Master d'informatique – M1

Cours de Réseaux

1998 - 2011

Zoubir MAMMERI

Chapitre 1

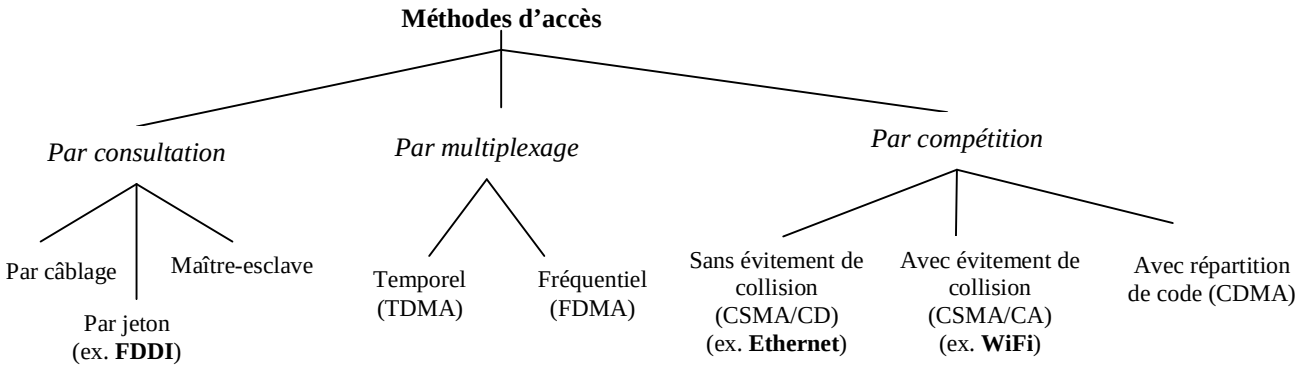
Rappels sur les techniques d'accès au support de transmission (MAC)

I. Introduction

La méthode ou technique d'accès au médium (souvent abrégée par MAC : "Medium Access Control") est une composante fondamentale des réseaux. C'est le protocole de la méthode d'accès d'un réseau qui permet de définir les règles d'attribution du support de transmission aux nœuds du réseau. En d'autres termes, la technique MAC définit les règles de partage du médium entre nœuds du réseau.

Beaucoup de méthodes d'accès au support de transmission ont été proposées et expérimentées pour répondre à différents besoins. Certaines de ces méthodes sont devenues des normes internationales et d'autres sont limitées à quelques réseaux propriétaires. Dans ce paragraphe, nous allons présenter les principes généraux des techniques MAC les plus connues.

Les techniques MAC peuvent être divisées en trois groupes, selon les règles auxquelles doivent se conformer les nœuds pour accéder au support de transmission : méthodes par consultation, méthodes par compétition et méthodes par multiplexage. Le choix d'une méthode d'accès dépend de plusieurs critères, notamment : la topologie du réseau, le type de support (filaire ou non filaire), le déterminisme d'accès, la possibilité de privilégier certains nœuds et la tolérance aux fautes.



Classes de méthodes d'accès dans les réseaux

II. Méthodes d'accès par consultation

Le principe de base des méthodes par consultation est que les nœuds "se consultent" pour décider de celui qui a le droit d'utiliser le support de transmission. Un nœud accède au support s'il y est autorisé. L'autorisation peut être gérée de plusieurs manières, notamment :

- **Par échange d'informations** : les nœuds s'échangent des informations de manière permanente pour connaître celui qui a le droit d'émettre. Généralement, le droit d'utiliser le support est géré soit de manière centralisée par un site privilégié (on parle dans ce cas d'une technique de type maître-esclave), soit par le passage d'un jeton.
- **Par utilisation d'éléments physiques dédiés :** les nœuds utilisent des lignes spéciales pour marquer leur intention d'utiliser le médium et un circuit permet d'autoriser les nœuds à émettre. Cette technique est rarement utilisée dans les réseaux où le nombre de nœuds est variable ou important.

Il faut souligner que les techniques par consultation sont surtout utilisées dans les réseaux locaux et/ou filaires.

II.1 Méthodes d'accès maître-esclave

Dans les méthodes maître-esclave, les nœuds sont regroupés en deux catégories : un nœud maître (ou nœud primaire) et des nœuds esclaves (ou nœuds secondaires). Un nœud esclave n'a le droit d'émettre que si le site maître l'autorise. Les règles appliquées par le nœud maître pour autoriser les nœuds esclaves à émettre sont diverses. En particulier, les deux manières suivantes d'autoriser les nœuds esclaves sont utilisées dans les réseaux que l'on trouve dans les installations industrielles automatisées (comme les laminoirs, les complexes pétrochimiques, les usines de montages de voitures...) :

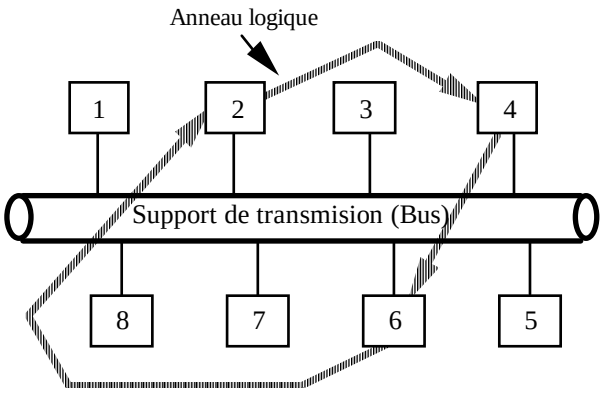
- *Méthode par scrutation régulière* : le nœud maître scrute un par un les nœuds esclaves selon un ordre préétabli ; lorsque le dernier nœud est scruté, le nœud maître reprend la scrutation à partir du premier nœud. Il donne le droit à émettre à un site esclave qui l'utilise le temps de transmettre une ou plusieurs trames, ensuite le nœud esclave rend le droit d'émettre au nœud maître. Si un site esclave n'a pas de trame à transmettre, il rend immédiatement le droit d'émettre au nœud maître. L'inconvénient de cette méthode est qu'il y a parfois une perte de temps à scruter des nœuds qui n'ont rien à transmettre. Différentes solutions ont été proposées pour atténuer ce défaut.
- *Méthode utilisant une table d'arbitre* : dans cette méthode, le nœud maître est appelé *arbitre du réseau* et possède une table qui lui indique à quel moment exactement il faut scruter un nœud esclave. Dans ce cas, la scrutation est interprétée par le nœud esclave comme une demande à émettre (et non une invitation à émettre, comme dans le cas de la méthode à scrutation). Un des réseaux de terrain qui utilise cette méthode est le réseau *WorldFIP* développé en France.

Pour des raisons de tolérance aux fautes, certains (ou tous les) nœuds esclaves ont le statut de nœud maître de secours et ils peuvent reprendre (mais un seul nœud à la fois) le contrôle du réseau lorsque le nœud maître courant tombe en panne.

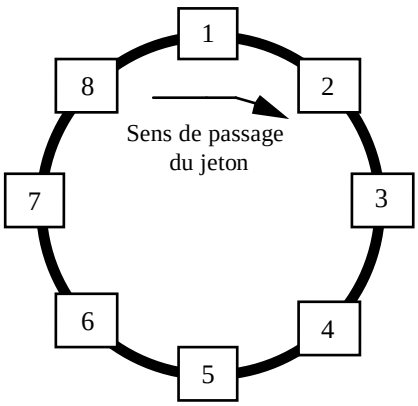
II.2 Méthodes d'accès à jeton

Les méthodes à jeton fonctionnent essentiellement sur des topologies en boucle ou en bus. Sur une boucle, l'ordre de passage du jeton est défini par l'ordre physique de connexion des nœuds. Sur une topologie en bus, on définit un anneau logique pour déterminer l'ordre de passage du jeton. Construire un anneau logique revient à indiquer (par un protocole d'initialisation de réseau) les adresses du prédécesseur et successeur de chaque nœud sur l'anneau logique. Lorsque le jeton fonctionne sur un bus, on parle de *bus à jeton* ("token bus") et de *boucle à jeton* ("token ring"), dans le cas d'une topologie en boucle. Une fois que l'anneau physique ou logique est établi, un jeton (un jeton est une trame qui a un format spécial) est créé (par une procédure de création du jeton), ensuite le jeton passe de nœud en nœud selon l'ordre de l'anneau. Chaque nœud n'a le droit d'émettre

que s'il reçoit le jeton, après quoi il peut transmettre pendant un certain temps limité et passe ensuite le jeton à son successeur. Le jeton tourne sur l'anneau tant qu'il y a au moins deux nœuds actifs.



Exemple de bus à jeton



Exemple de boucle à jeton

Techniques à jeton

III. Méthodes d'accès par compétition

Dans les méthodes d'accès par compétition, dites aussi *méthodes d'accès aléatoire*, il n'y a pas d'élément logique ou physique qui permet aux nœuds de se mettre d'accord pour utiliser le support ; chaque nœud peut commencer à transmettre dès qu'il le souhaite (à quelques exceptions près). Cela conduit évidemment à des situations de conflit d'accès quand deux ou plusieurs nœuds transmettent simultanément. Plusieurs solutions ont été proposées pour gérer les situations de conflit (*collision*). Dans le contexte des réseaux locaux et réseaux sans fil, c'est essentiellement la technique CSMA et ses variantes qui sont utilisées.

III.1. Méthode CSMA/CD

Dans la méthode CSMA/CD ("Carrier Sense Multiple Access with Collision Detection"), un nœud peut émettre dès qu'il le souhaite à condition de détecter que le support est libre. Si le support n'est pas libre, le nœud ajourne sa tentative jusqu'à la prochaine libération du support. Si le bus est libre, le nœud commence sa transmission et compare ce qu'il émet par rapport à ce qu'il reçoit, s'il y a une différence entre les deux signaux, il est fort probable qu'au moins un autre nœud soit en train de transmettre en même temps que lui. Dans ce cas, il arrête sa tentative de transmission, envoie un signal de brouillage pour signaler la collision aux autres nœuds, attend pendant un certain délai aléatoire avant de tenter une nouvelle fois sa transmission. L'inconvénient de cette méthode est que si le nombre de nœuds qui souhaitent transmettre est important le nombre de collisions devient tel qu'aucun de ces nœuds n'arrivent à transmettre sa trame. Par conséquent, la méthode CSMA/CD n'est pas adaptée aux applications temps réel. Il faut signaler que de par sa simplicité, la méthode CSMA/CD est utilisée par le réseau local le plus répandu au monde, à savoir le réseau Ethernet.

Pour éviter les collisions en chaîne conduisant à des temps de réponse excessifs, des améliorations de la méthode CSMA/CD ont été proposées, notamment les méthodes CSMA/CA, CSMA/CR et CSMA/DCR.

III.2. Méthode CSMA/DCR ("CSMA with Deterministic Collision Resolution")

La méthode CSMA/DCR intègre un algorithme de résolution de collision lancé par tous les nœuds qui détectent une collision. Cet algorithme utilise une technique de résolution en arbre binaire : en cas de collision, les nœuds sont partagés en deux groupes selon leurs adresses : un groupe des gagnants et un groupe des perdants. Les nœuds appartenant au groupe des perdants cessent d'émettre, les autres tentent de transmettre. Si le groupe des gagnants contient plus d'un nœud, il y a de forte chance qu'une nouvelle collision soit détectée, auquel cas le groupe est à nouveau scindé en deux, et ainsi de suite jusqu'à ce que le groupe des gagnants ne contienne qu'un seul nœud qui peut alors transmettre tranquillement sa trame. La méthode CSMA/DCR permet de borner le temps d'attente pour transmettre une trame.

III.3. Méthode CSMA/CA ("CSMA with Collision Avoidance")

Dans la méthode CSMA/CA, chaque nœud utilise les informations qu'il possède sur l'état d'activité du support pour calculer la probabilité d'entrer en collision s'il tente une transmission. Le nœud évite de transmettre pendant les instants où la probabilité de collision est jugée élevée. Cela conduit un nœud à attendre pendant un nombre (qui varie selon la charge du support) d'unités de temps même si le support est libre. La chance d'avoir deux nœuds qui tentent de transmettre en même temps est réduite par rapport à CSMA/CD. Les situations de collision ne sont pas complètement écartées en utilisant CSMA/CA. Alors, les nœuds utilisent les acquittements pour savoir s'il y a eu collision ; la non-réception d'acquittement au bout d'un certain temps conduit un nœud émetteur à considérer qu'il y a eu collision.

III.4. Méthode CSMA/CR ("CSMA with Collision Resolution")

Avant de commencer sa transmission, chaque station doit tester l'état du support et elle ne peut transmettre que si le support est libre. Pour éviter les collisions en chaîne, un nœud qui transmet une trame (sachant qu'une trame commence par une adresse unique), cesse d'émettre s'il reçoit un bit différent du sien. Ainsi, un nœud qui émet un bit à *1* s'arrête s'il voit passer sur le support un bit à *0*. En revanche un nœud qui reçoit un bit identique à celui qu'il a émis continue de transmettre. Comme les adresses diffèrent au moins par un bit, un seul nœud poursuit la transmission de sa trame jusqu'à sa fin. Cette technique est mise en œuvre sur le réseau CAN (Controler Area Network) qui est le réseau le plus utilisé dans le domaine de l'automobile.

III.5. Méthode à répartition de code : CDMA ("Code Division Multiple Access")

C'est l'une des techniques d'accès utilisées dans le domaine des réseaux sans fil, notamment dans les réseaux UMTS et dans les réseaux locaux sans fil. Elle est fondée sur un principe qui permet l'accès multiple mais aussi la sécurité des communications au niveau physique. L'idée de base de CDMA c'est comme si dans une salle, plusieurs personnes parlent en même temps mais dans des langues différentes. Pour chaque personne qui parle, seul son correspondant connaît la langue et arrive à extraire du 'vacarme ambiant' ce que dit son interlocuteur. Celui qui ne connaît pas une langue, reçoit le signal lié à cette langue mais ne peut pas le comprendre.

Techniquement parlant, CDMA est basée sur la répartition par codes qui permet à plusieurs sources d'émettre sur les mêmes fréquences. Chaque utilisateur est différencié du reste des utilisateurs par un code *C* qui lui a été alloué au début de sa communication et qui est orthogonal au reste de codes liés à d'autres utilisateurs. Dans ce cas, pour écouter l'utilisateur ayant le code *C*, le récepteur n'a qu'à multiplier le signal reçu par le code *C* associé à cet utilisateur. Chaque code est représenté sur *k* éléments. A chaque bit à transmettre, CDMA crée une séquence de *k* bits (on parle d'étalement de spectre) obtenu à partir de la valeur du bit initial et du code. Le signal ainsi transmis s'apparente à du bruit (car seul celui qui connaît le code de la source est capable de retrouver la chaîne de bits initiale). Dans ce sens, CDMA permet de renforcer la sécurité au niveau physique. A noter de CDMA été longtemps utilisée par les militaires.

Exemple :

Une source A utilise un code égal à <1, -1, 1, 1, -1, -1>. Pour transmettre un bit 1, elle transmet six bits <1 0 1 1 0 0> et pour transmettre un bit 0, elle transmet six bits <0 1 0 0 1 1>. Cela signifie que pour transmettre un bit initial égal à 1, la source génère 6 bits où le bit i (i=1, ..., 6) est égal à 1 si le ième élément du code est égal à 1 et 0 s'il est égal à -1. La séquence de bits transmis dans le cas où le bit initial à transmettre est 0 est obtenue en faisant le complément à 2 de la séquence obtenue pour un bit initial égal à 1.

La méthode CDMA repose sur l'orthogonalité des codes attribués aux sources. Mathématiquement parlant, si on a *n* utilisateurs avec *n* codes, alors tout ensemble de vecteurs dans le *n*-espace sont orthogonaux si tout point dans le *n*-espace peut être exprimé seulement avec une combinaison linéaire de ces vecteurs. On rappelle qu'un produit scalaire de deux vecteurs U et V de composantes u_1, u_2... u_n et v_1, v_2,... v_n est la somme $u_1v_1 + u_2v_2 + ... u_nv_n$.

Par exemple, les deux codes suivants sont orthogonaux :

Code A = <1, -1, -1, 1, -1, 1>

Code B = <1, 1, -1, -1, 1, 1>

Lorsque les signaux arrivent en provenance de plusieurs sources qui transmettent simultanément, ces signaux s'additionnent chez le récepteur. Le récepteur calcule la corrélation du signal avec le code de l'émetteur, ce qui permet de retrouver les bits du message (s'il n'y a pas eu d'erreur de transmission). Ci-dessous, nous expliquons à travers un exemple le principe simplifié de CDMA (attention les séquences composées de 1 et -1 sont utilisées pour comprendre le principe de CDMA, la transmission effective ne considère que des 1 et 0.

Exemple

- On considère deux sources S_A et S_B qui transmettent en même temps.
- La source S_A transmet le message A ayant pour information la chaîne binaire '100' mais codée par la séquence <1 -1 -1> où un bit initial à 1 est remplacé par un élément à 1 dans la séquence et un bit initial à 0 est remplacé par -1. La source S_B transmet le message B ayant pour information la chaîne binaire '001' mais codée par la séquence <-1 -1 1> avec la même règle que pour le message A.
- La source S_A utilise comme code la séquence C_A = <1 -1 -1 1>. La source S_B utilise comme code la séquence C_B = <1 -1 1 -1>. Les deux codes sont choisis pour être orthogonaux, c'est-à-dire que leur produit scalaire C_A*C_B est nul et le produit scalaire C_A*C_A est maximum.
- Le message A est multiplié par le code C_A pour obtenir le produit $A*C_A$:
  $A*C_A$ = {<1 -1 -1 1>, <-1 1 1 -1>, <-1 1 1 -1>}.
- Le message B est multiplié par le code C_B pour obtenir le produit $B*C_B$
  $B*C_B$ = {<-1 1 -1 1>, <-1 1 -1 1>, <1 -1 1 -1>}.
- Les séquences correspondant aux deux produits $A*C_A$ et $B*C_B$ traduites en termes de bits sont transmises. Une fois transmise simultanément, ces deux séquences produits, $A*C_A$ et $B*C_B$, sont additionnées car les signaux simultanés s'additionnent.
  $A*C_A + B*C_B$ = {<0 0 -2 2>, <-2 2 0 0>, <0 0 2 -2>
- A la réception, le destinataire du message A (et qui connaît le code C_A) multiplie la séquence reçue par le code C_A. On a : $(A*C_A + B*C_B)*C_A$ = {<0 0 2 2>, <-2 -2 0 0>, <0 0 -2 -2>}. On prend la moyenne des signaux reçus sur la durée d'un bit initial. C'est-à-dire (0+0+2+2)/4 = 1, (-2-2+0+0)/4 = -1, (0+0-2-2)/4 = -1. Ce qui permet de retrouver la séquence <1 -1 -1> et ensuite la chaîne initiale '100'.
- A la réception, le destinataire du message B (et qui connaît le code C_B) multiplie la séquence reçue par le code C_B. On a : $(A*C_A + B*C_B)*C_B$ = {<0 0 -2 -2>, <-2 -2 0 0>, <0 0 2 2>}. Avec la même procédure de calcul de la moyenne effectué pour le message A, on retrouve la chaîne '001'.

Le CDMA peut être combiné aux techniques de multiplexage temporel et fréquentiel pour donner lieu au WCDMA (wideband CDMA), TD-CDMA (Time Division CDMA)...

IV. Méthodes d'accès par multiplexage

Il y a trois catégories de multiplexage : multiplexage temporel, multiplexage fréquentiel et multiplexage d'ondes.

IV.1. Multiplexage temporel (TDMA : Time Division Multiple Access)

Cette technique est aussi connue sous le sigle TDMA ("Time Division Multiplexing Access"). L'allocation du support de communication fonctionne de manière cyclique. On fixe, selon les besoins des nœuds connectés, la durée d'un tour d'allocation du support (soit *TT* cette durée). Chaque nœud *i* connaît sa position exacte dans un tour et a la droit d'émettre au maximum pendant un temps H_i. La somme des H_i est égale à *TT*. Les valeurs de temps (quanta) des H_i alloués aux nœuds peuvent être identiques ou différentes selon l'importante et la quantité du flux de données généré par chaque nœud. Dans le premier cas, on parle de **TDMA synchrone** et dans le second, de **TDMA statistique**.

L'inconvénient majeur du TDMA synchrone est que lorsqu'un nœud n'a pas de données à émettre, le support reste libre, même si d'autres nœuds ont beaucoup de trames à transmettre. On notera que cette technique est utilisée dans le domaine de la téléphonie.

Le TDMA statistique améliore l'utilisation de la bande passante en fixant de manière dynamique la durée d'utilisation du médium par chaque station. La signalisation (gestion des demandes, notification à chaque station de la durée quelle peut utiliser) rend ce TDMA plus complexe à mettre en oeuvre.

IV.2. Multiplexage fréquentiel (FDMA : Frequency Division Multiple Access)

Dans ce cas, la bande de fréquences du réseau est subdivisée en sous-canaux et chaque nœud n'a le droit d'émettre que sur un seul sous-canal qui lui est réservé. On notera que cette technique est très utilisée dans le domaine de la radio (où chaque chaîne de radio émet sur une bande de fréquences qui lui est réservée).

IV.3. Multiplexage d'ondes (WDM : Wavelength Division Multiplexing)

Ce multiplexage n'est possible que pour les réseaux où la transmission ce fait par ondes lumineuses, c'est le cas essentiellement des fibres optiques. Il s'agit d'utiliser des faisceaux lumineux avec des bandes de fréquences différentes (dans ce sens, WDM est un cas particulier du multiplexage fréquentiel). Chaque couleur de faisceau constitue un canal. Actuellement, on peut avoir, sur une même fibre optique disponible dans le commerce, plusieurs centaines de faisceaux lumineux offrant chacun un débit de plusieurs dizaines de Gb/s sur plusieurs dizaines (centaines) de Km.

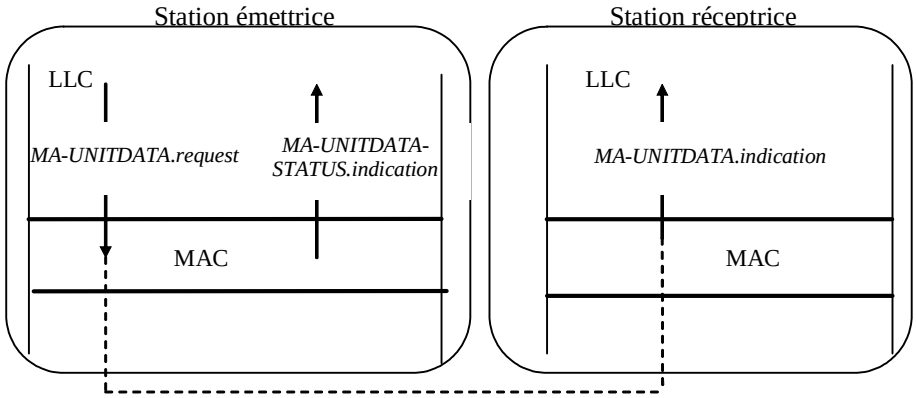
V. Interface MAC/LLC

L'interface entre MAC et LLC offre, en général, les services suivants :

1. *MA-UNITDATA.request* : permet à la sous-couche LLC de demander à la sous-couche MAC de transmettre une trame. Les paramètres de cette primitive englobent au moins : l'adresse source, l'adresse de destination et les données.
2. *MA-UNITDATA.indication* : permet à la sous-couche MAC d'indiquer à la sous-couche LLC qu'une trame est arrivée. Les paramètres de cette primitive englobent au moins : l'adresse source, l'adresse de destination et les données.
3. *MA-UNITDATA-STATUS.indication* : permet à la sous-couche MAC de rendre un compte à la sous-couche LLC concernant sa demande de transmission (réussite ou échec de la demande transmission). Les paramètres de cette primitive englobent au moins : l'adresse source, l'adresse de destination et le résultat d'émission. Un résultat positif signifie seulement que la trame a pu être envoyée, cela ne signifie pas

nécessairement que la trame a été effectivement reçue par son destinataire. Nous verrons que c'est la sous-couche LLC qui gère les acquittements.

Les initiales '*MA*' sont rajoutées aux primitives précédentes pour indiquer qu'il s'agit de primitives de niveau MAC.



Primitives de l'interface MAC/LLC

Exercices

Exercice 1
Combien d'émetteurs peuvent-ils transmettre en parallèle en utilisant CDMA avec des codes à 4 bits ? à 5 bits ? à n bits ?

Exercice 2
Quel est le débit maximum de mobiles qui transmettent en CDMA dans une bande de fréquence de 5 GHz avec des codes à 50 bits ?

Exercice 3
Quel est le nombre maximum de mobiles qui peuvent transmettre en même temps dans une cellule GSM en FDMA ?

Chapitre 2
Etude du réseau Ethernet

I. Introduction

Le réseau Ethernet est l'un des premiers réseaux locaux à voir le jour. Il est apparu au début des années 1970. Il a été inventé par Bob Metcalfe, ensuite il a été développé et commercialisé par DEC, Intel et Xerox. C'est incontestablement le réseau le plus vendu au monde. Il est quasiment le seul à être utilisé pour les communications qui n'ont pas de contraintes d'environnement ou de temps réel (il est utilisé en bureautique, dans le campus, dans les hôpitaux, etc.). Il fonctionne avec la technique CSMA/CD qui est très simple à comprendre et à mettre en œuvre.

Ethernet utilise une topologie en bus, en étoile ou en arborescence (ces deux dernières sont basées sur l'utilisation d'équipements de raccordement appelés hubs).

II. Technique CSMA/CD

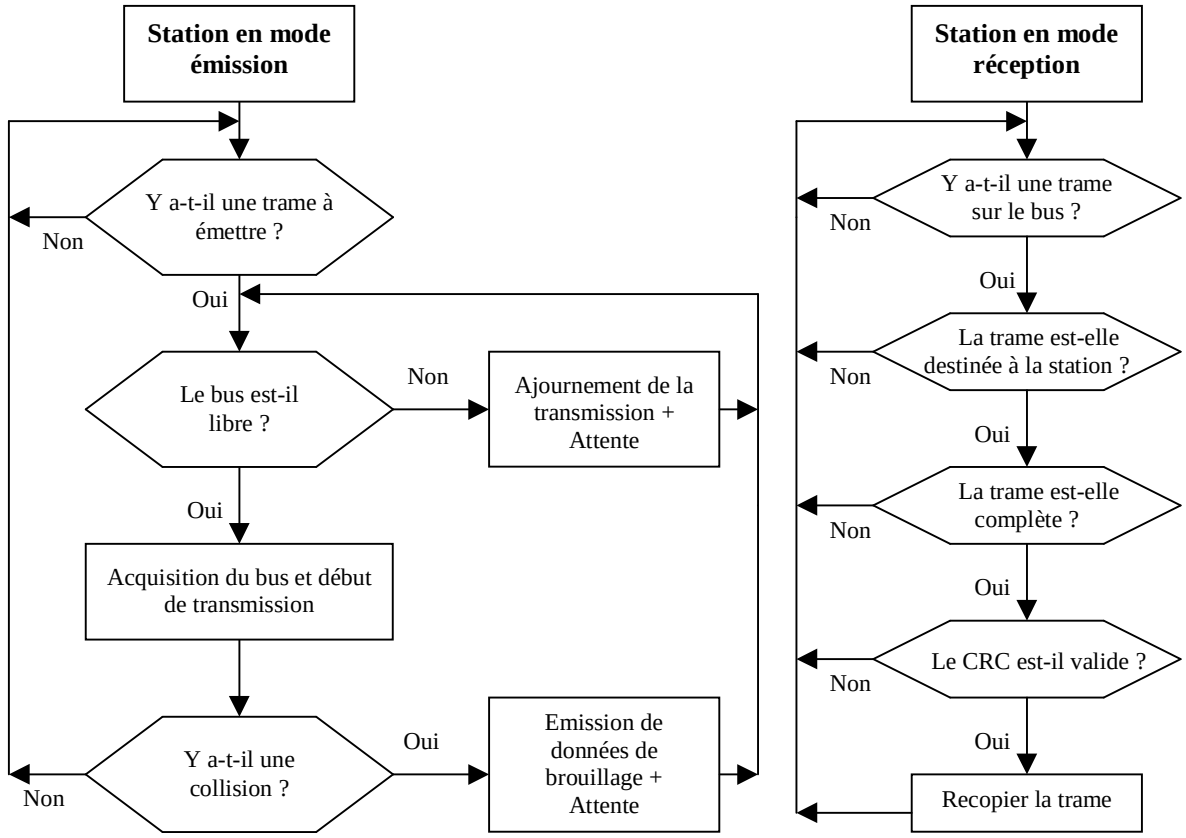
II.1. Principe

CSMA/CD = Carrier Sense Multiple Access with Collision Detection
= Accès multiple avec écoute de la porteuse et détection de collision.

CSMA/CD est l'une des techniques d'accès les plus anciennes dans le domaine des réseaux locaux. Son efficacité reste incontestable pour beaucoup de secteurs. Elle fonctionne selon un principe simple :

> *Toute station a le droit d'utiliser le support de transmission dès qu'elle détecte qu'il est libre, sous réserve de pouvoir détecter les conflits d'accès avec les autres stations.*

Le fonctionnement d'une station peut se résumer par la figure suivante :



Principe général de CSMA/CD.

II.2. Détection et résolution des conflits d'accès (collisions)

1) Détection de conflit

Il existe deux modes de transmission avec CSMA/CD : transmission en large bande et transmission en bande de base. La détection de collision dépend du mode de transmission.

- En bande de base, le signal physique se propage dans les deux sens du câble et chaque station qui se trouve en mode émission compare ce qu'elle émet par rapport à ce qu'elle reçoit sur le câble. Si le signal émis est différent du signal reçu, alors il y a collision.
- En large bande, le signal physique se propage de manière unidirectionnelle. La station émettrice attend le retour de son signal après que celui-ci ait été retransmis par la tête de câble pour effectuer la comparaison. Si le signal émis et différent de celui reçu, alors il y a collision.

2) Résolution de conflit

Après détection d'une collision, la station continue d'émettre des données dites « données de brouillage », pendant un temps dit « durée de brouillage », pour être sûre que toutes les autres stations concernées par la collision détectent celle-ci. Ensuite, la station attend un délai minimum avant de tenter de retransmettre. Le délai d'attente aléatoire avant retransmission est calculé comme suit :

*Si $N \leq Nmax$ alors Temps d'attente = R*Temps_de_base*
N : nombre de retransmissions déjà effectuées
Nmax : nombre maximum de tentatives de retransmissions autorisées
$R = Random(0, 2^L)$
$L = min(N, 10)$

II.3. Longueur minimale de trame

Pour pouvoir détecter une collision, la station émettrice doit comparer ce qu'elle émet par rapport à ce qu'elle reçoit. Un bit transmis par une station met un certain temps avant de passer devant toutes les autres stations (c'est le délai de propagation du signal). Le délai le plus long est celui qui correspond au cas où le signal émis par une station se trouant à une extrémité du câble doit arriver à la station se trouvant à l'autre extrémité du câble. Ainsi, lorsque la station émet sur un réseau en bande de base, sa trame doit durer au moins l'équivalent d'un aller-retour sur toute la longueur du câble. Si la station émet sur un réseau en large bande, cette durée est doublée (car la transmission est unidirectionnelle et les stations émettent sur un câble et reçoivent sur un autre). Par conséquent, les trames émises sur un réseau fonctionnant avec CSMA/CD doivent avoir une taille supérieure ou égale à une taille minimum pour que les stations puissent détecter les collisions en toute circonstance.

Si on note

D : le débit du bus

Tmax :le temps maximum d'aller-retour du signal entre deux stations du réseau

Lmin : taille minimale de trame

Alors

*Lmin ≥ D*Tmax pour un réseau en bande de base*

*Lmin ≥ 2D*Tmax pour un réseau en large bande.*

Si les données de la couche supérieure sont insuffisantes pour former une trame dont la taille est supérieure ou égale à *Lmin*, alors de **données de bourrage** sont rajoutées à l'émission et elles sont retirées à la réception avant de passer les données utiles à la couche supérieure.

L'introduction de données de bourrage dans les trames peut être considérée comme un inconvénient du réseau Ethernet. Dans la pratique, Ethernet est utilisé dans des applications (en bureautique en particulier) où on a rarement recours aux données de bourrage.

II.4. Paramètres de fonctionnement de réseau

Le standard IEEE 802.3 qui régit le fonctionnement de CSMA/CD fixe un ensemble de paramètres nécessaire au fonctionnement du protocole CSMA/CD. Il s'agit de :

- temps de base utilisé pour les diverses temporisations = temps pour émettre 512 bits ;
- temps intertrames : varie selon le débit (96 μs à 1 Mb/s, 9.6 μs à 10 Mb/s, …) ;
- nombre maximum de tentatives de retransmissions = 16 ;
- nombre de bits de brouillage = 32 ;
- taille maximale de trame = 1518 octets (1500 octets de données si l'on considère que les adresses sont codées sur 2 octets);
- taille minimale de trame = 64 octets (46 octets de données si l'on considère que les adresses sont codées sur 2 octets).

II.5. Format de trame

Préambule	7 octets
10101011	1 octet
Adresse de destination	6 octets
Adresse source	6 octets
Longueur de données utiles	2 octets
Données utiles + Bourrgae	n octets $46 \leq n \leq 1500$
Bits de contrôle (CRC)	4 octets

Format de trame Ethernet.

Le CRC est calculé par un code cyclique avec un polynôme générateur :

$$g(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

II.6. Primitives de niveau MAC

L'interface d'accès au MAC de CSMA/CD est définie par trois primitives :

- *MA-DATA.request* : pour demander la transmission d'une trame ;
- *MA-UNITDATA-STATUS.indication* : pour fournir le résultat de la dernière demande de transmission.
- *MA-DATA.indication* : pour délivrer à la sous-couche LLC une trame reçue.

III. Couche physique

III.1. Débits, support et longueur maximum

La couche physique du réseau Ethernet est définie par le standard IEEE 802.3 (qui est aussi une norme ISO : ISO 8802/3).

Un réseau Ethernet peut être constitué d'un seul segment ou de plusieurs segments interconnectés par des répéteurs. Le nombre maximum de segments est limité.

A l'origine, Ethernet ne pouvait fonctionner quasiment que sur du câble coaxial. Actuellement, Ethernet fonctionne sur quasiment tous les supports de transmission.

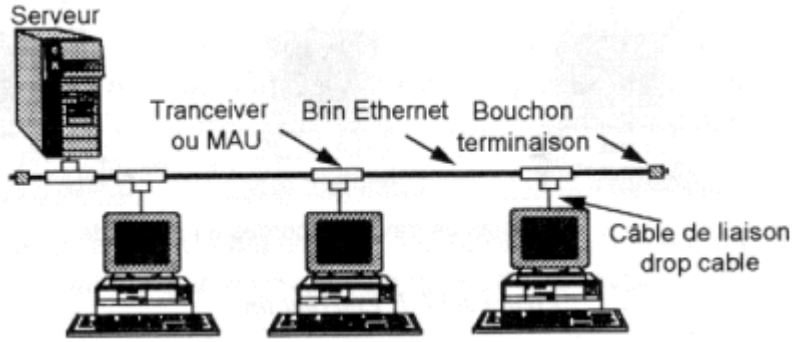
Norme IEEE	Débit (en Mb/s)	Support	Longueur maximum d'un segment
802.3 10Base5 Ethernet standard (Thick Ethernet)	10	Câble coaxial (50 Ω)	500 m
802.3 10Base2 Ethernet fin (Thin Ethernet)	10	Câble coaxial (50 Ω)	185 m
802.3 10BaseT	10	Paire torsadée (catégorie 3 ou 4)	100 m
802.3 10BaseF	10	Fibre optique	2 000 m
802.3u 100BaseTX (Fast Ethernet)	100	Paire torsadée (catégorie 5)	100 m
802.3u 100BaseT4 (Fast Ethernet)	100	Paire torsadée (catégorie 3 ou 4)	100 m
802.3u 100BaseFX (Fast Ethernet)	100	Fibre optique	2 000 m
100BaseVG (Fast Ethernet)	100	Paire torsadée Fibre optique	100 m 2 000 m
802.3z 1000Base-LX (Gigabit Ethernet)	1000	Fibre optique	500 m
802.3z 1000Base-SX (Gigabit Ethernet)	1000	Fibre optique	500 m
802.3z 1000Base-CX (Gigabit Ethernet)	1000	Paire torsadée	25 m
IEEE802.3ab :1000BaseT (Gigabit Ethernet)	1000	Câble coaxial, Paire torsadée	100 m
802.3ae 10GbaseCX4 (Gigabit Ethernet)	10 000	Fibre optique	15 m
802.3ae 10GbaseT (Gigabit Ethernet)	10 000	Fibre optique	100 m
802.3ae 10GbaseSR (Gigabit Ethernet)	10 000	Fibre optique	< 100 m
802.3ae 10GbaseLX4 (Gigabit Ethernet)	10 000	Fibre optique	10 000 m (MAN)
802.3ae 10GbaseLR (Gigabit Ethernet)	10 000	Fibre optique	40 000 m (MAN)

Classes de débits des réseaux Ethernet (standards IEEE 802.3x)

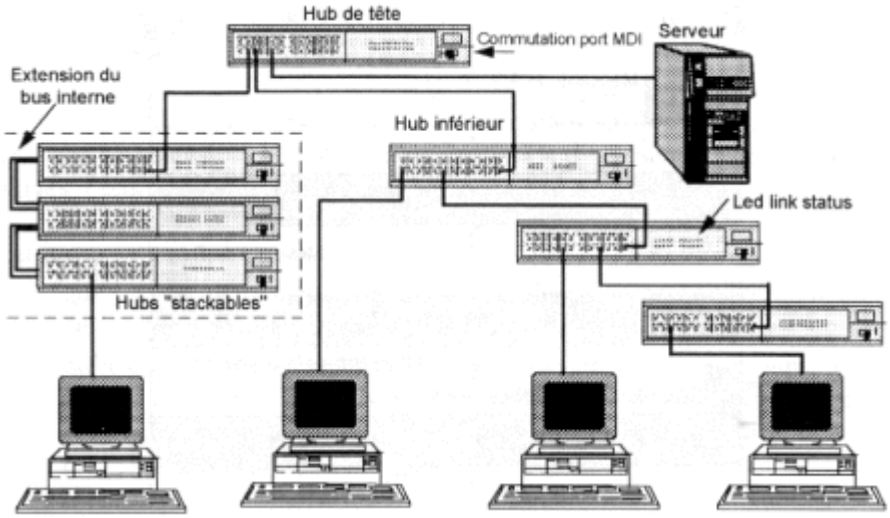
III.2. Topologies et raccordement

Un réseau Ethernet peut avoir une topologie en bus, en étoile ou en arborescence. Dans toutes les topologies, on peut utiliser des répéteurs pour accroître l'étendue du réseau. Dans les topologies en étoile ou en arbre, les raccordements se font via des équipements appelés **hubs**. Un hub n'a pas d'intelligence (c'est-à-dire qu'il ne traite pas les trames qui le traversent).

Les unités de raccordement de station au médium sont appelées MAU (Medium Access Units).



Exemple de réseau Ethernet avec une topologie en bus.



Exemple de réseau Ethernet avec une topologie en arbre (avec 4 niveaux de hubs).

IV. Ethernet commuté (« switched Ethernet »)

Dans la version initiale d'Ethernet, toute trame émise est entendue par l'ensemble des stations raccordées aux câbles du réseau, la bande passante disponible est partagée par l'ensemble des stations. Ce fonctionnement peut conduire à une perte de performance du réseau. En effet, lorsque le nombre de stations et/ou la longueur du réseau sont importants, le fait que toutes les stations doivent entendre chaque trame ralentit la vitesse globale du réseau. C'est un peut comme si dans une ville, à chaque fois qu'un véhicule veut franchir un carrefour tous les autres véhicules circulant dans la ville doivent attendre la fin du croisement de carrefour. C'est pour répondre à cette faiblesse d'Ethernet de base qu'a été introduit l'Ethernet commuté.

La topologie d'Ethernet commuté est l'étoile fondée sur l'utilisation de switch (commutateur). Le commutateur utilise un mécanisme de filtrage des trames. Il inspecte les adresses de source et de destination des trames, dresse une table qui lui permet alors de savoir quelle machine est connectée sur quel port du switch.

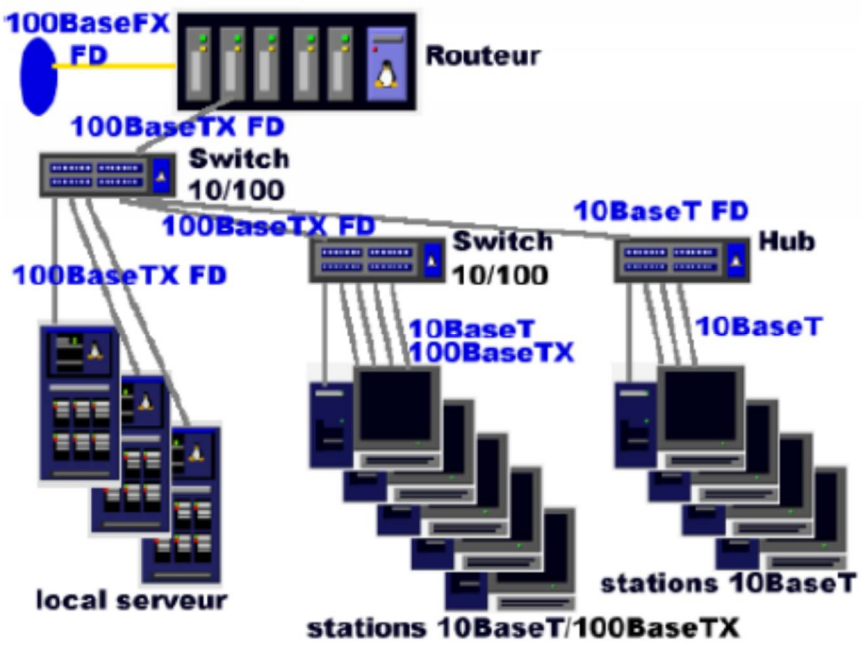
Connaissant le port du destinataire, le commutateur ne transmettra la trame que sur le port adéquat, les autres ports restant dès lors libres pour d'autres transmissions pouvant se produire simultanément. Il en résulte que chaque échange peut s'effectuer à débit nominal, sans collisions, avec pour conséquence une augmentation de la bande passante du réseau.

