DVB-S et DVB-T ; et/ou de type **bloc** (**Reed-Solomon**, **Golay**, etc) en couche 2. En couche 4, vu la correction et/ou détection d'erreur déjà effectuées en couche 1 et 2, on se borne souvent à une détection d'erreur résiduelle (checksum de TCP et UDP) et l'on applique des protocoles fiables, si demandé par l'application, pour retransmettre les blocs manquants (voir chapitre suivant).

On ajoute toutefois de la détection d'erreur plus solide dans certains protocoles de couches supérieures, comme par exemple l'intégrité d'un fichier à l'aide d'un hachage cryptographique (résistant aux attaques), ou des signatures électroniques (voir cours Cryptographie et Sécurité de 3^e année).

13. un mm^2 sur un CD correspond déjà à 1 Mbit !

14. il faut aussi noter qu'en choisissant bien le polynôme générateur du CRC, il est possible également de corriger jusqu'à $\frac{r}{2}$ erreurs.

3.5 Application à la haute disponibilité

3.5.1 Introduction

La haute disponibilité vise à ajouter de la redondance dans les systèmes, procédures et logiciels de manière à atteindre une meilleure **disponibilité**, ce qui ici implique également une certaine garantie sur l'**intégrité** des données *de bout en bout* et dans le stockage. Cette section a pour but de présenter quelques unes de ces techniques.

3.5.2 Redondance du matériel et des chemins

L'idée est d'avoir plusieurs unités redondantes (de stockage par exemple), accédées de préférence par plusieurs chemins différents. La panne d'un équipement (disque-dur, contrôleur de stockage, carte réseau, etc) ne devrait pas empêcher l'accès aux données. Le **multipath** et le RAID (voir section 3.5.4 en page 36) sont des techniques de ce type.

3.5.3 Intégrité des données de bout en bout

Ici, on entend par intégrité de bout en bout la garantie d'intégrité des données entre une application et son backend de stockage. Elle utilise notamment des techniques de détection et de correction d'erreur par redondance du chapitre, soient :

lors du stockage en RAM détecter, voire corriger les erreurs en RAM grâce à l'utilisation de RAM redondante et de circuits d'ECC

lors du transport détecter les erreurs par simple détection CRC sur les bus des différentes technologies le supportant (SATA, SAS, FC par exemple)

lors de la consultation ultérieure d'un média détecter, voire corriger, les erreurs du média lui-même grâce à l'ECC des disques-dur [15] et détecter les erreurs de méta-données ou encore mieux de données dans les systèmes de fichiers [16] ou les volumes logiques [17] grâce à des hachages ou des CRC – ou à le faire sous forme de hachages de fichiers dans l'application

Lorsque les techniques ne permettent que la détection d'erreur, il y aura retransmission lors du transport. Dans les 2 autres cas ci-dessus, soit la redondance doit être suffisante pour corriger les erreurs à une très grande fiabilité (RAM ECC), soit les données doivent être stockées de manière redondante elles-mêmes (p.ex. RAID, voir section suivante).

L'intégrité de bout en bout [18] est un sujet important pour les applications actuelles dont les données changent sans cesse. Il existe une approche standardisée : le standard T10 PI (Protection Information), implémenté comme une commande SCSI **DIF/DIX/EEDP** [19] et qui utilise 8 octets supplémentaires d'intégrité : on crée par exemple un CRC sur les données dans la RAM de l'ordinateur, que l'on stocke ensuite directement après chaque bloc de données de 512 (respectivement 4096) octets sur le périphérique. Ceci permet à la relecture de vérifier l'intégrité de bout en bout des données stockées.

15. p.ex. secteurs de données de 512 octets, suivi d'un **ECC** de 24 octets (détection et correction)
16. `btrfs`, `zfs` – `ext4` ne proposant que l'intégrité des méta-données
17. `dm-integrity` de `LVM`, ce qui permet ensuite d'y faire résider un système de fichiers qui détectera ainsi la corruption, ou même du RAID qui assurera la disponibilité des données
18. https://oss.oracle.com/~mkp/docs/lpc08-data-integrity.pdf
19. https://www.snia.org/sites/default/files/SDCEMEA/2020/3%20-%20Mikhail%20Malygin%20Yadro%20-%20Using%20Linux%20block%20integrity.pdf

En conclusion, la combinaison de méthodes assurant la redondance du matériel et des chemins et l'intégrité permet de prévenir les pertes de données, et surtout, la **corruption silencieuse** de données.

3.5.4 RAID

3.5.4.1 Introduction

Le **RAID** (*Redundant Array of Inexpensive/Independant Disks*) est une technique de stockage qui vise à arranger des parties d'unités de stockage (disques) de manière à répartir les données pour des bénéfices de performance ou de **HA** (*High Availability*, haute disponibilité ou fiabilité).

Les systèmes RAID se basent sur la supposition d'une intégrité de bout en bout, ou au minimum une détection de secteurs corrompus (voir section 3.5.3 en page 35) [20].

Le RAID ne remplace pas une bonne sauvegarde (en particulier en RAID 0), testée, ni un contrôle d'intégrité de bout-en-bout des données.

Les niveaux de RAID strictement *supérieurs à 0* peuvent protéger contre des pannes d'unités de stockage entières ou des pannes de secteurs individuels – tant qu'il reste suffisamment de redondance pour pouvoir retrouver les données. Les algorithmes utilisées vont de la simple copie multiple (RAID1, RAID10), à la parité par ou exclusif (**XOR**) pour les niveaux 3 à 5, jusqu'aux espaces-polynômes de Galois (niveau 6, permettant de résister à plus d'une erreur [21]).

On parle de *mode dégradé* lorsque l'ensemble RAID utilise déjà ses possibilités internes de correction et que peut-être une erreur supplémentaire empêchera l'accès aux données.

La performance maximale que l'on peut obtenir en RAID peut excéder la performance de chaque disque individuel : tout dépend du niveau de RAID considéré et de son arrangement, mais bien sûr aussi du débit total traversant du système (bus, mémoire).

3.5.4.2 Niveaux de RAID

Les niveaux de RAID sont détaillés ci-après. Le RAID0 – sans aucune redondance mais avec un avantage de performance – est aussi appelé *striping* et le RAID1 *mirroring*.

En pratique aujourd'hui, les systèmes d'exploitation et les contrôleurs proposent des variantes hybrides qui ne sont pas limitées à considérer des unités de stockage (disques) dans leur ensemble : les systèmes comme le RAID10 peuvent alors survivre à plusieurs erreurs simultanées sur plusieurs disques tant qu'elles ne concernent pas des blocs identiques. L'arrangement de disques de taille différente est également possible.

Le RAID10 est alors recommandé pour les applications multi-thread en lecture et si l'on a le budget ; dans le cas contraire du RAID5, du RAID6 ou des combinaisons (RAID50, RAID60) seront privilégiées.

3.5.4.3 Comparaison des types de RAID

Dans la figure 3.5 en page 37 ci-après, la colonne fiabilité indique le nombre de panne(s) n'aboutissant pas à une perte de données. Les coûts correspondent à un facteur par rapport aux données utiles. La performance est séquentielle, mono-thread. La performance multi-thread

20. attention donc à certaines optimisations qui consistent à réduire les possibilités de correction, voire de détection, des disques pour garantir des délais faibles

21. http://ftp.nluug.nl/ftp/ftp/os/Linux/system/kernel/people/hpa/raid6.pdf

est pour la lecture seulement. N représente le nombre de disques totaux ; W l'écriture et R la lecture.

niveau	fiabilité	performance	multithread	coût	principes
0	aucune ($\frac{1}{N}$)	N	non	1	données réparties par bandes sur les différents disques, accédées en parallèle (**striping**)
1	$N-1$	W ≤ 1, R $= 1$	N	N	écriture sur N disques, lecture d'un disque : tous les disques sauf un peuvent tomber en panne (**mirroring**)
10	$\frac{N-1}{2}$	W ≤ 2, R $= 2$	$\frac{N}{2}$	$\frac{N}{2}$	combinaison de RAID0 et RAID1, peut varier en fonction de l'arrangement : RAID01 ou RAID10 ou variante logicielle flexible
4	1	$N-1$	non	$\frac{N}{N-1}$	comme du RAID0, en plus chaque écriture provoque le calcul d'une parité (XOR) entre les groupes de données écrits séparément, et le résultat est écrit sur un disque dédié à la parité : la performance est celle du RAID0, avec la résistance à un disque en panne ; le mode dégradé nécessité la reconstruction du disque ou bloc absent en lisant les $N-2$ disques puis le disque de parité ; *goulet d'étranglement* : la performance en mode normal ou dégradé est liée à celle du disque de parité ; cycle *RMW* nécessaire si données à écrire trop courtes ou non alignées
5	1	$N-1$	non	$\frac{N}{N-1}$	comme du RAID4, avantage : plus de *goulet d'étranglement* car plus de disque de parité dédié : chacun des disques a ce rôle pour une partie des données.
6	2	$N-1$	non	$\frac{N}{N-1}$	calcul plus lent qu'en RAID5, mais meilleure résistance aux pannes.

FIGURE 3.5 – Comparaison de types de RAID

3.5.4.4 Perte de performance par RMW et/ou désalignement

L'effet **RMW** (*Read-Modify-Write*) est présent pour les niveaux de RAID supérieurs à 1 lorsque les nouvelles données ne remplissent pas entièrement un bloc logique du RAID et donc qu'il faut lire les données manquantes, appliquer la fonction de redondance et écrire la redondance, ce qui est très coûteux. Le coût du *Read-Modify-Write* peut être important si les systèmes de fichiers ne sont pas alignés aux blocs logiques du RAID ou que les unités d'entrées-sortie (facteur de blocage) diffèrent.

Ce type de problème se pose aussi avec les disques-dur à 4096 octets par secteur lorsqu'ils sont présentés en émulation 512 octets/secteur au système d'exploitation, ou pour les partitions des machines virtuelles[22]. La solution générale est d'assurer l'alignement et d'utiliser du cache (dans l'OS ou sur un contrôleur dédié, avec batterie) pour grouper les requêtes. A défaut et en particulier avec les disques SSD, une fatigue importante des cellules peut être créée (*wear-out*).

3.5.4.5 Types de panne et atomicité

Les OS modernes font tout pour garantir l'ordre des écritures et maintiennent des états persistants, même en présence de caches gérés par l'OS : par exemple, la journalisation permet d'assurer que les structures de données restent cohérentes même en cas de pannes franches aboutissant à ne pas stocker une partie des données de manière persistante.

Toutefois, certains systèmes de fichiers ne vont pas, par défaut, journaliser également les données pour des raisons de performance (exemple : `ext4`). Dans ce cas, des données récentes peuvent être perdues et remplacées par des séquences de NULs à la lecture.

Pour éviter ce genre de problèmes tout en conservant de la performance, diverses solutions sont appliquées à différents niveaux, par exemple des batteries pour les contrôleurs RAID disposant d'un cache non contrôlable par l'OS, ou des bitmaps permettant à `dm-integrity` de savoir quels blocs ont été hashés de manière persistante.

3.5.4.6 En pratique

Un disque supplémentaire de réserve (*hot standby*) permet une reconstruction pour passer assez rapidement à nouveau en mode normal après un passage en mode dégradé.

Le postulat de base que les disques sont *indépendants* n'est pas forcément vrai : des disques achetés ensemble (même fabricant, même modèle) et utilisés de la même manière peuvent bien tomber en panne simultanément, ou consécutivement à la charge supplémentaire induite par un mode dégradé, en particulier lors d'une reconstruction.

On peut résister à plus de pannes indiquées par le tableau en figure 3.5 en page 37 dans la mesure où les secteurs des disques en pannes sont différents (le tableau suppose des pannes totales, ce qui est rarement le cas). Les blocs en erreur peuvent d'ailleurs être reconstruits par parité puis ré-écrits, sans nécessiter de passer en mode dégradé : on recommandera un automatisme de relecture régulière des données.

Sans multipath ou d'autres types de redondances de chemin, la panne du contrôleur RAID va mener à l'indisponibilité des données.

22. étude pile de virtualisation NAS avec RMW : https://pdfs.semanticscholar.org/46e3/2ec7c96d200da62647d0c1ecfb0addd98642.pdf – notons également que les disques utilisant la technologie disque récente **SMR**[23] vont également exhiber des problèmes de performance similaire au RMW, sans ne pouvoir rien y faire et donc ne sont pas adaptés à des écritures aléatoires.

Chapitre 4

Protocoles fiables (protocoles à fenêtre)

Sommaire

Les couches de liaison (2) et de transport (4) ont comme tâche d'assurer une transmission fiable [1] 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** prénégociée (couche 2, voire 4).

Le rôle des protocoles assurant une transmission fiable 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 contrôle [2].

1. ou sûre, au sens de fiabilité
2. 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 4.2.1 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.

4.1 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 évidemment 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 [3] 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 : il doit décider s'il a reçu une répétition du dernier message ou un nouveau message.

On peut utiliser un bit qui est inversé à 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.

3. si cette numérotation revient souvent à sa valeur initiale, il faut peut-être tenir compte, en couche 4, des réinjections tardives dues au réseau, voir section 4.4.4.3.4 en page 52 ; en couche 2 on suppose en général que cela ne va pas se produire.

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).

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 délais de transmission à travers un réseau 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.

Le produit entre débit et délai[4] est un indicateur du nombre de données que l'on pourrait théoriquement faire circuler sur le réseau entre deux machines, et plus il est grand, moins Idle RQ sera performant. Dans ce cas là, des protocoles à fenêtre, vus dans la prochaine section, seront plus adaptés.

4.2 Continuous Request (Continuous RQ)

4.2.1 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** et est notée k[5].

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 évidemment 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 le rendement du protocole.

En plus de TCP, on peut citer **BBR** comme protocoles à fenêtre moderne, et **Z-Modem** et **UUCP** comme protocoles à fenêtre historiques.

4. en anglais, on utilise souvent le concept de *bandwidth-delay product*, qui est impropre
5. une taille $k = 1$ est le cas dégénéré Idle Request

4.2.2 Nombre de numéros de séquence

Optimiser le nombre de numéros de séquence au strict nécessaire est utile pour limiter la taille des entêtes (et donc améliorer l'efficacité intrinsèque, voir section 4.4.2 en page 44).

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	$2k$

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.

Il est évident que si le réseau peut réinjecter des trames en-dehors de la fenêtre courante, d'autres précautions doivent être prises (voir section 4.4.4.3.4 en page 52).

4.2.3 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.

4.3 Un exemple : HDLC (résumé)

Pas traité en détail cette année

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) [6]
— 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 :

Flag **01111110**	**Adresse**	**Contrôle**	**Données**	**FCS**	**Flag** **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**). Quelque soit la taille de fenêtre (dans les limites acceptables par le maximum de numéros de séquence), la numérotation est toujours 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.

6. 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 4.2.2 en page 42.

4.4 Rendement des protocoles

4.4.1 Introduction

La performance d'un protocole dépend notamment du débit, mais également du rapport entre débit utile ou net (celui que l'on obtient véritablement pour son application) et débit total ou brut (celui qu'on paie au fournisseur), ce qu'on appelle aussi l'**efficacité**[7]

Augmenter le débit payé n'est pas toujours une bonne solution pour obtenir une performance suffisante : on a souvent intérêt à utiliser le bon protocole avec des échanges optimisés. En effet, si le protocole est mauvais, doubler le débit brut payé ne va pas augmenter le débit utile pour l'application de manière significative. Chaque franc investi en plus ne le sera donc pas de manière rentable.

De plus, lorsque l'on dimensionne des liaisons pour des protocoles à données de faible taille (p.ex. pour la voix-sur-IP), le rendement intrinsèque (poids respectif des entêtes par rapport à des données audio de petite taille) peut être déterminant dans le rendement du protocole, et donc dans le coût de la liaison.

Le rendement des protocoles est donc la combinaison du calcul du rendement intrinsèque (structure) et du rendement des échanges (notamment délai et tailles des échanges, s'applique en particulier aux protocoles fiables).

4.4.2 Rendement intrinsèque

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

Il se définit comme :

$$U_{intr} = \frac{l_{utile}}{l_{totale}} \tag{4.1}$$

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).

Par exemple, si l'on envoie des datagrammes voix-sur-IP RTP sur UDP sur IP sur Ethernet, on peut calculer que les entêtes cumulés font 58 octets. De plus, pour éviter de mettre en tampon trop de données audio (et donc éviter des trop grands délais néfastes à l'interactivité de la téléphonie sur Internet), on se limitera à 20 ms d'accumulation avant envoi (bon compromis entre rendement et délai). Donc, en **G.711**, 20 ms correspond à 160 octets. On en déduit donc un $U_{intr} = \frac{160}{160+58} = 73\%$. La conséquence est qu'une communication à 64 kbit/s (net) nécessite en fait un peu moins de 88 kbit/s (brut).

7. l'efficacité est une mesure de la performance ces protocoles : ici, nous parlerons d'une grandeur sans unité, le **rendement**, soit une valeur entre 0 et 1, ou exprimable en pourcents.

4.4.3 Rendement des échanges

4.4.3.1 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, $\frac{bit}{\frac{bit}{s}} = s$)
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 :

$$U = \frac{T_{utile}}{T_{total}} \tag{4.2}$$

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 :

$$U = \frac{T_{ix}}{T_{ix} + 2T_p} = \frac{1}{1 + 2a} \tag{4.3}$$

avec

$$a = \frac{T_p}{T_{ix}} \tag{4.4}$$

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.

4.4.3.2 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 :

$$U = \frac{k}{1+2a} \tag{4.5}$$

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

4.4.3.3 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 plusieurs[8] 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 est la probabilité inverse à celle qu'il n'y ait aucune erreur sur tout les bits du message ($(1-P)^N$) :

$$P_f = 1 - (1-P)^N \tag{4.6}$$

Intuitivement, il est évident que si la moitié des messages sont en erreur (50%), il y aura en moyenne 2 transmissions pour chaque message à faire passer et on divisera donc le rendement par 2. Si un message sur 8 est en erreur (12.5%), on peut considérer que le débit résultant sera $\frac{7}{8}$ du débit initial. Il faut donc multiplier par $1 - P_f$.

Pour se convaincre mathématiquement de la justesse de ce raisonnement intuitif, on peut définir l'espérance de transmission (le nombre de transmission moyen) comme une moyenne pondérée :

$$E = \lim_{j\to\infty} \sum_{i=1}^{j} i\ (1-P_f)\ {P_f}^{i-1} = \frac{1}{1-P_f} \tag{4.7}$$

on en déduit que

$$U_{retransmission} = \frac{U}{E} \tag{4.8}$$

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

Les rendements sont alors les suivants :

Idle RQ :

$$U = \frac{1 - P_f}{1 + 2a} \tag{4.9}$$

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

Continuous RQ, selective repeat :

$k < 1 + 2a$:

$$U = \frac{k(1 - P_f)}{1 + 2a} \tag{4.10}$$

$k \geq 1 + 2a$:

$$U = 1 - P_f \tag{4.11}$$

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

Continuous RQ, go-back-N :

$k < 1 + 2a$:

$$U = \frac{k(1 - P_f)}{(1 + 2a)(1 + P_f(k - 1))} \tag{4.12}$$

$k \geq 1 + 2a$:

$$U = \frac{1 - P_f}{1 + P_f(k - 1)} \tag{4.13}$$

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.

4.4.4 Application à TCP

4.4.4.1 Formule approximative d'évaluation de débit maximum

Une formule approximative que l'on voit souvent pour le calcul du débit maximum de TCP, sans pertes de paquets, est la suivante :

$$\dot{D}_{utile} \leq \frac{window\ size}{RTT} \tag{4.14}$$

avec $window\ size$ la taille de fenêtre en octets, $\dot{D}_{utile}$ en octet/s et RTT délai aller-retour total (y compris temps de transmission, deux temps de propagation, etc) en secondes.

4.4.4.1.1 Utilité de la formule approximative Il s'agit d'une formule approximative, qui était plutôt utile lorsque la taille de fenêtre était limitée (65535 octets). Avec le window scaling (voir 4.4.4.3.2 en page 50), elle reste utile pour déterminer sur une capture réseau si un problème de débit rencontré est lié ou non au window scaling négocié ou à d'autres facteurs. Or, dans le cours Protocoles et Réseaux on s'intéresse plutôt au problème inverse : dimensionner un protocole ou calculer des efficacités. Elle donne d'ailleurs un maximum sans tenir compte des pertes de paquets ni d'autres aspects de TCP (voir paragraphe 4.4.4.1.5 en page 49).

4.4.4.1.2 Exemple d'application de la formule approximative Soit une taille de fenêtre 65535 et un délai RTT 17 ms. La formule approximative donne un débit maximum de 3.855 Moctet/s.