Puisque la commutation permet d'éviter les collisions et que les techniques 10/100/1000 base T(X) disposent de circuits séparés pour la transmission et la réception (une paire torsadée par sens de transmission), la plupart des commutateurs modernes permet de désactiver la détection de collision et de passer en full duplex sur les

ports. De la sorte, les machines peuvent émettre et recevoir en même temps (ce qui contribue à nouveau à la performance du réseau).

Les commutateurs Ethernet modernes détectent également la vitesse de transmission utilisée par chaque machine et si cette dernière supporte plusieurs vitesses (10 ou 100 ou 1000 megabits/sec) entament avec elle une négociation pour choisir une vitesse ainsi que le mode semi-duplex ou full-duplex de la transmission. Cela permet d'avoir un parc de machines ayant des performances différentes (dans une entreprise ou université par exemple, on ne change pas toutes les machines d'un seul coup, il faut donc pouvoir gérer les machines avec des débits hétérogènes).

Un autre avantage de la commutation dans Ethernet est qu'elle permet de construire des réseaux plus étendus géographiquement. En Ethernet partagé, une trame doit pouvoir atteindre toute machine dans le réseau dans un intervalle de temps précis (*slot time*) sans quoi le mécanisme de détection des collisions (CSMA/CD) ne fonctionne pas correctement. Ceci n'est plus d'application avec les commutateurs Ethernet. La distance n'est plus limitée que par les limites techniques du support utilisé (fibre optique ou paire torsadée, puissance du signal émis et sensibilité du récepteur, ...).



Exemple de topologie utilisant des commutateurs

Exercices

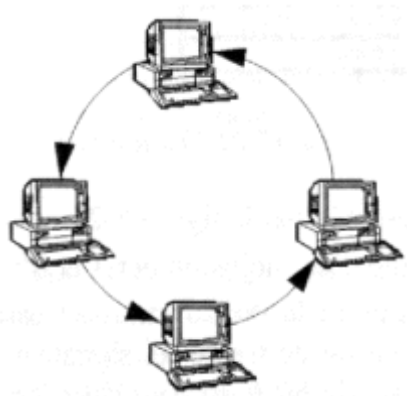
Exercice 1
On considère un réseau Ethernet à 100 Mb/s. Donner la longueur maximum de ce réseau si l'on considère que la longueur minimale de trame est de 100 octets.

Exercice 2
Etudier l'effet de la longueur d'un réseau Ethernet sur son rendement (rendement = nombre de trames reçues sans collision/ nombre de trames émises).

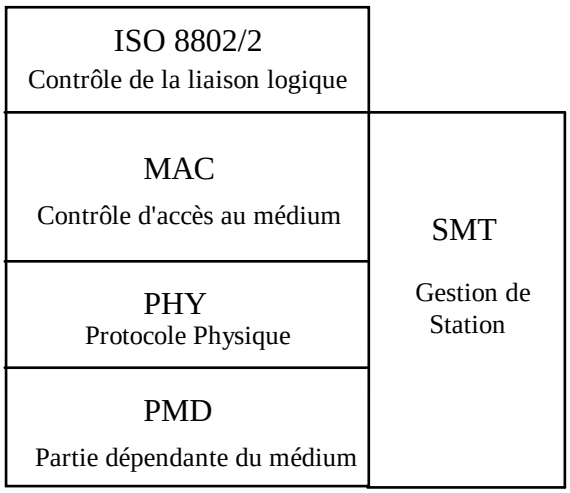
Chapitre 3
Etude du réseau FDDI

I. Caractéristiques générales

- FDDI (Fiber Distributed Data Interface) est un standard ANSI ;
- Utilise la fibre optique comme support de communication ;
- Topologie en boucle ;
- Débit de données = 100 Mb/s. Débit physique = 125 Mb/s
- Nombre maximum de nœuds = 500 (avec redondance du médium) et 1000 (sans redondance du médium)
- Longueur maximale de la boucle = 100 km (avec redondance du médium) et 200 km (sans redondance du médium)
- FDDI peut être utilisé comme réseau local ou comme réseau métropolitain.



Exemple de boucle FDDI.



Architecture du réseau FDDI.

II. Protocole MAC de FDDI

II.1. Principe de base de la technique MAC de FDDI

FDDI est fondé sur un protocole MAC à jeton temporisé. A tour de rôle, chaque station dispose d'une quantité de temps connue à l'avance pour transmettre ses trames. Une station ne peut commencer la transmission de ses données que si elle reçoit le jeton.

Le trafic généré par les utilisateurs de FDDI peut être classé en deux catégories :

- un trafic périodique (dit **synchrone**) dont les caractéristiques sont connues à l'avance ;
- un trafic apériodique dont les caractéristiques ne sont pas en général connues à l'avance. Les trames apériodiques peuvent être classées par ordre de priorité (des priorités allant de 1 à 8).

A l'origine, FDDI était introduit pour transporter en priorité du trafic synchrone (par exemple des images ou de la voix). Par conséquent, le protocole d'accès de FDDI privilégie le trafic synchrone.

A l'initialisation de la boucle, on définit :

- le TTRT (Target Token Rotation Time) qui indique le temps maximum de rotation du jeton sur la boucle.

- Un temps SA_i ("synchronous allocation") que chaque station *i* a le droit de consommer à chaque fois qu'elle reçoit le jeton.

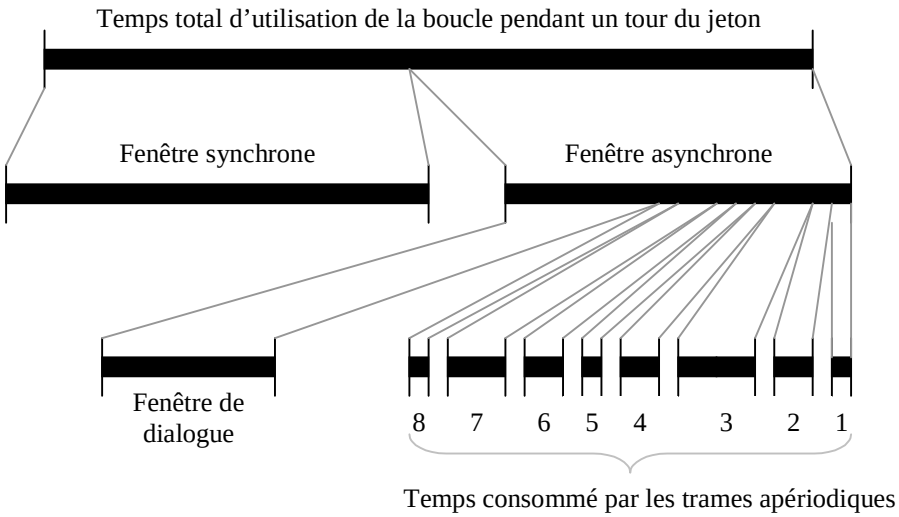
Lorsqu'une station *i* reçoit le jeton, elle transmet tout d'abord ses trames liées au trafic synchrone sans dépasser son quantum de temps SA_i. Ensuite, selon le temps pris par le jeton durant son dernier tour, la station peut transmettre des trames apériodiques, sans remettre en cause le TTRT fixé.

En plus des trames périodiques et apériodiques, une station peut monopoliser le temps destiné aux échanges de trames apériodiques (sans remettre en cause les quanta alloués au trafic synchrone des autres stations) pour ne communiquer qu'avec une seule autre station ; c'est ce que l'on appelle un **dialogue**. Ainsi, la boucle FDDI fonctionne avec un seul jeton (c'est le jeton normal) ou deux jetons, un jeton normal et un jeton de dialogue (dit **jeton restreint**).

Ainsi le temps d'utilisation de la boucle FDDI peut être décomposé comme suit :

- Une fenêtre synchrone dont la durée est égale à la somme des quanta alloués aux stations ;
- Une fenêtre asynchrone qui peut être décomposée en deux parties :
  + une fenêtre correspondant à un dialogue entre deux stations,
  + une fenêtre pendant laquelle des trames apériodiques (de la priorité 1 à 8) sont transmises.

Il faut noter que la fenêtre synchrone (ou asynchrone) n'est pas un intervalle de temps continu. Elle indique seulement la somme des quanta alloués pour le trafic synchrone (ou asynchrone).



Décomposition du temps d'accès à la boucle FDDI.

II.2. Algorithme d'utilisation de la boucle

L'algorithme suivant est exécuté par chaque station raccordée à un réseau FDDI :

`TTRT` (Target Token Rotation Time) : paramètre de fonctionnement de la boucle connu par toutes les stations après l'initialisation de la boucle. Avec une valeur X pour TTRT, cela signifie que chaque station reçoit le jeton au bout de X unités de temps en moyenne et au bout de 2X unités de temps au maximum.

`i` : numéro de station (i=1, ….n).

TRT_i (Token Rotation Time) : compteur qui permet de savoir si le jeton est reçu par la station en avance (si TRT_i expire avant l'arrivée du jeton, le jeton est en retard).

THT_i (Token Holding Time) : compteur qui indique le temps d'utilisation de la boucle pour transmettre des trames asynchrones.

LC_i (Late Counter) : compteur qui indique le nombre de fois que le jeton n'a pas été reçu pendant un intervalle de temps égal à `TTRT`. Normalement LC_i ne doit pas dépasser 1, sinon il y a réinitialisation de la boucle.

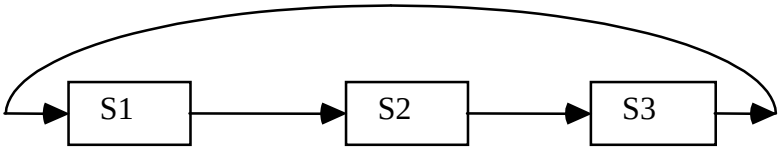
SA_i : temps d'allocation de la station `i` pour lui permettre de transmettre ses trames synchrones. Cette valeur est fixée à la configuration du réseau.

```
Début
    /* Phase d'initialisation */
    LC_i = 0
    Attendre_Jeton
    TRT_i = TTRT ; Déclencher TRT_i
       Passer le jeton  /* Dans le premier tour du jeton, il n'y a pas
       d'émission de trames ni synchrones, ni asynchrones */
    /* Phase de fonctionnement normal : utilisation du jeton pour
    transmettre des données */
    Tant que vrai Répéter
    {   Si TRT_i atteint la fin de temporisation Alors LC_i = LC_i +1
              Si LC_i > 1  Alors Réinitialiser la boucle
                           Sinon TRT_i = TTRT ; Déclencher TRT_i
              Finsi
       Finsi
       ||
       Si Arrivée du jeton
              Si LC_i = 0 alors
                    THT_i = TRT_i
                    TRT_i = TTRT
                    Déclencher TRT_i
                    Transmettre les trames périodiques, s'il y en a,
                    jusqu'à concurrence du SA_i
                    Déclencher THT_i
                    Tant que THT_i > 0 {Transmettre des apériodiques s'il y
                    en a}
                    Passer le jeton
               Sinon
                    LC_i = 0
                    /* TRT n'est pas réinitialisé dans ce cas, afin
                       d'accumuler le retard. */
                    Transmettre les trames périodiques jusqu'à concurrence
                    du SA_i
                    Passer le jeton
              Finsi
       Finsi
  }
 FIN
```

Exemple de fonctionnement de la boucle

Pour simplifier la présentation de l'exemple, on suppose que :

- les trames échangées sont de même taille, leur temps de transmission est égal à une unité de temps ;
- TTRT = 8 unités de temps ;
- SA_i = 2 unités de temps ;
- les temps de passage et de propagation du jeton sont négligeables ;
- la boucle vient d'être initialisée et un jeton arrive à la station S1 ;
- les stations ont suffisamment de trames pour utiliser, à chaque tour du jeton, tout le temps qui leur est imparti.



Passage du jeton numéro	Station S1			Station S2			Station S3		
	TRJ	TSyn	TAsyn	TRJ	TSyn	TAsyn	TRJ	TSyn	TAsyn
1	0	2	8	10	2	0	12	2	0
2	14	2	0	6	2	2	8	2	0
3	8	2	0	8	2	0	6	2	2
4	8	2	0	8	2	0	8	2	0
5	6	2	2	8	2	0	8	2	0
6	8	2	0	6	2	2	8	2	0
7	8	2	0	8	2	0	6	2	2

TSyn : nombre de trames synchrones émises par la station
TAsyn : nombre de trames asynchrones émises par la station

Exemple de déroulement de l'algorithme d'accès à la boucle FDDI.

II.3. Dialogue entre stations

La notion de dialogue introduite dans FDDI permet de réduire le temps de communication entre deux stations quand celles-ci souhaitent s'échanger des données rapidement, sans que ces données soient synchrones. Par exemple, une station effectue une transaction sur une autre station et veut recevoir le résultat de la transaction le plus rapidement possible. Il faut noter qu'il n'est pas obligatoire de passer par un dialogue pour permettre à deux stations de s'échanger rapidement des données. La notion de dialogue est un mécanisme efficace, mais rarement utilisé dans la pratique. Le principe de fonctionnement d'un dialogue selon le protocole FDDI est le suivant :

- La station qui désire commencer un dialogue reçoit un jeton normal.
- Elle émet sa première trame du dialogue et émet juste après un jeton restreint vers la station avec laquelle elle dialogue.

- Seule la station ayant reçu la dernière trame (c'est-à-dire la première trame du dialogue en cours) peut transmettre des trames asynchrones, en utilisant le jeton restreint.

- Les deux stations peuvent alors s'échanger des trames asynchrones et des jetons restreints pendant une période durant laquelle les autres stations ne peuvent pas émettre de trames asynchrones.

- Le dialogue se termine quand un jeton normal est mis en circulation.

- Le dialogue ne doit pas affecter la limitation imposée par le TTRT.

II.4. Procédures de gestion de la boucle FDDI

1) Initialisation de la boucle

Une station qui se connecte au réseau ou qui ne détecte aucun trafic sur le réseau lance une procédure d'initialisation de la boucle et création de jeton. Elle le fait en suivant les étapes suivantes :

- Elle émet une trame de *demande d'acquisition du jeton.*

- Elle émet sa demande de manière continue jusqu'à ce que sa demande ou une autre lui parvienne.

- Si une seule station fait la demande d'acquisition du jeton, elle obtient le jeton dès que sa demande lui revient.

- Si plusieurs stations demandent l'acquisition du jeton, en même temps, une seule station obtient le jeton, en respectant les règles suivantes :

  a)- Si la station ne demande pas à acquérir le jeton, elle répète la demande sur la boucle.

  b)- Si la station a demandé ou veut demander l'acquisition du jeton :
  + elle laisse passer la demande reçue, si elle est plus prioritaire que la sienne ;
  + elle absorbe la demande, si elle est moins prioritaire que la sienne.

  c)- Les règles de priorité sont fixées en tenant compte du TTRT demandé, de la taille des adresses et des adresses :
  + La demande ayant le TTRT le plus petit est prioritaire ;
  + En cas d'égalité des valeurs du TTRT, c'est la station qui a l'adresse la plus courte qui est prioritaire ;
  + En cas d'égalité des valeurs du TTRT et des longueurs des adresses, c'est la station qui a l'adresse la plus basse qui est prioritaire.

- La station qui s'impose crée un jeton et le fait circuler sur la boucle.

2) Contrôle de la boucle

Une station possédant le jeton peut tomber en panne sans libérer le jeton, un jeton peut se perdre durant son transfert de station à station, etc. Toutes ces anomalies peuvent compromettre le bon fonctionnement de la boucle. C'est la raison pour laquelle FDDI met en place un mécanisme de contrôle de la boucle de manière simple : toutes les stations contrôlent le bon fonctionnement de la boucle en utilisant des temporisateurs. Toute station lance la procédure d'initialisation de la boucle et de création du jeton (décrite précédemment) si 1) le temps de rotation du jeton dépasse 2*TTRT (sauf à l'initialisation), 2) le temps de propagation d'une trame est supérieur à celui d'un tour complet de la boucle.

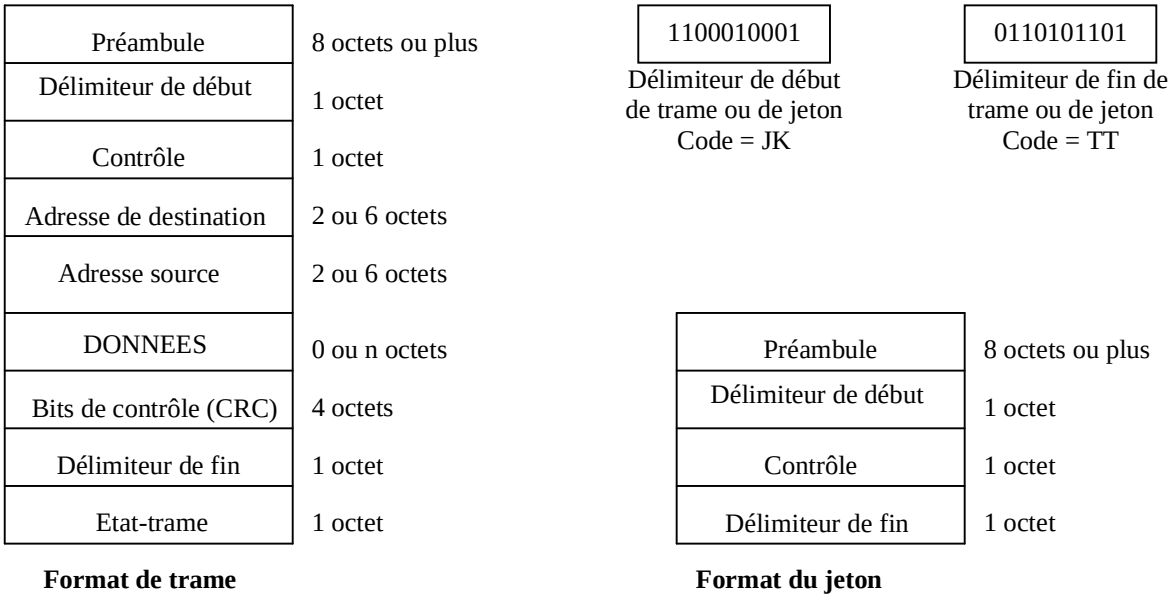
Lorsque la boucle est coupée, les stations lancent une procédure pour détecter l'endroit de coupure de la boucle qui fonctionne de la manière suivante :

- La station émet une trame *feu d'alarme* en continu ;

- La station arrête de transmettre sa trame feu d'alarme quand celle-ci lui revient ;

- Quand la station reçoit sa trame feu d'alarme, elle déclenche la procédure d'initialisation de la boucle.

II.5. Format des trames FDDI



Champ contrôle

CLFFZZZZ

C : indique si la trame est synchrone ou asynchrone
L : indique si les adresses sont sur 2 ou 6 octets
FF : indique si c'est une trame LLC ou une trame de contrôle MAC
ZZZZ : bits de contrôle

CLFFZZZZ	Signification
10000000	Jeton normal
11000000	Jeton restreint
1L000011	Demande de jeton
1L000010	Feu d'alarme
0L01rPPP	Trame asynchrone avec la priorité PPP
1L01 rrrr	Trame synchrone (rrrr : réservé pour extension future)

Etat-trame

EACx x .. x

E : indique s'il y a eu erreur de transmission ou non
A : indique si l'adresse de destination a été reconnue ou non
C : indique si la trame a été recopiée par son destinataire ou non
x..x : extension des bits d'état

Format de trame de FDDI

II.6. Primitives MAC de FDDI

L'interface d'accès au niveau MAC de FDDI offre des services similaires aux autres réseaux. Mais comme FDDI, au moment de son introduction, a été jugé très rapide par rapport aux autres réseaux existants à l'époque, on a introduit la possibilité de demander la transmission, à l'aide d'un seul appel de primitive, de plusieurs trames contenant des données destinées à plusieurs stations. Cette possibilité est un plus, mais on peut faire la même chose en appelant la primitive d'envoi de trame autant de fois qu'il y a de données à émettre à des destinataires différents.

Par ailleurs, pour accélérer les transmissions, le niveau LLC peut demander au niveau MAC de piéger le prochain jeton même s'il n'y a pas encore de données prêtes à émettre de manière à le garder (sans remettre en cause évidemment le fonctionnement de la boucle) jusqu'à ce qui des données soient prêtes. Si de telles données ne sont pas prêtes alors que le temps d'utilisation du jeton imparti à la station vient à échéance, celle-ci doit libérer le jeton.

Les primitives de service MAC FDDI sont :

- *MA-DATA.request (val-FC(1), adrs_destination(1), unité_données(1), classe_service(1), flot(1), val_FC(2), adrs_destination(2), unité_données(2), classe_service(2), flot(2), ... val_FC(n), adrs_destination(n), unité_données(n), classe_service(n), flot(n), classe_jeton)*

  Demande de transmettre une ou plusieurs données vers une ou plusieurs destinations en utilisant un jeton de classe normale ou de dialogue.

- *MA-DATA.confirmation(nombre_unités_données, état_transmission, classe_service_fourni)*

  La sous-couche MAC informe la sous-couche LLC du résultat concernant sa dernière demande de transmission. Elle lui indique le nombre d'unités de données, l'état de transmission (échec ou réussite) de ces unités de données et si elles ont été transmises, la classe de service fourni est indiquée.

- *MA-DATA.indication (val_FC, adrs_destination, adrs_source, unité_de_données, état_réception)*

  La sous-couche MAC passe à la sous-couche LLC une trame qui lui arrive du niveau physique avec un état de réception indiquant si la trame a été reçue avec ou sans erreurs de transmission.

- *MA-TOKEN.request (classe_jeton_à_capturer)*

  La sous-couche LLC demande à la couche MAC de bloquer le prochain jeton dont la classe est précisée, même s'il n'y a pas encore de données prêtes à émettre.

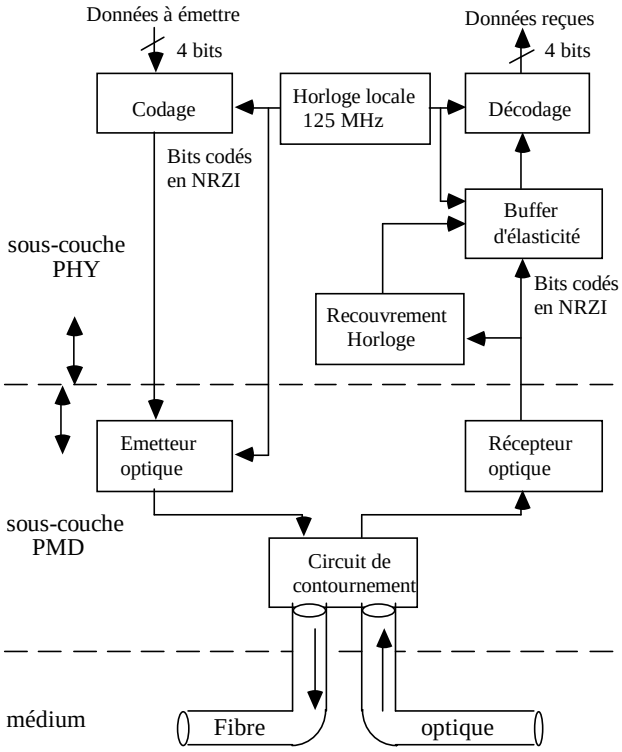
III. Couche physique

III.1. Code 4B/5B

La couche physique de FDDI utilise le codage NRZI + **4B/5B**. Cela signifie que chaque octet est transmis en NRZI (Non Return to Zero Inverted) sous la forme de 10 bits. Il y a donc une perte de 20% du débit du réseau. Le choix du code 4B/5B a été fait pour augmenter les possibilités de détection d'erreurs au niveau physique, puisque avec 5 bits on peut avoir 32 combinaisons binaires dont seulement 16 sont valides. La modulation au niveau physique se fait à 125 Mb/s (pour avoir les 100 Mb/s au niveau MAC). Ce choix, discutable aujourd'hui, a été fait fin des années 1980 où on considérait que même avec 80% du débit (c'est-à-dire 100 Mb/s au lieu de 125 Mb/s) FDDI restait un réseau très rapide. Ceci n'est plus vrai aujourd'hui.

Code	Valeur en Héxa	Code	Valeur en Héxa	Code	Symbole de contrôle
11110	0	10010	8	00000	quiet
01001	1	10011	9	11111	idle
10100	2	10110	A	00100	halt
10101	3	10111	B	11000	J
01010	4	11010	C	10001	K
01011	5	11011	D	01101	T
01110	6	11100	E	00111	R
01111	7	11101	F	11001	S
Les combinaisons binaires restantes sont invalides					

Code 4B/5B.

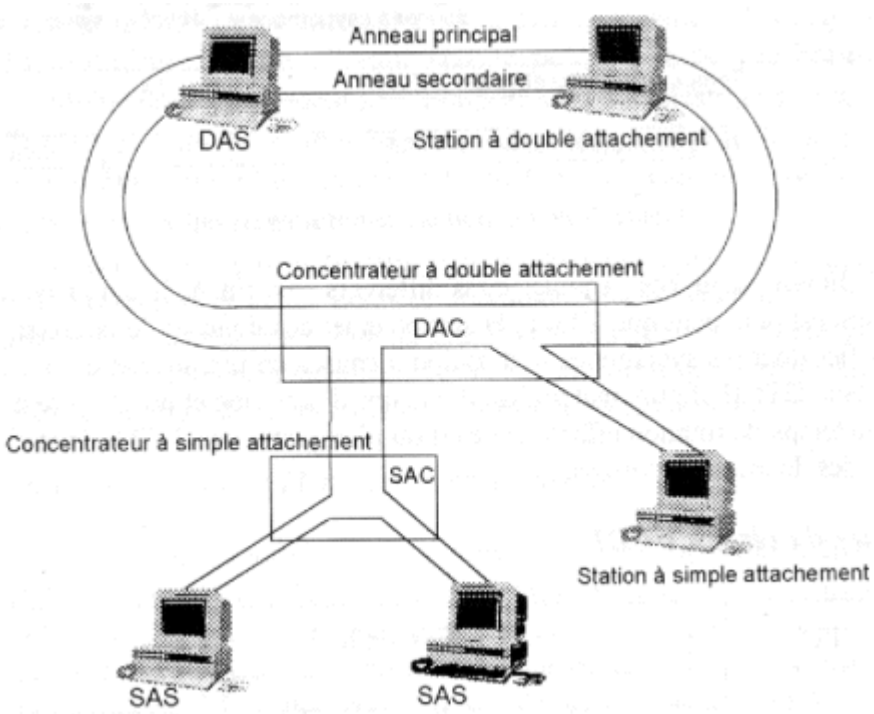


Principe de la transmission du réseau FDDI.

III.2. Câblage de boucle

Pour faire face aux coupures du support physique et aux pannes de stations, le réseau FDDI a introduit la redondance du médium en permettant d'avoir deux boucles (un anneau principal et un anneau secondaire). Quand des tronçons de l'anneau principal sont coupés, il y a basculement sur les tronçons adéquats de l'anneau secondaire. De même, lorsqu'une station tombe en panne, elle est court-circuitée en utilisant des tronçons de l'anneau secondaire.

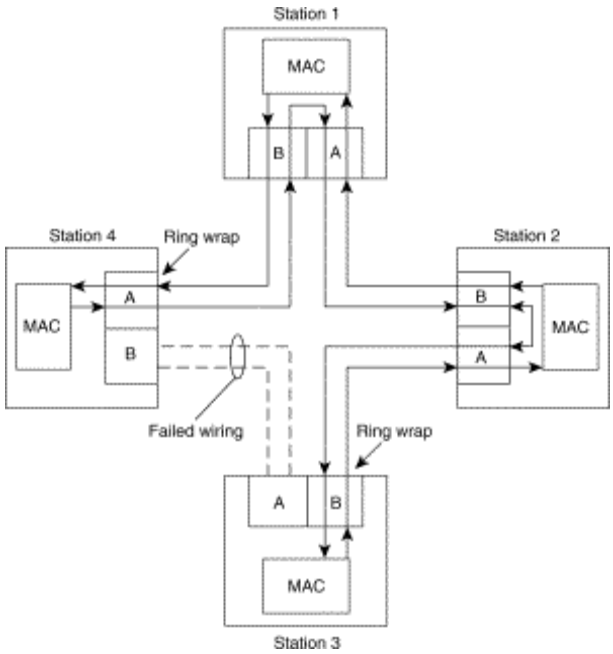
Un équipement peut être raccordé aux deux anneaux ou à un seul (cela permet de minimiser le coût du câblage).



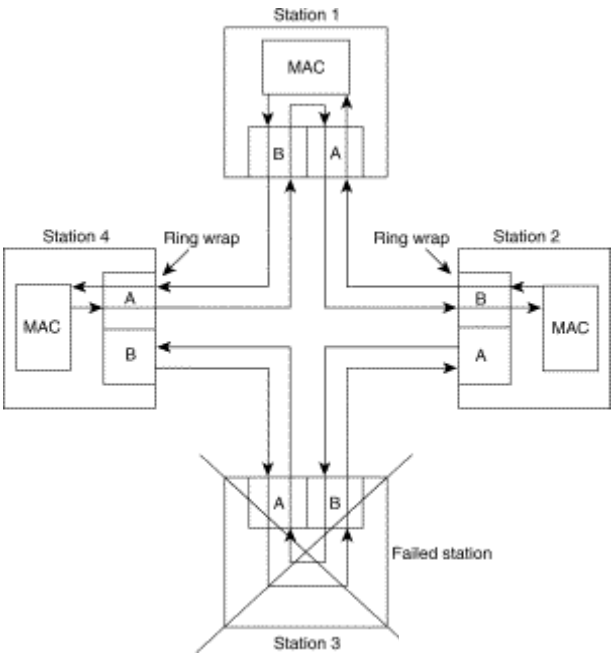
DAS : Double Attachment Station - SAS : Simple Attachment Station –
SAC : Simple Attachment Concentrator.

Topologie à deux anneaux et raccordement.

Les deux figures suivantes montrent des exemples de contournement de tronçon coupé ou de station en panne.



Exemple de contournement de tronçon coupé.



Exemple de contournement de station en panne.

Exercices

Exercice 1

On considère trois stations S1, S2 et S3 qui transmettent des trames de même longueur. On fixe TTRT à 8 ut et le temps d'allocation de la bande synchrone à 2 ut. On suppose que la durée de transmission d'une trame de données est égale à 1 ut et que le temps de transmission du jeton et le temps de propagation sont négligeables.

Q1. On suppose que chaque station (S1, S2 et S3) génère deux trames de manière périodique toutes les 8 ut : à t = 0, t = 8ut, t =16 ut… Etudier le fonctionnement de FDDI dans ce cas.

Q2. Que se passe t-il si une des stations génère 4 trames toutes les 8 ut ? Que se passe t-il si une des stations génère 5 trames toutes les 8 ut ?

Q3. On suppose que les stations ont suffisamment de trames synchrones et asynchrones pour utiliser le jeton aussi longtemps que possible. Calculer le temps maximum séparant deux passages successifs du jeton dans une station.

Q4. Montrer que le temps de rotation du jeton est toujours inférieur ou égal à 2TTRT.

Exercice 2

Un responsable de la communication et surveillance d'un grand magasin souhaite installer un réseau pour connecter différents appareils. Il veut opter pour l'utilisation de FDDI et s'adresse à vous pour l'éclairer.

Dans un premier temps, il souhaite raccorder au réseau FDDI des caméras de surveillance et des écrans. Chaque caméra envoie de manière périodique (toutes les 250 ms) une image composée d'une matrice de 200x200 pixels. L'information (couleur, brillance, etc.) représentant un pixel est codée sur un octet.

Q1. Quel est le nombre maximum de caméras de surveillance qu'il peut connecter au réseau ?

Dans un deuxième temps, il souhaite raccorder des caisses au réseau. Une caisse fonctionne de manière autonome, mais elle a besoin de télécharger (au démarrage) la table des prix des articles vendus par le magasin.

L'opération de téléchargement des prix est une opération non périodique. Le système de contrôle d'une caisse entre en communication (dialogue) avec un serveur qui centralise les prix pour lui envoyer la table des prix. La table des prix est stockée dans un fichier de 20 k octets (noter que la taille maximale d'une trame FDDI est de 4500 octets).

Le responsable souhaite que lorsqu'une caisse veut démarrer, elle ne doit pas attendre plus de *10* secondes pour obtenir la table des prix. Deux caissières au plus peuvent démarrer leur caisse dans un intervalle de temps d'un quart d'heure.

Q2. Quel est le nombre maximum de caisses et de caméras qu'il peut raccorder au réseau ?

Exercice 3

On considère deux stations S1 et S2 reliée via FDDI.
Préciser les appels de service et les trames échangées pour permettre aux deux stations :

- d'ouvrir une connexion de liaison de données
- d'échanger une dizaine de trames (sans erreur) de S1 vers S2 et 6 trames de S2 vers S1. On suppose que le niveau LLC utilise un mécanisme de contrôle de flux avec une fenêtre d'anticipation de taille égale à 4.

Chapitre 4
Réseau ATM
Asynchronous Transfer Mode

I. RNIS et ATM

Un réseau numérique à intégration de services (ou ISDN : Integrated Services Digital Network) est un réseau qui offre une connectivité numérique de bout en bout pour supporter une large gamme de services (voix, images vidéo, TV, fax, transfert de données, …) auxquels les utilisateurs accèdent par interfaces standards (définition du CCITT).

RNIS de base permet des canaux à 64 kb/s (canaux B), des canaux de signalisation à 16 kb/s (canaux D) et des canaux à accès primaires à 384, 1536 ou 1920 kb/s (canaux H0, H11 et H12). Les canaux H sont obtenus en utilisant plusieurs canaux B et D (ex. un canal H11 = 23B + 4D).
RNIS LB ou **B-ISDN** (broadband ISDN) permet de répondre aux besoins de débits élevés (obligatoires pour l'image, TV, …). Débits offerts : 32-34 (H2) 45, 70 (H3) et 135-140 (H4) Mb/s.
Plusieurs canaux H et D sont véhiculés par un canal H2, H3 ou H4.

Les services offerts via les B-ISDN sont de deux types les services interactifs (messagerie, vidéotex, télé-achat, news, télé-enseignement, télé-médecine, ..) et les services de distribution (TV). Ces services nécessitent de la communication d'informations de différents types : texte, données, documents, graphiques, images (fixes ou animés) et sons.

Type d'information		**Débit**	**Remarques**
Données		Débits très divers	Données avec des débits continus ou en rafales
Texte		Plusieurs kb/s	Transmission de texte de grand volume
Son	Téléphonie	64 kb/s 11 kb/s	Ligne téléphonique normale Voix sur IP (VoIP)
	Qualité CD	1,4 Mb/s	
Image	Télécopie	64 kb/s	Télécopie du groupe 4
	Vidéophonie	64-128 kb/s	Vidéo téléphonie (qualité réduite)
	TV std	120 Mb/s 1,5 Mb/s	TV standard non compressée TV standard compressée (MPEG-1)
	TV HD	1-3 Gb/s 10-30 Mb/s	HDTV non compressée HDTV compressée (MPEG-2)
	Traitement graphique	Très variable	Dépend des modèles 2D ou 3D, de la qualité des images, de l'animation…

Exemples de débits selon le type d'informations

ATM a été introduit pour être la principale technique utilisée pour implanter des B-ISDN avec une même technologie aussi bien pour raccorder les abonnés que les équipements des réseaux d'opérateurs. Aujourd'hui ATM est essentiellement utilisé au niveau des réseaux de cœur des opérateurs.

II. Principes et définitions de base de ATM

II.1. Principes de base de ATM

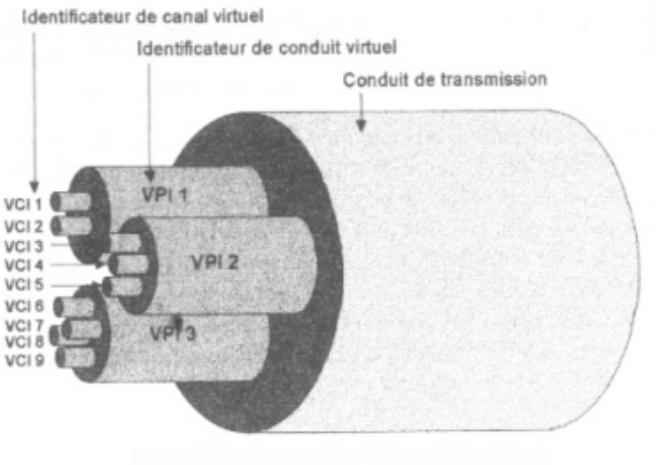
- ATM a été initié dans le cadre du projet PRELUDE du CNET en 1982.
- Des dizaines de normes existent.
- ATM forum (des centaines d'utilisateurs, fournisseurs et universitaires) pour promouvoir ATM.
- ATM est destiné au transport de tout type de données (sons, images, données, textes).
- ATM est indépendant de tout type de support et s'adapte à une large gamme de débit allant de 51,48 Mb/s à 9953,28 Mb/s.
- ATM permet le transfert de données isochrones (trafic périodique) ou asynchrones avec ou sans rafales.
- ATM offre un service orienté connexion par canaux virtuels (appelés aussi voies virtuelles) ou par chemins virtuels (appelés aussi conduits virtuels).
- ATM a été conçu par les opérateurs de télécommunication ce qui explique certains choix techniques pour faciliter le transport de la téléphonie numérique.

II.2. Canal virtuel, chemin virtuel et lien physique

Canal virtuel (virtual channel) : concept utilisé pour décrire un transport de données associées à un même identificateur (dit VCI : Virtual Channel Identifier).

Chemin virtuel (virtual path) : concept utilisé pour regrouper plusieurs canaux virtuels avec un même identificateur (dit VPI : Virtual Path identifier).

Chemin de transmission (transmission path) ou **lien physique** ou **conduit de transmission** : ensemble d'éléments (lignes, commutateurs, …) assemblés pour supporter une transmission de bout en bout. Un chemin de transmission peut supporter un ou plusieurs CV ou PV.

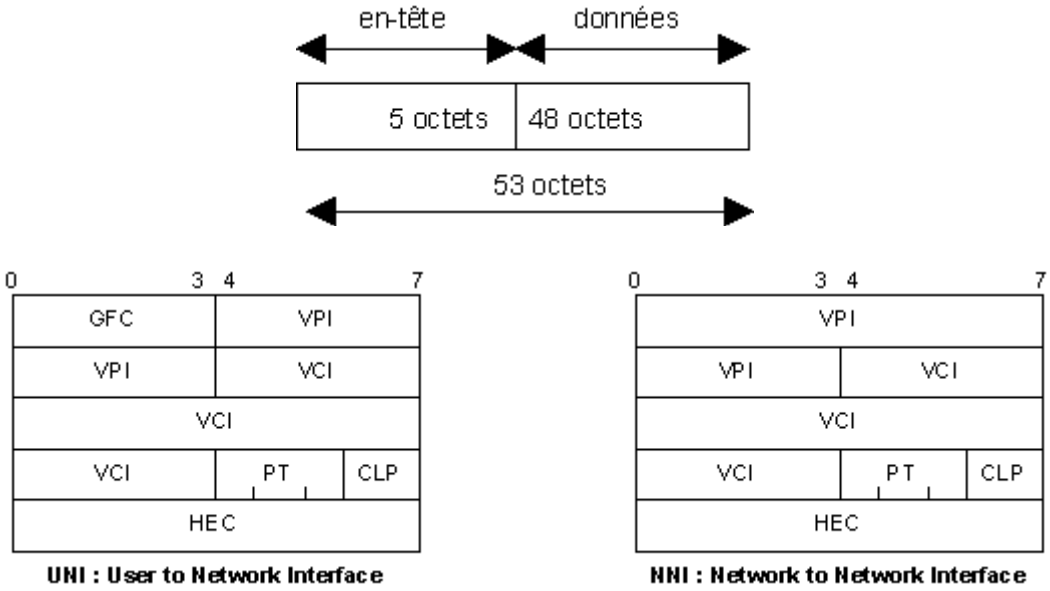


Relations entre VC, VP et lien physique.

II.3. Format des cellules d'ATM

Les cellules ATM ont une taille de 53 octets dont 48 octets de données au maximum (le nombre 48 est le résultat de longues discussions entre les européens et les américains, les uns voulaient 64, les autres voulaient 32, il a fallu contenter les deux parties en prenant la moyenne).

Le début de l'entête varie selon que l'on s'intéresse à l'interface entre l'utilisateur et le réseau (UNI : User Network Interface) ou l'interface entre deux nœuds ATM (Network to Network Interface).



Format de cellule ATM.

- GFC (Generic Flow Control) : ce champ est écrasé par le premier commutateur ATM rencontré par une cellule. Il ne réapparaît pas lorsque la cellule arrive à destination. Il est inutilisé dans les implantations actuelles. Il devrait servir au contrôle de flux.

- VPI : Virtual Path Identifier.

- VCI : Virtual Channel Identifier.

- PT (payload) : indique le type d'informations contenues dans la cellule (cellules de l'utilisateur, cellule de gestion de réseau, etc.).