4.4.4.1.3 Démonstration de la formule approximative Déjà, l'analyse dimensionnelle est correcte : $\frac{octet}{s}$. Ensuite, en supposant l'absence de pertes et un protocole Continuous Request inefficace (car limité en débit), on a : $U = \frac{k}{1+2a}$ et $\dot{D}_{utile} = \dot{D}U$.

On peut donc écrire :

$$\dot{D}_{utile} = \dot{D}\frac{k}{1+2a} \tag{4.15}$$

Mais on peut se rappeler la forme originale[9] du U, ou simplement multiplier en haut et en bas par un T_{ix}, temps de transmission d'un message N, supposé non nul : $\dot{D}_{utile} = \dot{D}\frac{kT_{ix}}{T_{ix}+2aT_{ix}}$. Comme $a = \frac{T_p}{T_{ix}}$, alors $\dot{D}_{utile} = \dot{D}\frac{kT_{ix}}{T_{ix}+2T_p}$. Il nous reste à se rappeler que $T_{ix} = \frac{N}{\dot{D}}$, donc : $\dot{D}_{utile} = \dot{D}\frac{k\frac{N}{\dot{D}}}{\frac{N}{\dot{D}}+2T_p}$, ce qui devient, en simplifiant le $\dot{D}$, $\dot{D}_{utile} = \frac{kN}{\frac{N}{\dot{D}}+2T_p}$

Si l'on pose window size $= kN$ et RTT $= \frac{N}{\dot{D}} + 2T_p$, on peut démontrer la formule. Avec N exprimé dans la même unité que la taille de fenêtre, c'est trivial.

Il faut toutefois affiner la définition du RTT, le *return-trip-time*, qui n'est pas ici deux fois le temps de propagation T_p. Il est plus proche du résultat de la commande ping[10] : celle-ci donne en fait la somme du temps de transmission du paquet ICMP, le délai aller-retour et le

9. qui contenait déjà des approximations en négligeant certains temps !
10. on peut toutefois évaluer le temps de propagation pur en faisant la différence du résultat de deux commandes ping utilisant des tailles de paquets différentes

temps de transmission du paquet ICMP en retour, au minimum. Le paquet aller (ICMP ECHO REQUEST) et le paquet retour (ICMP ECHO REPLY) pouvant être de la même taille, ou alors et dans certains cas, de taille inférieure.

Au maximum, ping vaut $2T_{ix} + 2T_p = 2\frac{N}{D} + 2Tp$. Pour une vitesse de transmission élevée ou un paquet de réponse bien plus petit que le paquet original, ce n'est pas si éloigné de l'expression $\frac{N}{D} + 2T_p$.

4.4.4.1.4 Comparaison avec la formule du cours Reprenons le cas pratique ci-dessus pour nous convaincre : taille de fenêtre 65535, délai RTT 17 ms. La formule approximative donne un débit maximum de 3.855 Moctet/s et la formule du cours montre une asymptote autour de 3.82 Moctet/s quelque soit le débit acheté.

Autrement dit, $\frac{N}{D}$ est négligeable par rapport à $2Tp$, comme 1 est négligeable par rapport à $2a$ pour de grandes valeurs de k (ce qui est encore plus le cas avec window scaling).

La formule approximative donne donc une borne supérieure au débit utile maximum, en faisant attention aux unités.

4.4.4.1.5 Et avec des pertes de paquets ? Dans ce cas, le modèle TCP de MATHIS [11], qui tient compte des erreurs et de certaines caractéristiques d'implémentation de TCP (notamment l'algorithme de *congestion avoidance*) peut être utilisé à la place des formules de ce chapitre :

$$\dot{D}_{utile} \leq \frac{MSS}{RTT} \cdot \frac{C}{P} \tag{4.16}$$

avec C souvent fixé à 1 (dépend de paramètres du modèle de Mathis), MSS étant la taille utile de paquet (sans les entêtes), RTT le délai aller-retour total et P le taux de perte de paquets.

4.4.4.2 Produit débit * délai

Le produit débit * délai, aussi appelé *bandwidth delay product* dans la littérature est particulièrement important pour les *Long Fat Networks pipes,* **LFN***s*, autrement dit pour les liaisons avec un grand débit et un grand délai – Internet dans la plupart des cas aujourd'hui. En effet, intuitivement, le canal ainsi modélisé peut contenir énormément de données : il faut absolument une grande taille de fenêtre en octet pour y être efficace et TCP classique n'y est plus efficace.

4.4.4.3 Application aux améliorations de TCP

4.4.4.3.1 Introduction TCP est un protocole très performant, qui est resté bon malgré les nombreux changements dans son environnement. Cependant, dans les réseaux actuels, en raison de matériel plus performant [12], des grands débits offerts par les réseaux (fibre optique, corollaire parfois aussi augmentation du MTU), de leurs caractéristiques en surcharge et des applications particulières, les limites du protocole TCP *original* sont atteintes :

11. https://www.thousandeyes.com/blog/a-very-simple-model-for-tcp-throughput et exemples pratiques dans https://www.switch.ch/network/tools/tcp_throughput/

12. le débit augmente, en particulier dans les **datacenter**s et les ordinateurs disposent de mémoire de grande taille

1. le produit `débit * délai` (voir section 4.4.4.2 en page 49) augmente de plus en plus, ce qui a comme conséquence que la taille de fenêtre maximum de TCP de 65'535 octets devient insuffisante pour être efficace (remplir le canal devient impossible)
2. en cas d'erreur de transmission, l'utilisation de la variante `go-back-N` implique des retransmissions inutiles : sur un WAN, ces erreurs sont dues, très souvent, à la **congestion**, qui s'aggrave si des retransmissions inutiles sont effectuées – surtout que plusieurs connexions TCP se font compétition
3. aucun support de détection de congestion n'était prévu dans IP : elle est détectée par TCP après coup en fonction des pertes de données, et via un algorithme heuristique (sans information sur la capacité réellement disponible)

C'est pour cela que des améliorations du protocole TCP ont été proposées [13], puis mises en pratique.

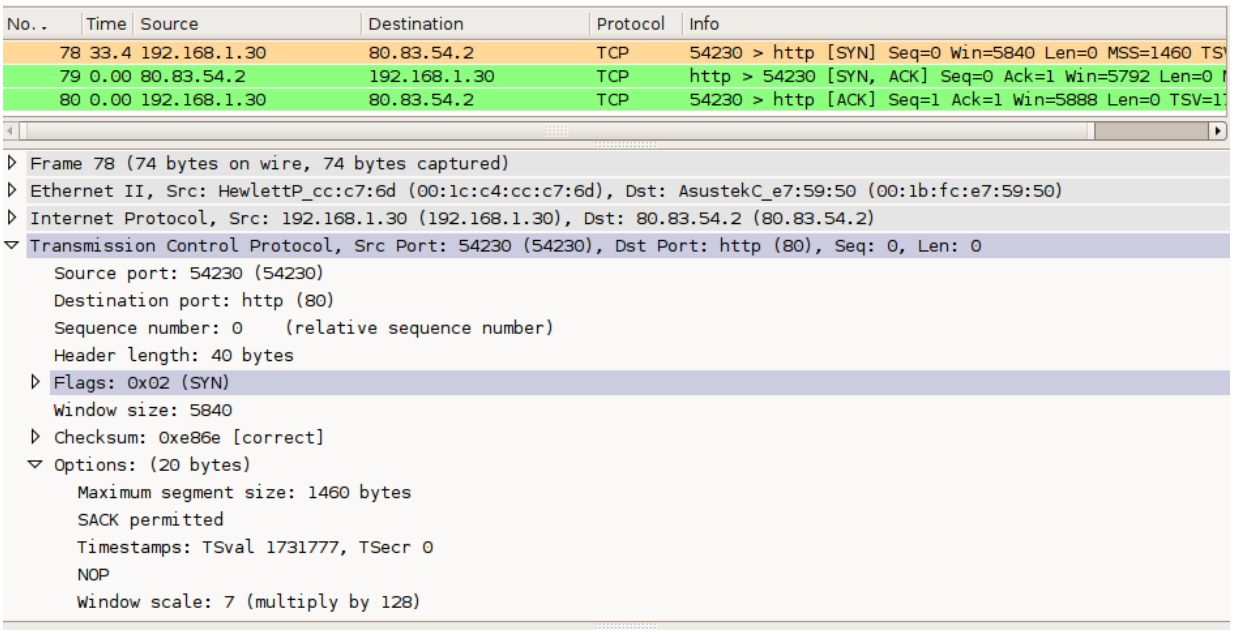


FIGURE 4.1 – Nouvelles options TCP

4.4.4.3.2 Amélioration de la performance : window-scaling

De manière à augmenter le volume de données envoyé sans confirmation (taille de fenêtre) pour atteindre le produit `débit * délai`, l'option **window-scale** permet de multiplier la taille de fenêtre [14]. Cette option TCP est utilisée uniquement dans un segment SYN (ouverture) et doit être confirmée, comme toute option, dans la réponse SYN/ACK (confirmation ouverture) pour être valable, et fixera le facteur d'échelle à appliquer durant toute la session, sans causer de perte d'efficacité intrinsèque durant l'échange de données.

L'exposant d'échelle maximum est défini dans le RFC-7323 à 14, soit un facteur de $2^{14} = 16384$, ce qui combiné avec le champ de taille de fenêtre maximum donne environ 1 Go de taille de fenêtre maximum effective [15].

Exemple : supposons un envoi de données au sein d'un LAN Ethernet, avec un MTU de 1518 octets (couche 2 y.c. entêtes Ethernet). Il reste donc à IP 1500 octets, dont 40 sont occupés,

13. voir p.ex. le RFC-7323
14. champ fenêtre de 16 bits de l'entête TCP, maximum $2^{16} - 1 = 65535$ octets
15. en pratique, c'est la taille du tampon de réception du kernel qui compte, par exemple sous Linux : http://www.acc.umu.se/~maswan/linux-netperf.txt

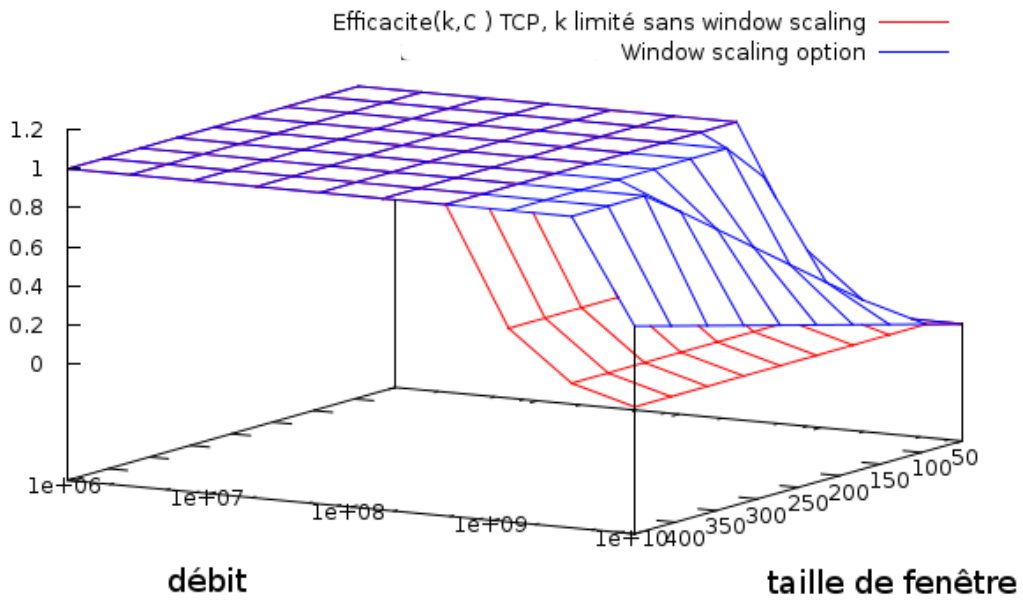


FIGURE 4.2 – Efficacité avec ou sans window scaling

au minimum, par les entêtes IP et TCP. Les données utiles sont donc, au maximum, $N = 1460$ octets $= 11680$ bits. Dans le cas de GBit Ethernet, $C = 1\ \frac{GBit}{s}$ et $Tp = 500\ \mu s$. On a bien entendu : $T_{ix} = \frac{N}{C} = \frac{1460 \cdot 8}{1 \cdot 10^9} = 11.68\ \mu s$ et $a = \frac{T_p}{T_{ix}} = \frac{500 \cdot 10^{-6}}{11.68 \cdot 10^{-6}} = 42.81$.

Donc, deux cas à considérer : (voir section 4.4.3.2 en page 46)

1. TCP classique (sans *window scaling*, $w_{max} = 2^{16} - 1$) : $k_{max} = \lfloor \frac{w_{max}}{N} \rfloor = \lfloor \frac{65535}{1460} \rfloor = 44$ (arrondi à l'entier inférieur), c'est le facteur limitant : dans ce cas, comme $k < 1 + 2a$, le protocole ne sera pas efficace et la formule $U = \frac{k}{1+2a}$ est à utiliser.
2. TCP avec *window scaling* actif : on peut augmenter la taille de fenêtre au-delà de k_{max} car on n'est plus limité à la taille de fenêtre w_{max} : il faut choisir k pour que $k \geq 1 + 2a$; il suffit donc de prendre un facteur de window scaling de 2 et donc $k = \lfloor \frac{2w_{max}}{N} \rfloor = \lfloor \frac{2 \cdot 65535}{1460} \rfloor = 89$; l'efficacité est donc parfaite avec $U = 1$, soit 100%.

On voit donc que même dans un cas assez favorable (LAN avec délais faibles et débits usuels), le window-scaling est nécessaire pour atteindre l'efficacité de protocole (ici un exposant de 1 suffirait).

En corollaire, il est possible de calculer la taille de fenêtre minimale TCP (en octets, représentant kN) nécessaire en fonction du produit `débit * délai` avec la formule :

$$l_{fenetre_{min}} \geq \dot{D} \cdot RTT \tag{4.17}$$

Par exemple, avec notre exemple de GBit Ethernet ci-dessus, on trouverait 125 kOctet (on peut démontrer cette formule en observant que $l_{fenetre_{min}} = kN \geq N + 2aN$ et que $RTT = \frac{N}{\dot{D}} + 2Tp$).

4.4.4.3.3 Maintenir la qualité de l'estimateur RTT : TCP timestamp option Les implémentations classiques de TCP mesurent le **RTT** (délai aller-retour) sur un datagramme au plus, sur toute une fenêtre. Avec des fenêtres de très grande taille (en cas d'option window-scale), cela est insuffisant. Une nouvelle méthode a été mise en place pour améliorer les mesures de RTT, en particulier en cas de pertes de datagrammes : l'option **time-stamp**.

La perte d'efficacité intrinsèque (voir 4.4.2 en page 44) introduite (10 bytes en plus d'options) est largement compensée par l'amélioration de la taille de fenêtre.

L'horloge utilisée doit changer suffisamment rapidement pour qu'elle soit utile, au moins une fois par fenêtre.

4.4.4.3.4 Maintenir la fiabilité : PROTECT AGAINST WRAPPED SEQUENCE NUMBERS (PAWS) Des délais dans le réseau peuvent réinjecter de vieux messages d'anciennes fenêtres ; de plus, dans une grande fenêtre, un numéro de séquence peut réapparaître : le flot de données sera alors corrompu. Ce repliement de numéros de séquence (*wrap around* – champ de 32 bits) est d'autant plus rapide pour TCP, qui compte le nombre d'octets !

Pour régler ce problème, on exige alors l'option time-stamp vue ci-dessus à *chaque* datagramme : le temps étant monotone croissant, deux numéros de séquence au départ identiques de la même fenêtre ou de deux fenêtres différentes seront alors vus comme différents grâce au time-stamp. Des contraintes supplémentaires sont à vérifier : notamment, l'horloge ne doit pas être trop rapide, sinon elle va également se replier : par exemple, 1 ms à 1 sec par coup d'horloge sont des valeurs admissibles.

4.4.4.3.5 Amélioration de la performance en présence d'erreurs : variante SELECTIVE REPEAT (SACK) TCP fonctionne par défaut en mode **go-back-N** (seuls les segments de données corrects et en séquence sont confirmés, les autres sont ignorés par le récepteur). Cela signifie qu'en cas d'une erreur sur un segment de la fenêtre, tous les segments pourtant corrects situés après (dans la même fenêtre) seront réémis ! Le problème se complique en cas de pertes lors des transmissions successives. Il est évident que des problèmes de pertes de paquets dus à des congestions ne vont être qu'amplifiés par ce protocole, en particulier vu la présence de *plusieurs* sessions TCP parallèles acheminées via un routeur.

Une amélioration existe : l'option TCP de confirmation sélective, documentée dans le RFC-2018. Elle est activée au moment du SYN par une option TCP (TCP SACK PERMITTED). En cas de réception de segments non contigus, le récepteur peut lui envoyer une option TCP SACK, indiquant les segments contigus reçus correctement (offsets des octets à gauche et à droite).

En pratique jusqu'à 4 (voire 3, si l'option time-stamp est activée) blocs discontinus peuvent être indiqués dans une option TCP SACK.

Pas traité en détail cette année

4.4.4.3.6 Détection de futures congestions : Explicit Congestion Notification

TCP incorpore depuis longtemps des mécanismes destinés à gérer les congestions[16] dans le réseau en particulier sur WAN (Wide Area Network), dans TCP.

Ces mécanismes ont cependant les inconvénients suivants :

— ils sont *aveugles* : ils supposent notamment que toute perte de datagramme est due à une congestion (ce qui est souvent vrai) mais surtout avancent en tâtonnant (estimation empirique de la taille de fenêtre problématique, augmentation prudente

16. p.ex. lignes ou routeurs saturé, aboutissant à des pertes de datagrammes sur un routeur

— ils ne peuvent pas prévenir des retransmissions en adaptant le flux d'information avant qu'un problème réel ne se produise.
— ils peuvent provoquer des pertes de données dans les *autres* flux (p.ex. flux UDP vidéo ou audio) en raison même de l'augmentation progressive des queues dans les routeurs induite par l'algorithme d'exploration de taille de fenêtre TCP.

L'idée d'**ECN** (*Explicit Congestion Notification*, RFC-3168) est de détecter les congestions et d'adapter le débit en fonction, grâce à des routeurs implémentant l'IP ECN (qui marquent les paquets sur le point de subir une congestion) et des noeuds terminaux implémentant l'IP ECN et le TCP ECN (la destination IP/TCP informant l'émetteur). Au sein du **datacenter**, ces améliorations sont particulièrement utiles [17].

Le principe est de modifier l'entête IP : Deux nouveaux bits remplacent les anciens bits réservés 6 et 7 du champ TOS (*type of service*) (ou du champ *differentiated services* dans les nouvelles définitions).

0	1	2	3	4	5	6	7
Differentiated Services						**ECN field**	

ECN field			
ECT	CE		**bit mis par**
0	0	inactif	
0	1	ECN-Capable Transport (ECT) 0	émetteur de données
1	0	ECN-Capable Transport (ECT) 1	émetteur de données
1	1	Congestion Experienced (CE)	routeur

Les deux codes ECT 0 et ECT 1 sont complètement équivalents et indiquent que l'émetteur supporte l'ECN. Ils permettent de tester si nécessaire que le bit CE n'est pas remis à zéro pendant le transit sur le réseau. Des provisions existent au niveau des protocoles couche 4 (transport) pour gérer ces codes séparément.

De plus, l'entête TCP est également modifié : deux bits réservés dans l'entête sont alloués :

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Header Length				Reserved				**CWR**	**ECE**	URG	ACK	PSH	RST	SYN	FIN

Le drapeau ECE (ECN-Echo) est positionné par le récepteur lors de sa confirmation de manière à informer l'émetteur de la congestion. Le drapeau CWR (Congestion Window Reduced) est positionné par l'émetteur de manière à informer le récepteur de la bonne réception du drapeau ECE.

Fonctionnement de l'ECN Les routeurs qui ont leur queue pleine laissent tomber le datagramme comme précédemment. Les routeurs qui sont en voie de devoir laisser tomber des datagrammes (près de la congestion) activent la combinaison 11 dans le champ ECN, dans la mesure où le champ valait autre chose que 00 précédemment (compatibilité pour le cas non supporté / réservé).

Si l'on s'approche de la congestion et que ECN est activé sur l'émetteur, et que le routeur en voie de saturation supporte l'ECN le récepteur IP/TCP reçoit alors un datagramme qu'il peut identifier comme ayant traversé au moins un routeur en congestion.

17. voir http://simula.stanford.edu/~alizade/Site/DCTCP.html

Dans le cas de TCP, le récepteur TCP peut donc confirmer le datagramme en y activant le drapeau ECE (ECN-Echo) du champ TCP, ce qui provoquera du côté de l'émetteur le même comportement qu'une perte de datagramme (sauf la retransmission), soit baisser la taille de la fenêtre de la même manière que si la confirmation n'avait pas été reçue par l'émetteur (sauvegarde de la valeur de la fenêtre divisée par deux comme limite entre *slow start* et *congestion avoidance*, redémarrage à partir de 1 segment).

Comme d'habitude, TCP étant bidirectionnel, chacun des émetteurs est également récepteur et réciproquement. Toutefois, chaque direction est gérée indépendamment pour la congestion.

L'ouverture de la connexion TCP est également modifiée afin de déterminer le support ECN du partenaire TCP : on active alors le bit CWR à l'ouverture.

4.4.4.3.7 Problèmes introduits par ces modifications

Ces modifications sont largement compatibles avec les noeuds et terminaux et interopérables avec des équipements antérieurs, dans la mesure où ils respectent les standards :

- les nouveaux drapeaux utilisés peuvent être filtrés par les firewalls (notamment les bits ECN) : le symptôme est alors un Internet dont certaines parties sont inaccessibles, la seule solution générale étant de corriger les firewalls ou de désactiver l'ECN ; ou d'attendre un meilleur support des ces drapeaux, ce qui semble être le cas en 2013
- les drapeaux concernant le **window-scale**, **SACK** et **time-stamp**, sont également compatibles avec l'ancienne version de TCP, dans la mesure où les extensions sont implémentées correctement et les options inconnues ne sont pas répétées par un récepteur non conforme aux standards

4.4.4.3.8 Autres améliorations

- la segmentation des données des couches supérieures par la couche 4, respectant le MTU couche 3 (qui lui-même dépend de la couche 2 correspondante) peut coûter cher en performance système : le **TSO** (*TCP segmentation offload*) permet de transmettre ce travail au pilote de la carte réseau, qui se charge alors de recréer les entêtes IP, checksum et séquences TCP en fonction (cette fonction ressemble à la fonction de scatter/gather implémentées en matériel pour les cartes de stockage p.ex.) ; la conséquence évidente est qu'un Wireshark sur la machine émettrice ou réceptrice ne montrera pas la réalité mais des tailles de trames étonnantes.
- l'utilisation de trames couche 2 de taille supérieure (**jumbo frames**) lorsque c'est possible
- et, en-dehors du domaine de la performance : des améliorations de sécurité pour rendre l'injection aveugle de données TCP moins facile (la solution correcte étant le chiffrement et la signature électronique), par exemple le RFC-5961.

De nombreux algorithmes[18] pour améliorer la performance du protocole à fenêtre lui-même ont été proposés et certains ont été largement déployés.

La difficulté est de toujours considérer que plusieurs flux TCP (et donc des algorithmes potentiellement différents) peuvent se cotoyer dans le même canal.

Une des dernières améliorations proposées est **BBR**[19], qui modélise et mesure le canal en terme de débit minimum (au goulet d'étranglement) et de délai, de manière à combattre le *bufferbloat* (trop de tampons un peu partout, qui créent du délai et peuvent amener à des pertes de paquets). L'idée est de remplir le tuyau au maximum sans augmenter le délai (car

18. https://en.wikipedia.org/wiki/TCP_congestion_control#Algorithms
19. https://queue.acm.org/detail.cfm?id=3022184

cela signifie que l'on utilise les tampons). BBR a en plus l'avantage de tenir compte du problème de l'asymétrie des liaisons, présenté ci-après.

4.4.5 Asymétrie des liaisons

En pratique et dans le cas des accès Internet des clients finaux, on doit également tenir compte de l'**asymétrie** des débits montants et descendants. Bien souvent, les débits montants sont le dixième du débit descendant.

Lors d'un téléchargement vers le client final, et de confirmations non groupées, l'instance TCP située sur le serveur sur Internet (en amont) envoie des données et l'instance TCP située sur la machine du client final envoie des confirmations. Ce n'est que si les confirmations arrivent assez vite que TCP peut travailler efficacement.

Au maximum, et en supposant un **MTU** couche 3 de 1500 octets (p.ex. sur Ethernet), les données utiles couche 7 feront 1460 octets, et les entêtes 40 octets : le rendement intrinsèque descendant maximal atteindra 97.3%.

Or, si le débit asymétrique de couche 3 est de 5 Mbit/s descendant, cela correspond au plus à 417 paquets par seconde descendant (en négligeant le tramage ADSL). Il faudra donc 417 confirmations par seconde, et donc un débit montant de 133'440 bit/s.

Le débit classique ADSL 5 Mbit/s descendant et 500 kbit/s montant semble donc dans ce cas largement suffisant. Mais cela ne serait pas le cas si tout à coup le MTU était largement inférieur sur une des liaisons entre le serveur et le client : p.ex. pour un MTU inférieur à 400 octets, le débit montant serait insuffisant [20] [21].

Le problème pourrait même se poser avant ce MTU très bas si la liaison montante sert également à des transferts de données dans l'autre direction : en effet, le **piggy-backing** ne peut être utilisé que sur la même session TCP et non pas en cas de sessions simultanées.

20. potentiellement moins si la couche 3 du client est directement connectée à une couche 2 disposant de contrôle de flux, car TCP pourrait alors remarquer que les queues d'envoi sont pleines et qu'il doit grouper les confirmations ; une autre tactique est de grouper systématiquement des confirmations proches lorsque les délais mesurés montrent un protocole efficace, voir [18].

21. il existe des équipements d'opérateur CATV qui peuvent effectuer des regroupements/dégroupements artificiels des confirmations TCP introduisant d'autres problèmes difficiles à debugger

Chapitre 5

Le dernier kilomètre (the last mile)

Sommaire

Autant les réseaux locaux (Ethernet, WiFi) que les réseaux publics à grande distance **WAN** (*Wide Area Network* : SDH[1], ATM[2], MPLS[3] ...) 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.

5.1 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 et de fiabilité des émetteurs/récepteurs, de créer un nouveau raccordement fibre pour chaque immeuble ou appartement. C'est pourquoi les solutions classiques font appel à l'**infrastructure existante** : les lignes téléphoniques, le

1. Synchronous Digital Hierarchy : voir section 6.2.3 en page 77
2. Asynchronous Transfer Mode, voir section 6.1.2 en page 74
3. Multi Protocol Label Switching : le réseau *core* actuel chez les opérateurs.

câble TV, l'alimentation électrique, et la transmission hertzienne (sans-fil) sont possibles, avec chacune ses avantages et ses inconvénients.

Les opérateurs se sont transformés par le **triple-play**[4], en rassemblant les offres télévision, téléphone et Internet dans un seul produit, en profitant du **dégroupage**, qui permet de s'affranchir, au moins en partie, des monopoles étatiques ou historiques pour l'accès aux réseaux WAN. Pour offrir un tel produit, un fournisseur peut (voir figure 5.2 en page 62) :

- raccorder lui-même l'abonné à travers son propre réseau national et ses propres équipements : vu son coût élevé, cela n'est proposé que par quelques fournisseurs nationaux s'ils disposent d'une infrastructure propre (par exemple Cablecom sur **CATV**, certaines offres hertziennes par satellite), ou par des fournisseurs locaux qui se concentrent sur des régions où ils disposent de points d'accès[5] (p.ex. Init7, VTX...)
- utiliser l'infrastructure Swisscom du dernier kilomètre (xDSL sur téléphone) et disposer ses équipements dans un central Swisscom[6] (possible depuis 2009) : une variante un peu plus économique en investissements mais moins flexible que la première et plus chère avec de nombreux abonnés
- utiliser l'infrastructure d'un autre opérateur, y compris transport des données : cette variante classique permet d'offrir un service national sans investissements lourds, avec en contrepartie une flexibilité très basse et un partage conséquent du chiffre d'affaire avec l'opérateur

Dans tous les cas, aujourd'hui la tendance pour le téléphone analogique est d'être remplacé par de la voix-sur-IP sur la connexion Internet de l'usager, les systèmes de téléphonies classiques n'étant plus développés.

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 [9]).

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 Genève ou Fribourg. Ces technologies **FTTx**, comme **FTTH** (*Fiber To The Home*) ou **FTTB** (*Fiber To The Building*, avec boîtiers de distribution VDSL ou d'autres technologies), relient 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**[7], notamment basés sur le sans-fil (p.ex. 802.11), ne visant pas seulement l'accès à Internet bon marché.

5.2 Entreprises

Les entreprises ont un choix plus important de connexions WAN, dans un contexte plus large que l'accès à Internet seul. Les différentes principales avec les accès privés étant la présence d'offres

4. voire quadruple-play avec la téléphone mobile
5. **POP**
6. il y a un central dans chaque zone géographique, par exemple village, et la carte http://hochbreitband.ch/fr/atlas-de-la-large-bande.html permet d'avoir une idée des solutions et débits de dernier kilomètre disponibles
7. réseau maillé, voir par exemple le téléphone coopératif http://www.craslab.org/bricophone/?page=FAQfr ou le projet SFNet à Genève.

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 de service des liaisons (voire une gestion/surveillance intégrée par le fournisseur).

- pour la téléphonie : réseau classique ISDN de base ou primaire (paire symétrique, utilisable également en **HDSL**), une solution certes courante mais désuète et qui sera de plus en remplacée par de la **voix-sur-IP** transportées sur des accès Internet (voir ci-dessous)
- pour l'accès Internet
  - les mêmes technologies que pour les usagers PME ou résidentiels (voir section 5.1 en page 57), complétées par des offres de gestion et surveillance, de qualité de service [8] (notamment en cas de transport simultané de la voix-sur-IP), des plages d'adresses IP fixes, de la **haute disponibilité** par redondance et souvent des offres à débit symétrique
  - des technologies spécifiques rapprochant du réseau de bord ou du réseau core **MPLS** de l'opérateur (voir section 6.3.5 en page 79), souvent en fibre optique
- pour le transport de données entre sites :
  - à courte distance : les lignes louées locales, reliant deux lignes téléphoniques existantes en cuivre via un central d'opérateur, ou des lignes propres (cuivre ou **fibre optique**) posées spécifiquement, puis exploitées par exemple en **SDSL** (*Symetric Digital Subscriber Line*) pour le cuivre, ou des liaisons sans fil (voir chapitre 6.3.7 en page 84)
  - à longue distance : outre les **VPN**s (*Virtual Private Network*, réseau privé virtuel, étendant le concept des classiques groupes fermés d'usagers **CUG**) construits sur les technologies d'accès Internet pré-citées, y compris **MPLS**[10], ou des services clés en main comme **CES** (*Carrier Ethernet Service*) ou les lignes louées numériques
  - pour les employés mobiles (**Road Warrior**), ayant besoin d'utiliser les services de l'entreprise, des services de VPN globaux ou **cloud**, accessibles dans le monde entier
- pour les réseaux de terrain / industriels pour la mesure et le contrôle, voire la commande, de plus en plus intégrés aux réseaux d'entreprises classiques (Internet) (voir section 5.4 en page 69)

Les entreprises ont tendance à intégrer leurs différents réseaux dans un réseau unique, parfois même basé sur des technologies d'opérateur comme **MPLS**. Les applications industrielles exploitent des données de réseaux de terrain et intègrent dont ceux-ci au réseau classique (Internet) de l'entreprise.

5.3 Réseaux d'accès

Par réseaux d'accès, on entend les technologies qui permettent l'accès à un réseau d'opérateur. Certaines de ces technologies sont toutefois mixtes et peuvent être exploitées sans opérateur (par exemple le WiFi), ou dans des cas particuliers pour des réseaux privés. La plupart du temps, ces technologies sont vues comme permettant l'accès Internet et c'est sur Internet que l'exploitation est faite (avec ou sans VPN, par exemple).

8. **QoS SLA** : *Service Level Agreement* : contrat de qualité de service (débit offert, délais, etc), géré par un opérateur et traversant son réseau

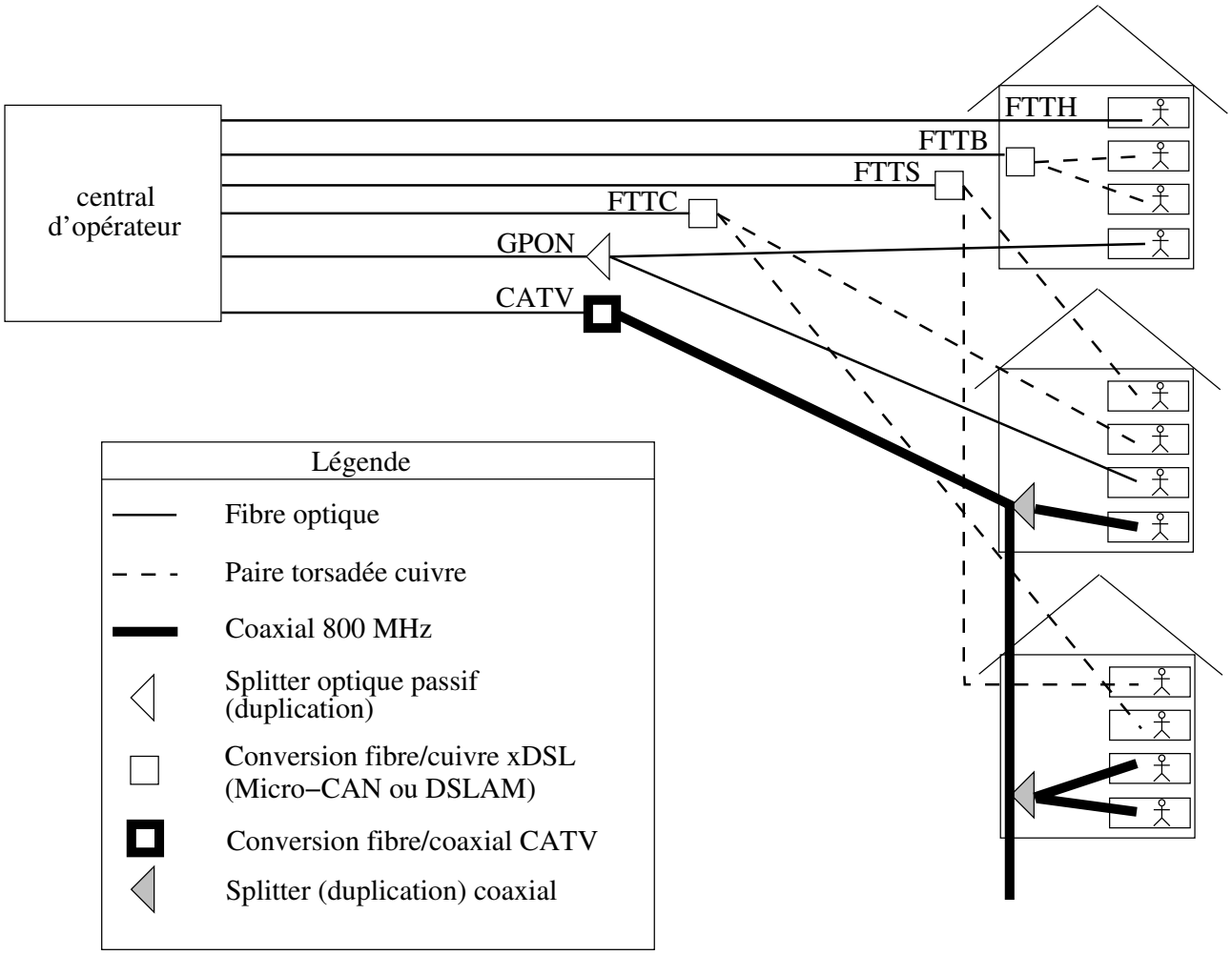


FIGURE 5.1 – Fibre pure (FTTH PTP), hybride (FTTx), hybride multiplexée (CATV) et fibre multiplexée/multipoint (GPON, soit FTTH PTMP)

5.3.1 FTTx

Câbler directement l'abonné en fibre optique pure (**FTTH**, *Fiber to the home*, en topologie point-à-point **PTP**) est aujourd'hui possible, en particulier dans les centres urbains et lorsque les pouvoirs publics soutiennent cette initiative. Cette technique est d'ailleurs ouverte au **dégroupage total**[9] en Suisse.

Toutefois, la plupart du temps, des solutions *hybrides* ou *multiplexées* sont déployées. L'avantage est bien sûr le coût et la réutilisation partielle des infrastructures cuivre des derniers mètres pour l'hybride, mais aussi que ces techniques ne sont actuellement pas entièrement dégroupables[10] et garantissent donc une rente de situation à l'opérateur.

Les solutions hybrides fibre/cuivre sont, par distance cuivre décroissante : **FTTC** (*Fiber to the curb*), **FTTS** (*Fiber to the street*), et **FTTB** (*Fiber to the building*[11]). Elles relient en fibre le central à un boîtier **MicroCAN** situé dans le quartier, la rue ou dans l'immeuble, et depuis ce

9. plus besoin de l'opérateur historique : infrastructures propres entre l'opérateur tiers et l'abonné

10. aucune obligation légale pour Swisscom de dégrouper – même si Swisscom le fait à ses tarifs – de plus, la gestion des diverses liaisons xDSL en mode **vectoring** pour compenser la **diaphonie** nécessite que toutes les liaisons d'un même tube soient gérées sur le même **DSLAM** : seul un **dégroupage partiel** est actuellement possible, et pas garanti

11. **FTTP**, *Fiber to the premises*, désigne les deux technologies **FTTH** (pure, dégroupable) et **FTTB** (hybride, pas forcément dégroupable)

point des technologies cuivre (**G.fast** ou VDSL2, voir la section suivante) sont exploitées. Les débits disponibles restent élevés, tant que la distance au boîtier reste limitée, mais l'évolutivité et la concurrence font défaut.

En multiplexé, avec partage de débit sur le dernier kilomètre, citons les technologies **GPON**[12] et CATV (voir section 5.3.3 en page 63), un hybride fibre/coaxial. Le GPON (FTTH point-à-multipoint **PTMP**) est un multiplexage optique (fibre) passif temporel (**TDMA**) où l'ensemble des abonnés, comme sur le câble coaxial du CATV, se partagent le débit total. Le multiplexage diminue le coût mais diminue automatiquement le débit disponible par utilisateur par rapport à de la fibre pure **FTTH PTP**.

5.3.2 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)[13]. 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, après un large repli chez les usagers résidentiels, a été désactivé par Swisscom en 2018, qui a migré ses clients à la **voix-sur-IP** sur **xDSL**.

La **paire torsadée** cuivre 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 fonctionne 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.

Sans migration à la **voix-sur-IP**, 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**[14]) ou encore via **USB**, si la liaison PPP se termine sur un équipement routeur séparé ou si le rôle est implémenté par le PC.

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 (tunnel) 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 (voir figure 5.2).

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

12. Swisscom déploie du XGS-PON, soit du GPON à 10 GBit/s pour tous les abonnés, symétrique.
13. frisant la limite théorique, grâce au fait qu'une partie de la liaison doit être numérique, en ISDN, du côté serveur et donc ne participe pas à la perte analogique de rapport signal sur bruit côté abonné et jusqu'au central.
14. ici Ethernet n'est utilisé que comme transport de trames **PPPoE**, et non pour IP directement

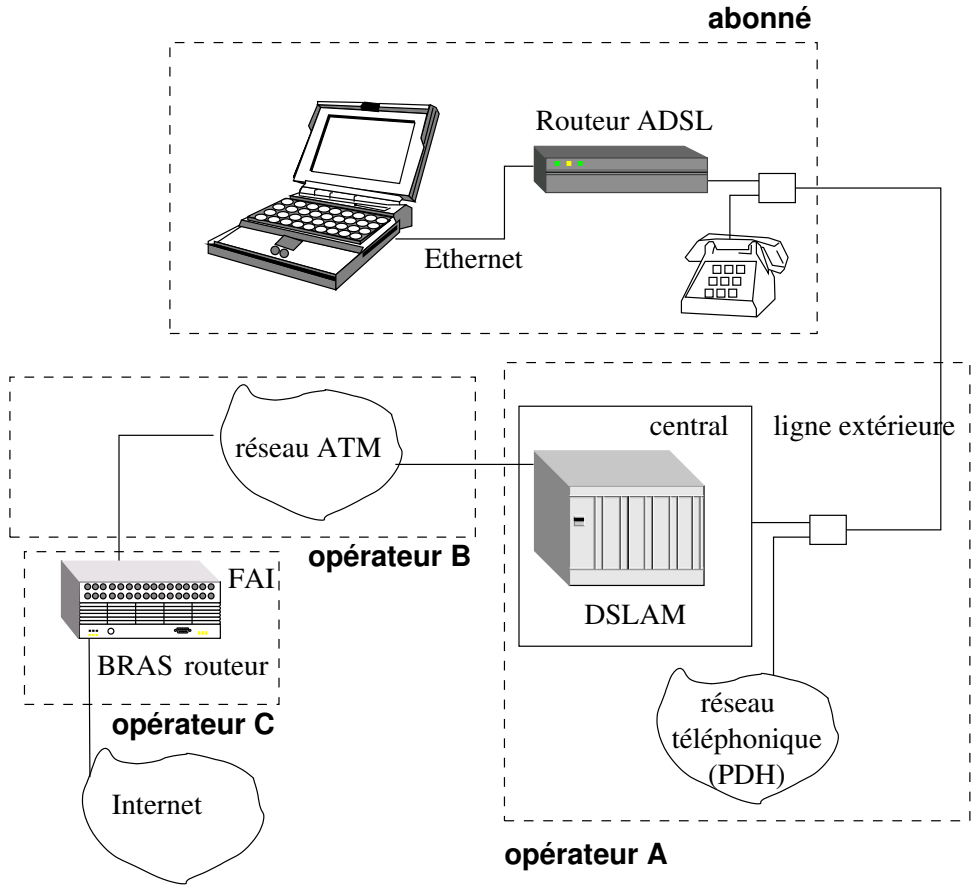


FIGURE 5.2 – xDSL : Liaison et transport par tunnel couche 2

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 a totalement modifié, du moins dans les zones bien desservies, les conditions d'accès [15].