- CLP (Cell Loss Priority) : les cellules marquées avec CLP=1 peuvent être détruites en cas de congestion de réseau.

- HEC (Header Error Control) : contrôle d'erreur sur l'entête de cellule seulement.

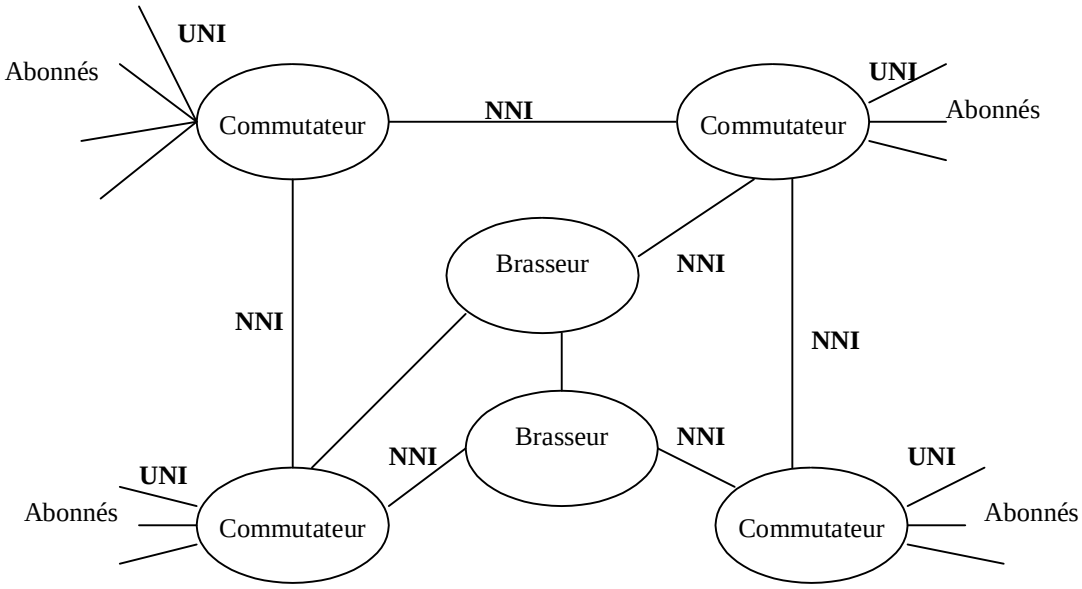
II.4. Architecture de réseau ATM

Toutes les topologies (bus, arbre, étoile, etc.) et tous les types de supports (fibre optique, onde radio, etc.) peuvent être utilisés pour mettre en place un réseau ATM. Dans tous les cas, un réseau ATM est composé de commutateurs (ou brasseurs) reliés entre eux.

Un commutateur (ou switch ATM) est un équipement qui commute les cellules en tenant compte de leur VCI et VPI.
Un brasseur (qui un commutateur ultra rapide) commute les cellules en tenant compte de leur VPI seulement. Les brasseurs se trouvent, en général, au centre du réseau pour accélérer la commutation.

Propriétés des commutateurs/brasseurs ATM

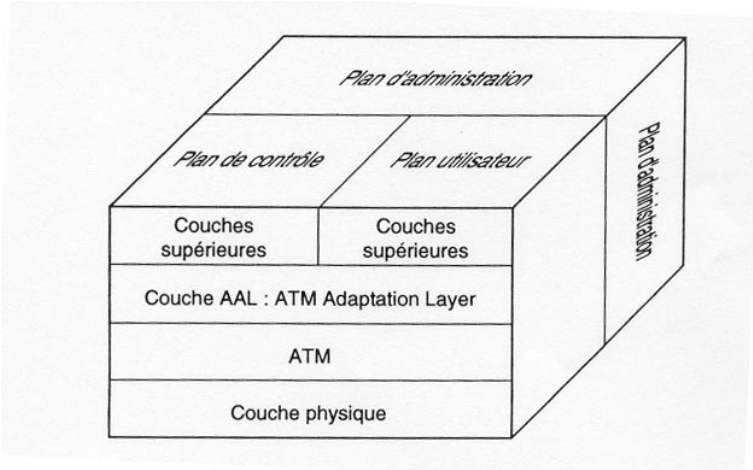
- très haut débit,
- très faible délai de commutation,
- très faible taux de perte de cellules,
- possibilité de diffusion,
- faible coût d'implantation.



Architecture générale de réseau ATM.

II.5. Architecture générale du réseau ATM

Les standards ATM couvrent essentiellement trois couches : couche AAL, couche ATM et couche physique. Le modèle du réseau ATM ne suit pas nécessairement les règles de structuration en couches du modèle OSI. En effet, on peut mettre directement la couche application au dessus de la couche AAL.



Architecture générale de ATM.

III. Couche physique

III.1. Principe général

La couche physique du réseau ATM n'est pas spécifique à un support ou une topologie particulière.

La couche physique est structurée en deux sous-couches : PM (Physical Medium) et TC (Transmission Convergence).

La sous-couche PM est liée au support physique et elle assure les fonctions suivantes :

- codage de bits,
- alignement de signaux,
- synchronisation,
- transformation électro-optique.

La sous-couche TC n'est pas directement dépendante du support physique et elle assure les fonctions suivantes :

- génération de trames et leur reconnaissance,
- adaptation des trames au support,
- délimitation (ou cadrage) des cellules,
- génération du HEC avec un polynôme $g(x) = x^8 + x^2 + x + 1$,
- détection des erreurs sur l'entête,
- insertion et suppression de cellules vides (pour adapter le rythme d'envoi de cellules au débit du support physique).

Plusieurs standards physiques existent, notamment les standards SDH (Synchronous Digital Hierarchy) utilisé en Europe et le standard SONET (Synchronous Optical Network) utilisé aux USA. Dans les deux standards, les cellules sont placées dans des trames émises de manière périodique ; ces trames sont émises même s'il n'y a pas de données à transmettre pour garder un rythme d'émission fixe.

III.2. Interface SONET

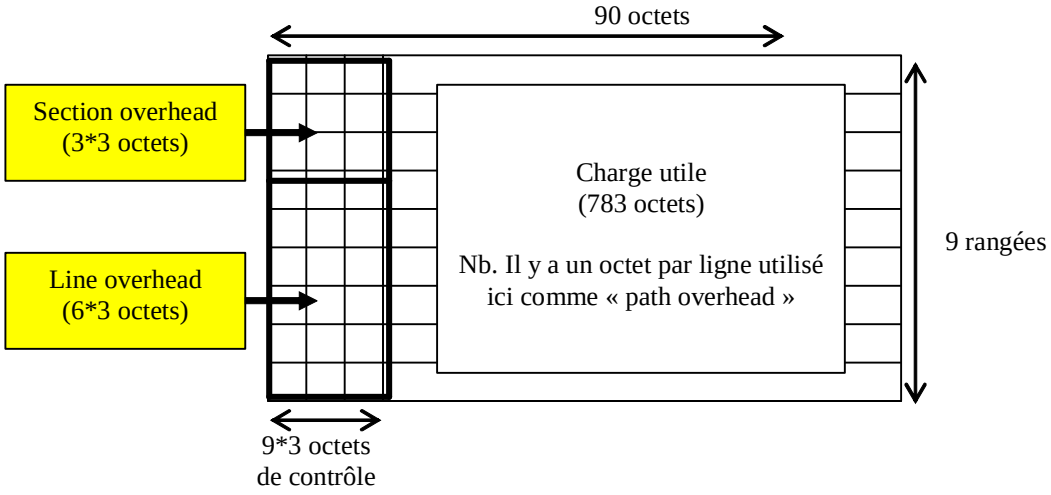
C'est une interface initialement proposée par Bell communication research.

Les trames de niveau physique sont émises à une fréquence de 8 kHZ (ce qui correspond à la fréquence d'échantillonnage de la voix qui rappelons-le était la principale cible des opérateurs de télécommunication).

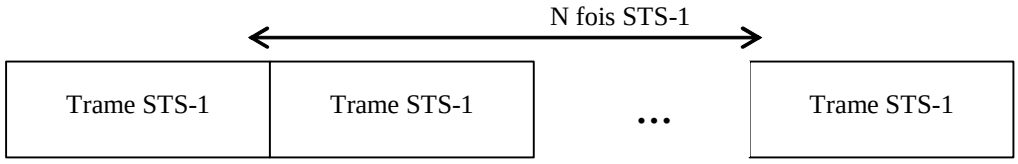
Les débits offerts sont des multiples de 51,48 Mb/s et sont identifiés par des codes OC-i (OC : Optical Carrier) :

OC-1 : 51,48 Mb/s, …… OC-192 : 9953,28 Mb/s.

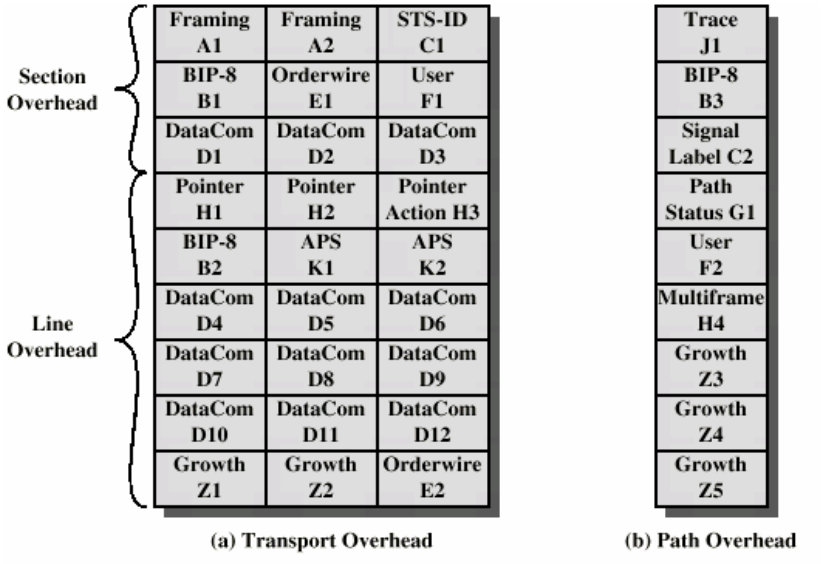
La trame SONET est structurée comme suit selon le format STS-1 (correspondant à 51,84 Mb/s qui est le débit minimal d'ATM) :



Format de la trame SONET STS-1
(on transmet dans l'ordre : 1ère rangée, 2ème rangée…, 9ème rangée)



Format de la trame SONET STS-N (il y a N trames STS-1 contiguës)

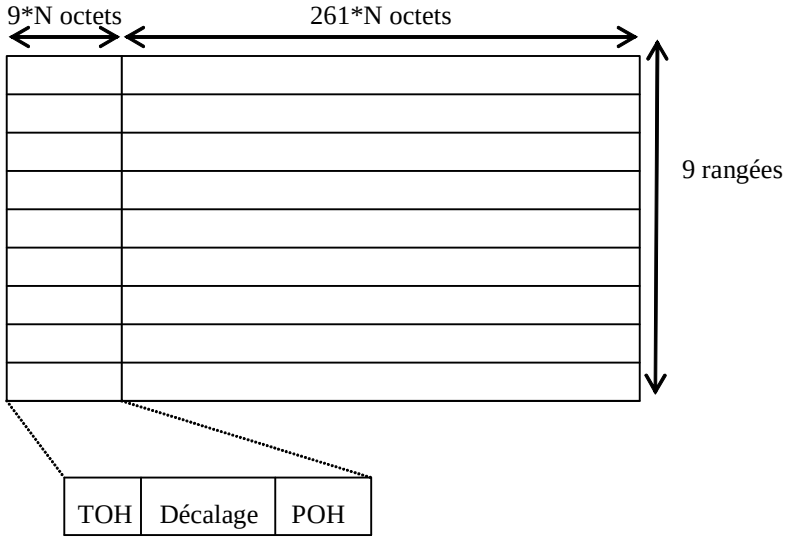


III.3. Interface SDH

Dans le standard SDH, les débits sont identifiés par des codes STM-i (STM : Synchronous Transport Module) :

STM-1 : 155,52 Mb/s = OC-3 de SONET
STM-64 : 9953,28 Mb/S = OC-192 de SONET

La trame SDH est structurée comme suit :



TOH (Transport Overhead) : informations sur les erreurs et autres
Décalage des données par rapport au début de la rangée,
POH (Path Overhead) : identificateur de chemin et autres

Format de la trame SDH (STM-N, N=1, 2, …, 64).

IV. Couche ATM

La couche ATM assure les fonctions suivantes :

- multiplexage et démultiplexage de cellules,
- commutation de cellules,
- génération et extraction de la partie entête de cellule,
- contrôle de flux,
- fourniture de la qualité de service demandée par l'utilisateur.

IV.1. Le routage ATM

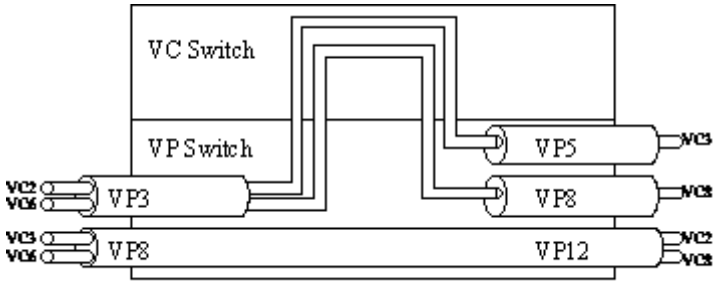
ATM est un réseau orienté connexion. Par conséquent, l'utilisateur doit établir une connexion avant de demander le transfert de données.

Il existe deux types de connexions que l'on peut demander à ATM :

- connexion de conduit virtuel : VPC (Virtual Path Connection)
- connexion de voie virtuelle : VCC (Virtual Channel Connection)

Les VPC sont permanentes ou semi permanentes. Les VCC sont généralement temporaires et établies selon les besoins de l'utilisateur pour une session de travail donnée.

Une VPC peut être considérée comme un agrégat de plusieurs VCC. Les VPC sont multiplexées sur les liaisons physiques.

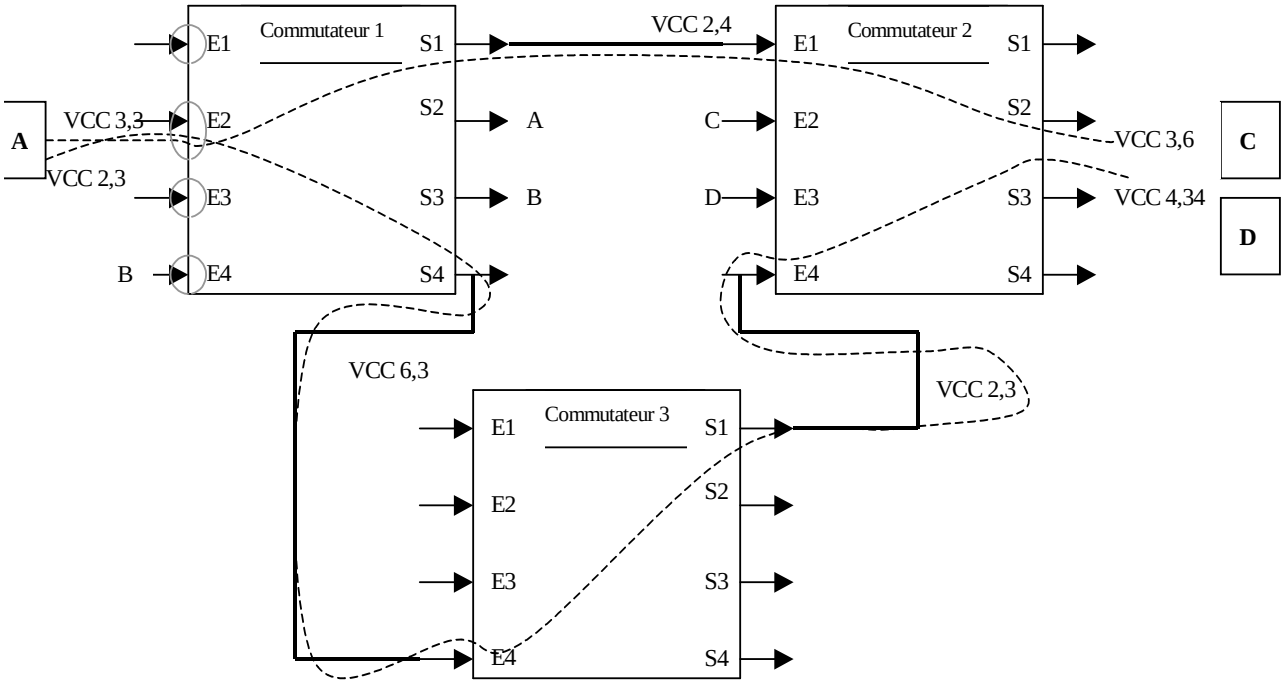


VP et VC dans un commutateur (switch) ATM.

Le routage dans les commutateurs peut être statique, dynamique ou spécifique. ATM ne spécifie pas un algorithme de routage particulier. Il existe une littérature abondante sur le routage dans ATM.

Pour faire le routage, chaque commutateur dispose d'une table de commutation. Cette table indique pour chaque voie d'entrée du commutateur quel couple VPI/VCI et quel lien de sortie faut-il appliquer à toute cellule entrante.

Exemple



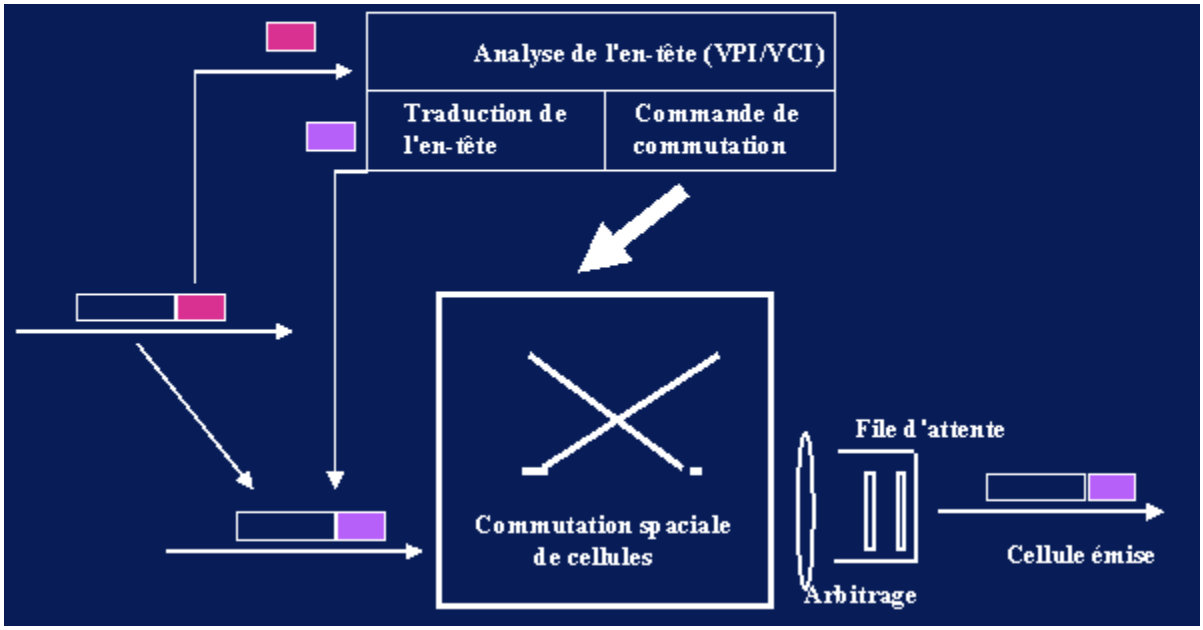
Exemples de circuits virtuels de bout en bout sur un réseau ATM à trois commutateurs. Le circuit de A vers C utilise successivement les couples VP//VCI *3,3* ; *2,4* ; *3,6* et le circuit de A vers D utilise les couples de VPI/CI *2,3 ; 6,3 ; 2,3* ; *4,34.*

En tenant compte de l'exemple de réseau précédent, la table du commutateur 1 est la suivante :

Voie d'entrée	VPI/VCI entrant	Voie de sortie	VPI/VCI sortant
E1			
E1			
E2			
E2	3,3	S1	2,4
E2	2,3	S4	6,3
E2			
E3			
E4			
…			

IV. Architectures des commutateurs ATM

Comme le montre la figure suivante la principale fonction d'un commutateur est de router les cellules sur les liaisons physiques.



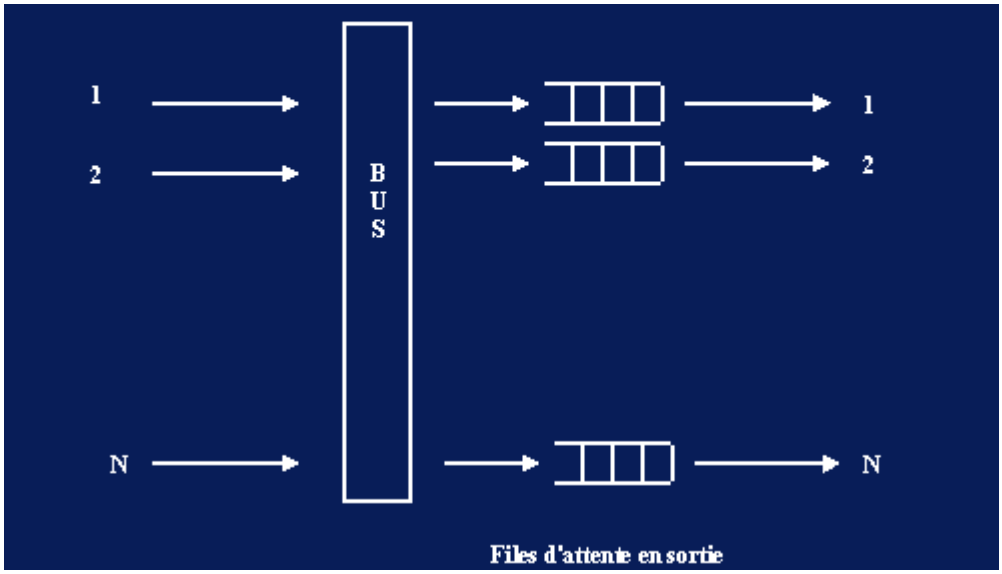
Principe général d'un commutateur ATM.

Il existe plusieurs architectures de commutateur dont les plus répandues sont :

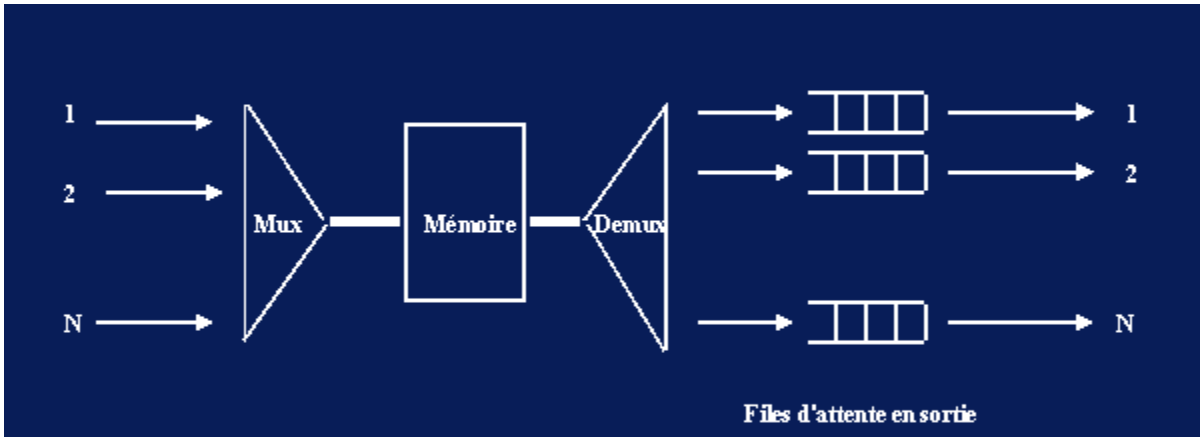
- **Les architectures de fond de panier :** elles utilisent le principe de raccordement des cartes d'entrées/sorties d'un processeur en utilisant le fond de panier de l'ordinateur. Chaque carte supporte un coupleur par voie d'entrée ou de sortie. Ces architectures sont adaptées à la diffusion de cellules.

- **Les architectures à mémoire partagée** : elles utilisent des mémoires partagées pour stocker les cellules dans des files d'attente. Ces architectures sont mal adaptées à la diffusion de cellules.

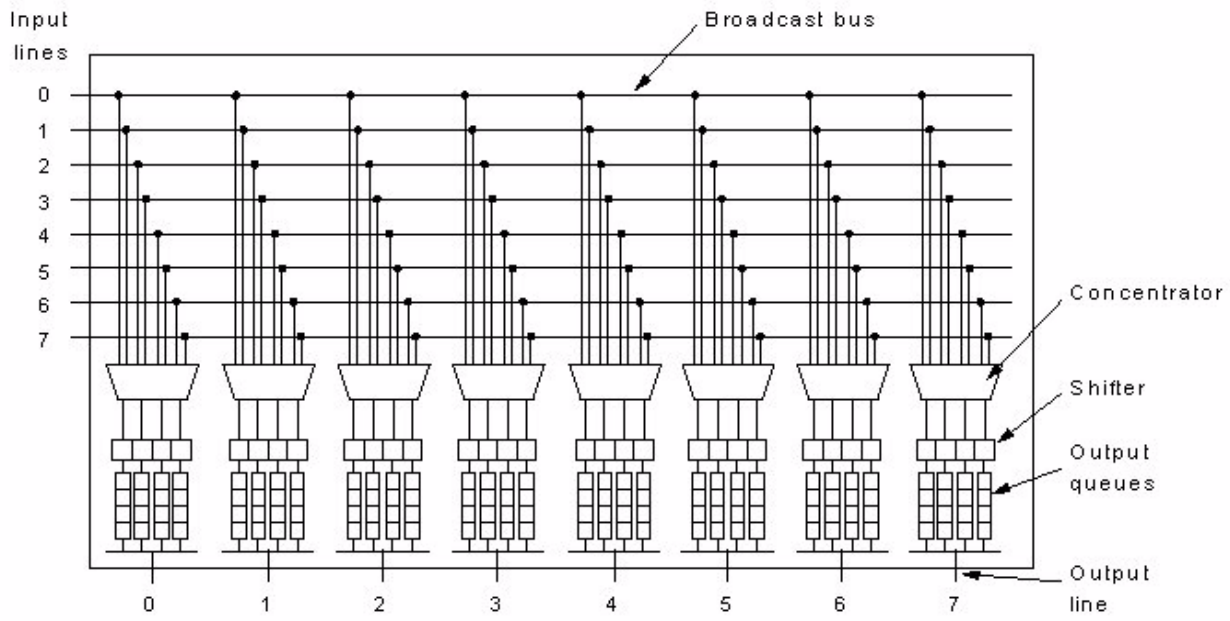
- **Les architectures à multiplexeurs** : elles sont essentiellement fondées sur une réalisation câblée des opérations de commutation. La voie de sortie est le plus souvent choisie, en cascade, selon les bits qui désignent les VPI/VCI. Chaque étage du multiplexeur peut disposer de capacité mémoire pour résoudre temporairement les conflits. Ceux sont les architectures les plus commercialisées. Les multiplexeurs les plus connus sont les *multiplexeurs crossbar* et les *multiplexeurs Banyan*.



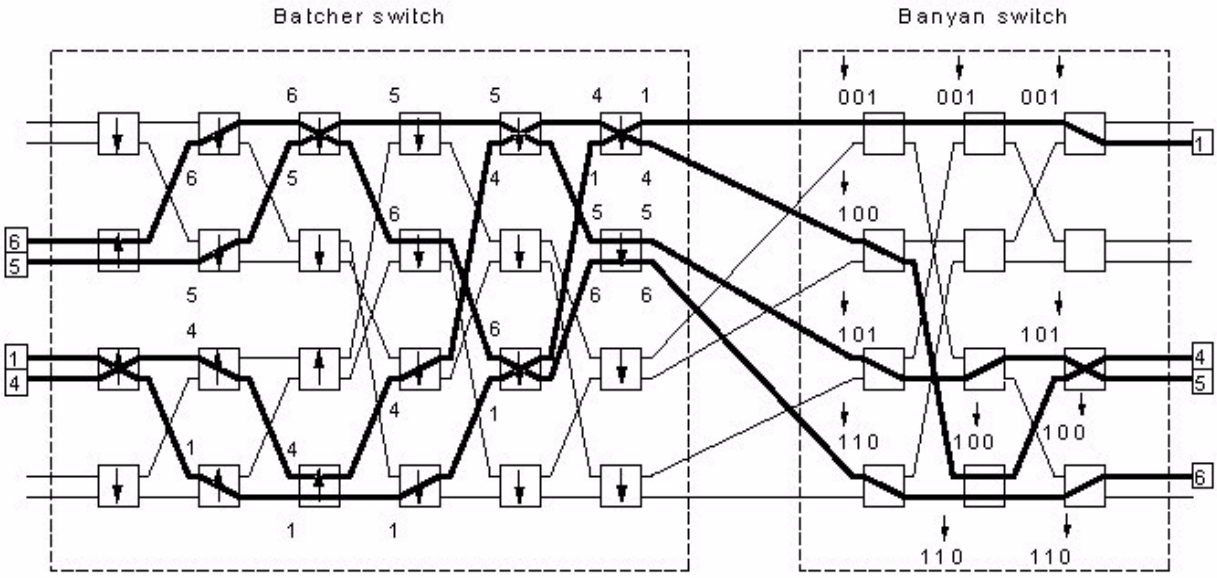
Principe de commutateur à fond de panier.



Principe de commutateur à mémoire partagée.



Principe de multiplexeur crossbar.



Principe de multiplexeur Banyan.

V. Couche AAL

La couche AAL a pour but de gérer l'interface entre ATM et les couches supérieures en tenant compte des besoins et exigences de ces couches. Certains considèrent que la couche AAL assure des fonctions similaires à celles de la couche transport ISO ou de TCP ou de UDP. Cependant, les principes de base de la couche AAL sont différents de ceux de la couche transport ISO ou de TCP/UDP, car ATM privilégie, à la base, le trafic avec débit périodique (téléphonie et vidéo qui sont des applications privilégiées des opérateurs de télécommunications) pour lesquelles le délai prime sur la fiabilité.

V.1. Types de trafics utilisateur et classes de services ATM

On distingue quatre types de trafic indépendamment du réseau ATM :

Type 1 : trafic isochrone à débit fixe

- Emission : débit constant et rythme fixe (périodique) ;
- Réception : même rythme que l'émission ;
- Pour la téléphonie.

Type 2 : trafic isochrone à débit variable

- Emission : débit variable, mais avec un rythme fixe (périodique) ;
- Réception : au même rythme que l'émission avec acceptation de gigue (la gigue c'est la variation du délai de communication) ;
- Pour la vidéo compressée.

Type 3 : Trafic asynchrone

- Emission avec débit variable ;
- Applications transactionnelles.

Type 4 : Trafic asynchrone avec rafales

- Emission : rafales d'informations pouvant être arbitrairement élevées ;
- Applications réseau local (transfert de fichiers, …).

Pour répondre à ces types de trafic quatre classes de services sont définies et ces quatre classes de services sont supportées par quatre AAL.

Classe de service	A	B	C	D
Contraintes de temps	OUI		NON	
Débit	Constant	Variable		
Mode connecté	OUI			NON

On ne parle plus de ces classes A, B, C et D depuis la version ATM 4.0 (mais beaucoup d'ouvrages antérieurs à la version 4.0 présentent ces classes). On parle des classes suivantes : CBR, rt-VBR, nrt-VBR, ABR, UBR et GFR. Les quatre classes sont donc devenues six.

Service	Caractéristiques	Applications types
CBR (Constant Bit Rate)	Débit constant, flux isochrone	Voix et vidéo non compressées
rt-VBR (real-time Variable Bit Rate)	Débit variable, flux isochrone	Voix et vidéo compressées
nrt-VBR (non real-time Variable Bit Rate)	Débit variable, mais prévisible	Transactions
ABR (Available Bit Rate)	Débit sporadique, sans contrainte temporelle	Interconnexion de réseaux locaux
UBR (Unspecified Bit Rate)	Trafic non spécifié (meilleur effort demandé)	Messagerie
GFR (Guaranteed Frame Rate)	Meilleur effort au niveau message et non au niveau cellule.	Transfert de fichiers et autres

Classes de services ATM

V.2. Classes de AAL et structure de la couche AAL

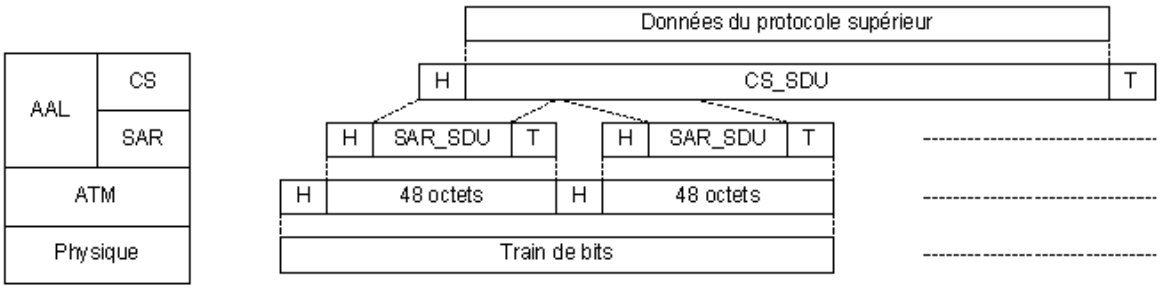
Les services AAL sont regroupés en 4 classes :

	Type de trafic 1	Type de trafic 2	Type de trafic 3	Type de trafic 4
AAL 1	OUI			
AAL 2		OUI		
AAL 3/4			OUI	OUI
AAL 5			OUI	OUI

La couche AAL est composée de deux sous-couches :

- CS (convergence sublayer) : les fonctionnalités dépendent du service rendu AAL1, 2, …, 5. CS traite des messages et non des cellules. Elle peut rajouter des en-têtes et en-queues selon ses besoins.

- SAR (segmentation and reassemly) : fragmenter les messages en cellules et les réassembler à la réception.



H : header (entête) T : tail (en-queue)

Principe d'encapsulation des données au niveau AAL.

V.3. AAL1

- Elle est destinée à la transmission d'audio ou de vidéo non compressée

- Elle permet de délivrer l'information à un rythme fixe.

- Elle gère les variations du délai de transmission (qui est variable de nature) pour limiter la gigue.

- L'entrée est un **flot de bits**, sans frontières de message (puisque c'est continu).

- Elle n'utilise pas de mécanisme de retransmission à cause des délais qu'il engendre. Elle détecte uniquement les cellules perdues ou mal insérées.

- Elle permet de mesurer les performances (taux de perte, débordement de buffer, taux d'erreur, …)

- Il n'y a pas de protocole de niveau CS.

- SAR a un en-tête composé des champs suivants :

+ Un bit C (généralement égal à 0) peut servir pour caler l'horloge du récepteur.

+ Un numéro de séquence (SN) qui permet de détecter les cellules manquantes ou mal insérées (c'est-à-dire qui ont été livrées à la station suite à une erreur de commutation).

+ Un champ de protection (SNP) obtenu avec $g(x) = x^3 + x + 1$ permettant de détecter les erreurs sur le champ SN.

+ Un bit de parité sur l'octet du SAR (pour réduire encore les erreurs non détectées).

+ Eventuellement un pointeur qui indique la position du message suivant.

Il reste 46 ou 47 octets de données utiles.

1	3	3	1 bits	
C	SN	SNP	parité	Informations utiles 47 octets

1	3	3	1	8 bits	
C	SN	SNP	parité	Pointeur	Informations utiles 46 octets

SN : Sequence Number SNP : Sequence Number Protection

Formats des cellules AAL1

V.4. AAL2

- Elle convient à l'audio et vidéo compressés (rafales et délais à garantir).

- Elle manipule des messages.

- Il n'y a pas de protocole de niveau CS.

- La sous couche SAR remplit des cellules avec 45 octets de données au maximum.

Les trames AAL2 ont la structure suivante :

+ Un champ SN (Sequence Number) : pour numéroter les cellules.

+ Un champ IT (Information Type) pour indiquer si la cellule transporte le début, le milieu ou la fin d'un message.

+ Un champ IL (Information Length) : indique le nombre d'octets de données utiles dans la cellule (ce nombre peut être inférieur à 45 pour la dernière cellule essentiellement)

+ Un champ CRC : pour le contrôle d'erreur sur toute la trame (avec $g(x) = x^{10} + x^9 + x^5 + x^4 + x + 1$).

Actuellement les deux champs IL et CRC tiennent sur deux octets, mais on ne connaît pas la taille exacte de chacun d'eux. AAL2 n'est pas encore tout à fait au point.

1 octet			2 octets	
SN	IT	**Informations utiles 45 octets**	IL	CRC

Format de cellule AAL2.

V.5. AAL3/4

Initialement il y avait un protocole orienté connexion AAL3 et un autre non orienté connexion AAL4. Ces deux protocoles ont été regroupés en un seul.

AAL3/4 peut fonctionner en deux modes : flots ou messages

AAL3/4 peut faire elle-même du multiplexage qui permet d'avoir plusieurs sessions (par exemple des terminaux) d'un hôte unique qui sont véhiculées par le circuit virtuel et qui ne sont séparées qu'à la destination. Cela permet aussi de réduire la facture pour les utilisateurs (en demandant un seul circuit virtuel au lieu de plusieurs).

Les applications peuvent soumettre à AAL3/4 des messages qui peuvent atteindre 65535 octets. Les messages sont complétés à des multiples de 4 octets puis on leur rajoute des en-têtes et des en-queues.

La CS et la SAR on chacune un protocole. La CS utilise le format suivant :

+ Le champ *CPI* (indicateur de partie commune) : donne le type de message et l'unité de compte pour les champs *Longueur* et *Taille Buffer*.

+ *Btag* et *Etag* permettent de délimiter les messages. Ils sont incrémentés de 1 à chaque émission de message. Ce mécanisme permet de détecter les pertes de cellules et les cellules mal insérées.

+ *Taille Buffer* : indique au récepteur la taille en octets qu'il doit réserver pour accueillir le message.

+ *Longueur* : donne la taille des informations utiles du message (ce champ doit être égal à *Taille buffer* en mode message, mais il peut être différent en mode flot).

1 oct	1	2	< 65 536 octets	0-3	1	1	2
CPI	Btag	Taille buffer	Charge utile	Bourrage	Non utilisé	Etag	Longueur

Format de message de CS d'AAL 3/4

La SAR insère chaque morceau de 44 octets du message de la CS dans la partie données utiles des cellules.

Le format des cellules de la SAR de AAL3/4 est le suivant :

+ Un champ ST : indique si la cellule commence un message, se trouve au milieu ou termine un message

+ Un champ SN : numéro qui sert à détecter les cellules manquantes ou mal insérées.

+ Un champ MID : utilisé en cas de multiplexage pour savoir à quelle session appartient une cellule.

+ Un champ IL : indique le nombre d'octets de données utiles dans la cellule.

+ Un champ CRC : pour le contrôle d'erreur sur toute la trame (même g(x) que pour AAL2).

2 bits	4 bits	10 bits	44 octets	6 bits	10 bits
ST	SN	MID	Charge utile	IL	CRC

Format des cellules de AAL 3/4

V.6. AAL5

- Elle est appelée aussi SEAL (Simple Efficient Adaptation Layer). Elle a été introduite pour remédier aux insuffisances de AAL 3/4.
- AAL 5 permet le choix entre un service fiable ou non fiable, entre le point-à-point et le multicast, entre le mode message et le mode flot.
- La communauté Internet considère que AAL 5 est la plus efficace pour supporter du trafic TCP/IP.
- AAL 5 rajoute des octets de bourrage (47 au maximum) pour avoir des messages dont la taille est un multiple de 48 octets.
- CS de AAL 5 n'a pas d'en-tête, mais elle a un en-queue composé de trois champs :
  + UU (user to user) : utilisé éventuellement par les couches supérieures.
  + Longueur : indique la longueur de l'information utile.
  + CRC sur 32 bits : porte sur tout le message y compris les données de bourrage, UU et la longueur.
  + Un octet réservé à des extensions futures.
- SAR de AAL 5 ne rajoute aucune information de contrôle. Elle récupère le message de CS et le place à raison de 48 octets par cellule.

< 65 536 octets	1	1	2	4
Charge utile	UU	Réservé	Longueur	CRC

Format de messages AAL 5

V.7. Récapitulatif des AAL

	AAL1	AAL2	AAL3/4	AAL5
Type de trafic	Périodique et fixe	Périodique avec rafales	Variable, sans contraintes de temps	Variable, sans contraintes de temps
Octets ajoutés par la CS par message	0	0	8	8
Octets ajoutés par SAR par cellule	1-2	3	4	0
Charge utile par cellule	46-47	45	44	48
Contrôle CS	Non	Non	Non	32 bits
Contrôle SAR	Non	Oui	10 bits	Non
Allocation anticipée de tampons	Non	Non	Oui	Non
Multiplexage	Non	Non	Oui	Non
Bourrage	Non	Non	≤ 3 octets	≤ 47 octets

Remarques

- Il existe trop de AAL (chaque corporation voulait son AAL) ;
- AAL 2, AAL 3, AAL 4 n'ont jamais vu le jour ;
- AAL3/4 est peu efficace aux yeux des utilisateurs ;
- Seules AAL1 et AAL 5 sont utilisés.