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, VDSL, VDSL2, G.fast, etc) correspondent à des débits et modulations différents. Ces variantes sont en règle générale prévues pour le grand public et sont bien adaptées à l'accès usager Internet car les débits sont asymétriques [16] (grand débit descendant, petit débit montant). Les technologies ont évolué en direction de la prise en compte des interactions diaphoniques entre les paires torsadées d'un même tube (**vectoring**) : la dernière technologie, le **G.fast**, permet jusqu'à 1 GBit/s maximum [17] et au total pour les deux directions [18], pour une distance de 250 mètres maximum. Contrairement aux autres technologies DSL et CATV qui utilisent une répartition des fréquences descendantes et montantes

15. rien n'oblige d'ailleurs le fournisseur dégroupé d'utiliser les technologies en voie d'obsolescence et coûteuses comme ATM dans son réseau.
16. le SDSL, utilisé plutôt en entreprise, est la variante symétrique
17. avec le NG.Fast à max 5 GBit/s
18. répartition 500/500 et 900/100 Mbit/s fréquentes

(**FDD**, *Frequency Division Duplexing* : basée sur un multiplexe fréquentiel **FDM**, aussi appelé la modulation large-bande **DMT**), G.fast utilise un multiplexe temporel (**TDM**), sous forme d'un demi-duplex (**TDD**, *Time Division Duplexing*) sans impact significatif sur le délai.

Il faut noter qu'ATM est utilisé en ADSL dans la couche 2 pour le transport des données [19], en général via la couche **AAL5** [20]. Un réseau ATM était originellement utilisé 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.

5.3.3 Câble TV

Le câble TV a été conçu au départ 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 utilisé 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 le débit total du coaxial [21], contrairement à l'xDSL où chaque usager dispose de son propre canal jusqu'au DSLAM [22]. 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. L'offre Internet CATV est donc une offre comparable à l'offre **fibre hybride** (**FTTB**, **FTTS**, **FTTC**) (voir section 5.3.1 en page 60).

Les opérateurs proposent en général un bouquet principal de chaînes de base et des chaînes payantes (p.ex. 300 chaînes numériques [23], dont quelques HD, encodées en DVB-C sur quelques dizaines de canaux de 10 MHz), et le reste des canaux servent à l'accès Internet multiplexé temporellement et transporté par une modulation **DMT**/**CAP**. L'accès se fait par boîtier-décodeur intelligent capable d'utiliser le multicast **DVB-C** et des fonctions avancées (pause, replay, enregistrement) via des services Internet.

Les technologies les plus modernes (p.ex. en 2023, ELLO à Neuchâtel) permettent des débits jusqu'à 1 GBit/s descendants (et 300 MBit/s montants) par abonné via des techniques de modulation avancées à près de 70 MBps par canal de 10 MHz.

5.3.4 Internet par réseau électrique (PLC/CPL)

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

19. dès le VDSL, l'ATM n'est plus utilisé, mais du **GBit Ethernet** puis le réseau MPLS core.
20. ATM Adaptation Layer 5 : aussi appelée **SEAL**, pour *Simple and Efficient Adaptation Layer*.
21. bande passante très élevée de l'ordre de 800 MHz, multiplexée en fréquence dans des canaux de 10 MHz
22. en FTTC/S/B, cet équipement se trouve assez près de l'abonné et en GPON il y a également partage du débit – en pratique le débit disponible dès qu'il y a un tronçon cuivre est similaire à CATV
23. l'offre analogique, avec un canal TV par chaîne, a été supprimée pour faire place

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 240V du réseau électrique interne d'un logement, voire d'un bâtiment.

Les signaux servant au transport des données ne traversant pas les transformateurs, il est nécessaire de 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 [24] inhérents, les câbles électriques agissant comme antennes.

5.3.5 Boucle locale sans fils (wireless local loop)

5.3.5.1 Types de technologies

On peut classer les technologies permettant l'accès Internet sans fil à moyenne distance [25] :

- basées au départ sur des technologies des réseaux locaux informatiques : WiFi 802.11, sans bande de fréquence réservée et donc à puissance plus faible (voir sections 7.2.5 en page 88 et 5.3.5.5 en page 67)
- basées sur des technologies de la téléphonie GSM (voir figure 5.3 en page 64) : **HSCSD**, **GPRS**, **UMTS** puis **LTE** et enfin **5G** : comme elles utilisent des bandes de fréquence réservées [26], elles disposent de plus de puissance ; suivant le maillage d'antennes et le nombre de mobiles, les performances peuvent varier
- conçues spécifiquement pour l'accès Internet : par exemple le **WiMAX**, qui utilise des bandes réservées mais n'a pas eu de succès en Suisse en raison des coûts et de la densité des autres technologies d'accès disponibles.

1980	1992	2001	2010	2020 →
1ère (1G)	2^e (2G)	3^e (3G)	2^e (4G)	2^e (5G)
analogique	numérique	numérique	numérique	numérique
voix	voix et textes (SMS)	voix et données	voix, données, vidéos	voix, données, vidéos et objects connectés
	80 à 100 kbit/s	→ 2 Mbit/s	→ 1 Gbit/s	→ 10 GBit/s

Source : [27], page 67 (basé sur : La Croix et Commission Européenne).

FIGURE 5.3 – Les générations de la téléphonie mobile (GSM)

24. bandes radio-amateur : le problème est aussi présent en xDSL moderne, mais l'effet est plus local
25. hors satellite par exemple
26. donc soumises à redevance OFCOM en Suisse

5.3.5.2 3G / UMTS

En ce qui concerne la 3G, la communication entre la station et le mobile se fait en multiplexe temporel dynamique au sein d'un **OFDM** (*Orthogonal Frequency Division Multiplexing*) qui suit le même principe que la transmission à large bande (**DMT**, *Discrete Multi Tone*), dont le but est de mieux supporter des sous-canaux bruités) mais en plus en utilisant des porteuses orthogonales entre elles pour qu'elles ne s'influencent pas. La communication entre le mobile et la station se fait par une émission sur un canal donné (multiplexage fréquentiel) à un moment réservé par la station (multiplexage temporel pseudo-statique : "slot").

L'UMTS voulait permettre divers types de qualité de service (débit notamment) en fonction de la densité d'antennes et donc de permettre aux opérateurs des services différenciés : beaucoup d'opérateur exploitent plutôt des réseaux 3G/4G/LTE mixtes dans ce but.

5.3.5.3 4G / LTE

La 4G amène une légère évolution dans le sens station vers mobile (OFDMA) : toutes les stations (et il peut y en avoir beaucoup dans un périmètre) émettent dans les mêmes fréquences : l'OFDM doit donc être meilleur *et* les stations doivent être très bien synchronisées (voir 6.3.7 en page 82).

Dans le sens mobile vers station, la modulation devient également OFDM avec un *frequency hopping* (changements de fréquences) ainsi que des slots. Le grand avantage par rapport à la 3G est que la puissance nécessaire à l'émission est bien plus faible en 4G (la répartition du signal sur plusieurs fréquences permet de le recevoir même à puissance basse). C'est mieux pour la batterie et peut-être pour les humains.

En plus du multiplexage ci-dessus, le code de couche 1 est souvent du QAM (QAM-16 p.ex.), soit une modulation phase-amplitude classique.

La téléphonie 4G peut être en l'un des deux modes suivants :

1. voix en 3G, données en 4G (LTE)
2. voix en voix-sur-IP sur données 4G, données en 4G (LTE-Advanced ou **VoLTE**), intégré NGN et plus performant

Swisscom est passé en mode 2 en 2014, avec un déploiement d'abord dans les grandes villes.

Parler de 4G pour le LTE (Long Term Evolution) est presque un abus : on a longtemps considéré que le LTE était de la 3.9G, en particulier en mode voix 3G.

5.3.5.4 5G

La bataille des bandes de fréquences à attribuer à la 5G a déjà commencé, y compris en Suisse, et un déploiement initial a été effectué dès 2019. Toutefois, les équipements compatibles (téléphones mobiles, chips IoT) ne sont pas encore majoritaires. De plus, les bandes de fréquences allouées jusqu'ici sont à des fréquences usuelles : les plages de fréquences les plus élevées (50 GHz, bande millimétrique – qui nécessitent encore plus d'antennes) ne sont pas encore utilisées. Un déploiement en plusieurs étapes est prévu, d'autant plus que pour atteindre tous les objectifs, la densité d'antenne doit être bien plus grande qu'en 4G/LTE.

Si la technologie est très performante, avec des codages de ligne améliorés atteignant 50 bits par Hertz (grâce à des constellations polaires, du MIMO (multiplexe spatial avec plusieurs antennes), du full-duplex via multiplexage temporel, de la conformation de direction d'émission

ou *beam-forming*), le *hype* n'est pas en reste : on nous présente une technologie permettant à la fois les LANs, les WANs, la basse consommation (**IoT**), des délais très faibles autorisant des applications très demandeuses en temps réel comme le contrôle à distance, les véhicules autonomes, la réalité augmentée et des débits dépassant ce qui est aujourd'hui disponible en dernier kilomètre filaire, à part en **FTTH**.

Si les documents de design demandent 8 propriétés [27] des technologies dites 5G :

1. débit élevé (jusqu'à 10 GBit/s)
2. 1ms de latence
3. bien plus de débit par antenne (par rapport à la 4G/LTE)
4. bien plus d'appareils connectés par antenne
5. disponibilité de 99.999%
6. couverture totale
7. 90% de réduction de consommation d'énergie
8. jusqu'à 10 ans de durée de vie de batterie IoT faible consommation

des *flavors* [28] sont déjà définies pour couvrir des domaines d'application spécifiques : dans les faits, on peut dire ceci sur les technologies 5G :

— Wikipedia [29] dit la vérité : c'est un terme marketing, principalement, qui recouvre de nombreuses technologies variées
— elles sont encore *en développement*, au sein de l'ITU-T mais également d'autres organisations comme l'IEEE ou 3GPP : citons le standard ITU-T IMT-2020, spécifiant un débit maximal descendant de 20 GBit/s, ou ceux du 3GPP – qui n'imposent par exemple pas de performance spécifique
— elles intègrent et feront évoluer, plutôt que remplacer, les technologies actuelles comme la 4G/LTE : elles utiliseront des plages de fréquences existantes, des nouvelles et des protocoles couche 1 et 2 actuels et futurs
— elles offriront une vision globale de toutes les technologies d'accès sans-fil existantes au sein d'un méta-réseau virtuel (logiciel) évoluant en fonction des besoins des applications et de la charge du réseau core et d'accès (**network slicing** [30])
— elles modifieront peut-être manière dont les documents seront accédés sur Internet : passant d'un modèle d'URL localisant les documents à un endroit donné à un modèle d'accès au contenu (recherche en cours, voir par exemple FIA [31] ou **ipfs** [32])
— certains aspects de sécurité qui auraient pu être améliorés, au sein du réseau core et 5G, par un déploiement des options de sécurité des NGN, semblent encore négligés : pourtant, des attaques récentes contre SS7 [33], voire contre les variantes IP utilisées en 4G/LTE, ont été publiées.

On pourrait dire que tel l'UMTS qui avait pour but de fournir un cadre pour une évolution et une flexibilité de l'offre télécom classique [34], la 5G veut elle rendre l'Internet adapté au mobile et aux applications modernes **all-IP**.

27. https://www.gemalto.com/france/telecom/inspiration/5g et https://www.sciencedirect.com/science/article/pii/S1389128616301918
28. https://techpinions.com/my-5g-explainer-there-will-be-five-flavors/57523
29. https://en.wikipedia.org/wiki/5G
30. des tests de ces fonctionnalités sont effectués régulièrement : par exemple un déploiement optimisé IoT : https://www.rcrwireless.com/20151022/carriers/5g-network-slicing-demo-by-sk-telecom-ericsson-tag23
31. *Future Internet Architecture* research project, http://www.nets-fia.net/
32. https://ipfs.io/
33. le protocole classique de signalisation inter-opérateur notamment utilisé en 2G/3G ou téléphonie fixe
34. et qui a été finalement dépassé par la 4G/LTE notamment car l'UMTS n'avait pas anticipé les besoins *data* et Internet

Toutefois, trois éléments à ne pas perdre de vue sont que l'entreprise qui a le plus contribué à la 5G, Huawei, est aussi celle qui est actuellement la cible du gouvernement US, et que les investissements nécessaires sont colossaux pour atteindre ne serait-ce qu'une partie des 8 propriétés mentionnées ; enfin, certaines critiques se font jour par rapport à une utilisation intensive des fréquences millimétriques (50 GHz) – le principe de précaution [35] pourrait limiter l'expansion en fréquence de la 5G.

5.3.5.5 WiFi 6

Cette évolution du WiFi veut intégrer des éléments de performance, de basse consommation, de *beam-forming* et utilise l'OFDMA aussi pour l'upstream. Il s'agit d'un sérieux concurrent, indépendant des opérateurs, face à la 5G, mais comme elle, c'est une évolution et non pas une technologie déjà déployée.

5.3.6 Satellites

5.3.6.1 Introduction

Les satellites sont un moyen de télécommunication adapté à certains types d'applications, en particulier globales ou pour des zones mal desservies, ou encore pour la diffusion vidéo simplexe sur une large région. On différencie les satellites géostationnaires (**GEO**) des satellites à orbite basse (*Low Earth Orbit*, **LEO**) : ces derniers peuvent offrir des fonctionnalités proches des réseaux terrestres.

5.3.6.2 Satellites géostationnaires (GEO)

Les satellites géostationnaires n'ont été pendant longtemps qu'un moyen alternatif ou de secours pour la transmission bidirectionnelle de données, en raison du coût élevé, de la taille des antennes terrestres et de la puissance d'émission nécessaires, du nombre de canaux limités et du délai très élevé. Toutefois, ils sont fort utiles pour certaines catégories d'applications (diffusion descendante de contenus simplexe, couverture de zones éloignées, etc).

Pour les applications interactives, le problème principal est le délai élevé, dont les causes sont :

- la distance élevée jusqu'au satellite géostationnaire (situé à l'équateur à 36'000 km d'altitude), à parcourir deux fois pour un seul bond (station terrestre – satellite – station terrestre), en fonction de l'emplacement réel des stations
- les protocoles d'accès aux canaux, en particulier en liaison montante et la surcharge des queues des satellites
- les débits relativement faibles à disposition (par exemple à 64 kbit/s, un paquet IP de 1500 octets prend 188 ms, mais seulement 7 ms pour un paquet d'ouverture TCP courant)
- l'asymétrie potentielle des débits montants et descendants

Pour deux stations situées à l'équateur juste en-dessous du satellite, le délai minimum est de 240 ms par bond (pour $h = 36'000$ km, aller seulement) :

$$t = 2\,\frac{h}{c} \tag{5.1}$$

35. voir aussi https://wiki.alphanet.ch/Sandbox/La5GOuPasLa5G

Pour avoir une estimation plus fiable, il faut tenir compte des distances plus grandes dès lors que l'on n'est ni directement au-dessous du satellite, ni à l'équateur. Ensuite, il faudrait ajouter les délais de multiplexage et d'acheminement vers Internet (entre la station de réception et le serveur réellement accédé). Une étude de 2021 [36] donne plus de détails sur les facteurs de délais et compare plusieurs mesures de trois opérateurs différents (intervalle de de 500 ms à 800 ms par bond).

En conséquence, l'ouverture d'une connexion TCP (3 phases), en pratique, peut durer plus d'une seconde et demi ! On peut limiter ces délais en partie grâce à des accélérateurs de protocole (*Performance Enhancing Proxy*, PEP [37]) ou en optimisant certains aspects du protocole (voir RFC-2499, page 13, table 1 ; QUIC (HTTP/3.0) pourrait avoir un avantage) ou de placement de la station terrestre.

5.3.6.3 Satellites à orbite basse (LEO)

Le lancement récent de plusieurs constellations de satellites à basse altitude, qui vont être complétées et renouvelées [38], permet vu la faible distance et le nombre de satellites, des communications bidirectionnelles avec des délais raisonnables ($<$ 100 ms) et le support d'applications exigeantes grâce à un tracking multi-satellite performant et l'utilisation de stations terrestres géographiquement proches du pays de résidence. Pour le moment, les applications envisagées sont plutôt de supporter des zones avec une faible densité d'accès Internet classiques. La concurrence avec d'autres technologies, comme la 5G, voire même des technologies IoT, pourrait venir plus tard.

36. https://cseweb.ucsd.edu/~schulman/class/cse291_f21/docs/satperf.pdf
37. https://ieeexplore.ieee.org/document/6181967
38. la durée de vie en orbite basse est plus courte

5.4 Réseaux de terrain

5.4.1 Introduction

Les réseaux de terrain (ou réseaux industriels) permettent la mesure, le contrôle d'équipements, voire la commande d'axe temps réel. Ces trois besoins ont bien évidemment des impératifs différents en ce qui concerne la qualité de service nécessaire (p.ex. des délais prévisibles et très courts en ce qui concerne la commande d'axe). Toutefois, beaucoup de processus lents sont commandables même avec des délais non garantis et certains réseaux de terrain sont optimisés principalement pour la mesure. On distingue divers domaines d'applications qui varient dans leur envergure et leurs types de besoins (mesure, contrôle, commande) : atelier, usine, domotique, cité intelligente, etc.

Si ce ne sont pas des réseaux d'accès Internet à proprement parler, certains protocoles, comme **LoRaWAN** permettent, via des **gateway**s, à des applications Internet de consulter des données rapportées par des équipements de terrain et d'interagir de manière limitée avec eux.

protocole	**énergie**	**débits**	**portée**	**topologie**	**opérateur ?**
WiFi	grande	1.3 GBit/s	$<$ 100m	étoile	sans
BLE	faible	1 Mbit/s	$<$ 100m	point à point, maillé	sans
Zigbee	très faible	256 kbit/s	$<$ 100m	point à point, étoile, maillé	sans
NB IoT	faible	250 kbit/s	30km +	cellulaire	avec
Sigfox	très faible	800 bits/s	10km +	cellulaire	avec
LoRaWAN	très faible	250 – 5470 bit/s	10km +	point à point, étoile, maillé	avec/sans
LTE-m	faible	qq bit/s à 1 MBit/s, voire 4 MBit/s	10km+	cellulaire	avec

La topologie étoile indiquée est logique, le média étant partagé. Le WiFi peut être exploité avec d'autres topologies et distances notamment dans le cadre de réseaux à technologies mixtes (voir chapitre 6.3.7 en page 84).
La norme WiFi 802.11ah (HaLow), utilisant des fréquences publiques dans la bande des 900 MHz, plus basses que le WiFi classique, passe mieux les obstacles et permet d'assurer la basse consommation (à un débit réduit). Il a été conçu comme concurrent direct du BLE ou du LoRaWAN. L'avantage du LoRaWAN reste qu'il s'agit d'un WAN dont l'opérateur peut être global et communautaire (voir ci-après). Le LTE-m utilise le réseau 4G, et la 5G IoT ne sera pas déployée tout de suite. Le BLE peut aussi être exploité au sein d'un Bluetooth Mesh Network, jusqu'à 1000 m, suivant les types de relais.

FIGURE 5.4 – Comparatif des technologies de terrain sans fil (voir [21])

Dans les réseaux de smart-meters sans-fil, les équipements raccordés sont souvent à faible puissance et ont une grande autonomie (voir figure 5.4 en page 69). Un élément important est donc de pouvoir stocker les messages dans des noeuds intermédiaires alimentés, que les équipements puissent consulter au besoin, sans maintenir une veille radio permanente.

5.4.2 A courte distance

Il existe des technologies de réseaux de terrain filaires comme le bus temps réel **CAN**, utilisé pour la mesure, le contrôle et la commande (moteurs, axes, etc), à courte distance, par exemple dans une voiture.

Pour interconnecter des automates et des équipements de terrain pour la supervision (mesure et contrôle), on peut utiliser des technologies comme **Modbus** ou **Profibus**, qui peuvent être étendues en distance et en compatibilité en les transportant sur TCP et sur Ethernet ou WiFi.

En sans-fil, il existe aussi les protocoles **Zigbee**, **BLE** (*Bluetooth Low Energy*) et **Z-Wave**, aussi pour la mesure et le contrôle.

Citons enfin le protocole **6LoWPAN**, utilisé dans des réseaux de terrain sans-fil IPv6 à basse consommation et courte distance, qui applique de la compression d'entête (pas au sens entropique : standard SCHC, *Static Context Header Compression*) utilisant le contexte particulier (entêtes toujours très similaires, ce qui permet de réduire l'entête IPv6 et UDP à un identificateur très court, voir https://en.wikipedia.org/wiki/Static_Context_Header_Compression).

5.4.3 A moyenne et longue distance

On peut toujours installer des gateways ad-hoc, reliant des réseaux de terrain à caractéristiques très variées à un réseau plus étendu (VPN sur Internet par exemple), les gateways étant elles-mêmes reliées grâce à diverses technologies d'accès classiques ou spécifiques (p.ex. modem radio 400 MHz).

Il existe toutefois plusieurs protocoles permettant la mesure et le contrôle à moyenne et grande distance. Un des plus intéressants actuellement est le **LoRa**, une technologie propriétaire à étalement de spectre. Ses fréquences étant libres (publiques) et partagées, les puissances sont faibles : les messages échangés entre deux équipements sont de petite taille et ne peuvent pas être très fréquents, de manière à ne pas dépasser le maximum de temps de transmission journalier légal. Le problème s'aggrave bien sûr si les messages à transmettre sont volumineux ou si la distance est grande (ce qui nécessite d'utiliser des modes de transmissions plus lents).

5.4.3.1 LoraWAN

Une extension du concept mène au **LoRaWAN**, où des **gateway**s sont installés pour intégrer les équipements LoRa à un réseau mondial (WAN), transporté par exemple sur Internet. Ces gateways peuvent même stocker des messages descendants de manière à limiter la nécessité d'un mode veille radio sur les équipements : ils seront lus à la prochaine transmission montante.

Ces gateways peuvent être privés [39], offerts par un fournisseur de télécommunications contre rémunération comme Swisscom [40], ou gérés au sein d'un réseau coopératif international relié à Internet, comme **TTN** [41]. Dans ce dernier cas, l'avantage est qu'un mobile pourra toujours communiquer avec l'application Internet, où qu'il soit dans une zone géographique très importante (toute l'Europe, voire le monde entier en respectant les fréquences et restrictions spécifiques).

39. contrebalançant ainsi l'avantage compétitif du **network slicing** de la 5G
40. Sigfox, lui, ne permet pas des gateways propres ou communautaires
41. *The Things Network* – il limite actuellement dans les meilleures conditions à environ 500 messages vers l'application (mesures, *uplink*) et à dix messages de l'application vers l'équipement embarqué (*downlink*, contrôle) par jour

En général, ces réseaux sont liés à des services cloud spécifiques (intégration d'équipements, stockage, traitement des données, alarme), et des cartes de couverture sont disponibles [42].

5.4.3.2 Apple AirTag

L'iPhone est un smartphone relativement populaire, en particulier dans les pays développés : mondialement c'est 28%, face à Android qui totalise 71% des OS mobiles [43]. Apple a donc développé un réseau de terrain propriétaire mondial qui utilise des iPhones de tiers qui communiquent en **UWB** [44] localement avec les objets connectés compatibles, puis relaient [45] ensuite les informations via Internet au smartphone et aux applications de l'utilisateur.

Malgré la non-interopérabilité et les risques de tracking, Apple a réussi à faire accepter sa solution, grâce à des fonctions de respect sphère privée [46].

42. pour TTN, voir https://ttnmapper.org/
43. https://gs.statcounter.com/os-market-share/mobile/worldwide, 2023
44. *Ultra Wide Band*, une technologie de transmission par étalement de spectre avec des objets connectés (surtout pour des fonctions de localisation, avec des piles durant de l'ordre d'un an) jusqu'à 50 à 200 mètres – les smartphones modernes peuvent le supporter
45. il est possible de configurer son iPhone pour ne pas participer au réseau (*opt-out*), mais alors on ne peut plus utiliser d'objets connectés AirTag
46. https://edition.cnn.com/2021/05/05/tech/airtags-apple-privacy-concerns/index.html

Chapitre 6

Hiérarchie des systèmes numériques

Sommaire

6.1 Aspects historiques

6.1.1 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 2.3.2 en page 9 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 le débit disponible et la fiabilité sont suffisants – est le codec **G.711** [1].

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

6.1.2 ATM : Asynchronous Transfer Mode

Pas traité en détail cette année

La question sous-jacente était celle de l'intérêt du mariage d'un réseau de paquets de données asynchrones[2] et d'un réseau de circuits de données synchrones. L'émergence des applications multimédias a rendu ce genre de réseau couche 2 indispensable.

Une solution possible est **ATM** : Asynchronous Transfer Mode [6]. 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 48[3] 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.

6.1.2.1 Couche liaison : sous-couche AAL

Ce qui a rendu ATM un temps très utile au transport de données multimédia est sa demi-couche d'adaptation ATM, qui 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.

2. 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.).

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

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) :

Notons qu'AAL5 est une encapsulation multi-cellule, dans laquelle les informations de contrôle se trouvent dans la dernière cellule. Elle est aussi utilisée dans les diverses variantes de télévision numérique (DVB-C/T/S) et de radio numérique (DAB/DAB+) comme format de super-trame de multiplexage temporel.

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 au débit 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.

6.1.2.2 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).

6.1.2.3 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 débit, 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** (*Multiprotocol Label Switching*), une technologie mixte couche 2/3 avec intégration d'IP, très générique, à commutation de paquet de couche 2 basée sur des labels, utilisé dans la plupart des implémentations actuelles de réseaux **core** des opérateurs pour implémenter des circuits virtuels à qualité de service.

6.2 Hiérarchies numériques classiques

6.2.1 Introduction

Une hiérarchie numérique est un ensemble de protocoles permettant de transmettre de manière numérique des informations (voix, vidéo, données) à différents débits, en combinant des affluents de différents niveaux. 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-continentales 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 possible au sein de SDH.

6.2.2 Hiérarchique 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 ou analogiques 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.

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

Niveau	nom	vitesse $[\frac{Mbit}{s}]$
0	E0	1 canal à 64 $\frac{kBit}{s}$ (p.ex. ISDN BRI)
1	E1	30 canaux sur 2.048 MBit/s (p.ex. ISDN PRI)
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

FIGURE 6.1 – Niveaux de la hiérarchie PDH

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 $\frac{Gbit}{s}$	

FIGURE 6.2 – Niveaux de la hiérarchie SDH

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.

6.2.3 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 μs).

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 sur DSL remplace les niveaux E0 et E1 de PDH.

Enfin, si SDH peut transporter **ATM**, c'est également le cas de **MPLS** : Swisscom a par exemple interfacé ses technologies ATM, SDH et Ethernet dans un grand réseau central (core) unique à base de technologie MPLS.

6.2.4 Obsolescence des technologies PDH et SDH

Dans les pays développés, on prévoit la *fin* du support des hiérarchies numériques classiques basées sur des canaux voix à 64 kbit/s (ISDN et PDH) entre 2016 et 2020. En effet, le développement de ces technologies est stoppé depuis quelques années et les services seront peu à peu intégrés dans du tout IP (**NGN-IMS**, *IP Management System*, voir section 6.3.5.3 en page 80), avec réseau d'accès haut débit **LTE** pour le mobile et fibre **FTTx** ou **VDSL** à 1 GBit/s chez l'abonné.

En particulier en Suisse et en Allemagne, l'ensemble des technologies PDH et téléphonie classique TDM SS7 (analogique et ISDN/RNIS) va être désactivée par les opérateurs nationaux historiques d'ici 2017. Les abonnés au téléphone fixe analogique pourront passer en voix-sur-IP grâce à une liaison VDSL. Les entreprises, qui utilisent encore souvent l'ISDN en externe et la voix-sur-IP en interne, se verront offrir des passerelles voix-sur-IP intégrant leurs centraux existants, ou une migration à une solution de central hébergé de type Swisscom Business Connect (anciennement Centrex VoIP).

Quant à SDH, on voit de plus en plus la migration à des technologies du monde IP comme **MPLS** sur **GBit Ethernet**, ce qui pose des nouveaux problèmes liés à la synchronisation du temps hors de la couche physique (voir section 6.3.7 en page 82).

Le coeur du réseau des opérateurs sera donc exclusivement et tout IP, transporté par **MPLS** et organisé au sein d'un réseau **NGN** (**IMS**, *IP Multimedia Service*), avec éventuellement des convertisseurs pendant la période de transition en bord du coeur (téléphonie 3G, voire ISDN et analogique SS7 pour les pays n'abandonnant pas ces technologies rapidement)

6.3 Hiérarchie des réseaux d'opérateurs

6.3.1 Introduction

Les réseaux d'opérateurs modernes font face à de nombreux défis : citons notamment l'intégration des anciennes technologies de hiérarchies numériques (PDH, SDH), l'intégration de services classiques (téléphonie analogique et ISDN, GSM, ...), l'intégration avec les protocoles de type Internet, la qualité de service, le transport de données y compris d'opérateurs tiers, etc.

Les solutions actuelles sont basées sur un mix technologique avec une interconnexion au sein du réseau **core** de l'opérateur, alors que les terminaux sont connectés via un **réseau d'amenée** multi-technologies : le **backhaul**.

L'intégration définitive des anciens (réseau téléphonique classique, ISDN, 3G, 4G-LTE) et nouveaux services avec Internet nécessitera la mise en place de réseaux de nouvelles génération (**NGN**).

6.3.2 Hiérarchisation

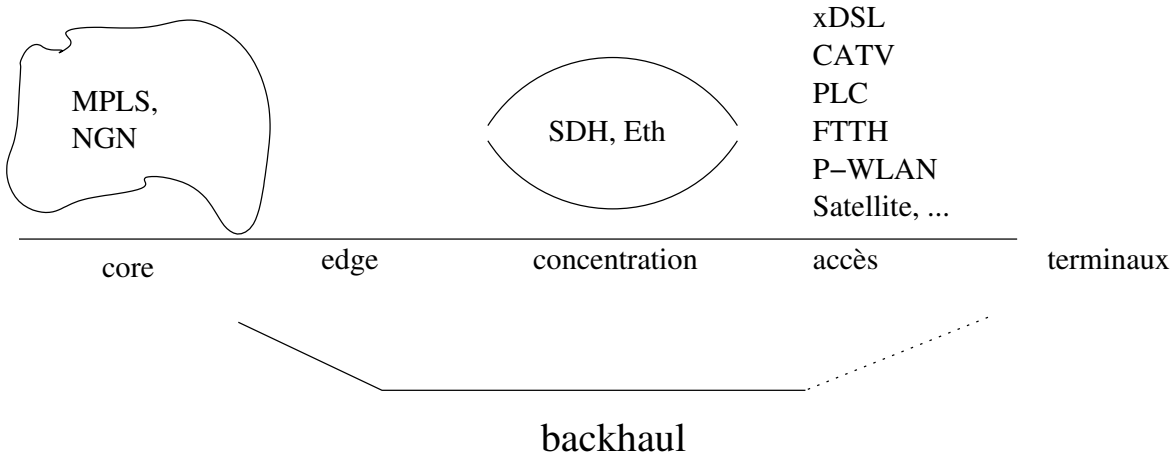


FIGURE 6.3 – Hiérarchie d'un réseau d'opérateur

On distinguera donc entre réseau **core**, de bord (**edge**), **de concentration** et **d'accès**.

La plupart du temps, aujourd'hui, la problématique technologique est différente entre **core**, **backhaul** (acheminement : relie les terminaux au réseau core) et éventuellement **accès** (dernier kilomètre) où la variation de technologies est la plus grande.

6.3.3 Accès

On retrouve ici les technologies d'accès principales traitées au chapitre 5 (dernier kilomètre).

6.3.4 Backhaul

Le réseau backhaul (littéralement : réseau d'acheminement ou **middle-mile**) relie les équipements terminaux aux points de présence (**POP**) de l'opérateur : aujourd'hui on considère qu'il s'étend en général entre le réseau d'accès et le réseau de bord.

Sa définition est rendue nécessaire notamment vu l'existence de réseaux de transports permettant l'offre de service par des opérateurs tiers.

Ses technologies sont variées (FTTH/C/B, Ethernet, SDH, ATM, xDSL, ...) mais l'on constate tout de même une standardisation vers les protocoles Ethernet (y compris qualité de service et VLAN) souvent exploités en **topologie anneau**, ce qui semble évident vu l'ubiquité du protocole IP.

6.3.5 Core

6.3.5.1 Rôles

Le réseau core d'un opérateur est le système d'interconnexion des différents backhaul. Il supporte des débits très élevés et de la qualité de service. Il est en général implémenté de manière redondante.

La solution actuelle pour les réseaux core est souvent MPLS (voir section 6.3.5.2 en page 80) et les problématiques typiques d'opérateur voulant intégrer ses télécommunications classiques

à Internet tout en offrant la palette de service généralement attendue seront résolues par les réseaux de prochaine génération (NGN, voir section 6.3.5.3 en page 80).

6.3.5.2 MPLS

ATM, du monde OSI, a résolu quelques problèmes des opérateurs concernant le transport efficace en **mode connecté** de données mixtes (voix, informatique) à grande distance et grand débit, tout en offrant de la qualité de service sur une infrastructure finançable. Ses idées de commutation rapide en couche 2 basées sur des étiquettes (VPI, VCI) potentiellement échangeables à chaque commutateur ont été à la base du développement d'un autre protocole, défini par l'IETF, qui est entrain de le remplacer : MPLS. En effet, ATM s'intégrait de manière complexe à IP et travaillait en mode connecté.

L'idée principale de MPLS (*Multiprotocol Label Switching*), est de reprendre la notion d'étiquettes d'ATM [4] tout en la généralisant, et de baser sa commutation sur la notion de *détection de flux* plutôt que connexion. Cela permet, aux bords du réseau MPLS, d'exploiter des routeurs-commutateurs (**LER**, *Label Edge Router*), qui, agissant comme des routeurs IP et des commutateurs MPLS, vont détecter des flux en fonction de critères (p.ex. qualité de service, adresses, ports, etc) et leur assigner des étiquettes (**label**s) qui seront utilisées et échangées par chacun des commutateur MPLS à l'interieur du réseau (**LSR**, *Label Switching Router*) [5].

L'arrivée de MPLS a été aussi un symptôme révélant que le monde IP avait largement gagné face aux monde des opérateurs télécoms (ITU-T). La plupart des développements modernes des opérateurs (p.ex. NGN), sont maintenant basés sur des technologies ouvertes du monde IP.

6.3.5.3 NGN

Les opérateurs de télécommunications, représentées par les organismes de standardisation comme l'ITU ou l'OSI n'ont pas forcément les mêmes intérêts commerciaux ou besoins que les opérateurs de réseaux Internet (voir section 6.3.5.2 en page 80), rien que par le type d'information transportée ou les modèles de facturation, qui peuvent s'opposer à la **neutralité du réseau**.

C'est pour cette raison que les protocoles de télécommunications classiques ont longtemps évolué parallèlement aux protocoles IP. Aujourd'hui, il devient de plus en plus coûteux pour les opérateurs de télécommunications historiques – devenus entre temps entre autre des fournisseurs d'accès Internet (**FAI**) en d'offrant simultanément des services **quadruple-play** (télévision, téléphone fixe et mobile, Internet) – de développer et d'exploiter deux infrastructures parallèles et de ne pouvoir facilement intégrer les produits proposés à leurs clients.

De plus, les réseaux IP généraux n'offrent pas les garanties suffisantes aux opérateurs télécoms classiques, notamment en ce qui concerne :

- la sécurité
- la facturation
- la qualité de service
- la mobilité (roaming)

Même s'il est possible au sein d'un réseau restreint d'opérateur (en particulier avec un réseau core **MPLS**) d'assurer ces besoins via des outils développés en interne et du personnel qualifié, le coût de maintenance peut devenir prohibitif, et l'interaction entre plusieurs opérateurs rendue

4. ce qui permet d'interfacer MPLS et ATM – encore que SDH ou Ethernet sont des candidats potentiellement plus intéressants

5. IPv6 a un support direct de ces labels de flux.

difficile. Enfin, les anciennes technologies (GSM 3G, téléphonie **SS7**[6] analogique ou ISDN) ne seront que difficilement intégrables, car elles n'utilisent pas le réseau core **MPLS**.

La vraie réponse est une intégration complète et standardisée de ces deux mondes : en effet, les grands réseaux IP sont en général intégrés et transportés par des technologies hybrides couche 2/3 (réseau **core**), comme p.ex. **MPLS** (un standard **IETF** qui dérive à la fois des techniques d'ATM et des résultats du monde IP) ; l'accès à la télévision par **VDSL** se base sur des standards IP et la téléphonie IP pourra remplacer à terme la téléphonie **SS7** analogique ou ISDN qui n'est plus développée.

Les *Next Generation Networks* (NGN) sont une évolution d'architecture de télécommunications visant à cette intégration. Leurs caractéristiques principales sont :

- transporte toutes les données et services sous forme paquet
- sont construits sur la pile de protocole IP (**all-IP**) – dans la plupart des cas
- touche les réseaux **core** et les réseaux **d'accès**
- interface les réseaux historiques (téléphonie SS7 analogique et ISDN, GSM) via des passerelles (**gateway**)
- utilise toutes les technologies d'accès possibles
- met en place de la qualité de service (**QoS**) via des contrats (**SLA**) garantis par l'infrastructure
- offre des services de mobilité (**roaming**)
- offre des plateformes d'identification et de facturation généralisées
- ajoute des services de chiffrement et d'authentification, tout en respectant les lois locales concernant les services de surveillance des télécommunications (pas de chiffrement de bout en bout des services et données de base du réseau) et assurant une confidentialité complète (données et entêtes) par *tronçon*.

Les NGN capitalisent sur de nombreux protocoles existants d'Internet comme IP, TCP, UDP, SIP, H.323, **RADIUS**/**Diameter**[7], **IPsec** ainsi que des technologies comme l'isolation (**firewall**, notion de zones restreintes, etc). Il s'y ajouteront des services d'activation multimédia (par exemple sous forme de services Web) et de passerelles entre technologies et des protocoles propres de signalisation NGN-IMS[8].

Le déploiement des NGN se fait en deux phases :

1. **NGN light** ou partiel : le NGN est déployé au sein d'un ou plusieurs opérateurs, mais les échanges entre opérateurs sont toujours effectués à l'aide de protocoles classiques comme **SS7**
2. **NGN full** : au niveau international, de plus en plus d'opérateurs exploitent uniquement les protocoles all-IP au sein des NGN.

Ce n'est qu'en déploiement complet que les avantages des NGN seront perceptibles, soit la qualité de service négociable de bout en bout sur Internet, la mobilité complète de couche 3 et le chiffrement par tronçon (assurant confidentialité par défaut et respect des lois nationales de la surveillance des télécommunications).

L'architecture Internet futuriste **SCION**[9] veut permettre un déploiement partiel entre systèmes autonomes (**AS**) participants, assurant une résistance aux pannes, la sécurité et la performance sur les NGNs, même en présence d'agents malveillants dans les équipements et chez

6. encore utilisé aujourd'hui pour la signalisation de l'interconnexion téléphonique entre opérateurs, peu sécurisé même lorsque transporté sur IP

7. protocoles **AAA**, *Authentication, Authorization, Accounting/Auditing* : **authentification**, **contrôle d'accès** et **traçabilité**

8. remplaçant SS7

9. *Scalability, Control and Isolation On NGNs*, https://www.scion-architecture.net/

les opérateurs. Elle fait partie des projets exploratoires de l'IETF dans le domaine des réseaux *path-aware* et est déjà en test dans des institutions suisses.