VI. Contrat de trafic

VI.1. Spécification de contrat de trafic

ATM est un réseau orienté connexion qui permet, le plus souvent, de garantir la qualité de service requise par les utilisateurs. Pour y parvenir, les exigences des utilisateurs doivent être exprimées sous forme de **contrat de trafic**.

Un contrat de trafic est défini par l'utilisateur et ATM devra le respecter le plus possible. Le contrat de trafic peut être fixé de manière permanente ou à chaque demande d'établissement de connexion.

Le plus souvent, la définition exacte d'un contrat de trafic est difficile et l'utilisateur se contente de donner une spécification approchée de son trafic.

Les paramètres qui définissent le trafic utilisateur sont :

- le débit moyen,
- le débit maximal (débit en rafale),
- le type de rafales et la durée de rafale,
- l'espacement minimum entre rafales,
- …

Une fois le trafic défini, l'utilisateur demande que son trafic soit considéré d'une manière spécifique à la nature de son application, on dit que l'utilisateur demande une certaine qualité de service (QoS : Quality of Service) au réseau.

Les paramètres qui définissent le trafic utilisateur sont :

- taux de perte de cellules,
- taux d'erreurs,
- délai maximal de transfert de cellule,
- délai moyen de transfert de cellule,
- gigue (variation du délai de transfert de cellule),
- …

Le réseau ATM peut accepter ou refuser une connexion selon sa charge instantanée. Par exemple, si le réseau est presque saturé, il ne va plus accepter de nouvelles connexions exigeantes en temps de transfert et en gigue.

L'interface d'accès à ATM définit un ensemble de paramètres pour spécifier le trafic et la qualité de service en fonction du type de service utilisé (CBR, rt-VBR, …, GFR).

	Service ATM					
Attribut	CBR	rt-VBR	nrt-VBR	UBR	ABR	GFR
Paramètres de trafic						
PCR and CDVT	Spécifié			Spécifié	Spécifié	Spécifié
SCR, MBS	Non spécifié	Spécifié		Non spécifié		
MCR	Non spécifié				Spécifié	Non spécifié
MCR, MBS, MFS, CDVT	Non spécifié					Spécifié
Paramètres de QoS						
Peak-to-peak CDV	Spécifié		Non spécifié			
MaxCTD	Spécifié		Non spécifié			
CLR	Spécifié			Non spécifié		

CDVT : Cell Delay Variation Tolerance
maxCTD : Maximum Cell Transfer Delay
MCR : Minimum Cell Rate
PCR : Peak Cell Rate
SCR : Sustainable Cell Rate
CLR : Cell Loss Ratio
MBS : Maximum Burst Size
MFS : Maximum Frame Size
peak-to-peak CDV : Peak-to-peak Cell Delay Variation

VI.2. Contrôle de trafic utilisateur, contrôle de flux, contrôle de congestion

Un utilisateur peut, de manière intentionnelle ou non, soumettre des cellules avec un rythme plus élevé que celui qu'il a demandé lors de l'établissement de connexion. Dans ce cas, l'utilisateur viole le contrat de trafic et peut conduire le réseau à ne plus respecter la qualité de service acceptée pour les autres utilisateurs (qui eux respectent leur contrat de trafic). Pour éviter de telles situations, ATM met en place un mécanisme de contrôle du trafic soumis par l'utilisateur. Un des algorithmes les plus utilisés pour contrôler le trafic de l'utilisateur est celui du seau percé ("leacky bucket") que nous verrons en TD.

Pour éviter des saturations intempestives, ATM met en place des mécanismes de contrôle de flux :

- contrôle de la charge actuelle du réseau avant d'accepter une nouvelle connexion,
- contrôle des cellules soumises par chaque utilisateur,
- évitement des situations de saturation,
- les cellules avec le bit CLP=1, sont éliminées en cas de saturation.

Ainsi, le réseau ne fait pas confiance aux usagers et contrôle le trafic par :

- des méthodes réactives (rejet de cellules non conformes),
- des méthodes préventives (utilisation de contrôle de flux par des mécanismes comme celui de la fenêtre coulissantes, refus des connexions avant même d'avoir atteint un seuil de saturation, etc.),
- méthodes hybrides.

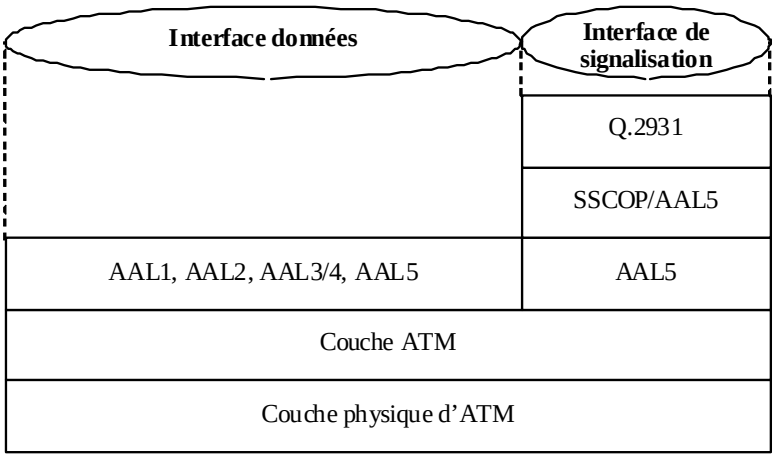
VII. Signalisation dans ATM

Il existe deux interfaces distinctes dans ATM :

- une interface pour le transfert de données,
- une interface pour la gestion des connexions (c'est l'interface de signalisation).

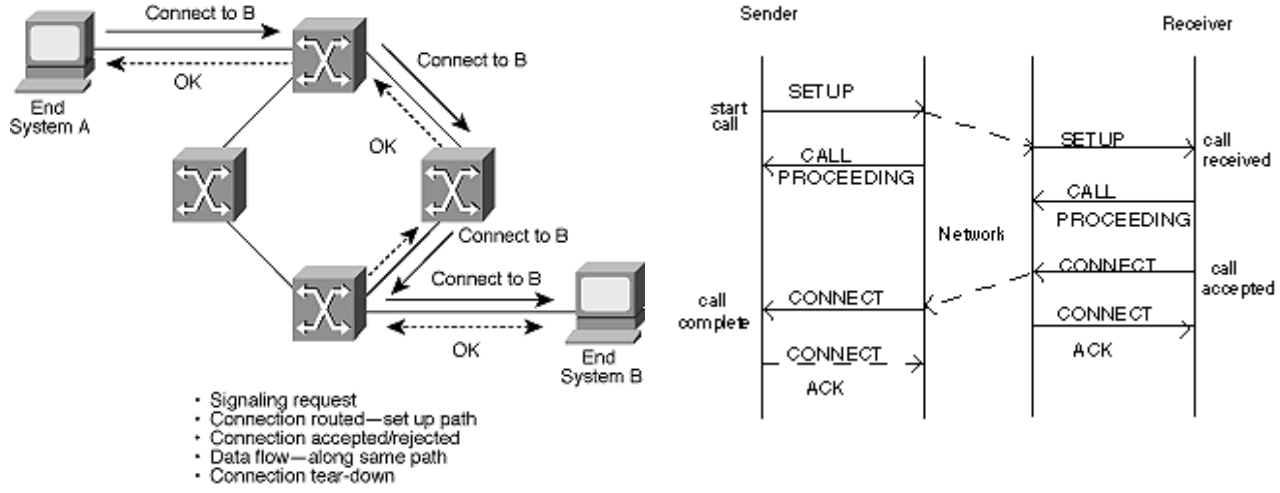
L'interface de signalisation offre les services suivants :

- demande d'établissement de connexion point à point ou multipoint,
- gestion des paramètres de qualité de service,
- affectation des valeurs de VPI/VCI aux connexions établies,
- réinitialisation de connexion,
- fermeture de connexion,
- mesure de paramètres d'une connexion (estimation du délai de transfert, de la gigue, etc.)



Interfaces d'accès au réseau ATM.

La connexion entre deux utilisateurs A et B s'effectue en utilisant les primitives *SET-UP* et *CONNECT* comme le montre la figure suivante :



Exemple d'établissement de connexion ATM.

VIII. ATM et les autres réseaux

ATM constitue une offre de réseau complète. On peut implanter directement des applications au dessus de ATM sans faire appel à d'autres protocoles. Au moment où ATM commençait à se développer on croyait qu'ATM allait s'imposer comme principale technologie de réseaux RNIS. C'était sans compter avec TCP/IP. En effet, la plupart des applications existantes ont été développées dans le cadre des protocoles TCP/IP et par conséquent, ATM ne pouvait ignorer TCP/IP. Mais le protocole IP a une approche complètement différente de ATM, il est non orienté connexion, alors que ATM est justement fondé sur le principe des connexions. Comment ATM peut-il donc coexister avec les autres standards ?

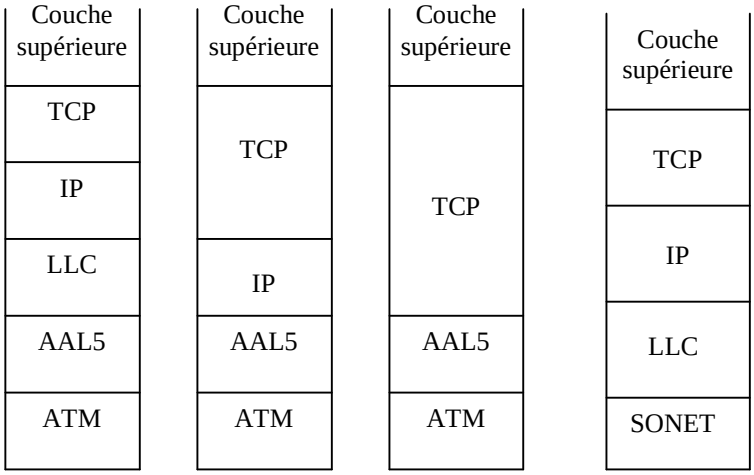
VIII.1. ATM et le monde OSI

En analysant de près les services offerts par ATM (couche AAL comprise), ATM peut couvrir les couches 1,2 et 3 du modèle OSI. Il y a certains qui considèrent que ATM est plutôt un réseau qui offre un service de niveau liaison de données, puisque ATM ne peut commuter que des cellules en provenance d'une même technologie, ATM, alors que la couche réseau OSI considère le routage à l'intérieur d'un ou de plusieurs réseaux, éventuellement hétérogènes, interconnectés.

VIII.2. ATM et le monde TCP/IP

Les utilisateurs des applications fonctionnant au-dessus de TCP/IP ne souhaitent pas en général que des modifications profondes soient apportées à l'interface qu'ils utilisent pour accéder aux services de communication. Cette interface d'accès à TCP est bien éprouvée. Par ailleurs, TCP et UDP sont fortement influencés par les choix de conception de IP. IP quant à lui peut fonctionner sur n'importe quel réseau physique. Même si en théorie plusieurs combinaisons entre ATM et les applications fonctionnant au-dessus de TCP/IP sont envisageables, dans la pratique ATM peut cohabiter avec IP (en étant un partenaire et non un rival, comme il l'a été pendant des années) en tant que réseau physique sur lequel vient se mettre le protocole IP.

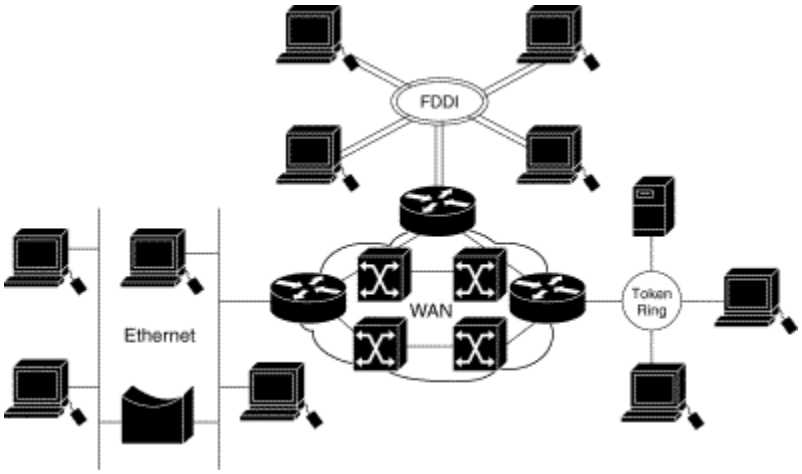
Comme IP ne garantit aucun transfert fiable (et encore moins des temps de transfert bornés), le protocole AAL le plus approprié pour supporter ATM est AAL5. Peut-être que dans le futur, avec l'émergence de nouveaux protocoles IP, l'utilisation des autres protocoles AAL serait envisagée.
On utilise souvent le terme "IP over ATM" pour parler de la combinaison IP et ATM.



Principales formes d'utilisation d'ATM avec TCP/IP.

VIII.3. ATM comme réseau d'interconnexion de réseaux locaux

Grâce à son débit très élevé, ATM peut servir de réseau fédérateur (ou d'interconnexion) d'autres réseaux (par exemple, des réseaux locaux dans une entreprise). En effet, le trafic entre deux réseaux est parfois important et nécessite des équipements d'interconnexion très rapides ; ATM peut jouer ce rôle.



Exemple d'interconnexion de réseaux locaux via un réseau ATM.

Exercices

Exercice 1
Calculer le débit maximal en cellules et en bits pour la communication d'un film couleur en utilisant AAL1 ou AAL5. Le flux correspondant au film est constitué de 25 images par seconde ; chaque image est composée d'environ 1024×768 pixels avec 16 bits par pixel.

Exercice 2
Etude de l'algorithme *Leacky bucket* (seau percé) utilisé pour le contrôle de trafic dans ATM.
L'algorithme dit *Leacky bucket* est utilisé pour le contrôle de trafic dans ATM. Son principe est le suivant :
Les cellules sont déposées dans un tampon à `M` places au maximum (`M` correspond au paramètre MBS – Maximum Burst Size). Les cellules sont vidées du buffer avec un débit constant de `S` cellules/s (correspond au paramètre SCR – Sustainable Cell Rate). Dans le cas où certaines avalanches contiennent trop de cellules pour tenir dans le tampon, il y a rejet de certaines cellules (on parle dans ce cas de débordement du seau).

```
TTA = ta(1)+I /* TTA initialisé à l'instant de l'arrivée de la première cellule */
```

Répéter indéfiniment

```
Arrivée de la cellule k (k>1)
Si TTA < ta(k)
      alors TTA = ta(k) + I
Sinon       Si TTA > ta(k) + L
                  Alors cellule non conforme
            Sinon TTA = TTA + I
            Finsi
                                    Finsi
```

FinRépéter

TTA : Temps théorique d'arrivée (à l'arrivée de la première cellule TTA = $t_a(1) + I$)
$t_a(k)$: temps d'arrivée de la $k^{ème}$ cellule
I : Incrément L : Limite

Q1.
On se place dans le cas où la source transmet deux cellules par milliseconde.
Dérouler l'algorithme avec I = 1 ms, L = 3 ms, ensuite avec I = 0 .5 ms et L = 3 ms.
Quelles sont les valeurs optimales pour I et L pour le trafic considéré ?

Q2.
Quelles sont les valeurs de I et L qui permettent de contrôler un trafic utilisateur défini de la manière suivante :
- Le délai minimum entre deux messages est de 4 ms.
- Un message est constitué de 6 cellules au maximum.

Utiliser un exemple de trafic qui respecte les conditions précédentes et un autre qui ne les respecte pas pour tester l'algorithme du seau percé avec les paramètres I et L calculés.

Exercice 3

Quelles sont les valeurs des paramètres du contrat de trafic (PCR, SCR, MBS,…) qui permettent à l'utilisateur de demander une connexion pour envoyer des données selon le modèle de trafic et contraintes de QoS suivants :

- Le délai minimum entre deux messages est de 4 ms.
- En moyenne un message arrive toutes les 100 ms. Lorsqu'un bloc de 4 messages successifs se présente (deux messages sont toujours séparés par au moins 4 ms), le prochain bloc de 4 messages successifs ne peut pas se présenter avant 50 ms.
- Un message est constitué de 6 cellules au maximum.
- L'utilisateur accèpte la perte d'une cellule sur 20.
- Le délai de bout en bout du transfert doit varier entre 250 ms et 300 ms.

Exercice 4

Une entreprise souhaite interconnecter tous ses équipements informatiques via un réseau local. Elle dispose d'un ensemble d'applications qui tournent au-dessus de TCP/IP et elle dispose déjà de la pile de protocoles TCP/IP. Elle s'adresse à vous pour l'éclairer sur le travail à réaliser. Elle vous pose les questions suivantes :

Q1. Si on abandonne TCP/IP, est-ce qu'il y a beaucoup de choses à modifier dans nos applications ?
Q2. En gardant TCP/IP, est-ce qu'il est plus intéressant de choisir un réseau ATM ou FDDI pour réaliser l'interconnexion en local ?

Q3. Si on choisit le réseau ATM au dessous de IP, en se limitant aux couche ATM et physique, que faut-il rajouter comme principales fonctions pour interfacer IP avec ATM ?

Q4. Si en plus de la couche ATM, on veut bénéficier des services d'une AAL, sur quoi se base-t-on pour choisir une AAL ?

Q5. Si on choisit une AAL (par exemple AAL5) quelles principales fonctions faut-il rajouter pour interfacer IP avec ATM ?

Q6. A quel niveau l'algorithme du seau percé est-il utilisé dans un réseau ATM ? Est-il fourni par le constructeur du réseau ATM ou bien c'est l'utilisateur qui le développe ?

Q7. Parmi les fonctions de gestion de réseau, quelles sont les fonctions qui doivent être fournies obligatoirement par le constructeur de réseaux ATM ? Et les autres fonctions (s'il en reste) dans quels cas l'utilisateur du réseau ATM doit-il les avoir ?

Il faut expliquer vos réponses pour toutes les questions.

Exercice 5

On considère trois stations A, B et C reliées par un réseau R. Les stations ont le même comportement : une station transmet un message de 100 octets par seconde de manière régulière et en plus, elle peut transmettre des messages apériodiques espacés d'au moins 5 secondes. Un message apériodique peut avoir une taille variant de 1 à 100 octets.

a) On suppose que les stations A, B et C sont reliées par un réseau FDDI. Donner les valeurs optimales des paramètres de configuration de FDDI pour que les messages soient émis en respectant les contraintes de temps. Donner un exemple de trafic pour illustrer le fonctionnement de passage du jeton sur FDDI. Indiquer la valeur maximale de rotation du jeton.
b) On suppose que les stations A, B et C sont reliées par un réseau ATM. Déterminer les valeurs optimales des paramètres I et L de l'algorithme du seau percé pour écouler correctement le trafic généré par les trois stations. Justifier votre réponse. Donner un exemple de trafic pour illustrer le fonctionnement de l'algorithme.

Exercice 6

a) Peut-on implanter IP au-dessus de ATM ? Si oui : dans quels cas une telle implantation est utile et quels sont les principaux problèmes à résoudre pour y parvenir ?

b) Peut-on implanter ATM au-dessus de IP ? Si oui : dans quels cas une telle implantation est utile et quels sont les principaux problèmes à résoudre pour y parvenir ?

Exercice 7

On considère une application constituée d'un client et d'un serveur. Le client effectue, en moyenne, cinq transactions par seconde. Mais il n'a pas le droit d'avoir plus de quatre transactions en attente d'émission. Chaque transaction nécessite un envoi de 96 octets par le client. Calculez (en justifiant votre calcul) les paramètres nécessaires au contrôle de trafic du client en appliquant l'algorithme du seau percé au réseau ATM. Donnez un exemple de trafic qui respecte le contrat de trafic spécifié à l'aide des paramètres de contrôle calculés et un autre trafic qui ne respecte pas le contrat.

Exercice 8

On considère un ensemble de 4 terminaux d'acquisition de données T_i *(i=1, 2, 3, 4)* reliés par un réseau qui peut être de type FDDI ou ATM. Chaque terminal T_i envoie des messages de longueur fixe L_i. Chaque terminal T_i génère 1 à 3 messages par période de temps égale à P_i. Si trois messages sont générés par un terminal T_i

pendant un intervalle de temps inférieur à P_i, alors le quatrième message n'est généré par ce terminal que $2P_i$ unités de temps après le troisième.

Q1 : Comment configurer un réseau FDDI pour pouvoir supporter le trafic généré par les quatre terminaux ?

Q2 : Quelles valeurs des paramètres *I* et *L* de l'algorithme du seau percé faut-il utiliser pour contrôler le trafic précédent ? Justifiez votre réponse.

Q3 : Quel service ATM convient-il le mieux pour véhiculer le trafic précédent ? Quelles sont les valeurs de paramètres de connexion ATM à choisir pour chaque connexion ?

Chapitre 5
Réseaux sans fil et mobilité

I. Introduction

I.1. Sans fil et mobilité

Depuis le début des années 90, les réseaux sans fil (wireless networks) prennent de plus en plus de place. Aujourd'hui l'essentiel du trafic acheminé par les réseaux sans fil correspond à de la voix (téléphone).
Les fabricants proposent de plus en plus d'appareils, de cartes, de logiciels pour la mise en place de réseaux sans fil. Les utilisateurs et les opérateurs investissent en masse (avec une certaine prudence). A titre d'exemple, on parle déjà de téléphone mobile de 4ème génération, alors que ceux de troisième génération (UMTS) sont à leur début et les investissements des opérateurs sont encore loin d'être amortis.

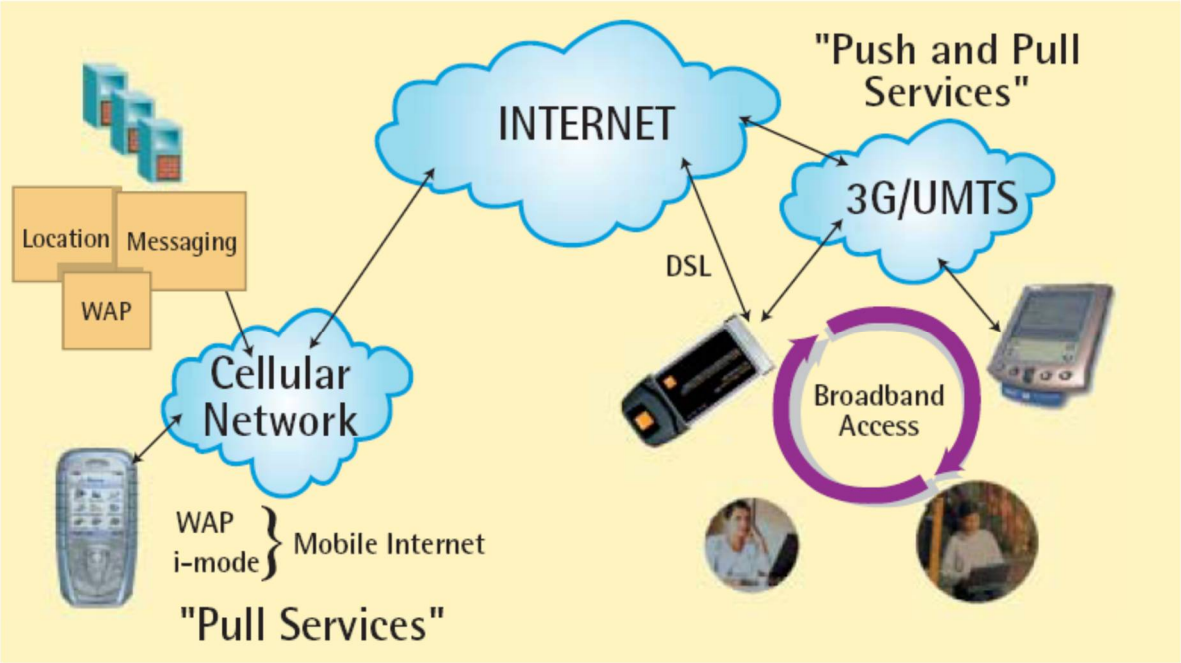
Rares sont les experts qui se hasardent à dire quelle sera exactement la technologie dominante pour les réseaux sans fil dans les prochaines décennies. Cependant, beaucoup s'accordent à dire que la notion de sans fil est un atout pour s'affranchir du cordon physique de raccordement au réseau.

Les différences fondamentales entre les réseaux filaires et les réseaux non filaires résident principalement dans les caractéristiques de transmission. Il faut noter en particulier :

- Spectre de fréquences limité et réglementé
- Qualité fluctuante à cause des mouvements, des interférences entre canaux voisins, des brouillages, la hauteur des bâtiments ou des montagnes, des tunnels, parkings, etc. En particulier, les réseaux de mobiles doivent faire face à des taux d'erreurs très élevés (10^{-3} en moyenne). Les applications doivent s'adapter à des délais de communication souvent variables.
- Le point d'accès de l'utilisateur au réseau est inconnu et variable dans le temps (nécessité d'une fonction dite de *handover* : maintien de la communication lors de changement de cellule ou zone).
- Sécurité plus difficile à garantir car les signaux transmis peuvent être captés par n'importe qui

La notion de **mobilité** désigne l'habitude d'un équipement à pouvoir continuer à communiquer en se déplaçant. Souvent, on confond *Réseaux sans fil* et *Réseaux de mobiles*. Il y a une nuance entre les deux.
Le tableau suivant résume les liens entre mobilité et sans fil.

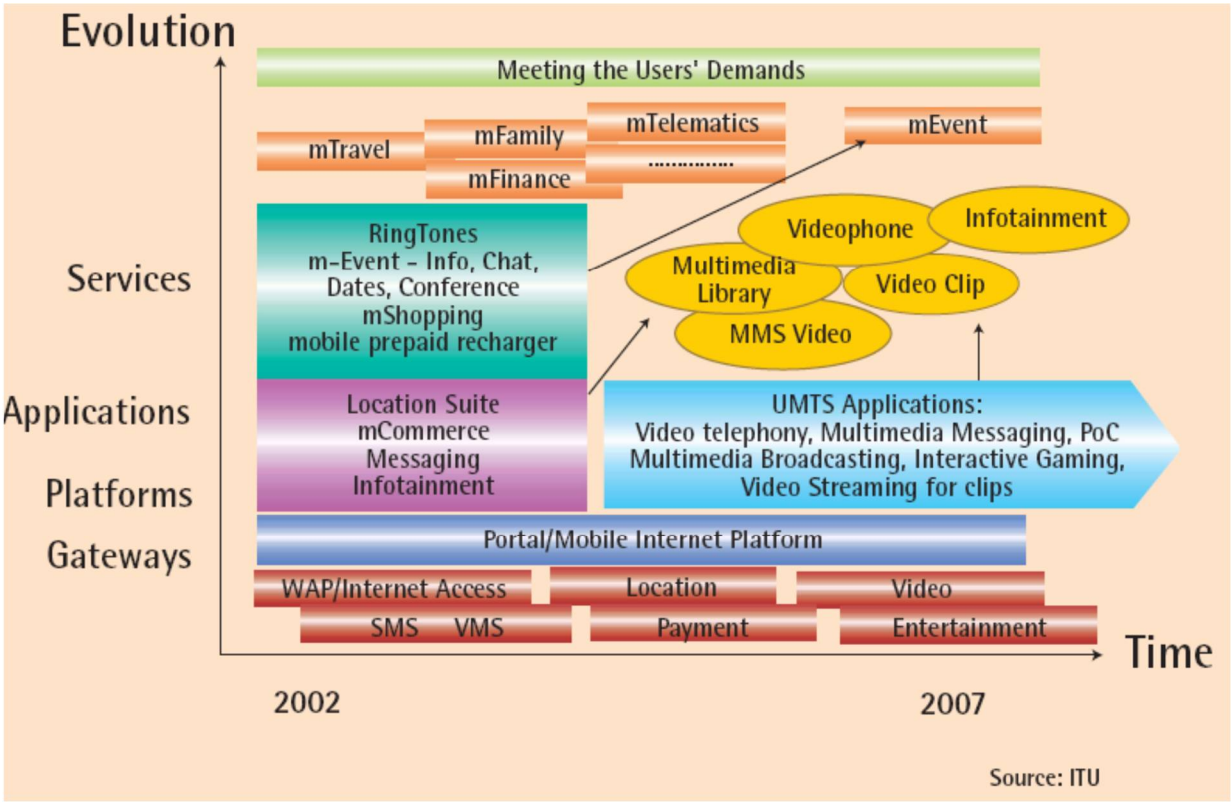
		Support de communication	
Mobilité		Avec fil	Sans fil
	Aucune mobilité	Oui	Incongru
	Mobilité du logiciel	On peut effectuer des migrations de programmes ou données sur un réseau filaire	On peut avoir du sans fil, mais sans mobilité du logiciel
	Mobilité du matériel	Impossible	Oui



Source UMTS-forum.org 2005
Vue simplifiée des moyens de communication pour mobiles

I.2. Les applications

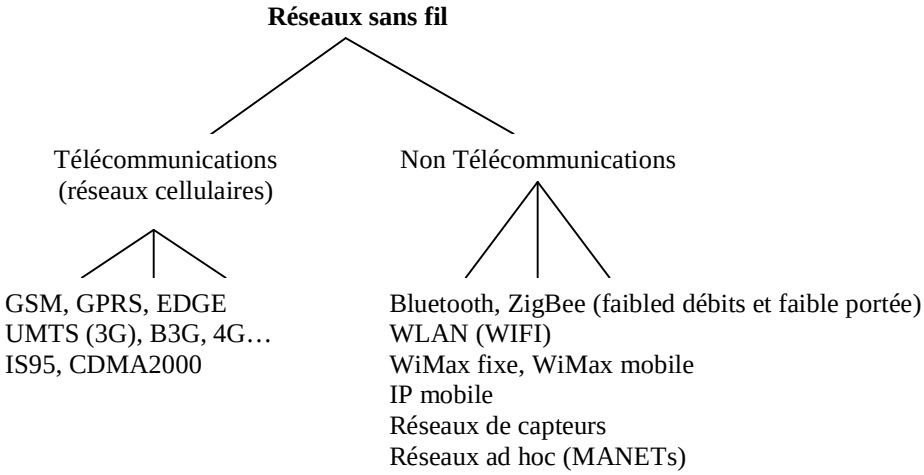
Comme le montre la figure ci-dessous les applications qui peuvent fonctionner au dessus des réseaux sont fil sont très nombreuses (il n'y a pas de limite à l'offre de services et applications !).



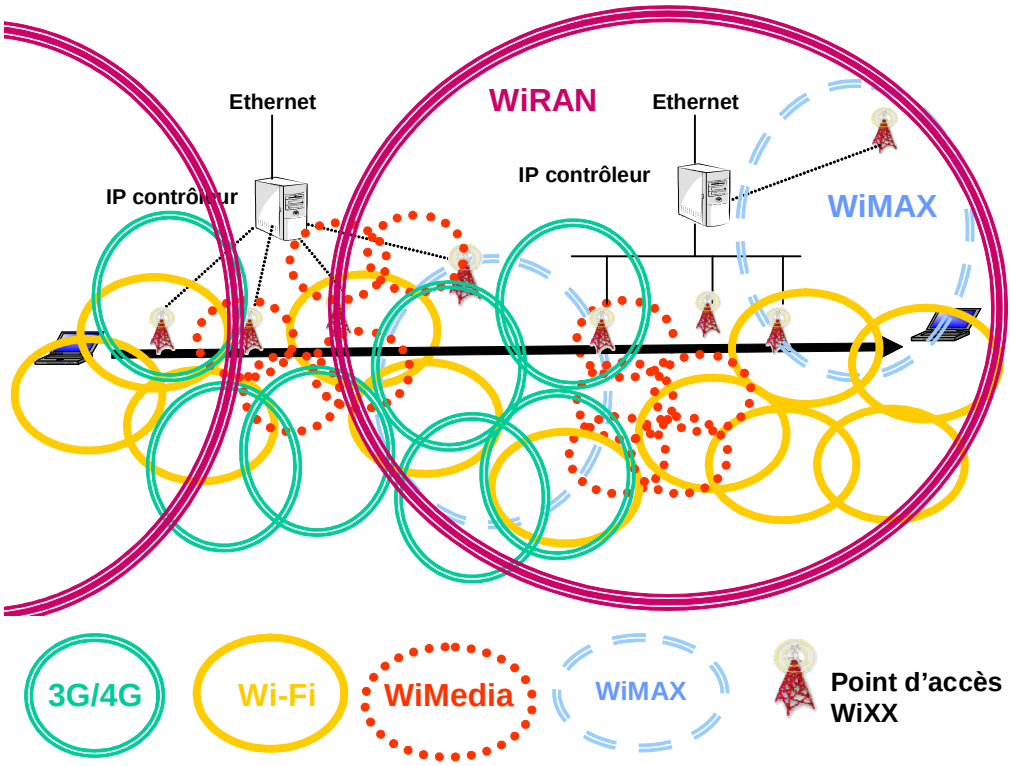
Source : ITU (2007)
Diversité des applications

I.3. Classes de réseaux sans fil

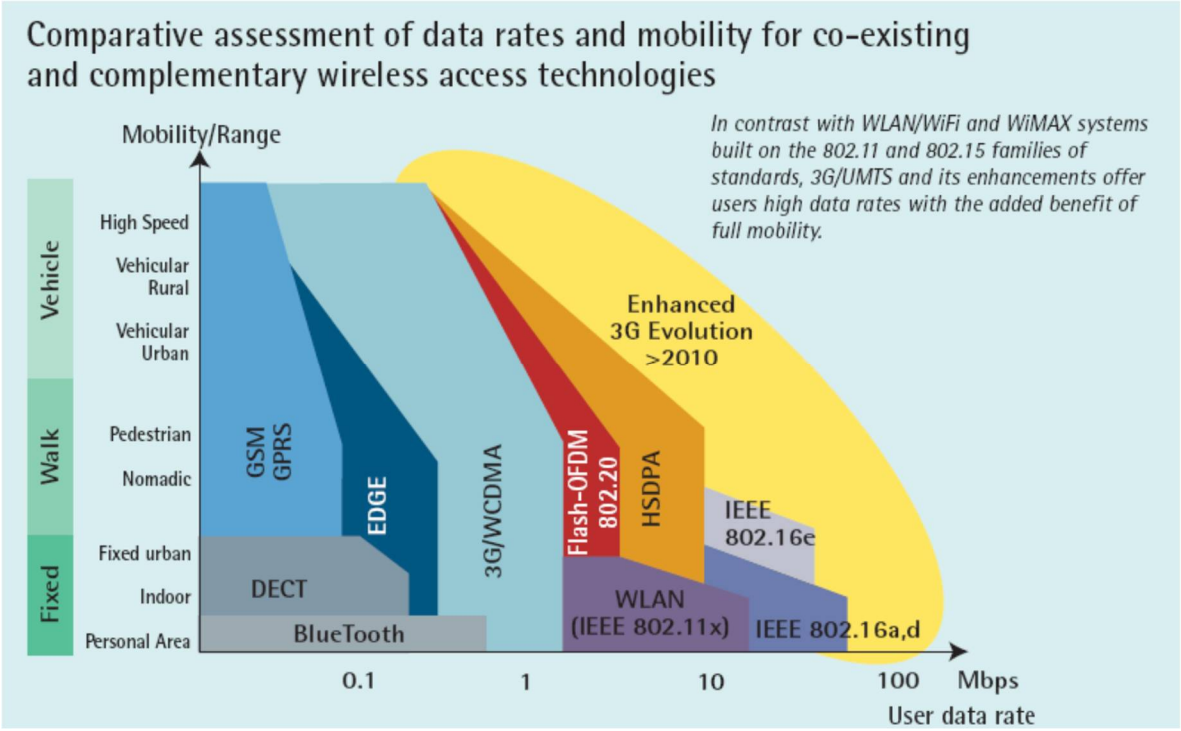
Les réseaux sans fil couvrent un très large spectre. Les communications entre deux stations (utilisateurs mobiles) peuvent passer par plusieurs réseaux sans fil interconnectés.



Catégories de réseaux sans fil



Diversité des réseaux sans fil – Interopérabilité entre réseaux



Source UMTS-forum.org (2005)

Diversité des réseaux de communication sans fil

I.4. Standards pour réseaux sans fil

Les produits que l'on trouve sur le marché sont essentiellement dominés par les standards de l'IEEE 802. Les opérateurs de Télécom utilisent d'autres standards pour offrir les mêmes services.

Standards pour PAN :

- IEEE 802.15.1 - Bluetooth
- IEEE 802.15.3 – UWB (Ultra Wide Band)
- IEEE 802.15.4 – ZigBee

Standards pour WLAN (IEEE 802.11 et Wi-Fi)

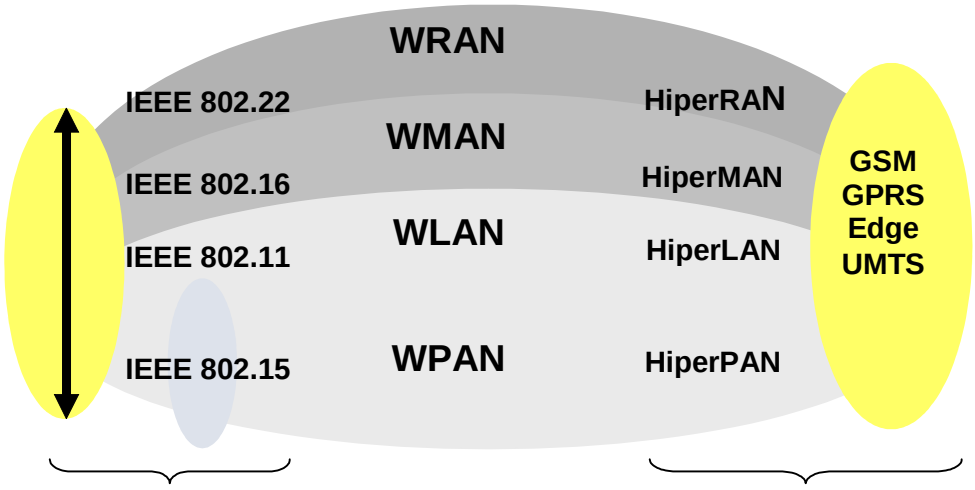
- IEEE 802.11b, a, g, e
- IEEE 802.11n
- IEEE 802.11s

Standards pour WMAN (IEEE 802.16 et WiMax)

- IEEE 802.16-2004
- IEEE 802.16e/IEEE 802.20 (Wi-Mobile)

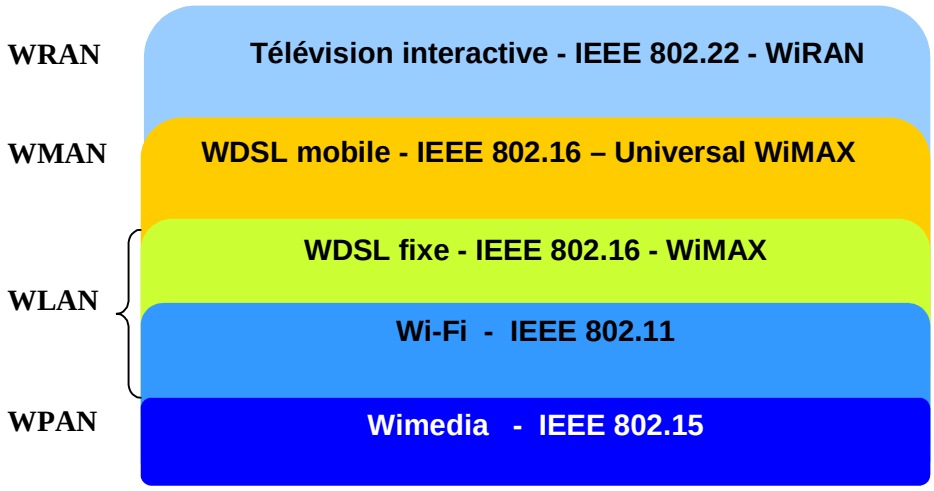
Standards pour WRAN

- IEEE 802.22 et WiRAN
- Utilisation des bandes TV 54-698



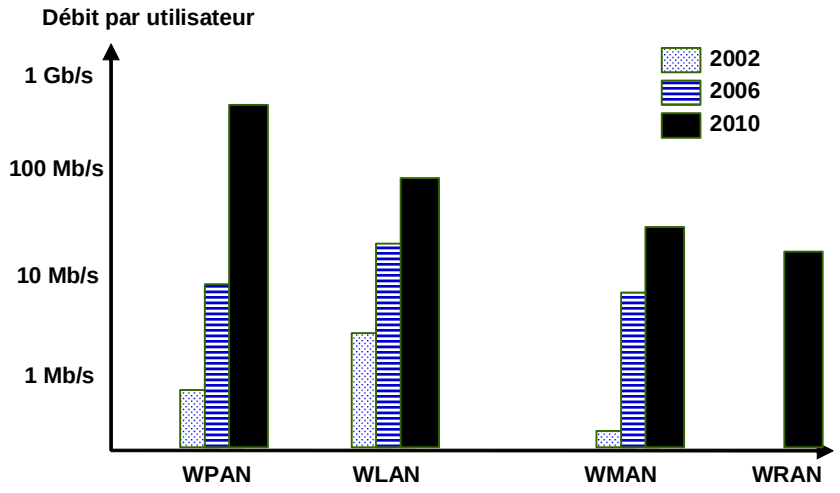
WRAN : Wireless Radio Access Network
WMAN : Wireless Metropolitan Area Network
WLAN : Wireless Local Area Network
WPAN : Wireless Personal Area Network

Principales normes pour réseaux sans fil



Tendances pour les réseaux d'accès sans fil

Les débits offerts par les réseaux sans fil sont en constante progression comme le montre le tableau ci-dessous.