Le triangle fournisseurs de contenu et de services (notamment les GAFAM), utilisateur et opérateurs télécoms modélise bien des intérêts divergents : la **neutralité du réseau** est centrale pour les fournisseurs de contenu et de services, la sécurité et la sphère privée essentielle pour les utilisateurs et les services offerts et leur facturation pour les opérateurs. Ce sont les interactions au sein de ce triangle qui décideront si et quand les NGN seront réellement exploités en NGN full.

6.3.6 Qualité de service

Le problème principal dans l'exploitation de qualité de service au sein d'un grand réseau d'opérateur est la correspondance inter-couches. En effet, si au sein d'un réseau interne Ethernet, il est facile de configurer de la qualité de service en couche 2 (**802.1q**) sur les équipements de commutation (switches Ethernet *manageables*), ou en couche 3 sur les routeurs (**DIFFSERV** et/ou anciens bits **TOS**), il faut ensuite assurer que les qualificateurs configurés en couche 3 sont reportés en couche 2, même si cette couche 2 change au travers du réseau d'opérateur (Ethernet, xDSL, boucle Ethernet, MPLS, ...).

Un autre problème est la vérification que le client a bien commandé le SLA qu'il utilise (ce qui correspond à la fonction **UPC** d'ATM) : cela doit se faire le plus près possible de l'utilisateur. On peut par exemple imaginer un effacement des bits TOS par le le routeur xDSL de l'abonné – sous contrôle de l'opérateur, ou sinon sur le premier routeur de ce dernier – si la prestation n'a pas été commandée.

Notons que la qualité de service à travers tout Internet ne sera réellement rendue possible qu'avec des mises en place globales de technologies comme **NGN**. Pour le moment, les bits TOS provenant d'Internet doivent être effacés ou traités comme moins prioritaires que le trafic sous SLA/QoS des utilisateurs internes. Une technique parfois utilisée est d'utiliser les bits de poids fort du champ **DIFFSERV** de l'entête IP comme indicateurs du **SLA** (*Service Level Agreement*) acheté, et les bits inférieurs comme les anciens bits TOS (délai minimisé, débit optimisé, taux de perte minimisé).

6.3.7 Synchronisation

Pas traité en détail cette année

La synchronisation d'horloges peut poursuivre plusieurs buts :

- synchronisation temps réel absolue : par exemple pour l'horodatage de transactions informatiques
- synchronisation relative ou absolue : interprétation des moments de transmission au bon instant, respect de slots de transmission TDM statique, positionnement GPS, etc.

On peut synchroniser en temps, en fréquence, ou en phase. La plupart du temps les synchronisations informatiques se font en temps ; les synchronisations de transmissions de données se font en fréquence, et parfois en phase notamment pour certains types de systèmes TDM (p.ex. en LTE lorsque le streaming vidéo est déployé).

Dans les hiérarchies numériques classiques (voir section 6.2 en page 76), la synchronisation fait souvent partie de la signalisation de la couche physique, avec parfois des informations de gestion dans les trames couche 2 (p.ex. avec SDH) : on peut parler de **synchronisation par canal** (physical layer).

Dans les hiérarchies numériques actuelles des opérateurs, il est toujours possible d'utiliser la synchronisation inhérente aux protocoles sous-jacent (p.ex. **SDH**), dans la mesure où l'on ne transporte pas des données d'autres opérateurs (dans ce cas une synchronisation globale n'est pas envisageable). On peut aussi utiliser de nouveaux protocoles de synchronisation couche physique Ethernet (**SyncEthernet**) qui tirent parti de l'utilisation de plus en plus universelle de protocoles **Ethernet** pour le **backhaul**, en raison de l'intégration facilitée avec IP et de la mise en place de qualité de service (**802.1q**). SyncEthernet ne supporte que la synchronisation en fréquence.

Une alternative indépendante des couches inférieures est la **synchronisation par messages** (packet-based) : l'utilisation de messages de couche 4 (datagrammes UDP sur IP) permettant la synchronisation. Le protocole le plus connu est **NTP** (*Network Time Protocol*), mais il ne permet pas une synchronisation à la précision requise ($<$ 1 us). Le protocole **PTP** (*Precision Time Protocol*), couplé à du support dans les équipements de commutation permettant de modifier les datagrammes couche 4 en transit (*transparent clock*) et d'y insérer des informations de délais effectifs, peut atteindre la précision requise, dans la mesure où les délais peuvent être estimés symétriques et constants durant les fenêtres de synchronisation. Un des avantages du PTP est la possibilité, si les équipements de commutation le supportent (*boundary clock*), de pouvoir supporter la synchronisation en phase.

Chapitre 7

Transmission sans fil

Sommaire

Les réseaux sans fils sont de plus en plus utilisés, à la fois comme prolongation du réseau local (LAN), pour des liaisons point à point ou point à multipoint ou multipoints en visibilité directe, dans la boucle locale sans fil (**WLL**, voir section 5.3.5 en page 64), mais également pour les réseaux de terrain, voir section 5.4 en page 69.

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 point à point plus éloignées qu'en utilisation classique, tout en donnant quelques pistes pour les réseaux classiques WiFi (intra-muros ou intra-campus) [11].

7.1 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 (voir section 5.3.6 en page 67)
— **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**, voir section 5.3.5 en page 64 et pour les réseaux de terrain la section 5.4 en page 69) : 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 [1] distance pour lesquels, notamment, la courbure terrestre peut être négligée. Les réseaux sans-fil régionaux et les liaisons point à point sont des cas d'application.

7.2 Calcul de liaison pour des faisceaux hertziens courts

7.2.1 Facteurs limitatifs

On doit tenir compte des facteurs suivants :

— puissance d'émission de l'équipement
— **rapport signal sur bruit** et sensibilité de 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
— normes légales sur la puissance à l'antenne

7.2.2 Niveaux et puissance

On définit un niveau absolu (en anglais : *level*), en décibels [2], comme un rapport logarithmique de la puissance à une puissance de référence :

$$L_x = 10 \ \log_{10} \frac{P_x}{P_{ref}} \tag{7.1}$$

Classiquement, cette puissance de référence est fixée à 1 milliwatt (soit 0 dBm). Un niveau absolu se notera alors en **dBm**.

Dans un bilan de liaison, on calculera le niveau de de réception en sommant en décibels le niveau d'émission et les gains d'antenne et en soustrayant les différents affaiblissement. On comparera ensuite le niveau de réception au niveau de sensibilité de réception pour déterminer la faisabilité de la liaison. Noter que plus le niveau de sensibilité de réception est faible, meilleure est l'adaptatibilité à de grands affaiblissements.

On s'intéressera aussi au niveau à l'antenne d'émission pour respecter les normes légales.

1. quelques centaines de mètres à quelques kilomètres
2. déci = facteur 10

7.2.3 Affaiblissement

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

$$A = 10 \ \log_{10} \frac{P_E}{P_R} = L_{emission} - L_{reception} \tag{7.2}$$

7.2.4 Bilan de liaison

Le **bilan de liaison** permet de vérifier la faisabilité théorique d'une liaison qui par ailleurs ne souffre pas de problèmes (visibilité directe assurée, obstacles suffisamment lointains[3], perturbations d'autres transmissions minimes, ...).

En effet, ce qui importe ici est que le récepteur reçoive suffisamment de signal pour qu'il puisse le comprendre. Le niveau de sensibilité de réception, qui dépend du matériel de réception considéré, est la borne inférieure qui permet de déterminer si la liaison est faisable : si le niveau du signal reçu est inférieur au niveau de sensibilité de réception, la liaison n'est pas faisable.

Le calcul est simplifié par le fait que si l'on indique les diverses puissances sous forme de niveaux en **dBm** et les affaiblissements ou gains en **dB**, seules des additions ou soustractions doivent être utilisées.

L'affaiblissement total sur une liaison est donc la somme des affaiblissements[4] :

$$A = A_{cables} + A_{antennes} + A_{espacelibre} + A_{obstacles} \tag{7.3}$$

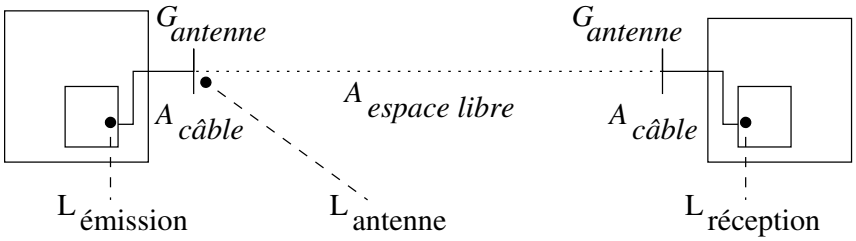


FIGURE 7.1 – Bilan de liaison

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

$$L_{reception} = L_{emission} - A \geq L_{sensibilite_{reception}} \tag{7.4}$$

3. voir l'ellipsoïde de Fresnel section 7.4 en page 92

4. $A_{antennes}$ vaut $-G_{antennes}$ (avec $G_{antennes} = 2\ G_{antenne}$ pour une liaison symétrique), car il s'agit souvent d'un gain, plus qu'un affaiblissement ; $A_{obstacles}$ est estimé zéro en cas de visibilité directe et lorsque les obstacles résiduels sont suffisamment éloignés ; et bien sûr A_{cables} vaut la somme des affaiblissements de tous les câbles d'antennes.

Le niveau de réception vaut le niveau d'émission moins la somme de l'affaiblissement de la liaison, et il doit être supérieur ou égal au niveau de réception minimal du matériel considéré, que l'on appelle la sensibilité de réception.

7.2.5 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 ou 5 GHz). Rappelons que ces plages de fréquences ne sont pas soumises à autorisation [5] et que de plus en 2.4 GHz, 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 2.4.6 en page 17).

7.2.6 Limitation de la puissance rayonnée

7.2.6.1 Motivation

On pourrait penser qu'il suffit d'augmenter la puissance d'émission – ou le **gain** des antennes – pour compenser tout affaiblissement. Si c'est théoriquement juste, c'est, en pratique, interdit par la loi pour deux raisons :

la cohabitation d'utilisateurs : l'usage de bandes de fréquences partagées (dites **publiques**, sans concession), en particulier en 802.11, limitent la puissance maximum à l'antenne : respectivement à 100 mW (20 dBm) en 2.4 GHz et 1 W (30 dBm) à 5 GHz [6] – attention, ces normes peuvent avoir des variations nationales en fonction du canal considéré

la santé publique : on s'intéresse ici aux champs électriques cumulés à un endroit donné (p.ex. un appartement), ou à l'absorption d'onde dans un tissu vivant ; en règle générale les puissances du WiFi sont largement en-deçà de ces normes ; toutefois l'accumulation d'émetteurs de divers types ou l'usage prolongé de téléphones sans-fil (GSM ou DECT) pourrait les dépasser. En habitation, la limite de rayonnement **ORNI** [7] s'applique. Le champ électrique ne doit pas dépasser 6 Volts par mètre. Les équipements utilisés près du corps humain sont évalués eux en **DAS** (Débit d'absorption spécifique, en Watts par kg, qui indique l'effet chauffant d'un téléphone p.ex.)

7.2.6.2 Puissance rayonnée équivalente

Dans tous les cas, on considérera non pas la densité moyenne de puissance à 360 degrés, mais bien la puissance équivalente d'une **antenne isotropique** [8] (émission uniforme dans toutes les directions, sans gain dans un secteur). En clair, on tient compte de la puissance maximum à

5. c'est pour éviter les interférences que la puissance **EIRP** est limitée à 100 mW en 2.4 GHz et 1 W en 5 GHz.

6. l'affaiblissement en espace libre dépend de la fréquence : l'avantage de puissance est vite contrebalancé par la distance ; d'un autre côté la bande des 5 GHz est moins polluée et dispose de beaucoup plus de canaux non recouvrants, même si elle comprend aussi certains usages spéciaux comme le radar, pour lequel le protocole spécial **DFS** a été conçu

7. Ordonnance suisse sur la protection contre le rayonnement non ionisant.

8. l'unité **dBi**, pour dB isotropique, permet de faire la différence entre une puissance dans un secteur ou une puissance isotropique.

l'antenne, autrement dit la puissance dans le secteur où l'antenne est la plus directionnelle : **PIRE** (puissance isotrope rayonnée équivalente, en anglais **EIRP**).

Donc, attention : contrairement au bilan de liaison qui s'intéresse au niveau à l'émetteur et au récepteur, le rayonnement maximum autorisé concerne le niveau maximum à l'antenne :

$$L_{antenne} = L_{emission} - A_{cable} + G_{antenne} \tag{7.5}$$

7.2.6.3 Principe de précaution

Pour le moment, les études ne *confirment pas*[9] de manière reproductible les effets nocifs d'une exposition *normale*[10] à la pollution électromagnétique. Les personnes qui se disent hypersensibles mentionnent la valeur du champ électrique et les trains d'ondes pulsés comme cause de leurs problèmes. Même sans preuves scientifiques formelles, le principe de précaution peut s'appliquer : il n'est pas difficile de limiter l'exposition et donc le risque potentiel avec quelques moyens simples, comme par exemple éloigner les émetteurs (y compris les appareils électriques de votre lit) ou désactiver les consommateurs inutiles.

Pour le sans-fil on commencera par calculer la puissance maximum à l'antenne et l'on vérifiera qu'elle ne dépasse pas la norme légale de puissance maximum de la bande de fréquence considérée (voir section 7.2.6.1 en page 88). C'est le plus important, et dans le cas du WiFi c'est très probablement suffisant pour limiter le risque.

Pour des émetteurs concessionnés pour lesquels la puissance maximum est beaucoup plus élevée (GSM p.ex.), ou par principe de précaution, on effectuera en plus des mesures du champ électrique dans les habitations très proches des antennes, pour détecter un éventuel effet d'accumulation : on se rappellera toutefois que les émetteurs situés à l'intérieur de ces habitations (téléphones DECT et GSM surtout) seront responsables de la majorité de la pollution électromagnétique et devront être adaptés en priorité pour une réelle réduction du risque potentiel, surtout que les antennes des opérateurs sont souvent montées à plus d'une centaine de mètres de toutes habitations.

Dans tous les cas des études scientifiques actuelles[11], l'effet déterminant est un effet chauffant qui a toujours été considéré comme bénin : toutefois lorsque le téléphone est utilisé très près d'organes (oreilles, yeux, voire cerveau), le principe de précaution suggère alors d'acheter un téléphone avec une valeur DAS faible (voir 7.2.6.1 en page 88), de ne pas abuser du téléphone sans-fil en particulier lorsque le signal est mauvais et d'utiliser une oreillette, ce qui devrait limiter le risque – même si une oreillette filaire est une antenne et une oreillette Bluetooth est elle-même un dispositif émetteur.

7.2.6.4 Calcul de champ électrique

Le champ électrique ($\frac{V}{m}$) à une distance r de l'antenne isotropique équivalente (**EIRP**) dépend bien sûr de la densité de puissance ($\frac{W}{m^2}$) du signal à cet endroit.

Par définition, on a les unités SI suivantes :

9. prouver l'inexistence d'un phénomène est difficile, voire impossible
10. par principe de précaution, on considère qu'une exposition normale doit être largement inférieure au maximum autorisé
11. voir par exemple les publications de l'OMS [20]

$1\ V$	$1\ \frac{kg\ m^2}{A\ s^3}$
$1\ W$	$1\ \frac{kg\ m^2}{s^3}$

et donc une étude dimensionnelle montre que $\frac{V}{m} = \frac{kg\ m}{A\ s^3}$ et $\frac{W}{m^2} = \frac{kg}{s^3}$ et donc en remplaçant que $\frac{V}{m} = \frac{W}{m\ A}$.

Avec $U = RI$ ou $I = \frac{U}{R}$ et donc en unités $\frac{V}{\Omega} = A$ [12], on peut réécrire l'expression en $\frac{V}{m} = \frac{W}{m\frac{V}{\Omega}}$.

Enfin, en multipliant par $\frac{V}{m}$ de chaque côté on obtient alors $\frac{V^2}{m^2} = \frac{W}{m^2}\Omega$.

On en déduit facilement la formule suivante :

$$E^2 = D\ R \tag{7.6}$$

avec :

- E : champ électrique en volts par mètre,
- D : puissance rayonnée par unité de surface ($\frac{W}{m^2}$) à la distance r de l'antenne en mètres (densité de puissance à distance r)
- R : impédance du milieu de propagation en ohms : c'est aussi le rapport entre le champ électrique ($\frac{V}{m}$) et le champ magnétique ($\frac{A}{m}$)

On suppose que l'on a des distances *suffisamment grandes* ($r >> \lambda$, la longueur d'onde du signal) – sinon de toute façon la notion de gain directionnel d'antenne n'est pas possible. Consultez également [19] pour d'autres simplifications sous-entendues ici.

En utilisant quelques résultats de physique, on peut rendre la formule plus simple à utiliser : comme $E = vB$ (B est l'induction magnétique, v la vitesse de propgation dans l'air soit $3 * 10^8\ \frac{m}{s}$) et $H = \frac{B}{\mu\mu_0}$ (avec μ_0 la permittivité du vide soit $4\pi * 10^{-7}$ et $\mu = 1$ la perméabilité de l'air), alors $R = \frac{E}{H} = 120\pi$ soit environ 377 ohms (impédance de l'air).

De plus, comme $D = \frac{P}{4\pi r^2}$ (puissance par unité de surface d'une sphère de rayon r, centrée à l'antenne), on peut alors réécrire l'équation 7.6 comme suit :

$$E = \sqrt{\frac{P}{4\pi r^2}120\pi} = \sqrt{30\frac{P}{r^2}} = \frac{\sqrt{30P}}{r} = \frac{\sqrt{0.03}}{r}10^{\frac{L}{20}} \tag{7.7}$$

En généralisant aux antennes à gain, cela nous donne le champ électrique E en $\frac{V}{m}$ à distance r d'une antenne dont la puissance calculée à l'antenne, dans ce secteur, est P (en Watts), ou L en dBm.

7.2.6.5 Exemple de calcul de champ électrique

On s'intéresse à une évaluation du champ électrique à $r = 20\ m$ d'une antenne de gain $G_a = 10\ dB$, reliée à un émetteur à $L_e = 14\ dBm$. Le champ est mesuré dans la direction prépondérante du gain (secteur à puissance maximum, **EIRP**).

Le niveau maximum à l'antenne est donc $L_a = L_e + G_a = 24\ dBm = 24\ dBi$ et la puissance correspondante est $P_a = 251\ mW$.

12. l'unité de R est l'ohm, noté Ω

La formule 7.7 nous donne : $E = \frac{\sqrt{30P_a}}{r} = \frac{\sqrt{0.03}}{r}10^{\frac{L_a}{20}} = 0.13\ \frac{V}{m}$ ce qui est largement inférieur à $6\ \frac{V}{m}$.

Conclusion : en règle générale, les émetteurs extérieurs ne contribuent pas de manière sensible au smog électromagnétique que l'on peut trouver en habitation, surtout que les murs ou vitres affaiblissent encore le signal reçu : il faut plutôt s'inquiéter des émetteurs situés à l'intérieur de l'habitation, en particulier des téléphones portables lorsque le niveau de signal reçu est mauvais, ou des téléphones DECT, ou encore tout équipement électrique classique (radio-réveil, lampes dites économiques). Toutefois, une distance de sécurité suffisante (30 cm à 1 m) suffit à réduire le risque de manière très significative.

7.3 Affaiblissements

7.3.1 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 :

$$A_{lin} = \alpha d \tag{7.8}$$

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).

7.3.2 Affaiblissement en espace libre

L'affaiblissement en espace libre (il s'agit donc d'une borne minimale) selon Friis [16] est :

$$A_{espacelibre} = 20\ \log_{10}\frac{4\pi d}{\lambda} = 20\ \log_{10}\frac{4\pi df}{c} \tag{7.9}$$

où :
c : vitesse de la lumière
d : distance en mètres
f : fréquence en Hz (ou λ : longueur d'onde)

Notons que cette formule n'est bien sûr valable que pour des faisceaux hertzien, donc quand la distance entre les antennes est suffisamment grande. La dépendance au carré [13] de la distance est due à la propagation sphérique des ondes, et la dépendance à la longueur d'onde provient de la captation de puissance à l'antenne (**ouverture d'antenne** ou **surface efficace de réception**). La formule néglige aussi les autres affaiblissements dus à la diffraction, aux échos parasites dus à des réflections d'ondes déphasées, les gains d'antennes, etc.