Débits selon les classes de réseaux sans fil

	Bluetooth	**802.11b**	**802.1a/g**	**WiMedia**
Puissance de transmission (mW)	70	350	700	405
Débit effectif (Mb/s)	0.75	6	25	170
Temps de transfert d'un fichier de 5 G octets (en minutes)	889	111	26.7	3.9

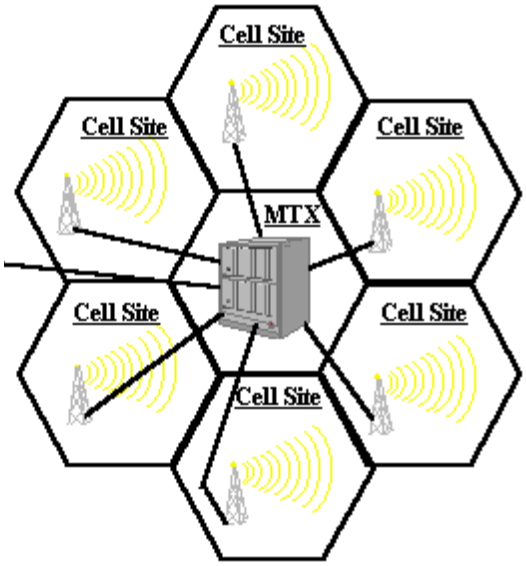
Quelques chiffres sur les performances des réseaux sans fil

Il difficile de présenter tous les réseaux sans fil, même de manière succincte, dans le cadre de ce cours. Nous avons choisi de présenter les concepts de base de systèmes cellulaires et du réseau sans fil le plus répandu, Wifi.

II. Réseaux cellulaires

II.1. Concept de cellule

Le concept cellulaire doit son origine au problème suivant : comment desservir une région de taille importante (un pays voire un continent) avec une largeur de bande limitée et avec une densité d'usagers importante ou qui peut changer au cours du temps ?
En mettant en œuvre le mécanisme de réutilisation de fréquences, le concept cellulaire permet de résoudre ce problème. En effet, la réutilisation des fréquences permet de couvrir des densités d'usagers et des zones de couvertures illimitées.
Le mécanisme cellulaire repose sur la propriété d'atténuation des ondes radioélectriques qui fait qu'une fréquence utilisée dans une zone peut être réutilisée dans une autre zone si celle-ci est suffisamment éloignée.
Chaque zone constitue une cellule dont la taille varie en fonction de la densité d'usagers à desservir.
La bande de fréquences est subdivisée en sous-bandes et chaque sous-bande est allouée à une station de base.



MTX (mobile telephone exchange)

Exemple de réseau à 7 cellules.

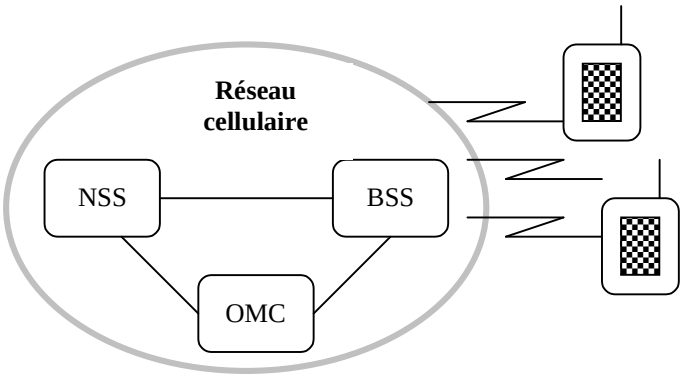
II.2. Architecture de système cellulaire

L'architecture générale d'un système cellulaire (par exemple le GSM : Global System for Mobile communications) repose sur deux sous-systèmes :

- un sous-système réseau (ou NSS : Network Sub-System) constitué de commutateurs permettant des gérer les appels (établissements, acheminement, facturation, utilisation de base de données sur les abonnés, etc.)

- un sous-système radio (ou BSS : Base Station Subsystem) contenant les équipements nécessaires à la gestion de l'interface radio.

Les NSS et BSS sont contrôlés et supervisés par un système d'exploitation et de maintenance (OMC : Operating and Maintenance Center).



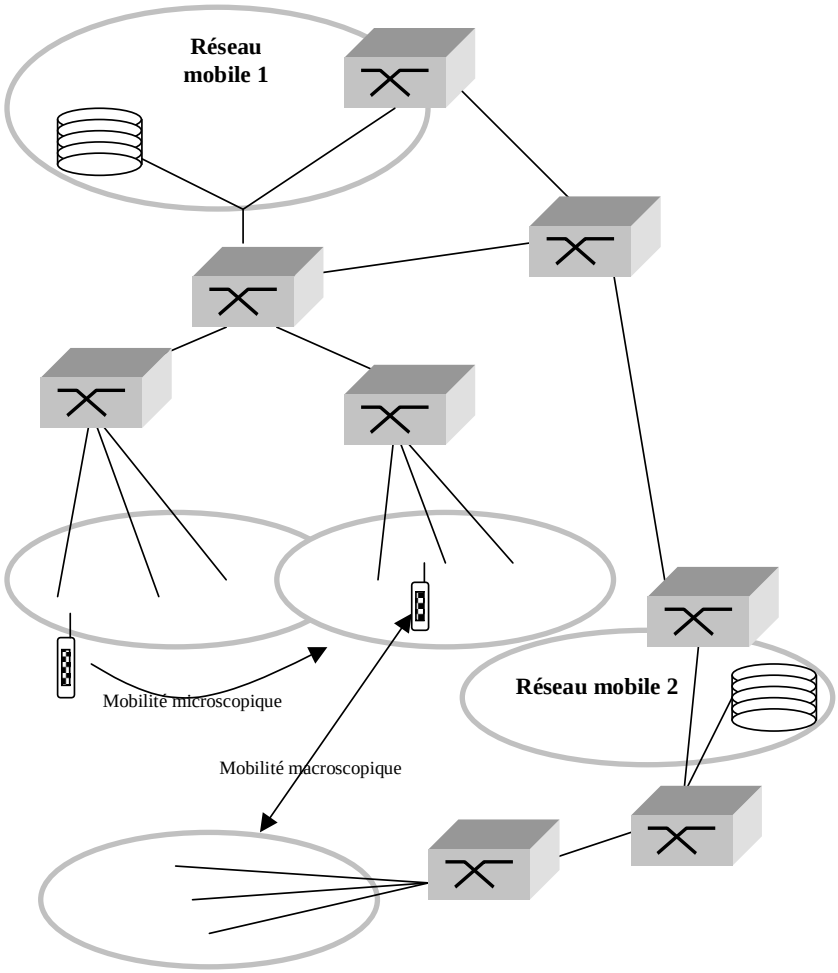
Exemple d'architecture de réseau cellulaire.

II.3. Gestion de la mobilité

Il existe deux niveaux de gestion de la mobilité : microscopique et macroscopique.

a) Niveau “microscopique” : c’est celui de la gestion de la mobilité radio qui permet à un abonné de changer de cellule tout en maintenant sa communication. Cette gestion fait appel au mécanisme de handover entre cellules dont le principe de base est le suivant : pendant la communication, le lien radio est mesuré et évalué périodiquement. La détection d’une situation anormale déclenche une alarme du contrôleur de station de base (BSS) vers le commutateur du service de mobile (BSC). A la réception d’une alarme, le BSC identifie la nouvelle cellule et déclenche le handover. Dans ce cas on parle de **handover horizontal**. Après un handover, l’ancien canal est libéré. A procédure décrite précédemment, il faut rajouter la gestion des états de veille et du passage de l’état inactif à l’état actif (qui permet à un mobile de se positionner sur une cellule pour recueillir les informations nécessaires à la connexion sur cette cellule). La localisation d’un abonné peut se faire selon plusieurs techniques : par recherche dans tout le réseau, mise à jour périodique de la localisation, mise à jour en cas de changement de zone, utilisation d’un réseau de signalisation déparé du réseau usager, etc. Une base de données distribuée ou centralisée permet de mémoriser la localisation des usagers. Des techniques issues de l’IA (par apprentissage) permettent d’accélérer la procédure de localisation.

b) Niveau “macroscopique” : c’est celui de la gestion de la mobilité du réseau qui permet à un abonné de bénéficier des services auxquels il a souscrit sous toute la couverture de son réseau nominal et éventuellement d’autres réseaux visités (par exemple, suivi d’appels, numéros abrégés, etc.). Dans ce cas on parle de **handover vertical**.



Mobilité dans les réseaux.

II.4 Méthodes d'accès

L'objectif est d'utiliser judicieusement la bande de fréquences allouées pour écouler un maximum de communications. Deux aspects complémentaires sont à distinguer : partage des fréquences par les techniques d'accès multiples et accès au réseau par les terminaux en utilisant des techniques d'accès aléatoire.

a) Techniques d'accès multiples

Les principales techniques d'accès multiples sont les suivantes :

- FDMA (Frequency Division Multiple Access) ou AMRF (Accès Multiple par Répartition dans les Fréquences). C'est la technique la plus ancienne. Elle consiste à utiliser un canal pour véhiculer un appel unique dans un sens donné.

- TDMA (Time-Division Multiple Access) ou AMRT (Accès Multiple par Répartition dans le Temps). Le temps est partagé en temps de base appelé *slots* qui sont alloués aux stations. L'allocation des *slots* aux stations peut être fixe ou non.

- CDMA (Code-Division Multiple Access) ou AMRC (Accès Multiple par Répartition par les Codes). C'est une technique initialement utilisée par les militaires pour des raisons tactiques. Plusieurs stations peuvent émettre simultanément dans la même bande de fréquence. A chaque station est associée une séquence aléatoire (un code) différente de celles des autres stations. Cette séquence est utilisée pour générer des sauts de fréquences. Les sauts de fréquences propres à chaque émetteur permettent de le

distinguer des autres même en cas de plusieurs transmissions simultanées un peu comme si dans une salle plusieurs couples de personnes se parlent simultanément deux à deux, mais dans des langues différentes ; chaque personne arrive à extraire du vacarme ambiant la voix de son interlocuteur.

Les trois techniques précédentes peuvent être combinées. L'efficacité des techniques d'accès multiple fait l'objet de nombreuses recherches et expérimentations.
Les techniques d'accès multiple sont surtout adaptées à la transmission de la voix.

b) Protocoles d'accès

Dans le cas de terminaux informatiques, les trafics sont souvent aléatoires, ce qui nécessite la mise en place de protocoles d'accès au réseau.

Les protocoles d'accès utilisés dans les réseaux de mobiles prennent, pour certains, leurs origines dans les méthodes d'accès utilisées dans les réseaux locaux fixes. Plus particulièrement, on distingue :

- les protocoles à accès aléatoires (ALOHA, S-ALOHA, CSMA),
- les protocoles à accès contrôle centralisé,
- les protocoles à accès contrôle distribué (jeton, …),
- les protocoles d'accès adaptatifs.

II.5 Protection contre les problèmes de transmission

Les signaux véhiculés par les réseaux sans fil subissent un nombre considérable de perturbations. Les taux d'erreur sont de l'ordre de 10^{-3} dans le cas des réseaux sans fil, alors que ce taux est de l'ordre de 10^{-10} à 10^{-12} dans le cas des réseaux fixes. Les systèmes de mobiles doivent donc être conçus pour détecter et corriger un nombre d'erreurs extrêmement élevé.

La chaîne de transmission de signaux numériques est la suivante :

- Codeur canal (ou codage correcteur d'erreur) : introduire de la redondance pour permettre la détection/correction d'erreur. Deux familles de codes sont utilisées : codes en bloc qui traitent les trames indépendamment les unes des autres et les convolutionnels qui traitent les trames par groupe (ils sont plus complexes, mais permettent de corriger plus d'erreurs avec le même nombre de bits de redondance).
- Entrelacement : comme les erreurs se produisent par avalanches (ou rafales), la fonction d'entrelacement consiste disperser au maximum les erreurs ce qui les rend plus facile à détecter et à corriger éventuellement. En général, un message est découpé en petits paquets.
- Multiplexage (fréquentiel ou temporel).
- Chiffrement : cryptage pour rendre l'information la plus illisible possible (cette fonction est importance, car dans un environnement de mobiles tout le monde peut écouter tout le monde, contrairement aux réseaux fixes où il faut se raccorder physiquement sur le réseau pour écouter).
- Modulation (MA, MF, MP ou modulation mixte).
- Egalisation : mécanisme de correction de distorsion de signaux (notamment de phase et d'amplitude).
- Les opérations inverses des six opérations précédentes sont appliquées pour retrouver, chez le récepteur, la chaîne de bits transmise.

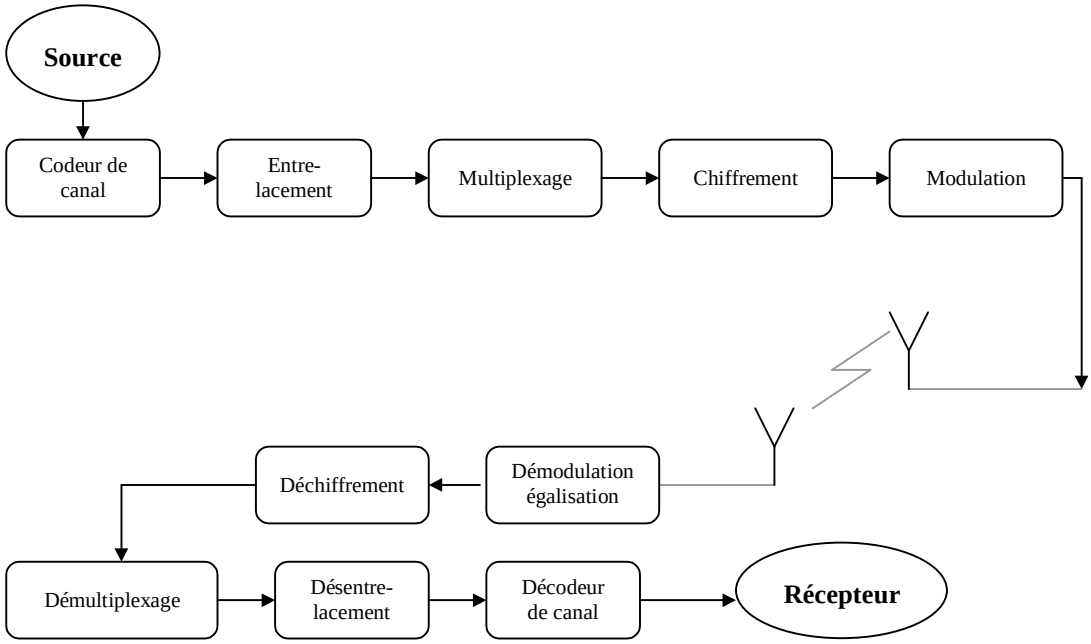


Schéma général d'une chaîne de transmission.

Les techniques de contrôle d'erreurs utilisées sont :

- Les techniques ARQ (Automatic Request for Retransmission) qui mettent en œuvre de la redondance pour détecter les erreurs et demander ensuite la retransmission explicite. Elles regroupent les techniques Stop-and-wait, Go-back-N (avec ou sans retransmission sélective).

- Les techniques FEC (Forward Error Correction) qui mettent en œuvre de la redondance pour détecter et corriger les erreurs au niveau du récepteur, sans retransmission. Elles permettent d'améliorer le débit effectif du réseau (en évitant d'inonder le réseau par les transmissions et acquittements).

II.6 Réseaux sans fil pour le transport de données

Dans le cas du réseau GSM, par exemple, les mécanismes de transmission de données nécessitent des délais d'établissement de communication de plusieurs secondes, durées ne répondant pas aux contraintes des modes d'échanges de données informatiques (applications transactionnelles, par exemple). Le trafic de données a une nature très différente de celle de la parole. Ce dernier est bien connu et donc prévisible. D'autre part, les communications de parole (une communication téléphonique) dure plusieurs minutes (en général) d'où un faible impact sur la qualité de durées d'établissement des communications de plusieurs secondes.

Dans le cas des communications de données, le trafic possède une nature souvent aléatoire (transfert de fichiers, messagerie, etc.). Cette différence a conduit à la définition de systèmes dédiés à la transmission de données.

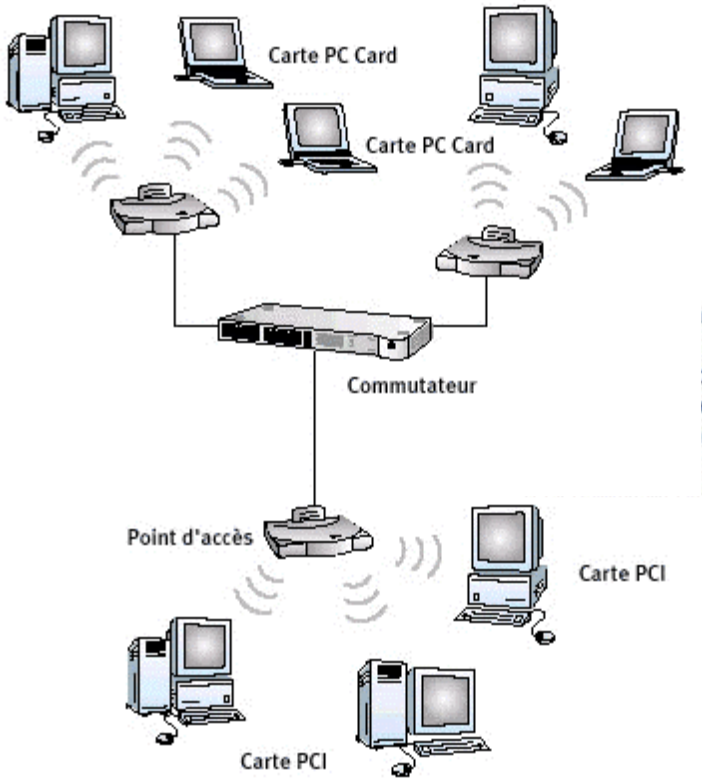
Les applications de transmission peuvent être classées en deux catégories : celles destinées au grand public (par exemple, le péage automatique, le guidage en ville, informations sur le trafic) et celles destinées aux professionnels (par exemple, le transport, le service après-vente, la télémaintenance, la vérification de cartes de crédit, suivi de stocks dans les magasins et supermarchés, les restaurants, les aéroports, etc.).

Les réseaux GSM ont apporté une réponse très basique (timide) aux besoins de transport de données (le plus souvent des textes SMS). Les réseaux 3G sont plus performants et répondent plus aux besoins de transport de données autres que la voix.

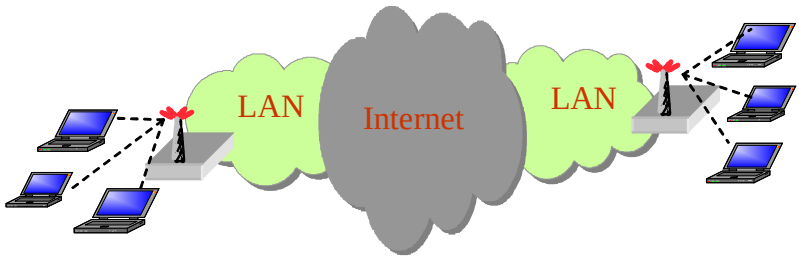
III. Réseaux locaux sans fil

III.1. Généralités

Le concept de réseau local sans fil (ou WLAN : Wireless Local Area Network) existe depuis le début des années 1980, mais la percée des WLAN n'a été significative que vers la fin des années 1990. Le principal objectif des WLAN est de permettre des communications haut débit entre des utilisateurs de faible mobilité dans une zone de couverture limitée (un bâtiment, un campus, etc.). Les utilisateurs peuvent appartenir à un même WLAN (et passent éventuellement par plusieurs points d'accès) ou à des WLAN distants (et passent par un réseau Internet -filaire ou non-).



Exemple de réseau sans fil composé de PC.



Exemple d'interconnexion de WLAN via Internet

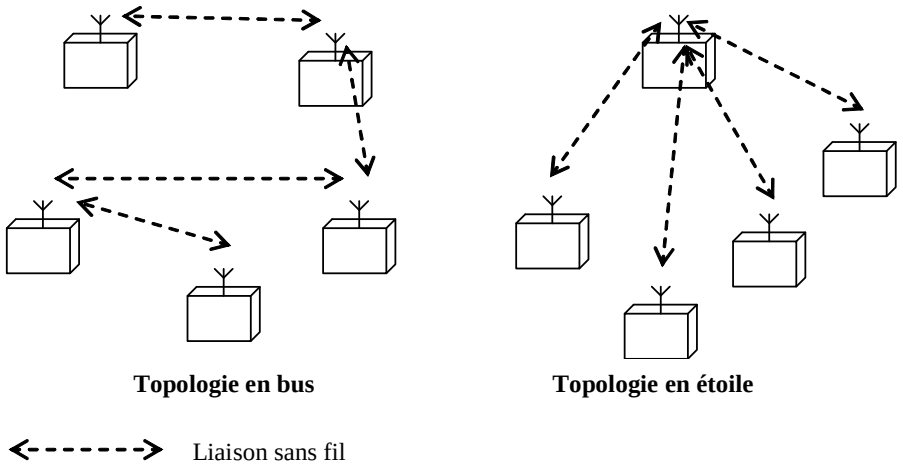
Généralement les applications des WLAN sont de trois types. La première application des WLAN est l'extension des réseaux locaux câblés existants. La seconde est l'interconnexion de réseaux locaux situés dans différents bâtiments. La troisième application est la fourniture de services de transmission de données pour usagers nomades munis d'ordinateurs portables dans des zones de type campus, gare, aéroport...

Les mécanismes de sécurité sont un aspect important dans les WLAN. La consommation de puissance est un facteur important aussi.

III.2 Topologies et couche physique

Généralement, les réseaux WLAN ont des architectures en bus ou en étoile. Généralement, les réseaux **sans infrastructure** sont en saut par saut (bus) et les réseaux **avec infrastructure** sont en étoile (les stations passent par un point d'accès pour communiquer).

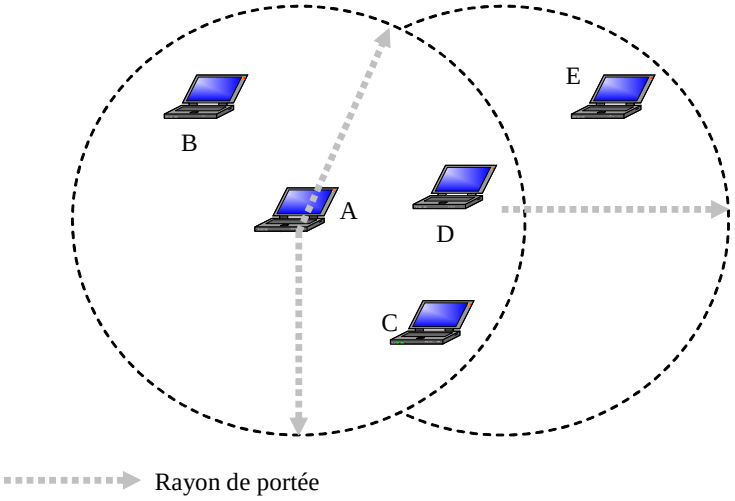
Les réseaux sans infrastructure sont généralement appelés **réseaux ad hoc**. Lorsque les stations sont mobiles on parle de réseaux ad hoc mobiles (MANETs : Mobile Ad hoc NETworks)



Principales architectures de réseaux locaux sans fil.

Pour ce qui concerne le médium, les WLAN utilisent soit l'infrarouge (IR) soit les ondes radio. Les WLAN IR sont en majorité dédiés à la communication entre terminaux fixes. Les WLAN radio offrent plus de mobilité.

Chaque station émet avec une puissance lui permettant d'atteindre directement un certain ensemble d'autres stations. On utilise le **rayon de couverture** (ou portée) pour déterminer un cercle (dont le centre est le nœud émetteur) sur lequel se trouvent les stations qui peuvent être atteintes directement par l'émetteur. Ce rayon varie de quelques mètres à quelques centaines de mètres selon la puissance de transmission. Les puissances généralement autorisées sont : ≤ 100 mW (pour les liaisons indoor), ≤ 10 mW (pour les liaisons outdoor).

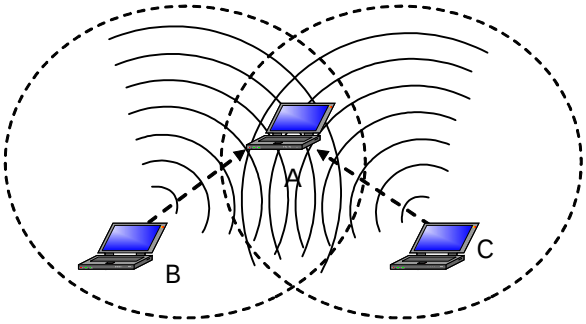


Rayon de couverture/portée

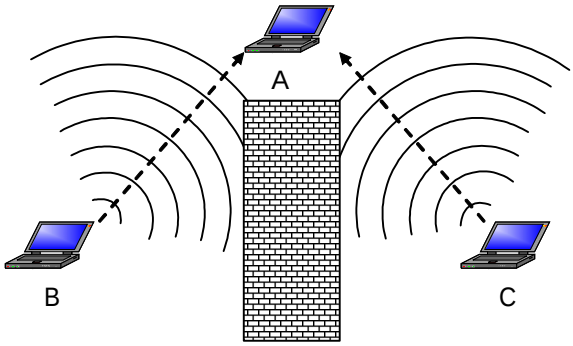
III.3 Station cachée et station exposée

Dans un réseau sans fil, les stations partagent une même bande de fréquences. Pour transmettre, toute station doit s'assurer qu'il n'y a pas une autre station en cours de transmission. Ce principe bien connu dans les réseaux Ethernet, est sujet à des nouveaux problèmes liés à la nature des réseaux sans fil. Il s'agit notamment des problèmes de la station cachée et de la station exposée.

Station cachée : dans la figure ci-dessous la station A est atteignable (visible) pour les stations B et C. La station B est soit éloignée de C, soit qu'elle ne voit pas C car il y a un obstacle (par exemple un mur) entre les deux stations. La station B commence à émettre vers A. La station C peut aussi (car elle n'entend pas la station B) émettre en même temps que B, ce qui conduit à des collisions et aucune des trames de B et C ne peut être reçue correctement par la station A. On dit que la station C est cachée par rapport à B et vice versa.



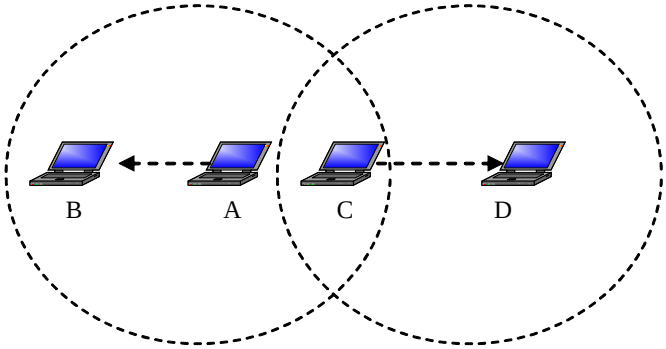
Cas de stations éloignées (affaiblissement de signal)



Cas de stations séparées par un obstacle

Problème de la station cachée

Station exposée : dans l'exemple de la figure ci-dessous, la station A émet vers B. C qui se trouve dans la zone de portée de A doit écouter la trame de A et s'abstenir d'émettre vers D. D est hors de portée de A et l'attente est donc inutile. La station D est dite exposée dans ce cas.



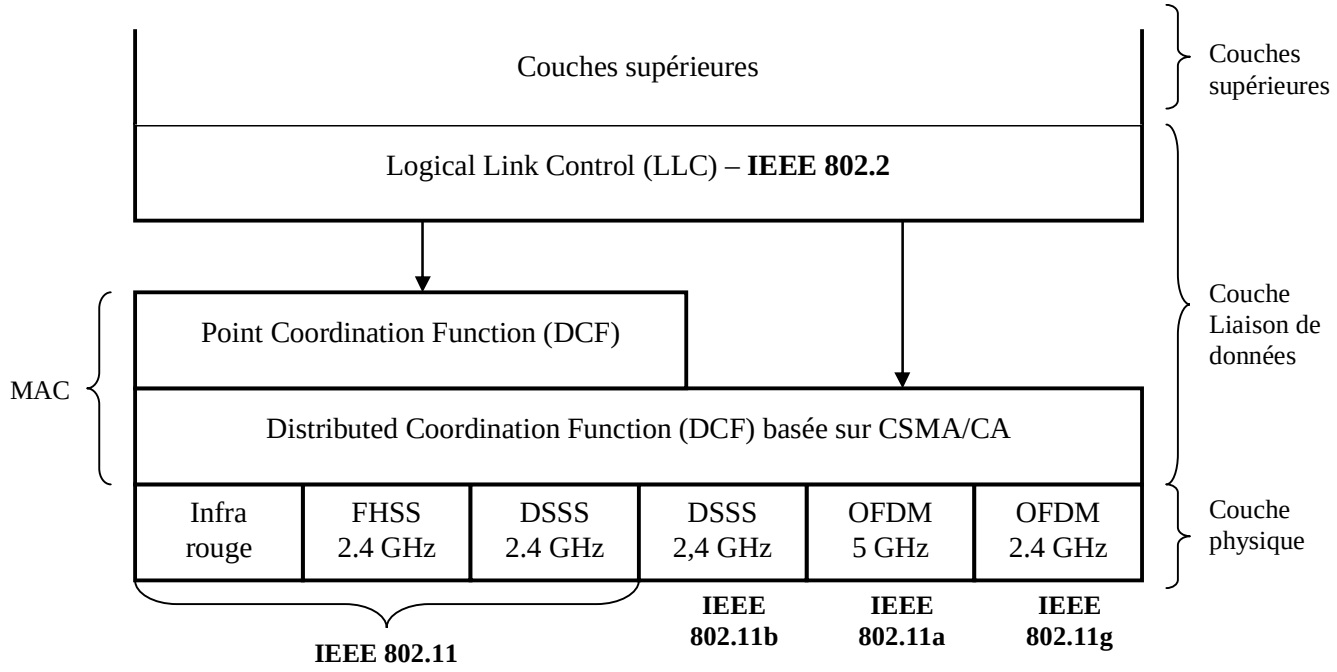
Problème de la station exposée

IV. Présentation du standard IEEE 802.11 (Wifi – WIreless FIdelity)

Il existe deux standards pour les réseaux WLAN : IEEE 802.11 et HIPERLAN (High Performance Radio Local Area Network). HIPERLAN, une norme européenne à l'origine, est soutenue essentiellement par les opérateurs de télécommunications. Très peu de produits compatibles aux normes HIPERLAN existent. IEEE 802.11 est le standard le plus répandu actuellement.

IV.1 Principes et concepts du standard IEEE 802.11

Le standard IEEE 802.11 est le standard dominant aujourd'hui dans le domaine de WLAN. Il a été adopté par l'IEEE en 1997. Ce standard décrit essentiellement la couche physique et la sous-couche MAC. Les autres couches sont reprises à partir des standards existants.



Architecture en couches d'un WLAN selon IEEE 802.11

a) Couche physique

Pour pouvoir s'adapter à différents contextes de transmission, l'IEEE 802.11 propose différents standards et techniques de modulation au niveau de la couche physique : Infra rouge, *Direct Sequence Spread Spectrum* (DSSS), *Frequency-Hopping spread spectrum* (FHSS), *Orthogonal Frequency Division Multiplexing* (OFDM). Concernant les produits commercialisés actuellement, les standards de niveau physique sont : 802.11a, 802.11b et 802.11g.

Le standard 802.11a fonctionne dans la bande de fréquence des 5 GHz. Théoriquement, ce standard offre un débit maximum de 54 Mb/s. Dans la pratique les débits observés sont de l'ordre de 20 à 25 Mb/s.

Le standard 802.11b fonctionne dans la bande de fréquences 2.4 GHz. Théoriquement, ce standard offre un débit maximum de 11 Mb/s. Dans la pratique les débits observés sont de l'ordre de 6 Mb/s.

Le standard 802.11g offre des débits de 54 Mb/s. Il utilise la bande de fréquences 2,4 GHz.

Les cartes réseaux Wifi sont disponibles pour différents formats de bus : PCI, PCMIA, USB…

Les tableaux suivants résument les principales capacités des réseaux IEEE 802.11.

Standard	Méthode de transmission	Bande de fréquences	Débits (en Mb/s)
802.11 (ancien)	FHSS, DSSS, IR	2.4 GHz, Infra rouge	1, 2
802.11b	DSSS, HR-DSSS	2.4 GHz	1, 2, 5.4, 11
802.11a	OFDM	5 GHz	6, 9, 12, 18, 24, 36, 48, 54
802.11g	DSSS, HR-DSSS, OFDM	2.4 GHz	1, 2, 5.5, 9, 12, 18, 24, 36, 48, 54

Débit 802.11b	Portée intérieure (indoor)	Portée extérieure (outdoor)
11 Mb/s	50 m	200 m
5 Mb/s	75 m	300 m
2 Mb/s	100 m	400 m
1 Mb/s	150 m	500 m

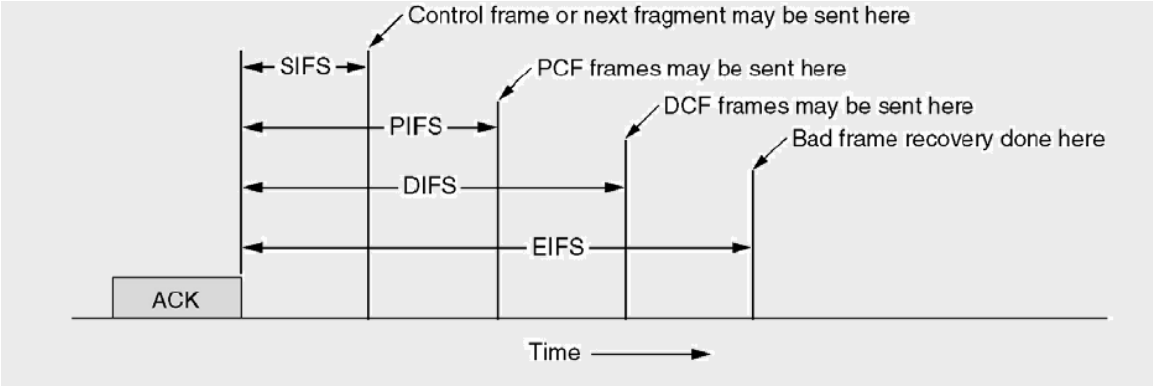
Débit 802.11g	Portée intérieure (indoor)	Portée extérieure (outdoor)
54 Mb/s	20 m	70 m
36 Mb/s	30 m	120 m
18 Mb/s	55 m	180 m
9 Mb/s	75 m	350 m
6 Mb/s	95 m	400 m

b) Temps d'espacement de trame

Le standard 802.11 définit quatre valeurs de temps pour espacer les trames transmises par les équipements :

- SIFS (Short Interframe Space) : temps minimum d'attente que doit observer toute station avant de tenter de transmettre une fois qu'elle a détecté que le canal est libre.
- PIFS (PCF Interframe Space) : temps d'attente que doit observer le point d'accès avant de transmettre (PIFS est utilisé en mode centralisé)
- DIFS (DCF Interframe Space) : temps d'attente que doit observer toute station avant de transmettre (DIFS est utilisé en mode décentralisé)
- EIFS (Extended Interframe space) : temps d'attente que doit observer une trame qui a émis une trame et qui attend un acquittement avant de conclure qu'il y a eu une erreur (sur sa trame ou sur l'acquittement).

La figure suivante montre les relations entre les différents espacements (SIFS < PIFS < DIFS < EIFS). Les valeurs sont configurables selon les spécificités de chaque réseau



Paramètres d'espacement de trames dans IEEE 802.11

c) Architectures de réseau Wifi

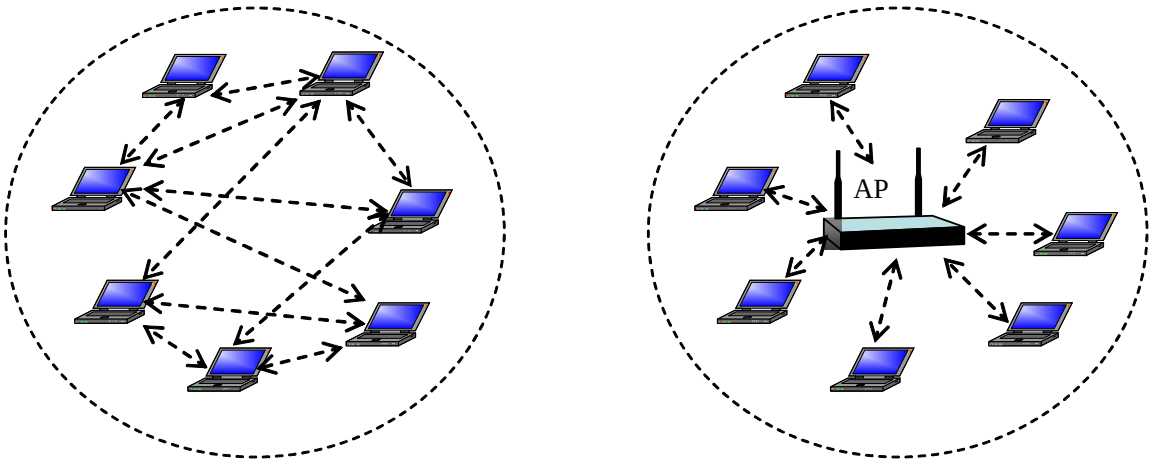
Station (STA) : tout appareil (équipement) compatible au standard pouvant transmettre/recevoir.

Point d'accès (AP) : tout équipement pouvant gérer l'accès au réseau (c'est-à-dire assurer l'attribution du canal aux autres équipements).

Basic Service Set (BSS) : un BSS est constitué d'un ensemble de stations. Chaque BSS est identifié par un BSSID (BSS Identifier). Un BSS est un réseau de base.

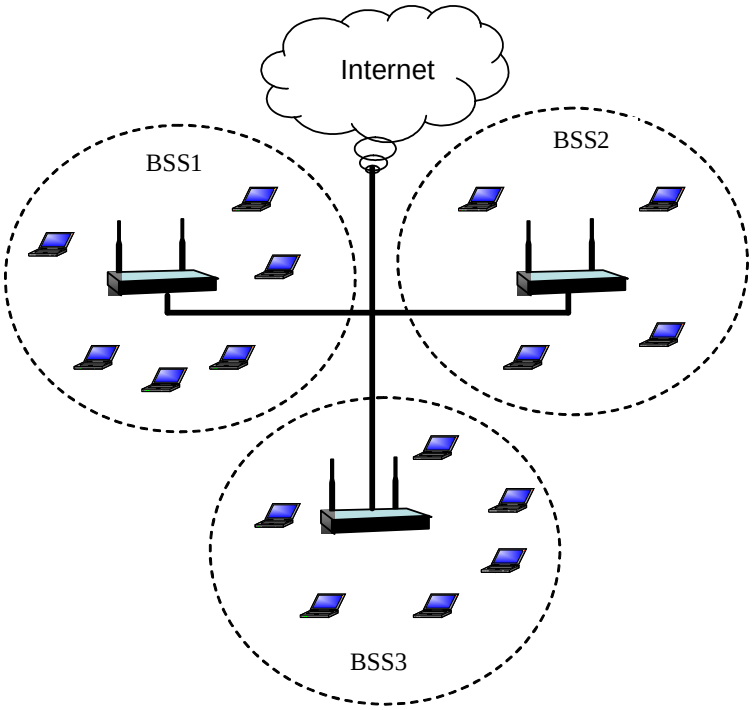
Infrastructure BSS : un Infrastructure BSS est constitué d'un ensemble de stations et d'un AP.
Independent BSS (IBSS) : un IBSS est un BSS sans AP. Ce type de réseau est dit aussi réseau ad hoc.

Réseau Wifi étendu (Extended Service Set ou ESS) : un ESS est formé d'un ensemble de BSS reliés entre eux par un réseau local de type Ethernet par exemple. Lorsqu'un utilisateur nomade passe d'un BSS à un autre en se déplaçant l'adaptateur réseau de sa machine est capable de changer de point d'accès selon la qualité des signaux provenant des différents AP du ESS (c'est la fonction de roaming au niveau WLAN).



Independent BSS Infrastructure BSS

Types de BSS de réseau IEEE 802.11



ESS - Réseau IEEE 802.11 étendu

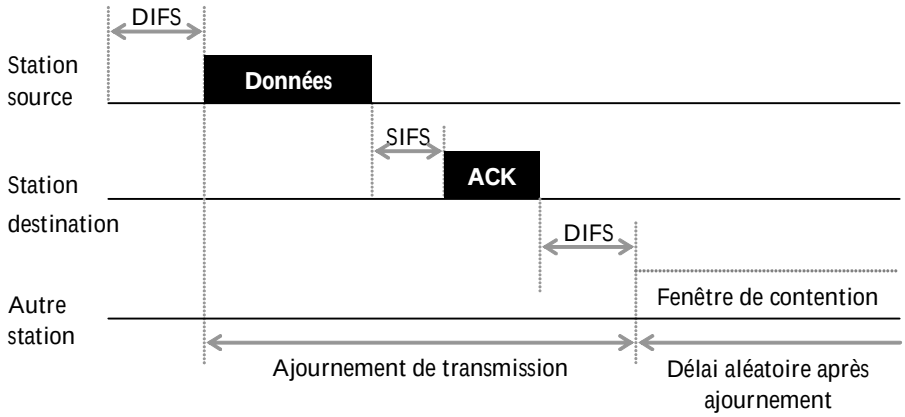
IV.2 Sous couche MAC

Un réseau IEEE 802.11 peut fonctionner selon le mode décentralisé (ou sans infrastructure) ou selon le mode centralisé (ou avec infrastructure). Les deux modes utilisent la méthode d'accès CSMA/CA.

La méthode CSMA/CA est une amélioration de CSMA pour s'adapter à des réseaux où la détection de collision n'est pas possible, comme c'est le cas des réseaux hertziens dans lesquels deux stations qui communiquent avec une troisième ne s'entendent pas forcément mutuellement en raison soit de leur rayon de portée, soit de la présence d'obstacle et entrent en collision.

a) Fonctionnement en mode décentralisé (DCF)

Comme son nom l'indique, dans le mode DCF (*Distributed Coordination Function*), il n'y a pas de station qui règle le contrôle d'accès au canal. Toutes les stations sont autonomes et sont concurrentes pour accéder au canal. Pour éviter/minimiser les situations collisions, La station voulant émettre une trame écoute le canal. Si le canal est libre pendant un temps *Distributed Inter Frame Space* (noté *DIFS*), alors la station transmet. A la réception des données, et après un délai *Short Inter Frame Space* (noté *SIFS*), le récepteur envoie un accusé de réception (*ACK*). Si l'accusé de réception n'est pas reçu dans les temps, la source tente une nouvelle retransmission, après écoute du canal évidemment. La figure suivante illustre le principe de fonctionnement de CSMA/CA.



Fonctionnement de la méthode CSMA/CA.

Si le canal est occupé au moment où la station le teste afin de savoir si elle peut transmettre sa trame, la transmission est ajournée. La station attend jusqu'à ce que le canal redevienne libre. Ensuite, si le canal reste libre pendant un temps égal à DIFS, elle tire un nombre aléatoire pour armer son temporisateur de retransmission. La station écoute le canal pendant que son temporisateur se décrémente. Lorsque ce temporisateur expire, la station commence à transmettre. Si le canal devient occupé avant l'expiration de son temporisateur, la station arrête (ou gèle) son temporisateur qui reprend sa décrémentation une fois que le canal redevient libre à nouveau.

La valeur du délai (ce délai est appelé *backoff*) d'attente avant retransmission est un multiple d'un temps de base (time slot) tiré de manière aléatoire dans la fenêtre de contention *[0, W]*. Lorsqu'une nouvelle trame arrive au niveau MAC pour être transmise, la variable *W* est initialisée à *CWmin*, ensuite elle est multipliée par 2 à chaque tentative de transmission (il y a retransmission quand l'émetteur ne reçoit pas d'ACK au bout d'un certain temps), mais sans dépasser un seuil égal à *CWmax*.

La figure suivante montre un exemple où 5 stations utilisent le canal. Au début, on suppose que la station *A* est en cours de transmission de sa trame. Pendant que la station *A* transmet, des nouvelles trames à émettre arrivent dans les sous-couches MAC des stations *B*, *C* et *D*, mais ces stations détectent que le canal est occupé et doivent attendre la fin de la transaction de la station *A*. A la fin de la transaction de la station *A*, les trois stations en attente tirent chacune une valeur aléatoire de backoff. C'est la station *B* qui prend le contrôle du canal car elle a tiré la plus petite valeur de backoff. Les stations *C* et *D* doivent continuer à attendre (en ayant leur timer gelé). A la fin de la transaction de la station *B*, c'est la station *D* qui a la plus petite valeur de backoff restant qui prend le contrôle du canal. La station *C* doit attendre encore. Pendant la transaction de la station *D*, une nouvelle trame arrive au niveau MAC de la station *E* qui devient à son tour candidate pour occuper le canal. A la fin de la transaction de la station *D*. La station *E* tire une valeur aléatoire de backoff plus petite que le backoff restant de *C* ; elle prend donc le contrôle du canal. A la fin de la transaction de la station *E*, la station *C* prend le contrôle du canal car elle est la seule à participer à la fenêtre de contention.

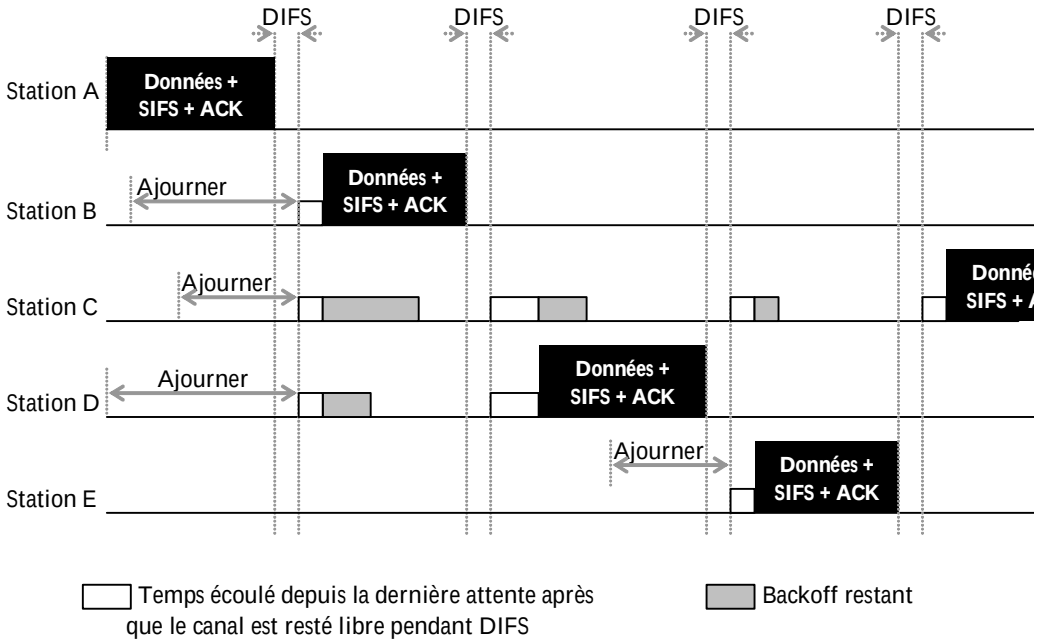


Figure. Exemple de tentatives de transmission et retransmission avec CSMA/CA.

b) Fonctionnement en mode centralisé (PCF)

Le mode PCF (*Point Coordination Function*) est un mode optionnel qui constitue un mécanisme basique pour répondre aux besoins de qualité de service exprimés par les applications transmettant du trafic avec des contraintes de temps (notamment les trafics audio et vidéo). Il y a d'autres protocoles de IEEE 802.11 plus puissants pour répondre aux besoins de qualité de service, ils ne seront pas présenter dans ce document.

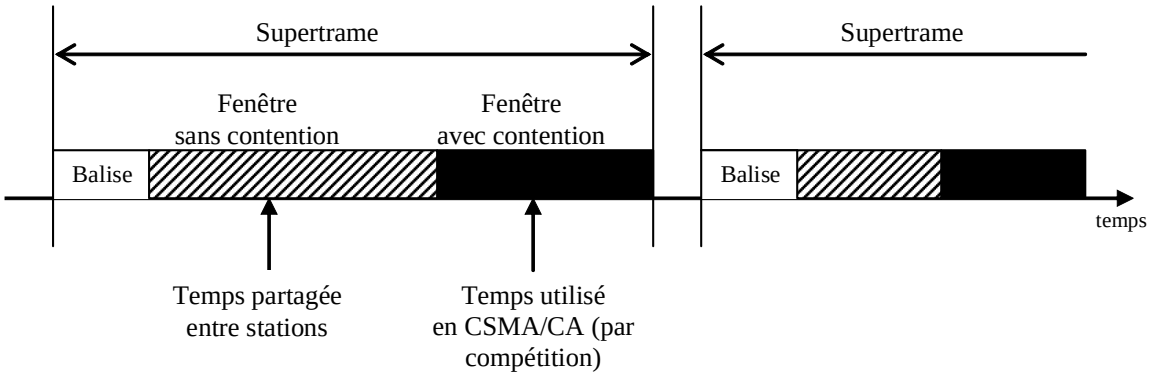
Le mode PCF est basé sur l'utilisation de point d'accès (appelé central **Point Coordinator**) qui régule l'accès au canal. Le mode PCF ne fonctionne donc qu'en mode infrastructure.

Un réseau peut fonctionner totalement en mode centralisé ou selon les deux modes centralisé et décentralisé. Dans le cas où les deux modes sont utilisés, on définit un intervalle de temps (appelé super trame) divisé en deux fenêtres : une pour le mode centralisé (c'est la **fenêtre sans contention**) et une autre pour le mode décentralisé (c'est la **fenêtre avec contention**).

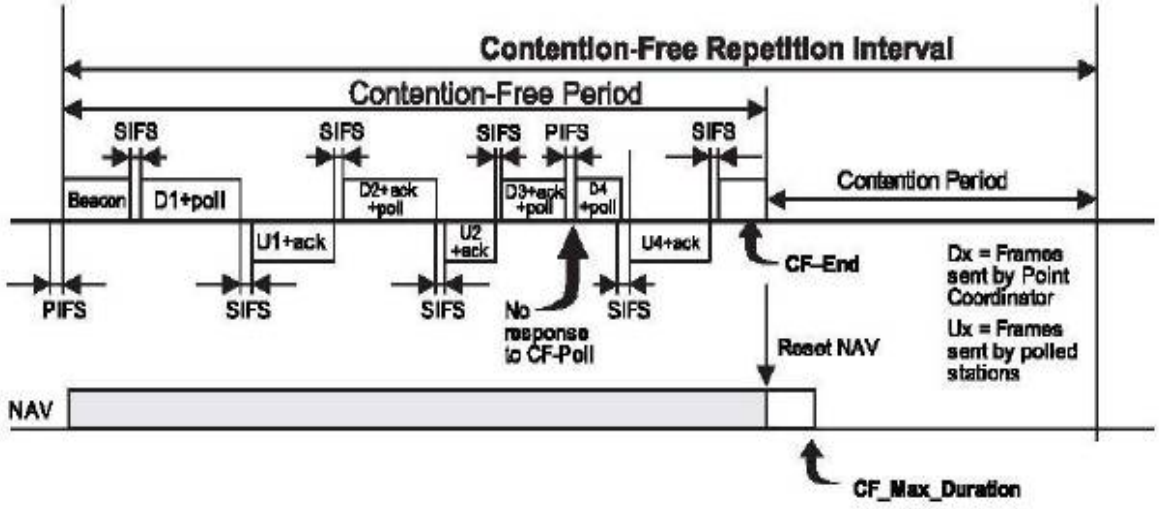
Durant la fenêtre avec contention, le réseau fonctionne en mode décentralisé et chaque station peut rentrer en compétition avec les autres pour transmettre à n'importe quel moment.

Chaque super trame commence par une trame balise (*beacon*). Au début d'une super trame, le point d'accès attend pendant un temps égal à PIFS et diffuse ensuite la balise. Pendant la fenêtre sans contention, le point d'accès scrute, à tour de rôle, une ou plusieurs stations pour les inviter à émettre. Le point d'accès transmet une trame CF-Poll par station scrutée. Chaque station scrutée transmet une trame de données suivie d'un ACK émis par le récepteur ou bien un ACK d'invitation (CF-ACK) pour indiquer qu'elle n'a pas de données. Si la station scrutée ne répond pas du tout (car elle ne fonctionne pas), le point d'accès attend pendant un PIFS et scrute la station suivante jusqu'à ce que la fenêtre sans contention soit consommée ou que toutes les stations ont été scrutées. A la fin de la fenêtre sans contention, le point d'accès diffuse une trame CF-END pour signaler aux stations qu'elles peuvent passer en mode décentralisé (c'est-à-dire chaque station peut transmettre en respectant seulement les règles de CSMA/CA).

Dans le mode PCF, les stations qui n'implantent pas ce mécanisme n'interfèrent pas sur le bon fonctionnement de ce mode. Elles ne répondent pas aux scrutations du point d'accès (car elles ne reconnaissent par les trames CF-Poll) et elles ne peuvent pas transmettre dans la fenêtre sans contention. En effet, en mode PCF, le temps d'attente avant de transmettre est PIFS (le canal n'est jamais libre pendant un temps supérieur à PIFS). Par conséquent, les stations qui ne fonctionnent qu'en mode DCF, ne peuvent pas transmettre car elles sont obligées d'attendre au moins un temps égal à DIFS (sachant que DIFS > PIFS).



Principe général du mode PCF



Exemple de fonctionnement du mode PCF
(quatre stations sont scrutées : les stations 1, 2 et 3 transmettent chacune une trame de données suivie d'un Ack, la station 4 n'a pas de données à transmettre et répond directement par un Ack)

c) CSMA/CA utilisée conjointement avec le mécanisme de réservation

La méthode CSMA/CA décrite précédemment fonctionne bien si la trame en cours de transmission est détectée par toutes les autres stations du réseau qui veulent prendre le contrôle du canal (comme dans le cas d'un réseau Ethernet câblé). Or, comme nous l'avons signalé précédemment, deux ou plusieurs stations qui ne s'entendent pas, à cause de problèmes de la station cachée et de couverture, peuvent prendre le contrôle du canal en même temps et finissent par entrer en collision. Pour réduire les situations de collision dues à ces deux problèmes, un mécanisme de réservation peut être utilisé conjointement avec CSMA/CA.

La station voulant transmettre écoute le canal. Si le canal est occupé, la transmission est ajournée. Si le canal est libre pendant un temps égal à *DIFS*, alors la station transmet une trame appelée *Ready To Send* (notée *RTS* signifiant *prêt à émettre*) contenant la durée nécessaire pour la transmission des données et de l'accusé de réception, pour avertir les autres stations de son intention de prendre le contrôle (i.e. réserver) le canal. Si la trame RTS parvient à son destinataire[1] (qui est généralement un point d'accès), celui-ci attend un temps égal

[1] *La station destinataire de trame RTS doit être visible (i.e. entendue par) des stations qui ne s'entendent pas et pour lesquelles la crainte de collision est forte pour permettre à ces stations de recevoir la trame CTS et de s'abstenir de transmettre en conséquence.*

à *SIFS* et répond par une trame appelée *Clear To Send* (*CTS*, signifiant *Le champ est libre pour émettre*), contenant les mêmes informations que la trame RTS à laquelle il répond. Après réception de la trame CTS, suivie d'un silence pendant un SIFS, la source transmet sa trame de données, suivie d'un silence pendant un SIFS, suivi d'un acquittement. Si d'autres stations tentent aussi de prendre le contrôle du canal, les trames RTS vont entrer en collision. Une station qui transmet une trame RTS et qui ne reçoit pas en retour une trame CTS, tente sa transmission de trame RTS plus tard.

Toutes les stations qui reçoivent une trame RTS ou CTS (et qui ne sont ni la source ni la destination de ces trames) utilisent le délai de réservation contenu dans ces trames pour armer leur temporisateur appelé *Network Allocation Vector* (noté par *NAV*). Le NAV contient donc une prédiction du trafic qui occupera le canal dans un futur immédiat pendant lequel ces stations sont invitées à s'abstenir de transmettre. Une station qui a armé son NAV ne peut devenir candidate que lorsque ce temporisateur atteint zéro. On dit que le mécanisme RTS/CTS constitue une *écoute virtuelle de la porteuse* effectuée par la sous-couche MAC et ce mécanisme est utilisé conjointement avec CSMA qui, elle, effectue l'écoute de la porteuse par la couche physique. La figure suivante illustre le fonctionnement de CSMA/CA avec RTS/CTS.

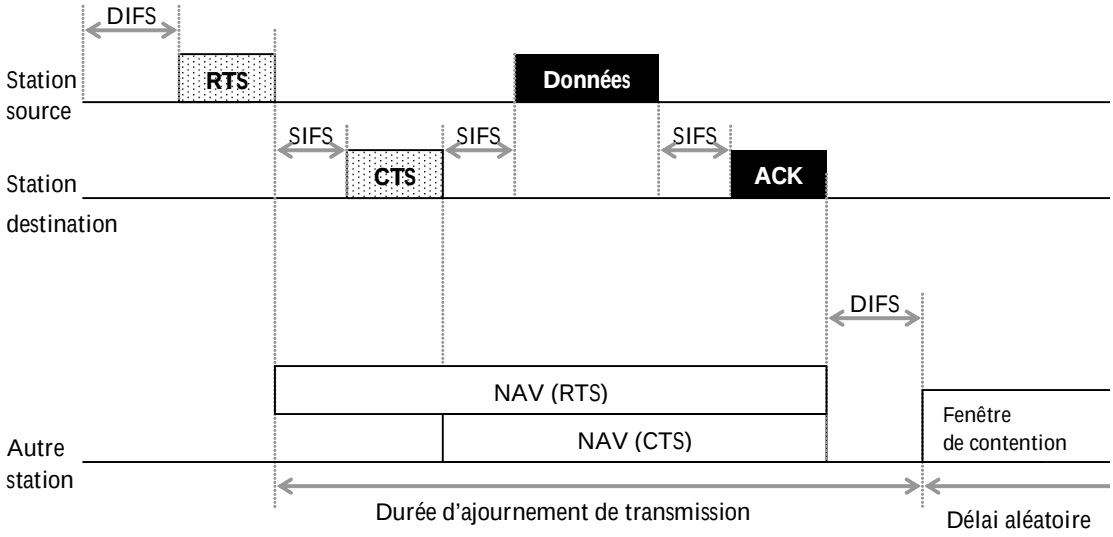


Figure. Exemple de fonctionnement de la méthode CSMA/CA + RTS/CTS

Il est important de souligner que les trames RTS et CTS sont très courtes (la trame RTS fait 20 octets et la trame CTS, 14 octets). Par conséquent, la période de vulnérabilité de ces deux trames est plus courte que celle des trames de données qui sont en général plus longues. Ainsi, au lieu d'envoyer directement une trame de données (comme le fait CSMA/CA de base), avec le risque d'avoir une collision de longue durée, on envoie d'abord une courte trame de réservation, avec le risque d'avoir une collision de courte durée, et si cette réservation réussit alors on transmet la trame de données. Il faut noter aussi que l'efficacité du mécanisme RTS/CTS dépend du ratio entre la taille de trame RTS/CTS et la taille maximale de trame de données, si ce ratio tend vers 1, le mécanisme RTS/CTS n'est plus utile. Enfin, il faut noter que les collisions ne sont pas complètement éliminées grâce au mécanisme de réservation RTS/CTS, car deux ou plusieurs stations, qui ne s'entendent pas, peuvent envoyer simultanément des trames RTS.

IV.3 Autres fonctionnalités de Wifi

Procédure d'entrée dans un BSS

Lorsqu'une station veut entrer dans un BSS (un réseau WIFI), elle doit se synchroniser avec le point d'accès. Cette synchronisation peut se faire de deux manières :

Par scrutation passive : dans ce cas, la station doit attendre que le point d'accès diffuse une trame balise. La trame balise est émise périodiquement par le point d'accès et contient : le SSID du réseau, les débits binaires supportés par le point d'accès, la période à laquelle la trame balise est émise, un TIM (Traffic Indication Map) et l'adresse MAC de l'AP et une estampille

Par scrutation active : dans ce cas, la station qui veut rentrer dans le réseau diffuse une trame sonde et attend qu'un AP réponde à sa trame. Si elle n'obtient pas de réponse, elle refait sa tentative quelques instants plus tard.

Procédure d'authentification

Une fois que la station a découvert un point d'accès, elle entre dans la phase d'authentification. Cette dernière peut se faire par : un système d'authentification ouvert (système par défaut) ou par un protocole d'authentification utilisant une clé partagée entre la station et le point d'accès.

Sécurité des données échangées

Comme les réseaux sans fil sont plus vulnérables que les réseaux filaires, la sécurité des réseaux sans fil est un point critique. Beaucoup de solutions ont été proposées pour sécuriser les réseaux Wifi notamment. Dans les premiers réseaux Wifi, il a été constaté que la plupart des réseaux n'étaient pas sécurisés (tout circulait en clair). Le problème vient du fait que les utilisateurs étaient habitués aux réseaux filaires qui étaient généralement sécurisés par des firewalls et les utilisateurs n'avaient rien à configurer sur leurs machines. L'utilisation des réseaux sans fil impose de nouveaux comportements aux utilisateurs en matière de sécurité.

Les premières solutions (basiques) de sécurité étaient basées sur WEP et ACL notamment.

WEP (Wired Equivalent Privacy)

- Principe : partage de clé secrète pour chiffrer les données
- Inconvénient : pas très robuste aux attaques (le protocole WEP est de moins en moins utilisé)

ACL (access control list) :

- Principe : une liste des adresses MAC des stations autorisées à utiliser le réseau est maintenue par l'AP.
- Inconvénient : il suffit de sniffer le réseau puis copier les adresses MAC de l'ACL.

D'autres protocoles plus puissants ont été aussi développés pour sécuriser les réseaux Wifi, notamment EAP-TLS, EAP-MD5, PEAP, WPA, WPA2. Une norme de sécurité a été proposée aussi par le comité IEEE 802.11, il s'agit de la norme IEEE 802.11i.

Procédure de roaming

Des protocoles, comme RADIUS (Remote Authentication Dial-In User Service) sont utilisés pour permettre la mobilité entre BSS interconnectés. Ces protocoles permettent (de manière transparente à l'utilisateur) de désassocier une station d'un AP et l'associer à un autre plus proche de sa nouvelle localisation.

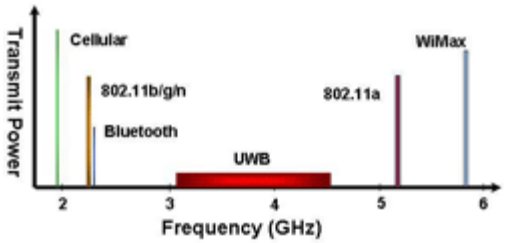
Procédure de gestion d'énergie

Une procédure est intégrée à Wifi pour optimiser la consommation d'énergie de la station. Les stations qui n'ont pas de données à émettre se mettent en mode veille.

V. Technologie UWB (UltraWideBand)

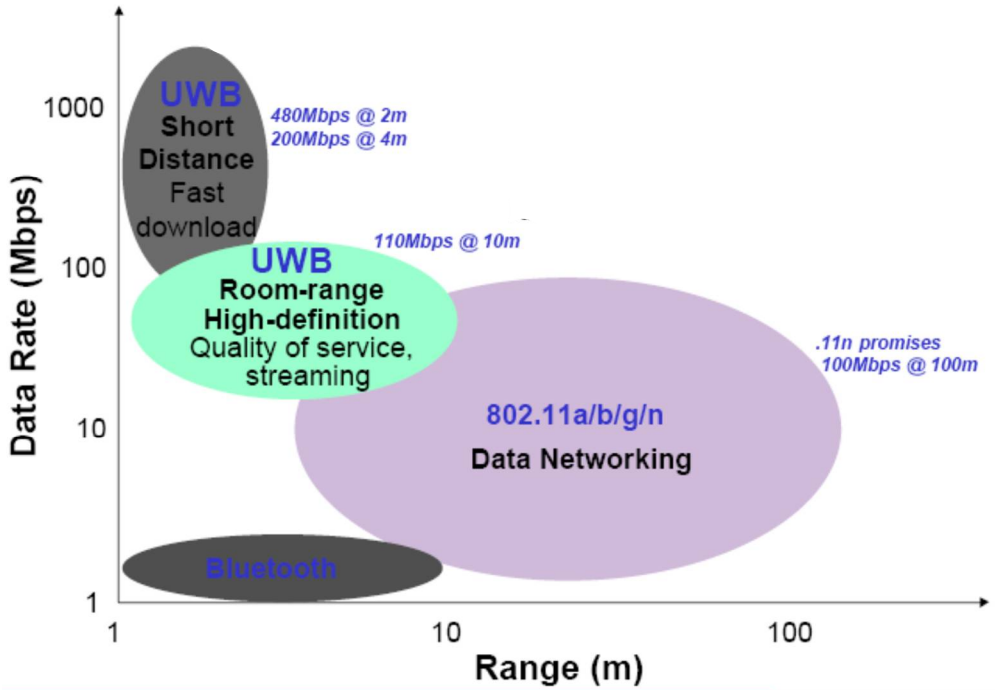
La technologie UWB offre un très large spectre de fréquences pour permettre la communication sans fil et consomme très peu d'énergie. Elle ne fonctionne que sur les (très) courtes distances. Elle peut donc être utilisée dans tout ce qui est communication personnelle à domicile sans interruption, ni attaque de sécurité par les voisins (comme c'est le cas avec la technologie Wifi).

UWB peut être utilisée, à la place de USB, pour raccorder des équipements locaux.



	Bluetooth	**802.15.4**		**UWB**	
		standard	**w/ZigBee**	**UWB Forum**	**WiMedia**
Spectre radio	2.4 GHz	868 MHz, 915 MHz, 2.4 GHz		3.1 GHz - 4.85 GHz, 6.2 GHz - 9.7 GHz	3.1 GHz - 10.6 GHz
Débit maxi	3 Mbps	250 kbps		2 Gbps	480 Mbps
Puissance radio	< 100 mW	> 1 mW		Dépend du débit : < 0.074 mW/GHz	
Distance maxi	1 m - 100 m	1 m - 100 m		varaible, très faible	
MAC	TDD,	CSMA-CA, TDD		Non connue	CSMA-CA, OFDM, TDD
Sécurité optionnelle	SAFER+, négociation de clé	AES-128	AES-128, autorité de clé	Non connue	AES-128, négociation de clé
Applications	Remplacement de câble faible débit	Capteurs, domotique		Remplacement de câble haut débit	

Comparaison des technologies de réseaux PAN



UWB vs. Autres réseaux PAN/WLAN

Chapitre 6

Couche Transport

Protocole TCP

I. Généralités

I.1. Objectifs et fonctions de la couche transport

La couche transport est une couche charnière entre les fonctions de traitement de données (couches 5, 6 et 7) et les couches orientées communication (couches 1, 2 et 3).

La couche transport a pour rôle de cacher aux applications les spécificités liées aux réseaux de communications (réseaux fiables ou non, …), en leur offrant un service avec la qualité de service requise -quand cela est possible- en gérant efficacement les ressources disponibles (lignes de communication spécialisées, réseaux, etc.). Elle permet la portabilité des applications.

La couche transport ressemble à la couche liaison de données, mais elles sont différentes : la couche liaison de données gère des connexions entre deux stations directement connectées, la couche transport, quant à elle, gère des connexions logiques établies le long d'un réseau ou de plusieurs réseaux interconnectés. Ces réseaux introduisent des retards de livraison de paquets, délivrent les paquets dans le désordre, … ce qui complique la tâche de la couche de transport.

Comme pour la couche réseau, on distingue deux classes de service de transport : ***Avec connexion*** ou ***Sans connexion***.

Les fonctions réalisées par la couche transport sont notamment les suivantes :

- ***Transport de bout en bout*** de messages en respectant la qualité de service requise par l'utilisateur.
- ***Fragmentation/réassemblage*** : les messages dont la taille dépasse la taille maximale de paquet autorisée par le réseau sont découpés (fragmentés) à l'émission et rassemblés à la réception.
- ***Contrôle de flux*** en utilisant généralement la technique de la fenêtre coulissante.
- **Contrôle d'erreurs** en utilisant généralement les techniques de fenêtre d'anticipation et retransmission.
- ***Séquencement*** de paquets : comme les paquets constituant un message peuvent être envoyés sur des chemins différents, ils peuvent être reçus dans un ordre différent de celui de leur émission. Le protocole de transport remet les paquets dans l'ordre avant de les délivrer au récepteur.
- ***Multiplexage*** de connexion (en fusion ou fission) : un connexion de transport peut utiliser deux ou plusieurs connexions de réseau (cela permet d'augmenter le débit de la connexion de transport ou sa sécurité). Plusieurs connexions de niveau transport peuvent emprunter la même connexion de réseau (cela permet d'optimiser le nombre de connexions réseau utilisées – il faut rappeler que les connexions réseau sont parfois payantes et il faut donc en minimiser le nombre).
- **Sécurité** : mise en place de mécanismes de sécurité des échanges si les niveaux inférieurs et supérieurs ne gèrent pas du tout (ou ne gèrent pas comme il se doit) la sécurité
- …

I.2 Notions de point d'écoute, extrémité de connexion, port, socket

Pour pouvoir communiquer avec ses correspondants, un processus applicatif se met à l'écoute d'éventuelles demandes de connexion ou de transfert de données. Les **points d'écoute** sont appelés :

- points d'accès au service transport (TSAP : transport access points) dans le monde OSI ;
- **ports** dans le monde TCP/IP.

De nombreux programmes peuvent être exécutés simultanément sur Internet (on peut par exemple ouvrir plusieurs navigateurs simultanément ou bien naviguer sur des pages HTML tout en téléchargeant un fichier par FTP). Chacun de ces programmes travaille avec un protocole, toutefois l'ordinateur doit pouvoir distinguer les différentes sources de données.

L'adresse réseau (par exemple l'adresse IP) sert donc à identifier de façon unique un ordinateur sur le réseau tandis que le numéro de port indique l'application à laquelle les données sont destinées. De cette manière, lorsque l'ordinateur reçoit des informations destinées à un port, les données sont envoyées vers l'application correspondante. S'il s'agit d'une requête à destination de l'application, l'application est appelée application **serveur**. S'il s'agit d'une réponse, on parle alors d'application **cliente**.

Aussi bien dans le monde OSI que dans le monde TCP/IP, les points d'écoute sont identifiés par des numéros. Certains numéros sont réservés.

Exemple de numéros réservés dans TCP/IP :

`ftp` : port 21 : `telnet` : port 23 `smtp` : port 25

Les points d'écoute sont affectés de manière statique (on parle de ports réservés) ou dynamique.

Un annuaire peut être utilisé pour retrouver le numéro de point d'écoute à partir du nom de service demandé.

Dans le cas d'une couche transport en mode connecté, on appelle ***connexion de transport*** le lien logique permettant à deux hôtes distants de communiquer.

Une connexion de transport est identifiée par chacun des deux correspondants par un identificateur local d'***extrémité de connexion***.

Dans le cas de TCP, une connexion est identifiée par chacun des deux hôtes par une ***socket*** (socket = joint, douille). Ainsi, une connexion TCP est identifiée à l'aide de deux sockets locales. Une socket est la combinaison d'une adresse IP et d'un numéro de port.

II. Couche transport OSI

II.1. Primitives de transport OSI

a) Etablissement de connexion

On utilise le service `T-CONNECT` (qui est un service confirmé).
Il y a négociation de paramètres de qualité de service pendant la phase d'établissement de connexion. Une telle négociation est importante pour certaines applications comme celles de type multimédia où il existe des contraintes sur la qualité du son et de l'image. Les principaux paramètres de qualité de service au niveau transport sont :

- *Temps d'établissement de connexion* : durée qui s'écoule entre l'émission d'une demande de connexion et la réception de sa confirmation.

- *Probabilité d'échec de connexion* : mesure le risque qu'une connexion ne puisse s'établir dans un délai maximum défini, à la suite, par exemple, de congestion de réseau ou d'indisponibilité du correspondant.

- *Débit de la liaison* : débit binaire estimé dans chaque sens de la liaison.

- *Temps de transit* : temps de transfert de message.

- *Taux d'erreur résiduel* : rapport entre le nombre de messages perdus ou mal transmis sur le nombre total de messages transmis. Ce nombre a une valeur non nulle sur de longues durées de communication à cause des erreurs qui affectent le niveau physique.

- *Protection* : possibilité de sécurisation des données durant le transfert.

- *Priorité* : possibilité de privilégier certaines données par rapport à d'autres (on parle de données express et normales).

- *Probabilité de résiliation* : probabilité que la couche de transport décide d'elle-même de la déconnexion suite à des anomalies répétées sur le réseau.

b) Fermeture de connexion

Elle consiste à libérer les ressources réservées pour répondre aux besoins de qualité de service d'une connexion. Avant de fermer une connexion, on doit s'assurer que tous les paquets et messages en instance d'acquittement sont bien acquittés.

On utilise le service `T-DISCONNECT` (qui est un service non confirmé)

c) Emission et réception de messages et contrôle de flux

Les messages sont numérotés pour faciliter le contrôle d'erreurs et de flux.

On utilise le service `T-DATA` (pour transmettre des données normales en mode connecté), le service `T-EXPEDITED-DATA` (pour transmettre des données urgentes en mode connecté) et le service `T-UNIT-DATA` (pour transmettre des données en mode non connecté).

II.2. Classes de transport OSI

Les réseaux utilisés pour la communication de paquets peuvent être classés en trois types selon leur degré de fiabilité :

- *Type A* : taux d'erreurs signalées acceptable et taux d'erreurs non signalées acceptable.

- *Type B* : taux d'erreurs signalées inacceptable et taux d'erreurs non signalées acceptable.

- *Type C* : taux d'erreurs signalées inacceptable et taux d'erreurs non signalées inacceptable.

Les erreurs signalées sont les erreurs détectées par la couche réseau et signalées à la couche transport. Les erreurs non signalées sont des erreurs éventuellement détectées par la couche réseau mais qui ne sont pas signalées à la couche transport pour qu'elle prenne les mesures adéquates face à ce type d'erreurs.

On dit qu'un taux d'erreurs signalées (respectivement non signalées) est acceptable quand ce nombre est trop faible pour nécessiter la mise en place d'un mécanisme de contrôle d'erreurs au niveau de la couche transport. En d'autres termes, on considère que le réseau est fiable.
On dit qu'un taux d'erreurs signalées (respectivement non signalées) est inacceptable quand ce nombre est jugé suffisamment élevé pour nécessiter la mise en place d'un mécanisme de contrôle d'erreurs au niveau de la couche transport. En d'autres termes, on considère que le réseau n'est pas fiable.

Pour tenir compte à la fois des besoins, notamment en termes de fiabilité, des couches utilisatrices de la couche transport et des types de réseaux sous-jacents, cinq classes de services de transport ont été définies par l'ISO :

- **Classe 0** (ou TP0 Transport Protocol 0) :
  - pas de multiplexage,
  - pas de reprise sur erreur,
  - adaptée aux réseaux de type A.

- **Classe 1** (ou TP1 Transport Protocol 1) :
  - pas de multiplexage,
  - reprise sur erreurs signalées,
  - adaptée aux réseaux de type B.

- **Classe 2** (ou TP2 Transport Protocol 2) :
  - multiplexage avec ou sans contrôle de flux,
  - pas de reprise sur erreur,
  - adaptée aux réseaux de type A.

- **Classe 3** (ou TP3 Transport Protocol 3) :
  - multiplexage avec ou sans contrôle de flux,
  - reprise sur erreurs signalées,
  - adaptée aux de type B.

- **Classe 4** (ou TP4 Transport Protocol 4) :
  - multiplexage avec contrôle de flux,
  - reprise sur erreurs signalées et non signalées,
  - adaptée aux réseaux de type C.

II.3. Détection des erreurs dans le protocole TP4 de l'OSI

Dans le protocole TP4, on définit un ensemble de paramètres qui permettent de détecter les situations d'erreurs. Une cascade d'erreurs peut conduire le protocole de transport à abandonner une connexion car la reprise est jugée impossible ou trop complexe à mettre en œuvre. Les paramètres utilisés sont essentiellement les suivants :

- temps maximum pouvant s'écouler entre l'émission d'un segment (c'est-à-dire un paquet contenant un morceau d'un message) et la réception de celui-ci par l'entité distante ;

- temps maximum entre la réception d'un segment et l'émission de l'accusé de réception associé ;

- temps maximum d'attente d'un accusé de réception avant de tenter une retransmission ;

- nombre maximum de retransmissions ;

- temps maximum d'oisiveté de la connexion avant de décider de fermer la connexion ;

- temps maximum d'attente avant de transmettre des informations de contrôle de la fenêtre d'anticipation.

On notera que le contrôle d'erreurs prend une part très importante dans le protocole de transport vu les nombreux types d'erreurs qu'il faut détecter et traiter.

III. TCP : Protocole de transport d'Internet

III.1. TCP vs UDP

Dans le monde Internet, deux protocoles peuvent être utilisés :

- TCP (Transmission Control Protocol)
- UDP (User Datagram Protocol)

Il faut noter qu'en plus de TCP et UDP (très utilisés tous les deux dans l'Internet), un protocole à mi-chemin entre TCP et UDP a été proposé par l'IETF en 2006. Il s'agit de DCCP (Datagram Congestion Control Protocol). DCCP n'est pas encore déployé de manière significative et ne sera pas étudié dans ce cours.

a) Quelques principes de TCP [Postel 1981 – RFC 793]

- TCP est conçu pour la transmission fiable d'un flux de données continu (ceci conduit à numéroter les octets et non les segments). Pour assurer la transmission fiable de segment TCP, le protocole TCP utilise la notion d'acquittement et de retransmission le cas échéant. Ce fonctionnement est similaire à celui de la couche liaison de données, mais il est beaucoup plus complexe.

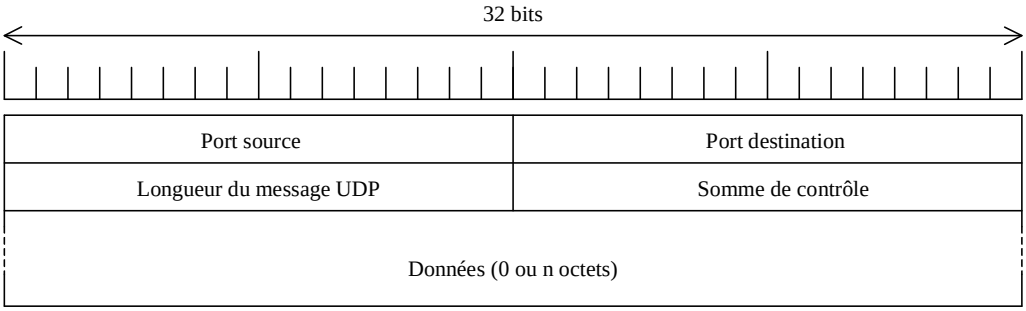
- TCP est orienté connexion.

- TCP limite la taille des segments à une valeur maximale de 64 K octets (dans la pratique on a souvent une valeur beaucoup plus petite, soit 1500 octets, ce qui permet d'éviter la fragmentation par le protocole IP, car la plupart des réseaux limitent la taille des paquets).

- Comme IP est non fiable, TCP doit disposer de tous les mécanismes nécessaires pour traiter les anomalies éventuelles (perte ou duplication de fragments de message, perte d'acquittements, arrivée dans le désordre des paquets, etc.)
- Le contrôle de flux est assuré grâce à un mécanisme de fenêtre d'anticipation.
- Protocole complexe et lourd en termes de temps de réponse.

b) Quelques principes de UDP [Postel 1980 – RFC 768]

- Protocole sans connexion et non fiable.
- Simple à mettre en œuvre (car il fait appel à un minimum de fonctions)
- Permet de gagner du temps, en évitant de passer par l'étape de connexion.
- Adapté aux applications de types client/serveur ou transactionnels (question-réponse).



En-tête de segment UDP.

III.2. Interface Utilisateur/TCP

L'interface d'utilisation de TCP (dite aussi interface socket) est définie par les primitives suivantes :

`OPEN``(local port, socket distante, mode Actif/passif, ....) --> nom local de connexion`
Primitive qui permet de demander l'ouverture d'une connexion. En mode `Actif`, on déclenche le dialogue avec l'entité distante pour ouvrir la connexion. En mode `Passif`, on attend que l'autre correspondant déclenche l'établissement de connexion.

`SEND``(nom local de connexion, adresse de buffer, nombre d'octets, indicateur d'urgence, timeout...)`
Primitive qui permet la demande de transmission d'un bloc d'octets.

`RECEIVE``(nom local de connexion, adresse de buffer, nombre d'octets) --> nombre d'octets, indicateur d'urgence`
Primitive qui permet de recevoir un ensemble d'octets (plus exactement de récupérer du buffer de réception un bloc d'octets). L'opération peut être bloquante ou non selon l'implantation et selon le nombre d'octets disponibles dans le buffer de réception de TCP.

`CLOSE``(nom local de connexion)`
Primitive de demande de fermeture de connexion.

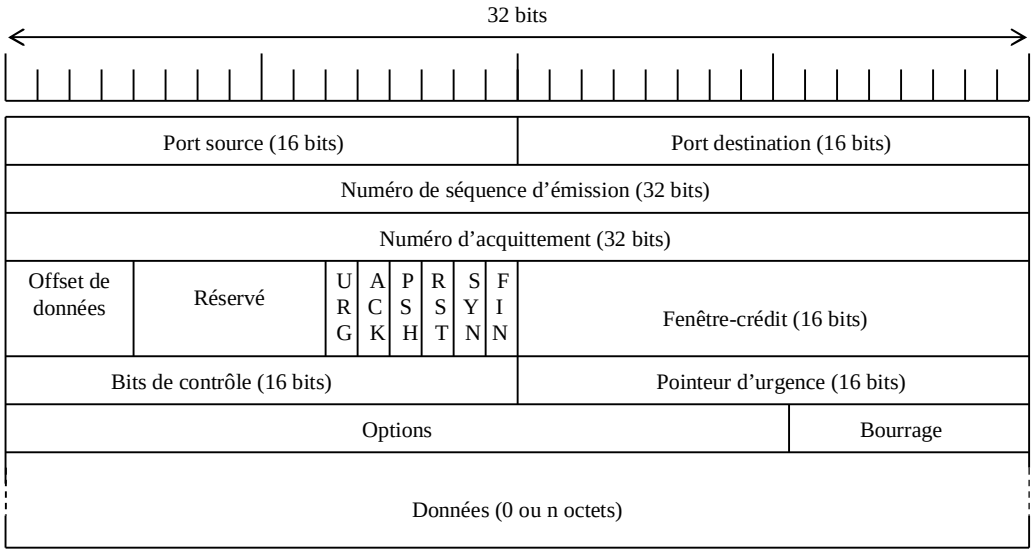
`STATUS``(nom local de connexion) --> Infos d'état`
Cette primitive dépend de l'implantation et permet d'obtenir des informations sur l'état d'une connexion : identificateur local de la socket, état de la connexion (établie ou non), fenêtre de réception, fenêtre de réception, nombre d'octets en attente d'acquittement, ...