13. ce qui s'exprime ici par le facteur 20 plutôt que 10 dans le logarithme car $10\ \log_{10}(\frac{4\pi d}{\lambda})^2 = 20\ \log_{10}\frac{4\pi d}{\lambda}$

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

$$A_{espacelibre} = 32.5 + 20\ \log_{10} f_{MHz} + 20\ \log_{10} d_{km} \tag{7.10}$$

On constate donc comme attendu que chaque doublement de la distance d_{km} augmente l'affaiblissement d'environ $2 * 3$ dB $= 6$ dB (division par 4 de la puissance du signal).

7.3.3 Autres affaiblissements

Quelques exemples (en négligeant l'impact de la fréquence et en supposant que l'obstacle est assez éloigné) [12].

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

7.4 Ellipsoïde de Fresnel

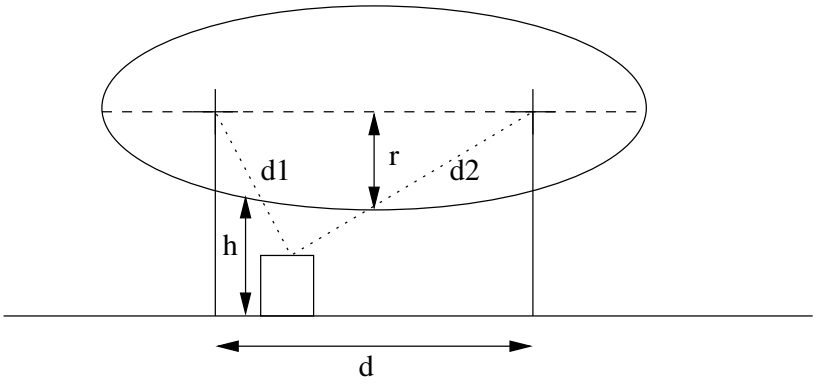


FIGURE 7.2 – Ellipsoïde de Fresnel

L'ellipsoïde[14] 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

$$r = \frac{1}{2}\sqrt{d\lambda} = \frac{1}{2}\sqrt{d\frac{c}{f}} \tag{7.11}$$

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

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) :

$$r = 17.32\sqrt{\frac{d_{km}}{4f_{GHz}}} \tag{7.12}$$

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.

$$h = 17.32\sqrt{\frac{d_1 d_2}{f(d_1 + d_2)}} \tag{7.13}$$

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

7.4.1 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 antennes [15]. 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 $R^c = 8500$ km [16]. Dans l'exemple ci-dessus, il faudrait relever les antennes d'un demi-mètre.

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

16. 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).

7.5 Application aux échanges sans-fil informatiques

Après avoir dimensionné théoriquement la puissance de transmission et les éventuelles antennes à gain[17] (voir section 7.2.4 en page 87) et vérifié que les affaiblissements supplémentaires envisageables sont limités, d'autres éléments peuvent encore influencer la performance d'une application sans-fil.

Par exemple, pour la plupart des technologies de transmission de données sans-fil comme le WiFi, la meilleure performance est obtenue lorsqu'il y a *visibilité directe* de toutes les stations même en **mode infrastructure**[18] (il n'y aura donc pas de collisions[19]), et qu'il n'y a pas trop d'autres usagers de cette plage de fréquence (quel que soit le protocole : un détecteur de réseaux WiFi n'est pas suffisant).

Ce qui peut également aider, en particulier avec les normes récentes WiFi (**WMM**, **802.11e**), c'est la possibilité de désactiver les confirmations en couche 2 (ACK) en particulier pour les protocoles temps réel ou à scrutation, d'éviter les échanges bidirectionnels, d'utiliser les modes directs ne passant pas par un access point et d'exploiter la qualité de service (classes de priorité).

Pour les réseaux informatiques sans fil de campus ou d'entreprise, les constructeurs offrent des outils qui permettent de déployer des réseaux de points d'accès WiFi, que cela soit pour définir leurs emplacements ou leurs fréquences.

17. même dans le cas d'une transmission sans-fil entre plusieurs partenaires, des antennes sectorielles peuvent être utilisées pour limiter le bruit
18. utilisation d'un **access point** central
19. dans le cas contraire, du **polling** – scrutation – ou de la réservation avec **RTS/CTS** est nécessaire

Chapitre 8

Sécurité dans les échanges

Pas traité en détail cette année

Sommaire

Le but de ce chapitre est de présenter la problématique particulière liée à la sécurité dans les échanges, y compris la problématique de la sécurité du périmètre d'un réseau interne d'entreprise ou de terrain.

8.1 Sécurité du périmètre

Le but de ce chapitre est de donner quelques informations de base sur la sécurisation du périmètre d'un réseau d'entreprise, en complétant et appliquant les notions des cours traitant de réseau dans le cursus, notamment en traitant les concepts de **VLAN**, les firewalls et les proxies.

8.1.1 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 [24]). 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 8.1.2.5), ou installer un système de détection d'intrusion et de réaction aux intrusions dans l'entreprise via un réseau de quarantaine (voir section 8.1.4.1) permettant de confiner les problèmes très rapidement.

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

8.1.2.1 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).

Signalons qu'avec l'Internet of things (**IoT**), il devient de plus en plus important de protéger les dispositifs techniques d'un accès non autorisé et de vulnérabilités de leur implémentation des couches inférieures : par exemple par des **proxy-application**s filtrants, souvent appelées **gateway**s de protocoles dans ce contexte.

Un cas particulier de proxy est le **reverse-proxy**, qui protège plutôt un serveur (p.ex. une application Web) d'attaques externes (voir section 8.2.1 en page 101).

8.1.2.2 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. La surface d'attaque (**attack surface**) augmente.

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 [26].

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).

8.1.2.3 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 datagrammes 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.

8.1.2.4 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'ECN [1]).

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).

8.1.2.4.1 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.

Enfin, un proxy ne peut détecter du trafic malicieux que s'il est transmis en clair : pour cette raison, on peut configurer soit une liste blanche des prestataires de services pour lesquels une connexion SSL/TLS est autorisée, ou déchiffrer le trafic grâce à une **MitM-box** – qui nécessite l'installation d'un certificat racine (CA ROOT) autorisant la création d'un demi-tunnel, sur l'ensemble des clients. Cette dernière option est assez dangereuse car une compromission de la clé privée de ce CA trafiqué peut mener à des attaques globales de type **MitM** (voir section 9.1.6.1 en page 110).

8.1.2.5 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 *extranet*.

(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.

1. Enhanced Congestion Notification

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.

8.1.3 Surveillance

8.1.3.1 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.

8.1.3.2 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 [22].

8.1.3.3 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 [23]).

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

Certains honeypots sont accessible sur Internet, dans le but d'aider la communauté à détecter des systèmes compromis, intégrés à un réseau de bots **command and control**, ou plus simplement des spammeurs (**spam trap**).

8.1.4 Réseaux privés virtuels (VPN)

D'une manière peut-être arbitraire, nous relions 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, ce qui comprend les équipements, logiciels et services cloud VPN (voir section 8.3.2 en page 103)
— les réseaux d'accès et d'entreprise globaux, gérés par des opérateurs (technologies : accès modem commuté, VPN ou **MPLS**)

8.1.4.1 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 8.1.3.2) est celui du *réseau de quarantaine* de l'EPFL[25] : 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*).

8.1.4.2 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 4 [2] 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 intéropérable de chiffrement et/ou authentification couche 3 pour IPv4/IPv6.

OpenVPN Logiciel libre, multiplateforme proposant chiffrement et authentification en couche 2 ou 3 pour IPv4/IPv6.

2. formellement en couche 6

8.2 La sécurité dans les échanges

8.2.1 B2B

Lorsqu'un réseau d'entreprise interne (hébergé ou non), offre des services à l'extérieur, que cela soit un simple serveur de mail ou Web, ou pour des échanges **B2B** entre entreprise (p.ex. accès applications métiers, bases de données, ...), des éléments complémentaires peuvent être mis en oeuvre, citons notamment :

— des reverse-proxies, chargés de confiner les piles IP des serveurs et de filtrer le contenu des requêtes, souvent après déchiffrement SSL ; ou plus globalement des systèmes de gestion de la menace (**Threat Management System**) qui centralisent le contrôle d'accès et s'intègrent avec des **IDS** et **IRS**
— des **VPN**s, pour relier des réseaux d'entreprise sur plusieurs sites ou des employés mobiles (**Road Warrior**)
— des systèmes d'authentification globaux et de single-sign-on (9.1.6.2 en page 111, p.ex. **OpenID**, Facebook, ...
— des infrastructures à clés publiques (PKI) permettant d'assurer la confiance (9.1.6.1 en page 110)

8.2.2 Echanges généraux

8.2.2.1 Risques généraux

Les échanges généraux sur Internet, en plus de tous les problèmes mentionnés précédemment, sont caractérisés par les risques suivants :

— le risque d'abus de vulnérabilités **zero-day**, beaucoup plus probable que dans un réseau interne ou un B2B, ce qui nécessite une surveillance proactive et une détection de comportements abusifs (IDS/IRS) et la capacité d'installer, voire d'adapter des correctifs extrêmement rapidement
— le risque de déni de service (**DoS**) distribué par un réseau **command and control**, par définition difficile à contrecarrer si l'on ne peut identifier l'attaquant, en particulier avec des attaques dites de **traffic amplification** DNS ou NTP
— l'impossibilité de faire confiance aux adresses IP, même dans les logs : par exemple, bloquer une adresse suite à une attaque peut être une stratégie élaborée d'un attaquant pour un déni de service envers un utilisateur particulier, tant que de la signature numérique (IPsec opportuniste) ou de l'anti-**spoofing** global (ECP38) n'est pas mis en oeuvre systématiquement sur l'Internet global
— l'acquisition de métadonnées sur les habitudes des clients, y compris en cas d'utilisation de HTTPS, par les FAI et fournisseurs de contenu

Garantir l'accessibilité et la non compromission de services généraux sur Internet est un métier : de plus en plus d'entreprise font donc appels à des services d'application hébergées (**SaaS**) ou au minimum à des hébergeurs disposant de l'expérience suffisante.

8.2.2.2 Confidentialité des échanges

8.2.2.3 Introduction

Par défaut, IPv4 n'incorpore pas de mécanisme de sécurité, que cela soit pour :

l'intégrité : un simple checksum d'entête : pas de signature numérique qui garantirait l'authenticité des messages

la confidentialité : tout est en clair par défaut

la disponibilité : pas de mécanisme de base

Les couches supérieures peuvent authentifier et chiffrer : mais les méta-données de couche 2, 3 et 4 ne sont pas protégées. Il existe toutefois des extensions pour la sécurité en couche 3 : citons par exemple les **VPN**s, comme le protocole **IPsec**, qui permettent d'assurer, dans une certaine mesure, l'intégrité des messages (méta-données et données), et la confidentialité des données. IPv6 oblige l'implémentation d'IPsec sans toutefois prévoir de mécanisme de configuration automatique[3]. Une certaine haute disponibilité (**HA**) peut être mise en place par le DNS, les protocoles de routage ou les **CDN**. Un **firewall** (niveau réseau ou applicatif, comme un **WAF**, *Web Application Firewall*) et de la détection d'intrusion (**IDS**) peuvent renforcer la sécurité.

8.2.2.4 Sécurité en couche 2

Certains switches peuvent être configurés pour appliquer quelques éléments de sécurisation de couche 2. Ils peuvent dans certains cas même inspecter les entêtes des couches supérieures pour bloquer certaines attaques. Parmi les fonctions de sécurité des switches déployés en entreprise, citons :

— des limitations pour chaque port où se trouvent uniquement des machines et non des liaisons avec d'autres switches : maximum d'adresses MAC, de trafic multicast ou broadcast, de nombre de requêtes du protocole ARP ou de protocoles de couches supérieures (p.ex. DHCP) ; blocages liés au protocole spanning tree[4], aux VLAN ou à des protocoles propriétaires Cisco (DTP, CDP)
— de l'anti-spoofing du protocole ARP (Dynamic ARP inspection) ou IP (IP Source Guard)
— des blocages contre des réponses de protocole réservées à des serveurs (DHCP Snooping, filtrage LLDP/NDP[5])
— de l'authentification pour l'accès à un réseau chiffré ou l'attribution à un **VLAN** (**802.1x**, **EAP**) en WiFi, voire Ethernet

8.2.2.5 Compatibilité des besoins de sécurité et de surveillance

Le tableau ci-dessous résume les moyens techniques de protection des méta-données (entête IP : adresses IP ; entêtes de couche 4 : numéros de ports) et des données suivant les protocoles utilisés et la compatibilité avec les besoins de surveillance des autorités légitimes[6] :

techniques	sécurisation		LSCPT		performance
	méta	données	méta	données	
HTTP			✓	✓	✓
HTTPS		✓	✓		✓
tor + HTTP	✓	(✓)			
tor + HTTPS	✓	✓			
VPN + HTTPS	(✓)	✓			(✓)
NGN full (voir page 81)	✓	✓	✓	✓	✓

3. concept de cryptographie opportuniste, voir RFC-7435 et le projet FreeS/WAN – non déployé
4. aussi pour protéger le réseau contre des boucles
5. Link Layer / Network Discovery Protocol, utilisé en IPv6
6. en Suisse : Loi fédérale sur la surveillance de la correspondance par poste et télécommunication (LSCPT) et loi sur le service de renseignement (LRens)

HTTPS chiffre le payload après la couche 4 (les entêtes de couche 4 et inférieures restent en clair) : les méta-données de couche 3 (adresse IP) ne peuvent pas être cachées car elles sont nécessaires pour le routage IP. On peut les cacher dans une certaine mesure et au prix de délais plus ou moins importants par des VPNs ou du routage par onion (**tor**).

En HTTPS, si l'on connaît l'adresse IP et le port du serveur consulté, on peut facilement télécharger le certificat X.509 TLS/SSL associé et donc le (ou les aliases) du nom de domaine accédé, car originellement un seul certificat était possible par adresse IP et port[7]. Le problème est bien sûr agravé si un DNS en clair[8], même avec les extensions d'intégrité **DNSSEC**, est utilisé.

Les **NGN**-full (voir section 6.3.5.3 en page 80) prévoient un chiffrement par tronçon (VPNs tunnelisés par IPsec), protégeant les données *et* méta-données, compatible, en déploiement complet *et* international des NGN, avec les besoins des autorités de surveillance, dans la mesure où du chiffrement de bout en bout n'est *pas* utilisé en plus. Pour le moment, le déploiement de cette solution n'est pas planifiée.

8.3 Protéger le grand public

8.3.1 Protéger ses données sur Internet

La FRC a publié un dossier (*Mieux protéger ses données*[9]), avec quelques recommandations sur la sécurisation des échanges.

8.3.2 Pourquoi utiliser un VPN ?

En ce qui concerne les VPN grand public, sous forme d'équipements (routeur avec fonction VPN par exemple), de services clouds ou de logiciels, ils ont plusieurs usages :

- créer des réseaux privés non accessibles d'Internet : réseaux de mesure, accès à un serveur de fichier local (NAS) depuis l'extérieur, etc : c'est l'usage classique technique et pour l'entreprise
- cacher ses actions au fournisseur local, voire aux autorités locales : utile par exemple pour un journaliste enquêtant sur des sujets sensibles ou dans un pays limitant ses droits
- passer outre certains types simples de censure / non respect de la **neutralité du réseau**
- changer son adresse IP et la faire provenir d'un autre pays, pour passer outre des restrictions d'offres nationales (catalogues de films ou séries par pays, par exemple)

Attention, en général si le trafic est chiffré entre votre ordinateur et le fournisseur du VPN, il ne l'est plus entièrement (**métadonnées**, voir section 8.2.2.5 en page 102) ni du tout (si vous utilisez des protocoles non chiffrés de bout en bout) dès lors que vous sortez de ce fournisseur et allez sur Internet. Par contre, votre IP est anonymisée pour le serveur que vous consultez. En général, les VPN vont également faire en sorte que le serveur DNS (conversion nom vers

7. l'extension **SNI** – *Server Name Indication* – qui permet d'héberger plusieurs certificats sur la même adresse IP, transmet *en clair* le nom du domaine désiré : seul **ESNI**, très récent, améliore la confidentialité sur quel site est consulté et seulement si l'adresse IP est servie par un reverse-proxy qui sert énormément de domaines différents (gros hébergeurs ou **CDN** – *Content Distribution Network* comme par exemple CloudFlare) !

8. des extensions pour DNS sur TLS (**DoT**) ou HTTPS (**DoH**) existent qui limitent les risques de surveillance par les autorités ou le fournisseur d'accès Internet : elles posent toutefois des problèmes aux DNS locaux et simplifient la collecte de métadonnées par le fournisseur de ce service

9. https://www.frc.ch/mieux-proteger-ses-donnees/

IP notamment) soit également interrogé via le VPN (sinon votre fournisseur et les autorités locales pourront savoir quels sites vous consultez).

Enfin, certains logiciels VPN contiennent également d'autres logiciels comme des anti-virus, des systèmes pour limiter le tracking, etc. Tout est finalement question de confiance face à l'éditeur du logiciel et à l'exploitant. Il y a déjà eu des bugs graves où on pouvait très facilement désanonymiser les utilisateurs d'un VPN spécifiques. De plus, si vous utilisez le même navigateur via le VPN et hors du VPN, les sites distants pourront vous désanonymiser, d'autant plus si vous êtes loggué sur un GAFAM.

Dans certains cas extrêmes, on recommandera l'utilisation de **tor**, au prix d'une performance réduite et du risque de blocages.

Chapitre 9

Authentification

Pas traité en détail cette année

Sommaire

Le but de ce chapitre est de présenter les méthodes d'authentification les plus courantes, une classification et leurs forces et faiblesses.

9.1 Authentification et identification

9.1.1 Introduction

Une entité quelconque (utilisateur, agent, etc) est identifiée par un processus simple d'**identification** (par son numéro unique, par son nom, etc).

Mais cela ne *prouve pas son identité* : pour ce faire, il faut un processus d'authentification. Sans ce processus, il manque donc une confirmation, ou preuve, d'identité et aucune confiance n'est possible.

9.1.2 Le besoin d'authentification

L'**authentification** est le processus (ou protocole) par lequel on vérifie la preuve d'**identité** d'une entité, dans le but de lui permettre l'accès à des ressources (via le contrôle d'accès, pour assurer l'intégrité et la confidentialité) et à assurer la **traçabilité**. L'authentification est également nécessaire pour assurer la non-répudiabilité de transactions, en particulier combiné à un **horodatage**, voire un **tiers-garant** ou tiers de confiance.