`ABORT``(nom local de connexion)`
Primitive qui permet l'avortement d'une connexion avec suppression de toutes les données en cours d'émission ou de réception.

III.3. Format de segment

Les PDUs de données gérés par TCP sont appelés ***segments***. Un segment TCP peut transporter jusqu'à 65 515 octets de données (cette limite est due au format de paquet IP qui code la longueur de segment, y compris l'entête de segment, sur 16 bits).

Comme le montre la figure suivante, le format de segment TCP est constitué d'une partie fixe de 20 octets et d'une partie variable (optionnelle) qui contient des options et des données de bourrage éventuellement.



En-tête de segment TCP

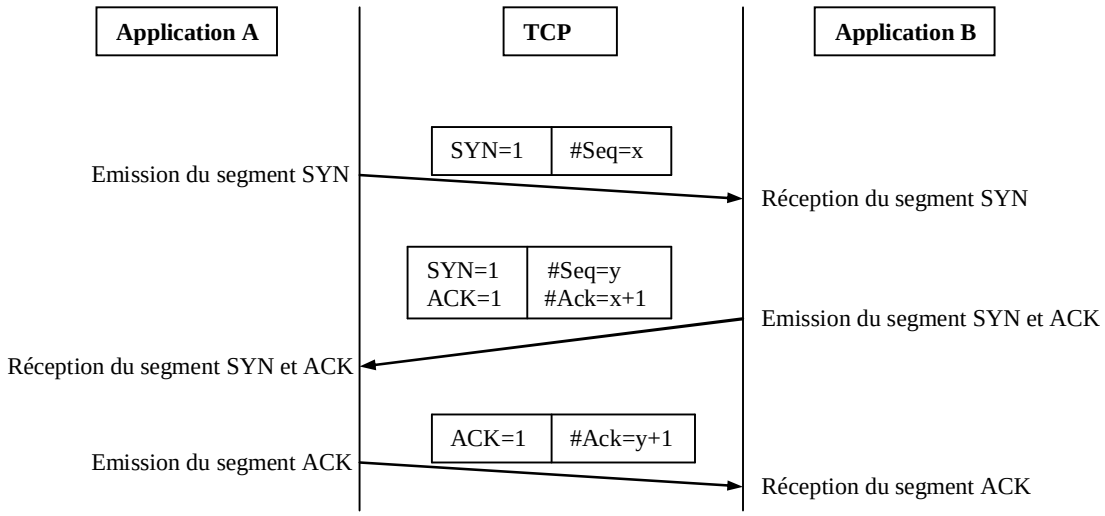
- *Port source* et *Port destination* : indiquent les ports utilisés par la couche Application.
- *Numéro de séquence d'émission* : numéro du premier octet de données dans le segment (sauf pour un segment avec *SYN*=1). Si le bit *SYN* est à 1 (c.-à-d. si le segment est une demande de connexion), le numéro de séquence signale au destinataire que le prochain segment de données qui sera émis commencera à partir de l'octet *Numéro de séquence d'émission + 1*.
- *Numéro d'acquittement* : indique le numéro du prochain octet attendu par le destinataire. Si dans le segment reçu *FIN*=1 et *Numéro de séquence d'émission* = x, alors le *Numéro d'acquittement* renvoyé est x+1 (x est interprété comme étant le numéro de l'octet *FIN* et que cet octet est acquitté).
- *Offset de données* : des options (de taille variable) peuvent être intégrées à l'entête et des données de bourrage peuvent être rajoutées pour rendre la longueur de l'entête multiple de 4 octets. L'Offset indique la position relative où commencent les données.
- *Fenêtre-crédit* : indique le nombre d'octets que le destinataire peut recevoir. La notion de crédit est utilisée pour gérer le contrôle de flux entre émetteur et récepteur. Si *Fenêtre-crédit* = F et que le segment contient un *Numéro d'acquittement* = V, alors le récepteur accepte de recevoir les octets numérotés de V à V + F -1.
- *Bits de contrôle* : séquence de contrôle (CRC) portant sur l'entête de segment.
- Flags (*URG*, *ACK*, *PSH*, *RST*, *SYN*, *FIN*)
    - *URG* = 1 si le segment contient des données urgentes et *URG* = 0 sinon.
    - *ACK* = 1 indique que le numéro d'acquittement est valide et il peut être pris en compte par le récepteur. *ACK* = 0 si l'accusé de réception est non valide.
    - *PSH* = 1 indique que les données doivent être remises à l'application dès leur arrivée et de ne pas les stocker dans une file d'attente.
    - *RST* = 1 pour demander la réinitialisation d'une connexion TCP.
    - *SYN* = 1 et *ACK* = 0 servent à demander l'établissement de connexion TCP.
    - *SYN* = 1 et *ACK* = 1 servent à accepter une demande de connexion TCP.
    - *FIN* = 1 indique que l'émetteur n'a plus de données à émettre et demande de rompre la connexion de son côté.
- Pointeur d'urgence : indique l'emplacement (numéro d'octet) des données urgentes dans un segment. Il es valable uniquement si le bit URG=1.

- *Options* (taille variable) : options nécessitant des traitements particuliers.
- *Données (*de taille variable) : données provenant de la couche Application.

III.4. Etablissement de connexion TCP

La communication de données avec TCP nécessite l'établissement de connexion. L'établissement de connexion peut être déclenché par un des deux correspondants ou les deux simultanément. L'établissement de connexion TCP se fait en trois étapes :

- L'émetteur envoie un segment avec un bit SYN=1 et un numéro de séquence *x*.
- Le récepteur renvoie un segment avec bit SYN=1, un numéro de séquence = *y*, bit ACK=1 et un numéro d'accusé de réception = *x +1*.
- L'émetteur envoie un segment avec bit ACK=1 et un numéro d'accusé de réception = *y +1*.



Etablissement de connexion TCP

Pendant les trois étapes précédentes, les deux correspondants se mettent d'accord sur les numéros de séquence d'émission et sur les numéros d'acquittement. Comme les deux correspondants peuvent émettre des données dans les deux sens, chacun spécifie son numéro de séquence et notifie à son correspondant le numéro à partir duquel il accepte de recevoir. Les numéros de séquence dans les deux directions commencent normalement à la même valeur, mais ce n'est pas obligatoire. Si un des deux correspondants ne transmet pas de données, son numéro de séquence en émission n'a pas d'importance.

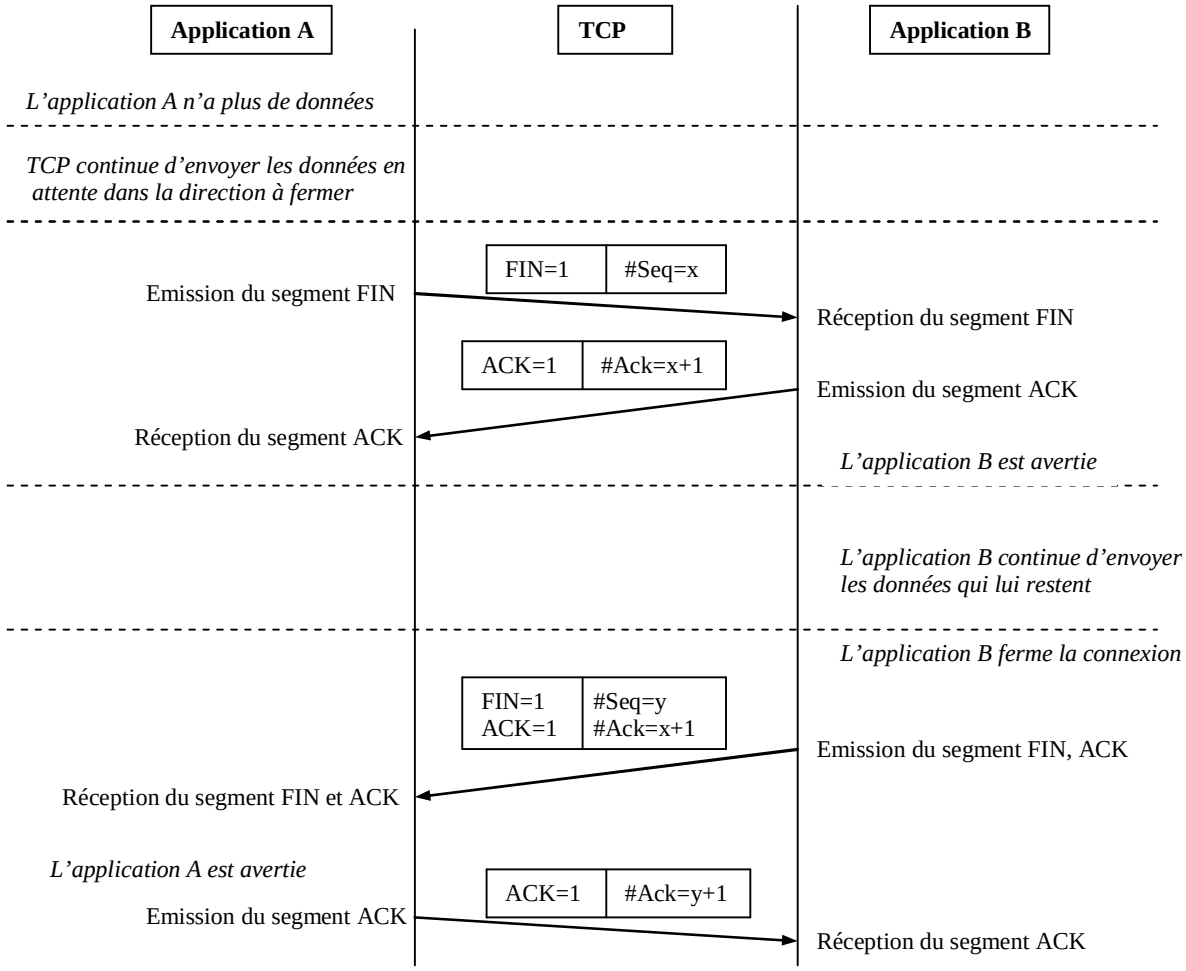
Sur la figure précédente, l'ordinateur *A* indique un numéro de séquence égal à *x*. L'ordinateur *B*, qui accepte la connexion, indique un numéro d'acquittement égal à *x+1* (pour signaler à l'ordinateur *A* qu'il est prêt à revoir des données à partir de l'octet numéroté *x+1*). L'ordinateur *A* enverra ses données à partir du numéro *x+1*. Le même principe est utilisable pour définir le numéro *y*.

III.5. Libération de connexion TCP

A la fin de transmission des données, l'un des deux correspondants doit fermer la connexion (à noter que TCP gère aussi le cas où les deux correspondants demandent simultanément la fermeture de connexion). Comme les connexions TCP sont bidirectionnelles, il faut faire attention lorsqu'on veut fermer une connexion pour ne pas perdre les données en cours de transmission ou d'acquittement.

Lorsque, sur un site, l'utilisateur informe TCP qu'il n'a plus de données à émettre, TCP libère la connexion dans un seul sens (pour ne pas perdre les données qui viennent dans l'autre sens). Pour ce faire, TCP émetteur achève la transmission des données en attente, attend que le récepteur accuse réception, puis envoie un segment avec bit FIN=1. Le TCP récepteur acquitte le segment FIN et informe son utilisateur local qu'il n'y a plus de données à recevoir. Pendant ce temps, les données dans l'autre sens de la connexion continuent à circuler normalement. Lorsque la connexion est fermée dans les deux sens, TCP détruit toutes les informations liées à la connexion.

La figure suivante illustre les étapes de fermeture de connexion.



Fermeture de connexion TCP

III.6. Gestion de la fiabilité

Chaque segment émis par la couche TCP est conservé en mémoire jusqu'à ce que l'acquittement correspondant soit reçu par l'émetteur. Pour détecter la perte de segment (de données ou d'acquittement), un temporisateur est armé au moment de la transmission du segment. Si le temporisateur se déclenche, l'émetteur retransmet le segment concerné.

Les segments TCP sont numérotés en séquence à l'émission et il en est de même pour les acquittements. Ainsi, si l'émetteur a transmis plusieurs segments avant de recevoir un acquittement, à partir du numéro contenu dans le dernier acquittement reçu, il sait quel segment est concerné.

***Remarque** : les numéros de séquence (placés par l'émetteur) et les numéros d'acquittement (placés par le récepteur) donnent une position en octets par rapport au début du flux de l'émetteur. Par exemple, un acquittement contenant le nombre 1021 signifie que le récepteur est prêt à recevoir le segment commençant à partir de l'octet 1021. Il ne faut donc pas confondre cette numération avec celle utilisée par la couche liaison de données qui compte en nombre de trames émises ou reçues.*

Comme la couche liaison de données, la couche TCP utilise le principe de la fenêtre coulissante pour émettre une suite de segments avant d'attendre un acquittement. Lorsqu'un acquittement est reçu, la fenêtre d'émission glisse vers la droite d'un nombre d'octets en fonction du numéro d'accusé de réception reçu.

Dans les réseaux filaires actuels (attention ce qui suit n'est absolument pas le cas des réseaux non filaires), le taux d'erreurs est très faible (surtout sur les fibres optiques). On considère alors (à juste titre) que la perte d'un segment est due à une congestion de réseau et non à une erreur de transmission. Les algorithmes de retransmission et de contrôle de congestion dans Internet sont fondés sur cette hypothèse.

Algorithme de retransmission adaptatif

TCP est un protocole destiné à fonctionner au-dessus d'un protocole de réseau éventuellement non fiable (généralement ce protocole c'est IP). Le nombre de réseaux traversés par un segment est généralement inconnu par la source. Par conséquent, le temps de transfert d'un segment peut varier considérablement d'un segment à l'autre. TCP doit donc tenir compte de ces variations pour déterminer les valeurs de temporisateurs liés à l'attente des acquittements. Il faut rappeler que cette notion de temporisateur est utilisée aussi par la couche liaison de données, mais la valeur d'armement du temporisateur est fixe, car on peut facilement estimer le délai de traverser d'un réseau local.

Pour prendre en compte la variation des délais d'acheminement de paquets par le réseau sous-jacent, TCP utilise un algorithme de retransmission adaptatif qui consiste à surveiller le comportement du réseau et en déduire les valeurs de temporisation. L'algorithme le plus répandu pour calculer les valeurs de temporisation de retransmission est celui de Jacobson (1988) dont le principe est résumé ci-dessous.

Pour obtenir les informations nécessaires à l'algorithme de retransmission, TCP enregistre l'instant auquel chaque segment de données est émis et l'instant de réception de l'acquittement correspondant. Ces deux instants permettent de déterminer une valeur (un échantillon) de délai d'aller-retour (ou *RTT : Round Trip Time*). Pour estimer avec précision le délai d'aller-retour, plusieurs échantillons sont nécessaires. Ainsi, TCP calcule et conserve les deux dernières valeurs (la dernière valeur notée *Nouveau_RTT* et celle qui la précède notée *RTTestimé*) de délai d'aller-retour pour les différents segments transmis. Le temps de RTT moyen est calculé à tout instant par la formule suivante :

$$RTTestimé = (\alpha * RTTestimé) + (1- \alpha) * Nouveau\_RTT)$$

Le coefficient α ($0 \leq \alpha < 1$) est choisi de manière protéger l'ancienne valeur moyenne contre une influence trop forte de la nouvelle, c'est-à-dire ne pas réagir brutalement aux pics de trafic. Le choix d'une valeur proche de 1 rend la moyenne insensible aux variations brutales de trafic.

Lorsqu'un segment est émis, TCP calcule une valeur de temporisation *RTO* ("Retransmission TimeOut"), pour armer le temporisateur, à partir du *RTTestimé* de la manière suivante :

$$RTO = Min\{BorneSuppRTO, Max\{BorneInfRTO, RTTestimé * \beta\}\}$$

BorneSuppRTO désigne la plus grande valeur que peut prendre la temporisation (par exemple 1 minute). *BorneInfRTO* c'est la plus petite valeur que peut prendre la temporisation (par exemple 10 ms). Le coefficient β ($\beta > 1$) est utilisé pour mieux tenir compte du délai d'aller-retour. Si β est proche de 1, l'attente de l'acquittement dure l'équivalent du dernier RTT et dans ce cas on détecte rapidement la perte du segment

données ou de son acquittement. Si β est nettement plus grand que 1 (par exemple $\beta = 2$), cela permet éventuellement d'être un peu plus 'patient' pour tenir compte d'une éventuelle hausse du délai d'aller-retour, mais avec cette solution on peut perdre du temps avant retransmettre. Le choix d'une valeur optimale de β est difficile ; il dépend de beaucoup de facteurs et a fait et fait encore l'objet de nombreux travaux de recherche et d'expérimentation.

Problème de calcul du temps de boucle (aller-retour)

Le calcul décrit précédemment serait simple si TCP utilise un acquittement par segment de données. Rappelez-vous que TCP utilise une technique d'acquittement cumulative : un acquittement concerne un nombre d'octets bien reçus et non un segment de données particulier. Ainsi, lorsque TCP retransmet un segment de données et puis il reçoit un acquittement, se pose alors la question : est-ce que cet acquittement (éventuellement tardif) correspond au segment de données initial ou bien est-ce qu'il correspond au segment de données retransmis ? TCP n'a aucun moins de distinguer les deux cas. Associer l'acquittement au segment de données le plus ancien peut provoquer une augmentation significative du délai d'aller-retour surtout si plusieurs tentatives de retransmissions ont été effectuées pour le même segment initial. Associer l'accusé de réception au segment de données (retransmis) le plus récent peut aussi être incorrect, car on peut recevoir l'acquittement du segment initial juste après l'avoir retransmis (dans ce cas, on attribue l'acquittement au segment retransmis et non au segment initial) et trouver un délai d'aller-retour très petit. Par conséquent, ni l'association de l'acquittement à la transmission du segment de données initial, ni l'association de l'acquittement à la transmission du dernier segment de données retransmis ne peuvent fournir une bonne estimation du délai d'aller-retour. Par conséquent, TCP ne doit pas mettre à jour la valeur du *RTTestimé* quand il s'agit de segments retransmis ; cette idée est connue sous le nom d'algorithme de Karn. Pour mieux prendre en compte les situations générées par les pertes de segments de données et des acquittements, une des solutions c'est de combiner les temporisations de retransmission avec une stratégie d'augmentation des temporisations. La *stratégie d'augmentation des temporisations* calcule une valeur initiale de temporisation *RTO*, comme vu précédemment, et en cas de retransmission, on augmente la valeur de *RTO*. Les implantations de TCP utilisent différentes manières d'augmenter la valeur de *RTO*. La plupart des implantations choisissent un facteur multiplicatif λ qui vaut généralement 2 :

$$Valeur\ de\ Tempo\ augmentée = RTO * \lambda$$

Il existe d'autres techniques plus récentes, mais plus complexes, car elles exigent plus de paramètres et de calcul, pour changer dynamiquement la valeur de temporisation.

III.7. Contrôle de congestion *(principe "Additive Increase Multiplicative Decrease")*

1. Principe général d'utilisation de fenêtre

Dans TCP, les numéros de segments sont codés sur 32 bits. Si aucune précaution n'est prise, un émetteur peut envoyer en rafale jusqu'à 2^{32} octets (il envoie par exemple un fichier très volumineux) avant de se bloquer. Un tel comportement présente plusieurs inconvénients :

1) S'il y a des pertes ou des erreurs, l'émetteur risque d'effectuer un nombre élevé de retransmissions (d'où une perte de temps et une mauvaise utilisation du réseau) ;
2) Le récepteur risque d'être saturé, ce qui le conduit à rejeter des segments (car il n'a plus d'espace mémoire pour les stocker) ;
3) Le réseau traversé risque d'être saturé par le comportement en rafale de l'émetteur.

Pour éviter ces problèmes, le protocole TCP assure une fonction de contrôle de congestion. Ainsi, on "responsabilise" la source de données afin d'éviter la congestion de réseau.

Le récepteur annonce son crédit dans le champ *Fenêtre-crédit* des segments qu'il envoie pour indiquer à son correspondant le nombre d'octets qu'il est prêt à recevoir. On dit *Fenêtre* ou *Crédit* ou encore *Fenêtre de réception*. Quand la *Fenêtre* est nulle, l'émetteur ne doit pas plus envoyer de segments sauf dans deux cas particuliers :

1) Il peut envoyer des données urgentes, pour permettre l'exécution de certaines opérations importantes sur le site destinataire (par exemple, arrêter une application distante).
2) Il peut envoyer un segment contenant un seul octet pour obliger le récepteur à ré-annoncer le prochain octet attendu ainsi que la taille de la fenêtre. Ce mécanisme permet d'éviter les situations d'interblocage en cas de perte d'annonce de fenêtre.

La taille maximum de segments que la source peut émettre avant d'attendre un acquittement est appelé *Fenêtre de congestion* (*cwd* : congestion window). Au lieu d'avoir une valeur fixe pour *cwd* (comme on le fait dans le cas de la couche liaison de données), le protocole TCP gère de manière sophistiquée la taille de la fenêtre de congestion. De manière simplifiée, cette gestion consiste à émettre beaucoup de segments de données quand la source juge que le réseau est sous-chargé et, au contraire, à réduire son rythme de transmission quand elle juge que le réseau est surchargé (voire saturé). La manière dont on ajuste dynamiquement la taille de la fenêtre de congestion a donné lieu à différentes implantations de TCP (Reno, New Reno, Vegas, Tahoe…).

Pour réagir dynamiquement à la charge du réseau, la couche TCP utilise des indicateurs de mesure.

Il faut bien noter que dans beaucoup de situations, les retransmissions aggravent la congestion au lieu de la résorber. En effet, la congestion de certains routeurs conduit à la perte de segments (car ils sont rejetés par des routeurs saturés). Ensuite, les nœuds d'extrémité qui ont perdu leurs segments retransmettent, ce qui augmente la charge du réseau et donc à plus de pertes et ainsi de suite jusqu'à ce que le réseau se bloque complètement (congestion totale). Pour éviter de tomber dans de telles situations, TCP doit réduire le débit de transmission en cas de congestion constatée. Il faut rappeler que les routeurs utilisent les paquets ICMP pour avertir certains ordinateurs afin qu'ils réduisent leur débit. Donc l'initiative de réduction de débit peut être entreprise sur la base des RTT observés par la source ou suite à la réception de paquet ICMP.

La figure 1 montre les informations mémorisées par chaque site pour suivre l'évolution de l'utilisation des fenêtres chez l'émetteur (qui envoie les données) et chez le récepteur (qui envoie les acquittements).

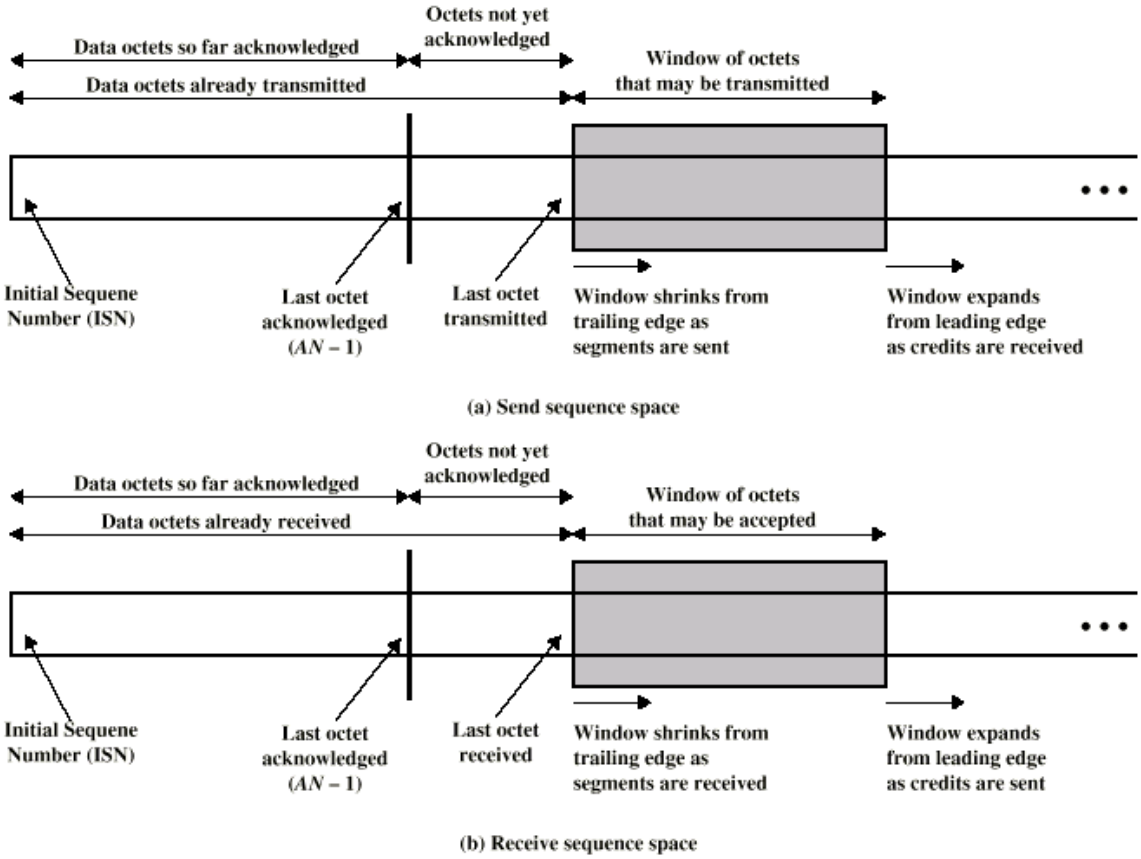
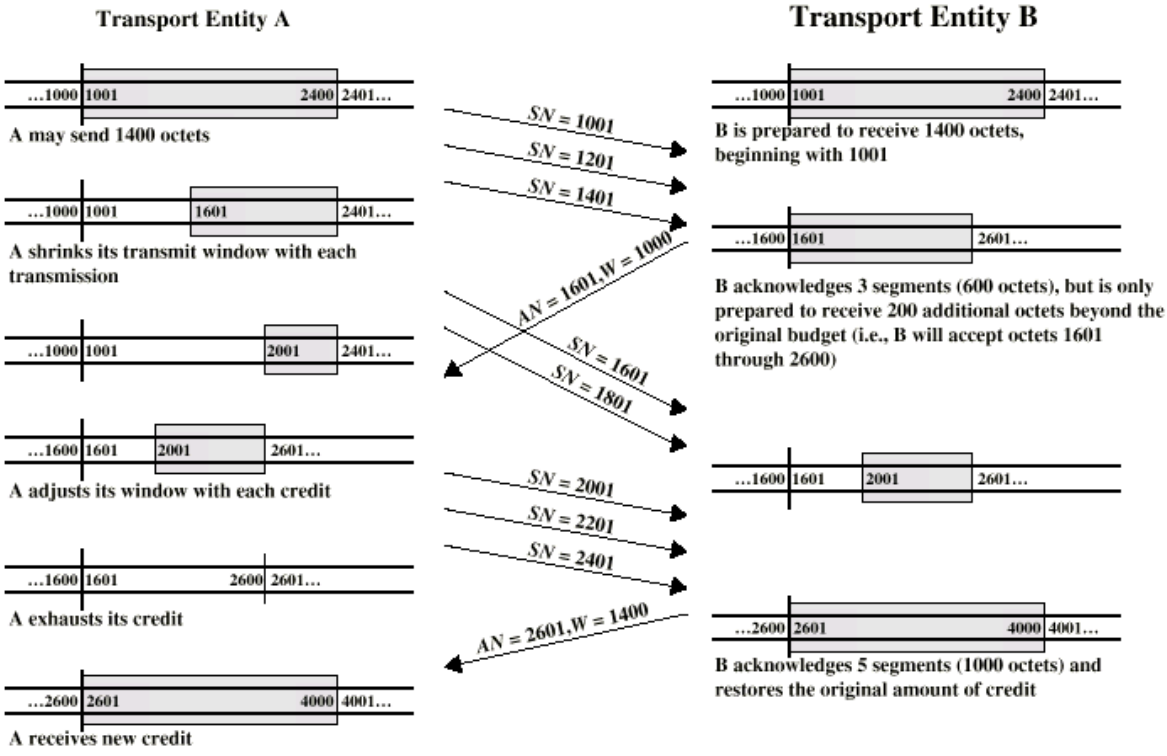


Figure 1. Principe d'utilisation de fenêtres

La figure 2 montre un exemple d'échange de données et acquittement avec changement de la taille de la *Fenêtre*.



SN: Numéro de séquence à l'émission AN : Numéro d'acquittement W : Fenêtre

Figure 2. Exemple d'échange de données et acquittement.

2. Principe "Additive increase – Multiplicative decrease"

Le standard TCP recommande l'utilisation combinée de deux techniques : ***démarrage lent*** et la ***diminution dichotomique***. Ces deux techniques combinées sont souvent désignées par *Additive increase – Multiplicative decrease*.

A tout instant, la source peut au maximum un nombre d'octets déterminé par la valeur de *Fenêtre de crédit* incluse dans l'acquittement. Pour contrôler la congestion, TCP gère un seuil appelé *Fenêtre de congestion*. A tout moment, TCP fonctionne comme si :

Fenêtre autorisée = minimum (Crédit restant, Fenêtre de congestion)
Crédit restant = Fenêtre de crédit – quantité d'octets envoyés et non encore acquittée

Dans les situations où il n'y a pas de congestion sur une connexion, la valeur de *Fenêtre de congestion* est égale à celle de Crédit restant. Dans ce cas, le rythme de transmission est limité uniquement par les capacités du récepteur.

Pour réduire la *Fenêtre de congestion*, TCP applique la technique de diminution dichotomique (dite aussi 'multiplicative decrease'), en considérant que la plupart des pertes de segments sont dues à des routeurs saturés. Cette technique fonctionne de la manière suivante :

> *En cas de perte d'un segment de données, réduire la fenêtre de congestion de moitié jusqu'à atteindre le minimum, c'est-à-dire un seul segment à la fois. Pour les segments qui restent dans la fenêtre d'émission, augmenter la temporisation de manière exponentielle (en la multipliant à chaque fois par deux).*

Lorsque la situation de congestion se résorbe, TCP doit revenir à un état de fonctionnement normal et augmenter son débit. Pour éviter un changement brutal du débit de transmission, TCP utilise la technique dite de *Démarrage lent* dont le principe est le suivant :

> *Au début du trafic sur une connexion ou lorsque que le trafic reprend après une phase de congestion, commencer une fenêtre de congestion limitée à un seul segment et incrémenter la fenêtre de congestion d'un segment à la fois chaque fois qu'un acquittement est reçu.*

Avec un démarrage lent, la source peut atteindre rapidement un rythme de croisière (c'est-à-dire où elle peut transmettre un maximum de segments) si les acquittements lui parviennent rapidement.

La figure 3 montre un exemple d'évolution du débit d'une connexion en fonction des situations de non-réception d'acquittement. Le débit observé fonctionne en dents de scie.

3. Syndrome de la fenêtre stupide

C'est une situation où les tampons d'émission ou de réception sont vidés à raison d'un octet (ou d'un petit nombre d'octets) par segment. Dans ce cas, du côté émetteur, TCP peut envoyer des segments qui ne contiennent qu'un octet de données par segment et du côté récepteur, TCP peut envoyer un acquittement pour chaque octet livré au destinataire. Comme chaque segment TCP est encapsulé dans un paquet IP, il y a un gaspillage des ressources du réseau. Les implantations de TCP doivent éviter de tomber dans de telles situations. Pour cela, TCP émetteur essaie de retarder le plus possible la transmission d'un segment tant qu'il n'a pas un nombre suffisant d'octets à émettre et le récepteur retarde l'envoi d'acquittement tant qu'une quantité suffisante d'octets n'a pas encore été délivrée au destinataire.

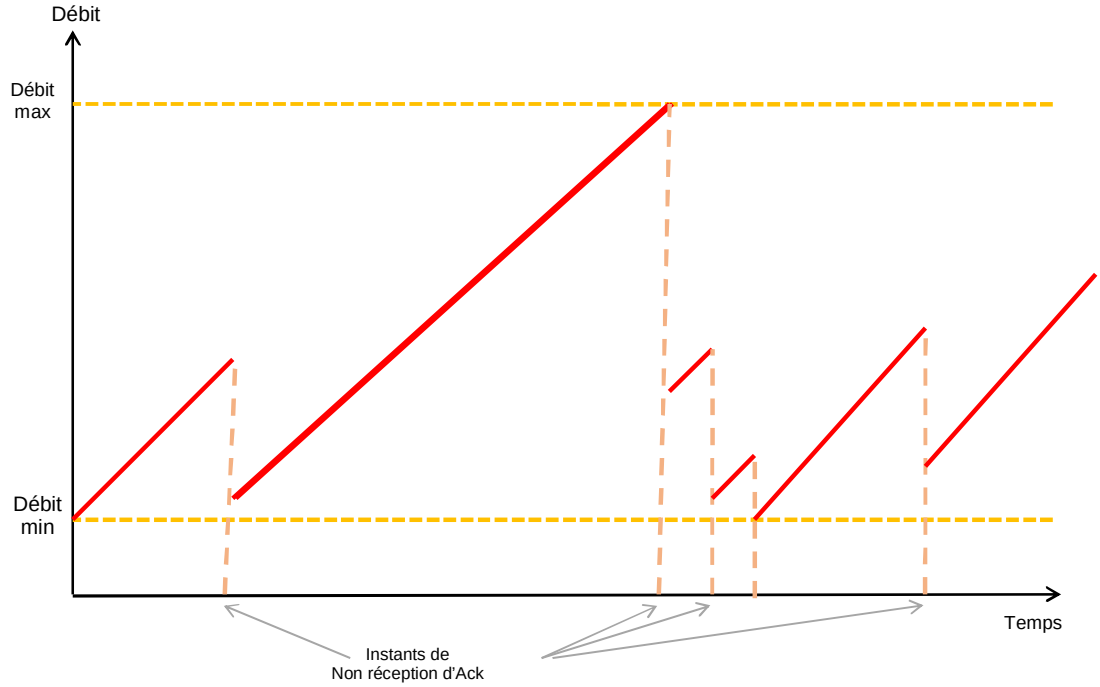


Figure 3. Evolution du débit d'une connexion en fonction à cause de la prte des acquittements.

4. Principe de retardement de segment

Beaucoup d'implémentations de TCP utilisent le principe de retardement des accusés de réception et les indications de *Fenêtre-crédit* pendant un certain délai (500 ms en général) dans l'espoir d'avoir plus de données à transmettre sur lesquelles on pourra greffer les accusés de réception et les informations de *Fenêtre-crédit* (qui seront alors transportés sans coût supplémentaire, c'est-à-dire sans utiliser un segment exprès pour ces informations). Ce principe permet d'améliorer l'utilisation de la bande passante, mais ce n'est pas toujours vrai.

5. Algorithme de Nagle

Cet algorithme fonctionne de la manière suivante : quand une application génère les données octet par octet (par exemple, une application où on saisit les caractères tapés par une personne), on envoie le premier octet et on accumule les autres jusqu'à ce que l'acquittement de l'octet envoyé soit reçu. Puis on envoie les octets accumulés dans un seul segment, ensuite on accumule les octets en attendant l'acquittement du segment envoyé. Et ainsi de suite... Cependant, pour plus d'efficacité si l'application génère des octets avec un rythme élevé, l'algorithme prévoit d'envoyer un segment s'il y a eu assez de données pour remplir la moitié de la fenêtre de crédit ou si les données accumulées ont atteint la taille maximale de segment.

L'algorithme de Nagle (1984) est disponible sur beaucoup d'implantations de TCP. Il peut être activé ou désactivé selon les spécificités des applications.

III.8. TCP et le monde sans fil

Si on respecte à la lettre le principe des couches (ISO ou TCP/IP), le fonctionnement d'une couche *N* doit être indépendant et transparent à la couche *N+1*. Ceci est vrai lorsque l'on ne considère pas les problèmes de performances. En effet, si une couche *N+1* veut avoir des traitements et des temps de réponse sous une certaine limite, elle doit connaître certaines caractéristiques de la couche *N*. Ainsi, certains paramètres de fonctionnement et algorithmes sont sélectionnés en fonction des spécificités du réseau sous-jacent.

Dans le domaine des réseaux sans fil, le taux d'erreurs est très élevé. La perte d'un segment est rarement due à la congestion d'un noeud intermédiaire, mais le plus souvent à une erreur de transmission. En cas de perte d'un segment si on attend beaucoup de temps avant de retransmettre, on ne va pas augmenter les chances de transmissions.

Pour prendre en compte les spécificités des réseaux non filaires, des extensions et modifications de TCP ont été proposées.

Exercices

Exercice 1

On souhaite étudier la modélisation de TCP à l'aide d'automates à états finis.

Rappel : Nous signalons que les automates à états finis constituent un des modèles les plus utilisés pour modéliser et ensuite analyser les protocoles de communication (par exemple, vérifier qu'il n'y a pas d'interblocage).

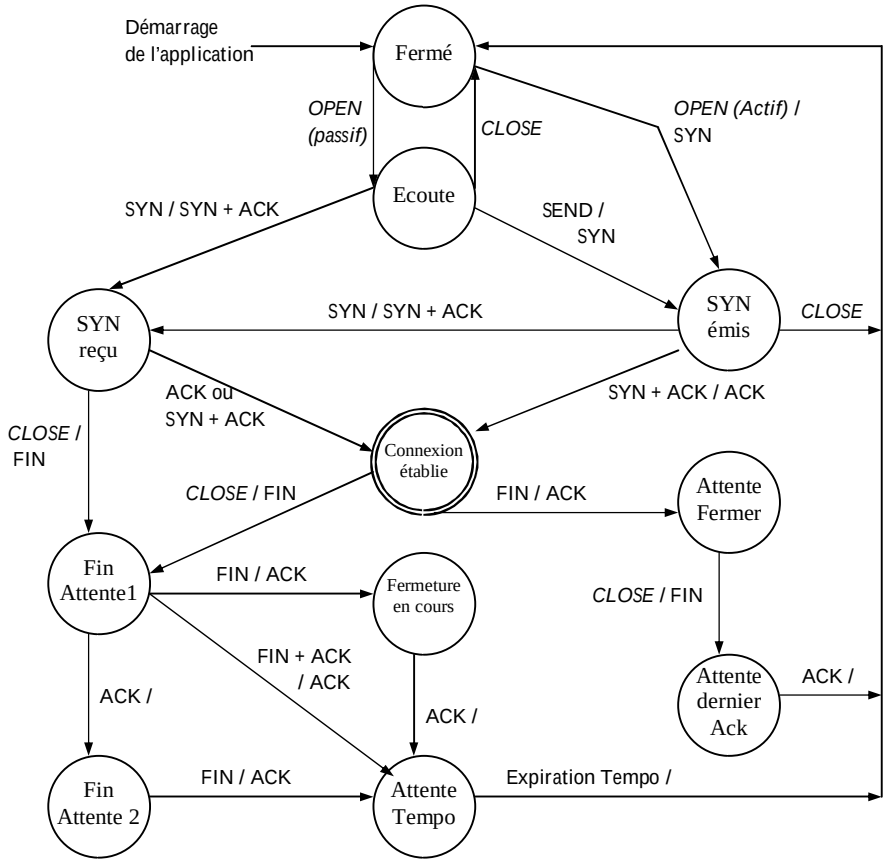
Un automate à états finis est composé d'états (désignés par des cercles à l'intérieur desquels on marque les noms des états) et de transitions entre états matérialisées par des flèches. Sur une transition on indique une étiquette de la forme Evt / Act qui signifie que sur occurrence de l'événement Evt, on exécute une série d'actions Act ensuite on passe de l'état source de la transition à l'état destination. Dans le cas de la modélisation de protocole, un événement peut être un appel de service (exemple ouvrir ou fermer connexion), la réception d'un segment (ACK, FIN...) ou un déclenchement de temporisateur. Une action peut être l'émission d'un segment ou l'exécution d'une opération (armement ou annulation de temporisateur, incrémentation d'un compteur...). Si le champ Act est vide, cela signifie qu'aucune action (significative pour la modélisation) n'est exécutée.

L'automate à états finis modélisant les phases d'établissement de connexion et de fermeture de connexion TCP est donné par la figure suivante. Les états de l'automate sont les suivants :

- Fermé : représente un état fictif (connexion inexistante).
- Ecoute : attente de demande de connexion déclenchée par le correspondant.
- SYN émis : un segment SYN a été envoyé et on attend la confirmation de connexion par le correspondant.
- SYN reçu : attente de confirmation de connexion après avoir reçu et émis une demande de connexion.
- Connexion établie : représente une connexion ouverte sur laquelle des données peuvent être échangées.
- Fin Attente 1 : représente l'attente de requête de fermeture de connexion du TCP distant ou l'attente de réception de l'acquittement concernant la demande de fermeture précédemment émise.
- Fin Attente 2 : représente l'attente de demande de fermeture de connexion par le TCP distant.
- Attente Fermer : représente l'attente de la demande de fermeture par l'utilisateur local.
- Fermeture en cours : représente l'attente de l'acquittement (en provenance du TCP distant) de la demande de fermeture par l'utilisateur local.
- Attente Dernier ACK : représente l'attente de l'acquittement de la demande de terminaison de connexion (qui inclut aussi l'acquittement des données en attente).
- Attente Tempo : attente d'un temps suffisant pour être sûr que le TCP distant reçoit bien l'acquittement pour sa demande de terminaison de connexion. Le temps d'attente est généralement fixé à deux fois MSL (Maximum Segment Lifetime) qui fixe le temps maximum de vie d'un segment TCP dans le réseau Internet (il peut être fixé à 30 secondes par exemple).

En suivant l'automate de la figure ci-dessous, le logiciel TCP démarre, à chaque extrémité, dans l'état Fermé. Chaque programme utilisateur qui veut utiliser TCP doit émettre une commande Open(actif) s'il veut initier l'établissement de connexion ou Open(passif) s'il veut se mettre à l'écoute de son correspondant.