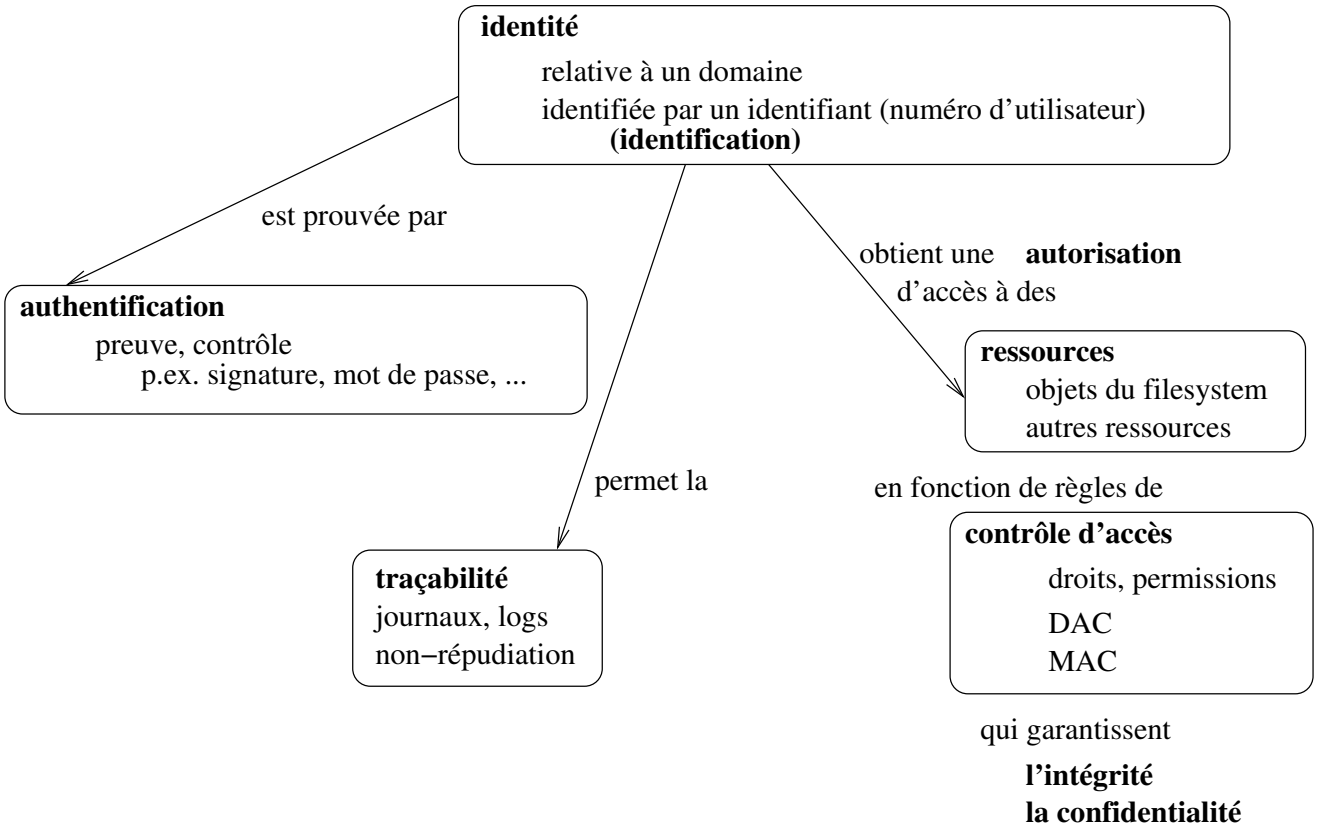


FIGURE 9.1 – Authentification, identité, identification et contrôle d'accès

9.1.2.1 Les facteurs d'authentification

Les preuves, aussi appelés **facteurs d' authentification**, les plus courants sont :

- un mot de passe en clair (ce que vous connaissez)
- une carte d'identité, une carte magnétique, un téléphone portable (ce que vous avez)
- un élément biométrique (ce que vous êtes)
- un comportement (ce que vous savez faire)

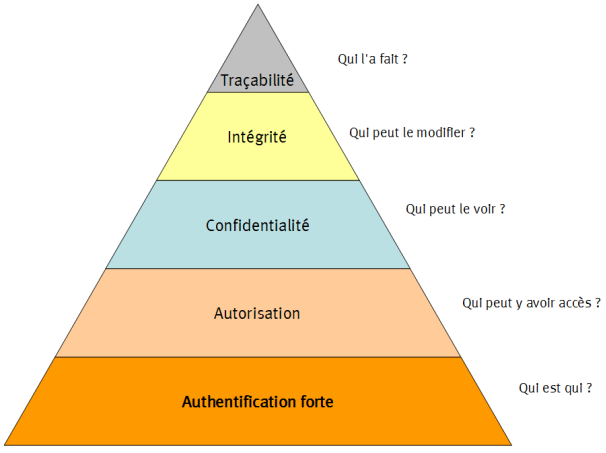
Les meilleurs preuves sont souvent obtenues par le biais d'un **agent**, sous forme d'un terminal, d'une carte à puce, d'un ordinateur, auquel vous faites confiance et à lequel vous confiez quelque chose de secret et qui réalisent une opération, comme par exemple :

- un calcul (basé de préférence sur un élément externe comme le temps ou un token) prouvant la connaissance d'un élément secret (p.ex. clé, mot de passe), un comportement (ce que vous savez faire)
- un échange bidirectionnel, où le token est configuré par le serveur sous forme d'un nombre unique ou **nonce**

9.1.2.2 L'authentification forte

La combinaison de plusieurs facteurs (authentification multifactorielle, au moins deux) permet ce que l'on appelle l'authentification forte.

L'authentification forte devrait permettre de satisfaire à tous les critères nécessaires énoncés ci-contre (source : Wikipedia).



On considère également les mots de passe à usage unique (**OTP**) et les certificats numériques (voir 9.1.6.1 en page 110) sont également des techniques d'authentification forte.

La **biométrie** peut aussi être considérée comme une authentification forte si l'on analyse des comportements complexes récurrents d'une personne, par exemple, mais pas s'il ne s'agit que d'un élément simple biométrique, à fortiori souvent copiable.

9.1.3 Quelques exemples de protocoles

9.1.3.1 Introduction

Un protocole d'authentification consiste en la vérification de la validité de la preuve d'identité présentée.

Les meilleurs protocoles d'authentification sont ceux qui évitent

- d'envoyer un élément d'authentification en clair, en utilisant par exemple un **hachage cryptographique**
- de répéter un message déjà envoyé (sinon **replay-attack**, attaque du rejeu), par exemple en utilisant un élément externe (temps, jeton, **nonce**, compteur) ou plus simplement un mot de passe unique à jeter (**OTP**, *One Time Password*).
- une attaque du relais (**MitM**, man in the middle), en assurant l'intégrité du canal de communication, par exemple par vérification de certificats ou chiffrement par secrets partagés.
- les attaques dites de **social engineering**, dont le **phishing** est un cas particulier, par un bon design des interfaces et une bonne (in)formation.

et qui combinent plusieurs preuves obtenues par divers canaux (ou au minimum par différents facteurs : voir **authentification forte** ci-dessus).

Ci-après on parlera de serveur d'authentification pour le partenaire qui assure l'authentification et de client pour l'entité qui veut s'authentifier.

9.1.3.2 Exemple : PAP

Ce protocole consiste simplement à envoyer un mot de passe en clair (**PAP**, *Password Authentification Protocol*). Son seul avantage réside dans la possibilité de stocker le mot de passe haché [1] côté serveur et donc de limiter l'impact de vols de fichiers de mot de passe.

Il ne devrait être utilisé que si un canal sûr [2] a déjà été établi : par exemple avec un tunnel sûr EAP (voir EAP en section 9.1.4 en page 108) ou de **auth/basic** en HTTPS.

On pourrait croire qu'ajouter un simple hachage à la transmission PAP est une bonne solution, mais utilisé seul, on diminuera la sécurité (les mots de passe devront être stockés en clair côté serveur et le mot de passe haché devient de fait le véritable mot de passe).

1. éventuellement perturbé par un sel (**salt**, graine) pour rendre les attaques de **compromis temps/mémoire** difficiles
2. confidentiel, sans risque d'attaque de type **MitM**

9.1.3.3 Exemple : CHAP

Le protocole **CHAP** (*Challenge-response Authentification Protocol*) permet une authentification sans risque de rejeu, car elle utilise une communication bidirectionnelle et l'envoi d'un **nonce** (nombre à usage unique) par le serveur qui sera utilisé par le client pour perturber un hachage cryptographique du mot de passe en clair.

Toute la difficulté est d'assurer que le hachage soit cryptographique (non facilement inversible) et que le nonce soit suffisamment long et bien choisi (répétitions peu probables, voire garanties inexistantes).

Par contre, ce protocole ne protège pas contre l'attaque du relais et nécessite le stockage en clair du mot de passe côté serveur. Des extensions à CHAP existent pour éviter de stocker le mot de passe en clair, mais leur sécurité n'est pas bonne (p.ex.le très compliqué MS-CHAPv2 [3]).

Une utilisation de ce protocole résistant aux attaques **MitM** se fait également au sein d'un tunnel sûr comme pour PAP ci-dessus, ou de toute autre méthode garantissant l'authenticité des partenaires, ou au moins du serveur (p.ex. **auth/digest** en HTTPS).

9.1.3.4 Exemple : canaux supplémentaires

Toute authentification qui utilise un canal supplémentaire (p.ex. confirmation par SMS en plus d'une authentification où une attaque **MitM** peut être exclue), renforce la sécurité, notamment vis-à-vis d'agents (programmes) malicieux.

Si le canal séparé aboutit à un terminal séparé qui dispose d'un affichage, on peut valider plus d'informations que juste l'existence de la transaction (p.ex. texte et numéro dans un SMS).

9.1.4 Extensible Authentification Protocol (EAP, 802.1x)

9.1.4.1 Principes

EAP est notamment utilisé dans le contexte du filaire (accès aux ports d'un switch : 802.1x) et du sans-fil (WiFi) et propose une infrastructure standard (plug-in) pour l'authentification définissant les clients d'authentification (*supplicants*), les composants réseaux effectuant l'authentification (*authenticators*) et les systèmes d'authentification proprement dits situés sur des serveurs distants.

De nombreux protocoles, par exemple ceux vus dans la section précédente, sont supportés, ainsi que des combinaisons de protocoles par tunnelling qui assurent tout d'abord un canal (tunnel) sûr (chiffré *et* évitant le **MitM**), puis démarrent une authentification classique (appelée *intérieure*) : p.ex. CHAP (EAP-MD5), PAP, ...

9.1.4.2 Protocoles à tunnels

Les deux protocoles standards pour assurer un tunnel sûr et autoriser ensuite une authentification intérieure sont les suivants :

EAP-TLS obligatoire pour le WPA/WPA2, utilise un certificat serveur et un certificat client ; crée un tunnel sûr ; l'identité de l'utilisateur est en clair

3. https://www.schneier.com/paper-pptpv2.html

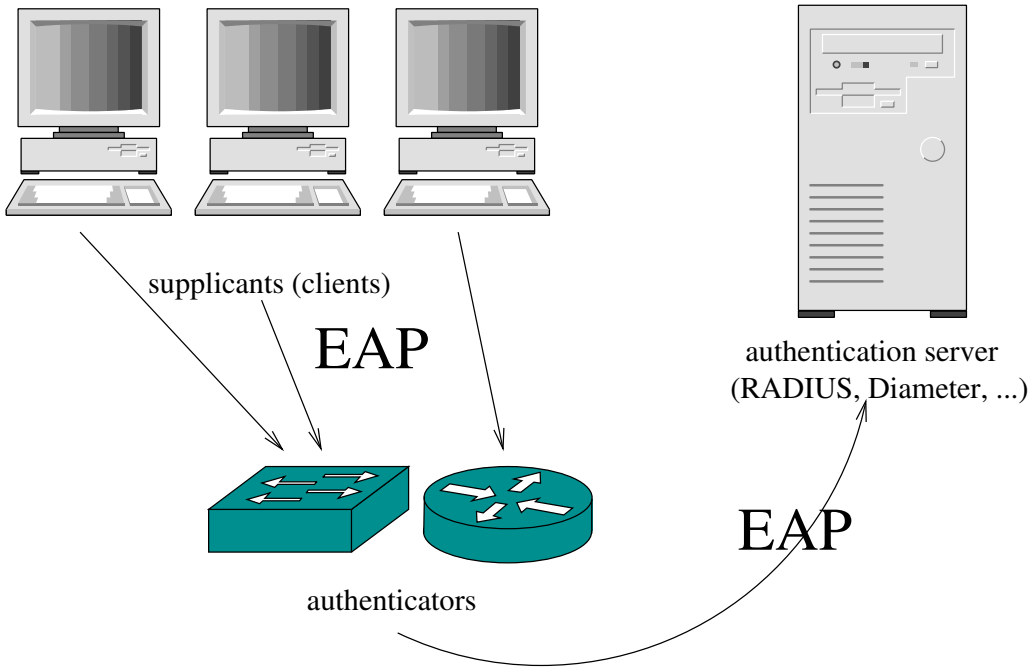


FIGURE 9.2 – Composants d'un EAP

EAP-TTLS nécessite uniquement un certificat serveur ; crée un tunnel sûr ; l'identité réelle de l'utilisateur peut n'être montrée qu'au sein du tunnel sûr

Il est bien évident que pour les protocoles à tunnels sûrs (p.ex. EAP-TLS et EAP-TTLS ci-dessus), il faut authentifier le(s) certificat(s) présenté(s) : on remarque que cette authentification est optionnelle sur certains smartphones, probablement vu la difficulté d'installation manuelle des certificats du CA correspondant : un risque d'attaque MitM peut donc subsister.

9.1.4.3 EAP-SIM

En alternative, **EAP-SIM** permet une authentification sûre avec la carte SIM d'un équipement GSM de deuxième génération, basée sur un challenge-response multiple et bidirectionnel, permettant de générer une clé de session suffisamment longue, partagée entre le centre d'authentification et la carte SIM (RFC-4186). L'absence d'authentification mutuelle est donc compensé par ce double échange. EAP-AKA est quant à lui exploité dans des réseaux de 3^e génération, et permet une véritable authentification mutuelle.

9.1.4.4 Protected EAP (PEAP)

Cisco et Microsoft ont développés des protocoles permettant d'établir des tunnels sûrs, mais plus fermés que ceux présentés ci-dessus. Ils sont toutefois pour le moment souvent déployés vu la compatibilité des variantes Microsoft avec Microsoft Active Directory. Il s'agit des protocoles à tunnels sûrs suivants :

PEAP très similaire à EAP-TTLS (tunnel sûr avec certificat serveur), supporté par Cisco et Microsoft certificat serveur et un certificat client ; crée un tunnel sûr

PEAP-EAP-TLS très similaire à EAP-TLS : certificat serveur *et* certificat client : propriétaire Microsoft (avec implémentation récente chez Cisco).

Une fois qu'un de ces tunnels sûrs est établi, une authentification est effectuée par un des protocoles suivants :

PEAPv0 (MS-CHAPv2) authentification MS-CHAPv2, compatible Microsoft Active Directory (voir la section 9.1.3.3 en page 108 pour une mise en garde sur la sécurité de ce protocole)

PEAPv1 (GTC) support d'une carte à jetons (token card), avec protocole challenge-response (RFC2284, RFC-3748), pas supporté par défaut par Microsoft, supporté notamment par Cisco.

Le marché semble actuellement s'orienter plutôt vers EAP-TTLS, avec un certificat serveur, et le choix entre un protocole d'authentification intérieur PAP ou CHAP ou, pour assurer la compatibilité Microsoft, MS-CHAPv2.

9.1.5 Comparatif des protocoles d'authentification

En particulier en ce qui concerne les attaques citées à la page 107 :

protocole	**mot de passe visible sur le réseau ?**	**mot de passe haché côté serveur ?**	**rejeu ?**	**MitM ?**
PAP	oui	possible	oui	oui
CHAP	non	non (sinon, le mot de passe haché est en fait le vrai mot de passe)	non (nonce)	oui
EAP-TTLS avec PAP	non – tunnel chiffré	possible (PAP)	non (clé chiffrement symétrique aléatoire)	non (si le certificat du serveur est vérifiable)
certificats X.509 serveur & client	pas utilisé	pas utilisé	non	non

D'autres attaques sont toujours possibles.

9.1.6 Authentification distribuée

9.1.6.1 Annuaires

L'authentification est souvent basée sur une source d'information, qui peut être un annuaire comme p.ex. **LDAP** (implémentation légère de DAP - X.500). LDAP lui-même peut être utilisé comme protocole d'authentification via la notion d'association (binding) à un serveur LDAP.

Les annuaires peuvent contenir des informations administratives sur l'utilisateur, p.ex. son UID, ses GIDs, son adresse e-mail, et parfois des éléments avancés comme des politiques de sécurité.

9.1.6.2 PKI

Une PKI (*Public Key Infrastructure*) permet de résoudre le problème de la distribution des clés publiques et de la **confiance** au sein d'un réseau plus ou moins étendu. Par exemple, les certificats X.509 TLS/SSL de serveurs Web permettent de les authentifier si un chemin de confiance (**trust path**) mène à un certificat racine préinstallé (par la signature numérique. Le système **OpenPGP**, lui, est plutôt basé sur une notion de réseau de confiance (**WoT**, *Web of trust* ou encore **trust net**), où il faut obtenir, à la façon d'un réseau communautaire, le plus d'utilisateurs possibles qui signent votre clé publique.

Avant les cryptosystèmes à clé révélée, des PKI ont été mises en place en utilisant du chiffrement symétrique relativement faible et la notion de ticket, valide pendant une fenêtre courte évitant une attaque d'authentification. Par exemple, **Kerberos**[4] implémente une PKI en utilisant le mot de passe de l'utilisateur dans sa phase initiale envers le serveur d'authentification, puis des tickets horodatés obtenus d'un serveur de tickets pour l'accès sécurisé à chaque service (via un chiffrement symétrique et une clé par association).

Une PKI moderne, basée sur la cryptographie à clé publique, se compose en général des éléments suivants :

- une autorité de certification (CA), qui délivre (signe) les certificats des clients (pour une authentification par certificat) et des serveurs (pour l'authentification classique, p.ex. CHAP sur SSL/TLS), et qui permet d'éviter une attaque **MitM**) à l'aide du CA ROOT préinstallé – cette autorité est aussi appelée un **tiers de confiance**
- un certificat racine de la PKI, préinstallé sur tous les clients et serveurs, appelé ci-après le CA ROOT.
- un serveur de temps, pouvant délivrer des horodatages authentifiés (**TSP**)
- une liste de révocation, permettant d'annuler des signatures délivrées, de préférence en-ligne (**OCS**).
- de clients et de serveurs auprès desquels ont veut s'identifier

En règle générale, le véritable CA ROOT, dont la durée de vie peut être longue, est stocké de manière sécurisée et sert à signer régulièrement le CA *opérationnel* (p.ex. de 2^e niveau) utilisé pour la certification (qui lui a une durée de vie limité). Ce processus permet de limiter l'impact d'une compromission de la clé privée du CA ROOT.

Si l'on désire augmenter cette PKI pour exploiter du B2C ou du B2B, il est possible de faire signer le CA ROOT par une autorité de certification reconnue par défaut par les navigateurs : cette autorité peut limiter la portée des identités validables par le certificat.

9.1.6.3 Sur Internet : single sign-on

Divers protocoles d'authentification unique (**SSO**, *single sign-on*) existent, comme par exemple **OpenID**[5]. On notera aussi l'existence du langage **SAML**[6], norme OASIS, qui propose un format XML qui permet l'interopérabilité entre applications SSO.

4. dès Kerberos 5, on peut utiliser des certificats à la place
5. http://www.openid.net/
6. https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

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] *ATM : Télécommunications à large bande*, Ecole d'ingénieurs de Fribourg, 1996

[7] 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)

[8] R. W. HAMMING, *Error-detecting and error-correcting codes*, Bell System Technical Journal, vol 27, pp. 147-160, 1950

[9] Ulysse BLACK, *MPLS and Label Switching Networks*, Prentice Hall, ISDN 0-13-015823-2

[10] Ivan PEPELNJAK & Jim GUICHARD, *MPLS and VPN architectures*, Cisco Press, ISBN 1-58705-002-1

[11] Matthew S. GAST, *802.11 Wireless Networks : the definitive guide*, 2nd Edition, O'Reilly, ISBN-0-596-10052-3

[12] 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

[13] W. Richard STEVENS, *TCP/IP Illustrated volume 1 : The Protocols*, Addison-Wesley, ISBN 0-201-63346-9

[14] Christian HUITEMA, *Le routage dans l'Internet*, Eyrolles, 10/1994, 418p, ISBN 2-212-08902-3 (10/1994)

[15] Alain PELAT, *Signaux numériques, protection contre les erreurs*, Ellipses, 2005, ISBN 2-7298-2367-0

[16] Harald T. FRIIS, IRE Proceedings, vol. 34, p. 254, 1946.

[17] Marc SCHAEFER, *Gitlab des cours*.

[18] Hao SHANG & Craig E. WILLS, *Making Better Use of All Those TCP ACK Packets*, http://web.cs.wpi.edu/~cew/papers/isast07.pdf, ISAST, 2007.

[19] Calcul du champ électrique rayonné, http://blog.f6krk.org/wp-content/uploads/2013/06/Calcul_Champ_E.pdf.

[20] Que sont les champs électromagnétiques, publication de l'OMS, http://www.who.int/peh-emf/about/WhatisEMF/fr/ ainsi que les aides-mémoire résumant les connaissances scientifiques sur l'hypersensibilité, comme par exemple http://www.who.int/peh-emf/publications/facts/fs296/fr/.

[21] Guillaume SCHREINER (Université de Strasbourg) et al, *Retour d'expérience d'un déploiement LoRaWAN*, https://conf-ng.jres.org/2019/document_revision_5202.html?download, JRES, 2019.

[22] Les systèmes de détection d'intrusion classiques et comportementaux, voir http://www.securiteinfo.com/conseils/choix_ids.shtml

[23] Les HoneyPots (voir aussi Wikipedia.org), http://www.honeypots.net/

[24] Voir http://www.informit.com/articles/article.asp?p=350391&seqNum=5

[25] Réseau de quarantaine (snort IDS, VLAN, proxy squid) http://dit.epfl.ch/page59201.html

[26] 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/

[27] Cyrille P. [Coutansais, *La (re)localisation du monde*, CNRS Éditions, 2021, 281p, ISBN 978-2271-12709-9

Index des concepts

Table des figures

Table des matières

ISBN 978-2-940387-09-0