Une requête locale Open(actif) provoque l'émission d'un segment SYN. Ensuite, la machine TCP se met dans l'état SYN Emis. Dans cet état, TCP local reçoit un segment avec bits SYN et ACK positionnés (qui indiquent que son correspondant est d'accord pour établir la connexion), il émet un segment avec bit ACK positionné pour confirmer à son correspondant la connexion et passe dans l'état Connexion établie. Différents enchaînements sont possibles à partir de cet état.



SNY + ACK : signifie un segment avec les bits SYN et ACK positionnés à 1
FIN + ACK : signifie un segment avec les bits FIN et ACK positionnés à 1

Figure Fonctionnement simplifié de connexion TCP modélisé à l'aide d'automate à états finis (sans temporisation, ni réinitialisation)

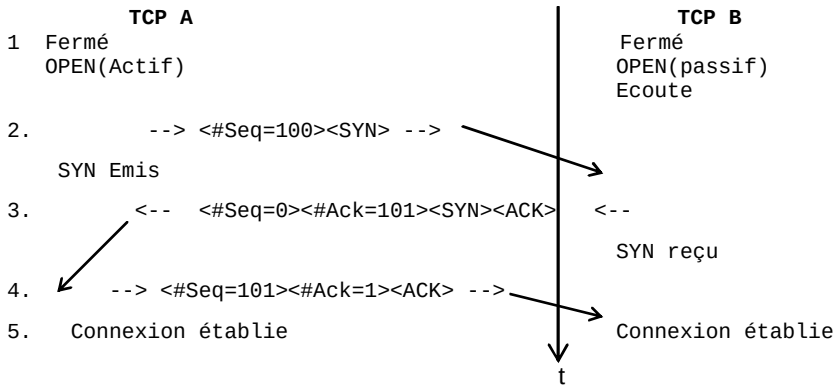
Question 1

En reprenant l'automate précédent, monter parmi les 5 enchaînements suivants quels sont ceux qui sont corrects et ceux qui nécessitent des modifications de l'automate.

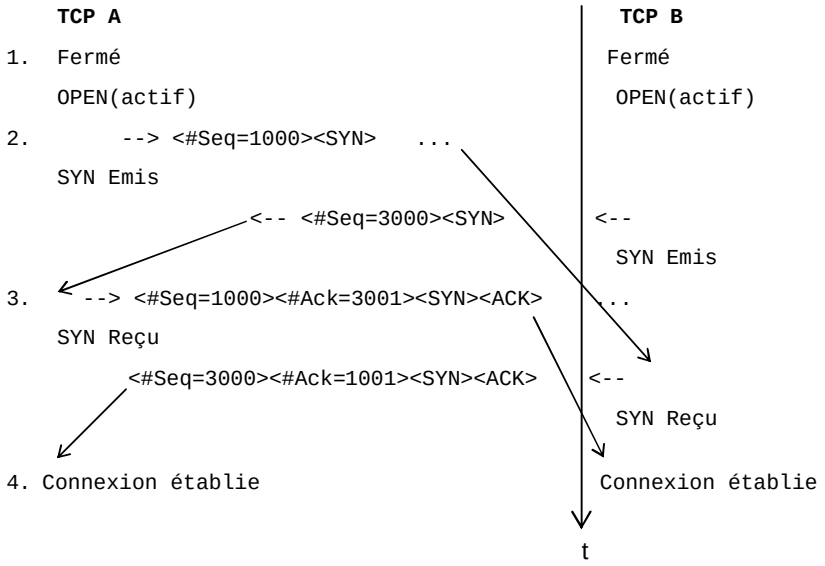
On utilise les notations suivantes :

<#Seq=x> : le numéro de séquence véhiculé par le segment est égal à *x*.
<#Ack=x> : le numéro d'acquittement véhiculé par le segment est égal à x.
<SYN> : le bit SYN du segment est positionné à 1.
<ACK> : le bit ACK du segment est positionné à 1.
<RST> : le bit RST du segment est positionné à 1.
<FIN> : le bit FIN du segment est positionné à 1.
--> | <-- : émission/réception de segment
... : signifie un segment en cours d'acheminement (retardé éventuellement)
flèche oblique : sa pointe indique à quel moment le segment est reçu

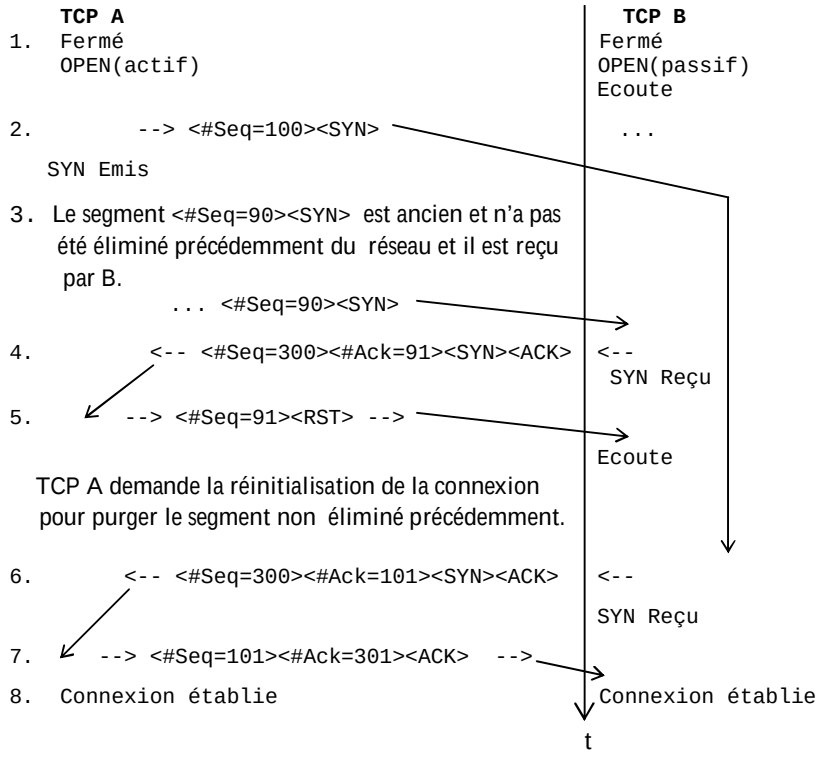
Enchaînement 1 : Etablissement de connexion par une seule entité TCP



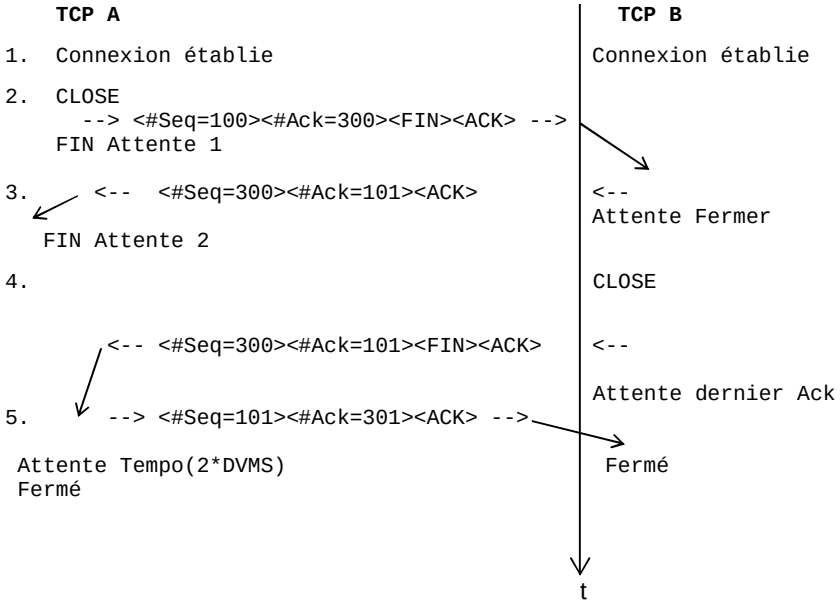
Enchaînement 2 : Demandes de connexion simultanées



Enchaînement 3 : Recouvrement en cas de segment SYN *qui restait dans le réseau*

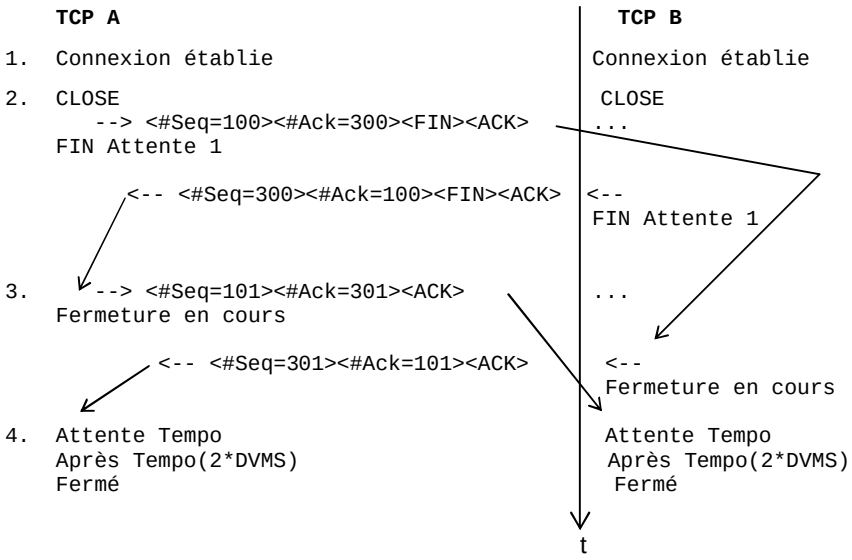


Enchaînement 4 : Séquence de fermeture normale initiée par un seul correspondant



DVMS : durée de vie maximum de segment TCP

Enchaînement 5 : Demandes de déconnexion simultanées



DVMS : durée de vie maximum de segment TCP

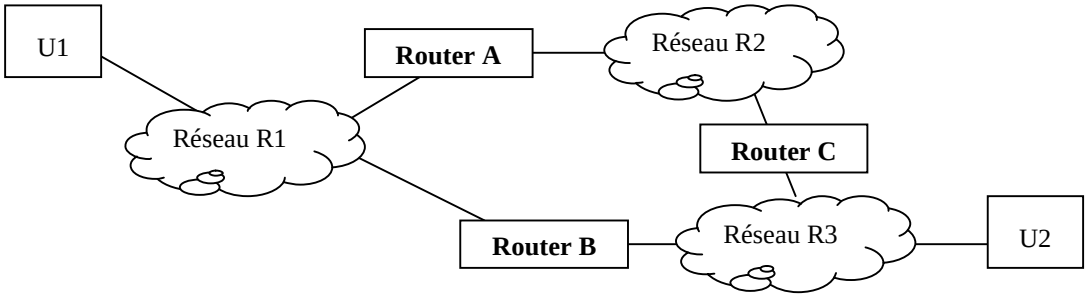
Question 2 :
2.1. Est-ce TCP décrit par l'automate précédent peut se trouver en situation d'interblocage ? Si oui, que faut-il rajouter pour éviter les situations d'interblocage ?
2.2. En supposant qu'il n'y a jamais de perte de segments durant les phases d'établissement et de fermeture de connexion, supprimer des transitions pour que deux machines TCP peuvent se trouver en interblocage.

Question 3 :
3.1. Modéliser le comportement TCP émetteur (c'est-à-dire qu'on ne s'intéresse qu'à l'émission de données et la réception des acquittements correspondants). On suppose ici que le contrôle de flux est statique (les tailles des fenêtres sont fixées à 10 k octets) et qu'il n'y a jamais de perte ni de segments de données ni d'acquittement.

3.2. Modéliser le comportement de TCP émetteur-récepteur (c'est à dire qu'une machine TCP émet des segments de données et en profite pour accuser réception à chaque fois où cela est possible). Mêmes hypothèses que précédemment.

Exercice 2

On s'intéresse à un utilisateur U1 de la couche TCP qui émet un segment de données de 3 K octets vers son correspondant U2. Ce segment traverse des routeurs interconnectés via 3 réseaux Ethernet R1, R2 et R3. Les MTU (Maximum Transfer Unit) des trois réseaux sont respectivement 1 k, 1.5 k et 0.5 k octets. Expliquer le principe de fragmentation et réassemblage du segment émis par U1.



Exercice 3

TCP assure le contrôle de congestion selon le principe "Additive Increase – Multiplicative decrease".
- en partant d'un réseau en sous-charge, au bout de combien de transmissions, une source peut-elle atteindre son rythme maximal ?
- en partant d'un réseau surchargé, au bout de combien de transmissions, une source se retrouve t-elle à son rythme de transmission le plus bas ?

On suppose que les segments ont une taille fixe Z.

Exercice 4

Q1. On suppose qu'à un instant *t* le RTT estimé est égal à 12 ms. Ensuite, trois accusés de réception sont reçus aux instants *t+10, t+80* et *t+100*.
Quelle est la nouvelle valeur du RTT estimée après le dernier Ack si on prend $\alpha = 0.9$?
Quelle est la nouvelle valeur du RTT estimée après le dernier Ack si on prend $\alpha = 0.1$?
Quelle conclusion peut-on tirer de la variation de α ?

Q2. Dans un réseau où un segment de données ne peut pas transporter plus de 128 octets, le temps minimum de traversée du réseau est de 25 ms et le numéro de séquence est sur 8 bits, quel est le débit maximal par connexion ?

Exercice 5

On se place dans le cadre d'une couche transport qui a besoin d'établir des connexions multipoint fonctionnant de la manière suivante :
- une connexion est établie entre une station émettrice de données et un groupe de stations réceptrices de données,
- une connexion multipoint est refusée si une ou plusieurs stations membres du groupe ne peuvent pas se connecter,
- tout paquet émis doit être reçu par toutes les stations faisant partie du groupe.

La couche réseau permet à la couche transport d'établir des connexions multipoints, d'émettre des données sur ces connexions et de les fermer à la fin de la communication. La couche transport demande, à la couche réseau, l'établissement d'une connexion multipoint en lui spécifiant les adresses des stations qui constituent le groupe de réception. En réponse à cette demande, la couche réseau fournit, à la couche transport, un identificateur de groupe qu'elle utilisera pour émettre des données (la couche transport n'utilise pas les adresses des stations du groupe pour leur émettre des données), si toutes les stations du groupe acceptent la connexion.
La couche liaison de données ne permet d'établir que des connexions en point-à-point (c'est-à-dire qu'une connexion de niveau liaison de données est établie entre deux stations seulement).

On suppose que les couches réseau et liaison de données sont en mode connecté et assurent un contrôle de flux d'une fenêtre de taille égale à *n* pour les deux couches.

Question 1 : En supposant qu'il n'y a pas d'erreurs de transmission et que la taille des fenêtres d'anticipation, pour les couches 2 et 3, est égale à *3* (*n=3*), montrer les échanges de trames nécessaires pour que la couche transport d'une station *A* établisse une connexion multipoint avec quatre stations *B*, *C*, *D* et *E* et diffuse 6 paquets de données à ces quatre stations, ensuite elle ferme la connexion multipoint.

Question 2 : Reprendre la question 1 en supposant qu'il y a des erreurs de transmission mais qu'au niveau liaison de données une trame est toujours bien reçue au bout de deux tentatives au maximum.

Chapitre 7
Couche Session

I. Introduction

Le rôle de la couche session est de fournir aux entités de présentation les moyens nécessaires pour organiser et synchroniser leurs dialogues et pour gérer les échanges de données.

La couche session a été introduite pour gérer de manière efficace les transferts de données volumineuses. En particulier, dans les réseaux à débit faible, les pannes peuvent conduire à annuler des transfert et les relancer plusieurs fois, d'où une perte de temps considérable. Par exemple, si on effectue le transfert d'une base de données dont la taille se chiffre en dizaines de méga octets, sur un réseau à quelques kb/s, si une panne du site récepteur survient juste avant le transfert du dernier paquet de la base de données, on peut imaginer le temps de travail perdu.

Il faut noter qu'actuellement les réseaux ont des débits très élevés et la transmission de données même volumineuses ne prend que peu de temps, cela rend les services de la couche session un peu inutiles. D'ailleurs la couche session n'est quasiment jamais implantée dans sa globalité, c'est le plus souvent une forme simplifiée de la couche session (dite noyau) qui est implantée.

II. Concepts liés à la couche session

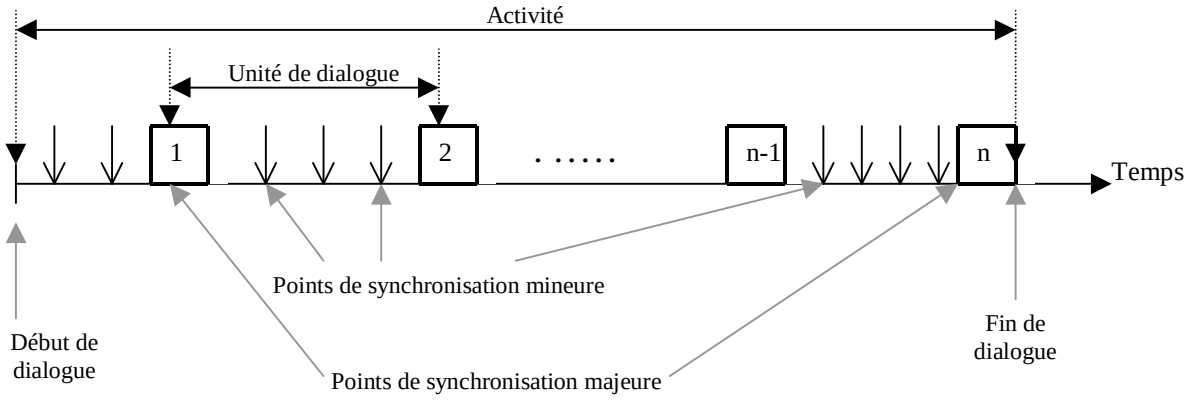
II.1. Synchronisation, unités de dialogue

Pour comprendre le fonctionnement de la couche session, prenons le cas de transfert d'un document. Si le document est transmis en un seul bloc, en cas d'erreur ou de panne, il faut retransmettre tout le document, cela ne gène pas trop si le document est de petite taille ou si le réseau a un débit élevé. Pour minimiser le temps pris par les retransmissions, on peut structurer le document en chapitres et les chapitres en pages. On transmet un certain nombre de pages et on demande au récepteur de valider la réception (de stocker les pages sur disque) avant de transmettre les pages suivantes, puis le chapitre suivant.

Un transfert est géré par une **activité**. Une activité est divisée en **unités de dialogue**. Chaque unité de dialogue permet le transfert d'une partie des données.

Exemple : une activité est une saisie d'opérations entre une agence et le siège d'une banque et une unité de dialogue est la saisie des opérations sur un seul compte bancaire.

La **synchronisation** est utilisée pour mettre les entités de session dans un état connu des deux interlocuteurs. En cas de panne, il est possible de remettre la session dans un état correspondant au dernier **point de synchronisation** connu par les deux correspondants et reprendre le transfert à partir de ce point. Chaque point de synchronisation est identifié par un **numéro de série**. Lorsqu'un utilisateur émet une requête pour poser un point de synchronisation, l'autre reçoit l'indication correspondante.
Les points de synchronisation mineure servent à la reprise à chaud (après la coupure d'une ligne ou une erreur mémoire, par exemple) et les points de synchronisation majeure à la reprise à froid (après le crash du système par exemple).



Activité et unité de dialogue

II.2. Jetons

Pour coordonner les échanges entre émetteur et récepteur de données, la couche session introduit la notion de **jeton**. Le jeton représente le droit, pour une entité de session, de pouvoir effectuer des opérations que son correspondant ne peut pas effectuer en même temps qu'elle.

Il y a quatre types de jeton :

- jeton de données (utilisé pour émettre des données),
- jeton de libération (utilisé pour mettre fin à une session),
- jeton de synchronisation mineure (utilisé pour fixer des points de synchronisation mineure),
- jeton de synchronisation majeure (utilisé pour fixer des points de synchronisation majeure).

L'utilisation des jetons dépend de ce que les entités de session souhaitent faire et la manière dont elles veulent cordonner leurs dialogues. En particulier, deux entités de session peuvent dialoguer sans faire appel aux jetons ou au contraire utiliser un ou plusieurs types de jeton.

III. Services de session

Les services de la couche session sont nombreux et ils ne sont pas tous utiles pour tous les types d'applications. Ainsi, pour répondre de manière efficace aux besoins des applications, les services de la couche session sont regroupés en classes (non disjointes) dites **unités fonctionnelle**s. Un système peut implanter une ou plusieurs unités fonctionnelles selon les besoins.

Primitive	Fonction	Mode
S-CONNECT	Etablissement de connexion de session	Confirmé
S-DATA	Emission de données normales	Non confirmé
S-EXPEDITED-DATA	Emission de données urgentes	Non confirmé
S-TYPED-DATA	Emission de données typées	Non confirmé
S-CAPABILITY-DATA	Emission de données de capacités	Confirmé
S-TOKEN-GIVE	Passage de jeton	Non confirmé
S-TOKEN-PLEASE	Demande de jeton	Non confirmé
S-CONTROL-GIVE	Contrôle du jeton	Non confirmé
S-SYNC-MINOR	Pose de point de synchronisation mineure	Confirmé
S-SYNC-MAJOR	Pose de point de synchronisation majeure	Confirmé
S-RESYNCHRONIZE	Resynchronisation	Confirmé
S-P-EXCEPTION-REPORT	Notification d'anomalie par le fournisseur de service	Non confirmé
S-U-EXCEPTION-REPORT	Notification d'anomalie par l'utilisateur	Non confirmé
S-ACTIVITY-START	Début d'activité	Non confirmé
S-ACTIVITY-RESUME	Reprise d'activité	Non confirmé
S-ACTIVITY-INTERRUPT	Interruption d'activité	Non confirmé
S-ACTIVITY-END	Fin d'activité	Confirmé
S-ACTIVITY-DISCARD	Annulation d'activité	Confirmé
S-RELEASE	Libération de connexion de session	Confirmé
S-U-ABORT	Avortement de session par l'utilisateur	Non confirmé
S-P-ABORT	Avortement de session par le fournisseur de service	Non confirmé

Unités fonctionnelles de la couche session

- Unité fonctionnelle *Noyau* : elle contient les primitives : S-CONNECT, S-DATA, S-RELEASE, S-U-ABORT, S-P-ABORT.

- Unité fonctionnelle *Négociation de fermeture de connexion :* elle contient les primitives S-RELEASE, S-TOKEN-GIVE, S-TOKEN-PLEASE.

- Unité fonctionnelle *Half duplex* : elle contient les primitives S-TOKEN-GIVE, S-TOKEN-PLEASE.

- Unité fonctionnelle *Données urgentes :* elle contient la primitive S-EXPEDITED-DATA.

- Unité fonctionnelle *Données de capacités :* elle contient la primitive S-CAPABILITY-DATA.

- Unité fonctionnelle *Données typées* : elle contient la primitive S-TYPED-DATA.

- Unité fonctionnelle *Synchronisation mineure* : elle contient les primitives S-SYNC-MINOR, S-TOKEN-GIVE, S-TOKEN-PLEASE.

- Unité fonctionnelle *Synchronisation majeure* : elle contient les primitives S-SYNC-MAJOR, S-TOKEN-GIVE, S-TOKEN-PLEASE.

- Unité fonctionnelle *Synchronisation resynchronisation* : elle contient la primitive S-RESYNCHRONIZE.

- Unité fonctionnelle *Synchronisation Exceptions* : elle contient les primitives S-U-EXCEPTION-REPORT, S-P-EXCEPTION-REPORT.

- Unité fonctionnelle *Gestion d'activités* : elle contient les primitives S-ACTIVITY-START, S-ACTIVITY-RESUME, S-ACTIVITY-INTERRUPT, S-ACTIVITY-DISCARD, S-ACTIVITY-END, S-TOKEN-GIVE, S-TOKEN-PLEASE, S-CONTROL-GIVE.

Phase d'établissement de session

Les paramètres de qualité de service sont négociés durant la phase d'établissement de session. Il s'agit notamment de :

- délai maximum d'établissement de connexion de session,
- probabilité d'échec d'établissement de connexion de session,
- débit nécessaire à la session,
- temps de transfert,
- taux d'erreurs résiduelles,
- probabilité d'incident de transfert,
- délai de libération de connexion de session,
- probabilité d'échec de libération de connexion de session,
- protection de connexion de session,
- priorité de connexion de session,
- transfert avec optimisation.

Exercices

Exercice 1
En utilisant les primitives de service de la couche session, écrire les algorithmes permettant d'effectuer le transfert de fichiers :
a) cas de transfert non fiable,

b) transfert fiable : dans ce cas, on utilise un point de synchronisation mineure tous les dix enregistrements et un point de synchronisation majeure tous les 100 enregistrements. On suppose que le fichier est constitué d'enregistrements.

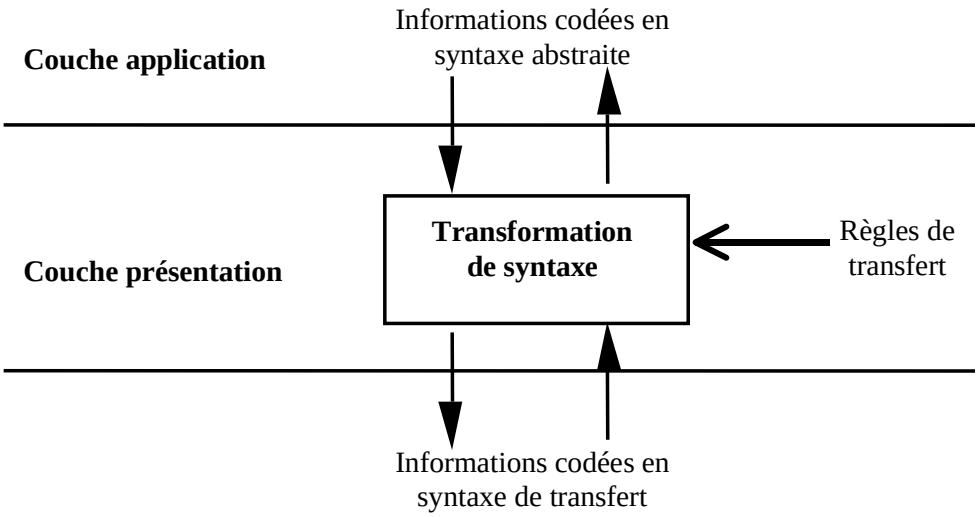
Exercice 2
Utiliser des automates à états finis pour représenter :
a) le fonctionnement de l'unité fonctionnelle synchronisation mineure
b) le fonctionnement de l'unité fonctionnelle Half duplex,
c) le fonctionnement de l'unité fonctionnelle resynchronisation.

Chapitre 8
Couche Présentation
ASN.1 et XDR

I. Introduction

La couche présentation offre des services permettant l'indépendance des applications vis-à-vis de la représentation des données (codage des entiers, des réels, …), types de caractères (ASCII ou autre), et des formats de fichiers.

Le principal apport existant actuellement, au niveau présentation, est la syntaxe abstraite numéro 1 : ASN.1 (Abstract Syntax Notation 1)



II. Services et unités fonctionnelles de la couche présentation

Comme pour la couche session, les services de la couche présentation sont regroupés en unités fonctionnelles.

Primitive	Fonction	Mode
P-CONNECT	Etablissement de connexion de présentation	Confirmé
P-DATA	Emission de données normales	Non confirmé
P-EXPEDITED-DATA	Emission de données urgentes	Non confirmé
P-TYPED-DATA	Emission de données typées	Non confirmé
P-CAPABILITY-DATA	Emission de données de capacités	Confirmé

P-TOKEN-GIVE	Passage de jeton	Non confirmé
P-TOKEN-PLEASE	Demande de jeton	Non confirmé
P-CONTROL-GIVE	Contrôle du jeton	Non confirmé
P-SYNC-MINOR	Pose de point de synchronisation mineure	Confirmé
P-SYNC-MAJOR	Pose de point de synchronisation majeure	Confirmé
P-RESYNCHRONIZE	Resynchronisation	Confirmé
P-P-EXCEPTION-REPORT	Notification d'anomalie par le fournisseur de service	Non confirmé
P-U-EXCEPTION-REPORT	Notification d'anomalie par l'utilisateur	Non confirmé
P-ACTIVITY-START	Début d'activité	Non confirmé
P-ACTIVITY-RESUME	Reprise d'activité	Non confirmé
P-ACTIVITY-INTERRUPT	Interruption d'activité	Non confirmé
P-ACTIVITY-END	Fin d'activité	Confirmé
P-ACTIVITY-DISCARD	Annulation d'activité	Confirmé
P-RELEASE	Libération de connexion de présentation	Confirmé
P-U-ABORT	Avortement de connexion par l'utilisateur	Non confirmé
P-P-ABORT	Avortement de connexion par le fournisseur de service	Non confirmé
P-ALTER-CONTEXT	Gestion de contexte	Confirmé

Les primitives *P-TOKEN-GIVE*, *P-TOKEN-PLEASE*, *P-CONTROL-GIVE*, *P-SYNC-MINOR*, *P-SYNC-MAJOR*, *P-RESYNCHRONIZE*, *P-P-EXCEPTION-REPORT*, *P-U-EXCEPTION-REPORT*, *P-ACTIVITY-START*, *P-ACTIVITY-RESUME*, *P-ACTIVITY-INTERRUPT*, *P-ACTIVITY-END* et *P-ACTIVITY-DISCARD* sont utilisées pour permettre à la couche application d'accéder aux services de gestion des activités et de synchronisation fournis par la couche session. Elles sont mises en correspondance directe avec les primitives *S-TOKEN-GIVE*, *S-TOKEN-PLEASE*, *S-CONTROL-GIVE*, *S-SYNC-MINOR*, *S-SYNC-MAJOR*, *S-RESYNCHRONIZE*, *S-P-EXCEPTION-REPORT*, *S-U-EXCEPTION-REPORT*, *S-ACTIVITY-START*, *S-ACTIVITY-RESUME*, *S-ACTIVITY-INTERRUPT*, *S-ACTIVITY-END* et *S-ACTIVITY-DISCARD* de la couche session. La couche présentation n'apporte aucun traitement additionnel (par rapport à celui effectué par la couche session) à ces primitives.

Unités fonctionnelles de la couche présentation

- Unité fonctionnelle *Noyau* : elle contient les primitives : P-CONNECT, P-DATA, P-RELEASE, P-U-ABORT, P-P-ABORT.

- Unité fonctionnelle *Gestion de contexte :* elle contient la primitive P-ALTER-CONTEXT.

- Unité fonctionnelle *Négociation de libération de connexion :* elle contient les primitives P-RELEASE, P-TOKEN-GIVE, P-TOKEN-PLEASE.

- Unité fonctionnelle *Half duplex :* elle contient les primitives P-TOKEN-GIVE, P-TOKEN-PLEASE.

- Unité fonctionnelle *Données urgentes :* elle contient la primitive P-EXPEDITED-DATA.

- Unité fonctionnelle *Données typées :* elle contient la primitive P-TYPED-DATA.

- Unité fonctionnelle *Données de capacités :* elle contient la primitive P-CAPABILITY-DATA.

- Unité fonctionnelle *Synchronisation mineure* : elle contient les primitives P-SYNC-MINOR, P-TOKEN-GIVE, P-TOKEN-PLEASE.

- Unité fonctionnelle *Synchronisation majeure :* elle contient les primitives P-SYNC-MAJOR, P-TOKEN-GIVE, P-TOKEN-PLEASE.

- Unité fonctionnelle *Synchronisation resynchronisation :* elle contient la primitive P-RESYNCHRONIZE.

- Unité fonctionnelle *Synchronisation Exceptions :* elle contient les primitives P-U-EXCEPTION-REPORT, P-P-EXCEPTION-REPORT.

- Unité fonctionnelle *Gestion d'activités :* elle contient les primitives P-ACTIVITY-START, P-ACTIVITY-RESUME, P-ACTIVITY-INTERRUPT, P-ACTIVITY-DISCARD, P-ACTIVITY-END, P-TOKEN-GIVE, P-TOKEN-PLEASE, P-CONTROL-GIVE.

III. Introduction à ASN.1

Dans le domaine des réseaux, la norme OSI/ITU-T dite ASN.1 est utilisée depuis longtemps, pour décrire les données -les données des services et protocoles de la couche application, en particulier- et pour fixer les règles de codage en bits des données transmises via un réseau de communication. La notation ASN.1 est indispensable pour la normalisation des services de niveau application offerts par les réseaux, puisqu'elle permet de rendre plus indépendants les protocoles de niveau application de ceux du niveau transport de données.

ASN.1 est basé sur la notion d'ensembles de valeurs. Les opérateurs n'y sont pas définis. Un type est défini par l'ensemble des valeurs qu'il prend seulement. De nouveaux types peuvent être définis à partir de types existants. En utilisant des règles de codage, on peut spécifier la manière dont les données sont représentées en chaînes de bits pour les transmettre d'un équipement vers un autre. ASN.1 ne définit pas d'opérateurs sur les types et par conséquent, aucune propriété ne peut être associée à un type.

III.1. Définition et inclusion de module ASN.1

Afin d'utiliser les données ASN.1 dans différents contextes, ces données doivent être définies dans des librairies appelés modules(`MODULE`). Un module ASN.1 contient des définitions de types de données ou des valeurs destinés à être importés par d'autres modules. La clause `EXPORTS` (resp. `IMPORTS`) de ASN.1 permet d'exporter (resp. d'importer) des définitions.

```
<définition de module ASN.1> ::=
  <module> DEFINITIONS [<attache par défaut>] ::=
  BEGIN [<corps de module>] END

<module> ::= <nom de module> [<valeur d'identificateur d'objet>]

<attache par défaut> ::= EXPLICIT TAGS | IMPLICIT TAGS
  | AUTOMATIC TAGS

<corps de module> ::= [<exporter>] [<importer>]
  <entité dans librairie>*

<exporter> ::= EXPORTS [<liste de sélection de définition>] <fin>

<importer> ::= IMPORTS <symboles importés>* <fin>

<symboles importés> ::= {<liste de sélection de définition>
  FROM <module>}*
```

Exemple

```
Mod1 DEFINITIONS ::=
  BEGIN
    EXPORTS Oui, Non ;
    Oui BOOLEAN ::= True ;
    Non BOOLEAN ::= False ;
  END
```

III.2. Types prédéfinis de ASN.1

ASN.1 offre plusieurs types dont certains sont identiques à ceux connus dans des langages de programmation tels que Pascal. Ces types peuvent être regroupés en quatre catégories : types simples, types de chaînes, types structurés et autres types.

Types simples

Les types simples de ASN.1 sont les suivants :

– `BOOLEAN`

– `INTEGER`

– `REAL` : nombres réels exprimés à l'aide de triplets {mantisse, base, exposant} ;

– `ENUMERATED :` ensemble de valeurs ordonnées ;

– `OCTET :` chaîne de huit bits.

Exemple

```
Couleurs ::= ENUMERATED {bleu(2), blanc(1), rouge(0)} ;
```

Types de chaînes

Pour répondre à divers besoins de manipulation de données structurées sous forme de chaînes, la notation ASN.1 fournit plusieurs types de chaînes :

– `BIT STRING` : suite de bits (`0` ou `1`) représentant des nombres en base `2` ; les bits peuvent être aussi regroupés par blocs de quatre bits pour représenter des nombres en base `16` (par exemple, les deux chaînes suivantes sont égales : `'1100'B` et `'C'H`) ;

– `OCTET STRING` : suite de bits regroupés en octets pour représenter des données dont la structure n'est pas connue, mais dont la taille est un multiple de 8 bits. Les chaînes d'octets s'écrivent de la même manière que les chaînes de bits (en binaire ou en hexadécimal) ;

– `IA5String` : chaînes composées de tous les caractères ASCII ;

– `NumericString` : chaînes composées de chiffres décimaux et d'espace ;

– `PrintableString` : chaînes composées de lettres majuscules et minuscules, chiffres décimaux, `","`, `"."`, `" :"`, `"="`, `"?"`, `"'"`, `"("`, `")"`, `"+"`, `"-"`, `"/"` ;

– `VisibleString` : chaînes composées de caractères imprimables ASCII et d'espace ;

– `GraphicString` : chaînes composées de tous les ensembles G de caractères enregistrés et d'espace ;

– `UniversalString` : chaînes composées de tous les ensembles de caractères C et G enregistrés plus l'espace et "delete".

Types structurés

ASN.1 permet de définir des enregistrements, des ensembles et des listes. En particulier, ASN.1 offre les constructeurs de types suivants :

– `SEQUENCE` : équivalent à la notion d'enregistrement du langage Pascal. L'ordre des différents champs d'un type `SEQUENCE` est important et ces champs peuvent appartenir à des types différents. Le mot clé `OPTIONAL` indique qu'un champ de la structure est optionnel. Le mot clé `DEFAULT` permet d'initialiser un champ en utilisant une valeur par défaut.

– `SEQUENCE OF` : ensemble d'éléments ordonnés de même type. En particulier, `SEQUENCE OF` permet de définir des listes et des tableaux ;

– `SET` : constructeur similaire à `SEQUENCE`, mais où l'ordre entre les éléments n'est pas important et où les types des éléments doivent être distincts deux à deux ;

– `SET OF` : constructeur similaire à `SEQUENCE OF`, mais où l'ordre des éléments n'est pas important ;

– `CHOICE` : constructeur qui permet de définir un type qui peut prendre un (et un seul) type parmi plusieurs types spécifiés.

Exemple

```
-- Définition d'un type liste d'entiers en ASN.1.
    Liste_entiers ::= SEQUENCE OF INTEGER ;

-- Définition d'un type NumeroTelephone à 10 chiffres en ASN.1.
   NumeroTelephone ::= SEQUENCE SIZE(10) OF INTEGER (0 : 9) ;

-- Définition d'un type FichierX dont les éléments peuvent être du texte, du
binaire ou vides. */
   FichierX ::= SEQUENCE OF ContenuFichier ;
   ContenuFichier ::= CHOICE {    texte IA5String,
                                  binaire BIT STRING,
                                  Vide NULL } ;
```

Autres types

– NULL : type qui n'a qu'une seule valeur (NULL) ;

– OBJECT IDENTIFIER : type utilisé pour identifier les informations échangées sur un réseau de manière unique. Les identificateurs d'objets sont gérés par des institutions internationales (telles que l'ISO ou l'ITU-T). Un identificateur d'objet est représenté par une suite ordonnée de nombres. Par exemple, l'identificateur d'objet {1 3 100 10 2 7 0} pourrait représenter un terminal (0) d'un équipement (7) d'une application(2) d'une organisation (10) gérée par une autorité (100) reconnue(3) par l'ISO (1).

– ANY : indique que le type d'une information n'est pas défini par l'utilisateur. Par exemple, si on ne connaît pas la structure des données d'une PDU échangée dans un protocole, on peut spécifier un type de PDU de la manière suivante :

```
PDUIncomplet ::= SEQUENCE {   adresse_source INTEGER,
                              adresse_destination INTEGER,
                              contenu ANY}
```

– ANY DEFINED BY : indique que le type d'un paramètre dépend de la valeur d'un autre paramètre. Par exemple, dans le type PDUComplet suivant, le type du champ contenu dépend de la valeur du champ PDU_type :

```
PDUComplet ::= SEQUENCE {    adresse_source INTEGER,
     adresse_destination INTEGER,
     PDU_type T_PDU_type,
     contenu ANY DEFINED BY PDU_type}
     T_PDU_Type ::= ENUMERATED {    DemandeConnexion(0),
                              AcceptationConnexion(1),
                              DemandeDeconnexion(2),
                              AcceptationDeconnexion(3),
                              Donnees(4)}
```

– GeneralizedTime : permet de spécifier un instant sous l'une des trois formes suivantes (où : AAAA désigne l'année, MM, le mois, JJ, le jour, HH, l'heure, mm, la minute, ss, la seconde et fff, le millième de seconde) :

- temps local : AAAAMMJJHHmmss.fff,
- temps universel : AAAAMMJJHHmmss.fffZ,
- temps local avec le décalage par rapport au temps universel UTC (Universal Time Coordinated.) : AAAAMMJJHHmmss.fff±HHmm

– UTCTime : permet de spécifier un instant sous l'une des trois formes suivantes

- temps local : AAMMJJHHmmss,
- temps universel : AAMMJJHHmmssZ,
- temps local avec le décalage par rapport au temps universel UTC : AAMMJJHHmmss±HHmm

– ObjectDescriptor : utilisé pour avoir des identificateurs mnémoniques au lieu de simples codes de type entier.

– External : permet de définir des types externes à ASN.1.

III.3. Utilisation d'attaches

Lorsque des données sont échangées entre deux entités distantes, des attaches ("tags") peuvent être associées à ces données pour faciliter leur décodage par le récepteur. Une attache permet d'indiquer le type d'une donnée. Par exemple, pour distinguer deux entiers qui correspondent à une température et une pression, deux attaches distinctes peuvent être associées à ces données.

On peut utiliser soit un mode d'attaches dit explicite (indiqué par `EXPLICIT`, qui signifie que les attaches de l'utilisateur accompagnent les attaches standards) ou bien un mode dit implicite (indiqué par `IMPLICIT`, qui signifie que les attaches de l'utilisateur remplacent les attaches standards).

Les attaches sont regroupées en classes : `UNIVERSAL` (pour les types prédéfinis de ASN.1) et `APPLICATION` et `PRIVATE` (pour les autres types). Il existe des attaches standards pour les types prédéfinis de ASN.1 (par exemple, `1` pour `BOOLEAN`, `9` pour `REAL` et `23` pour `UTCTime`).

Quelques règles syntaxiques de ASN.1

```
<sorte existante> ::= [<nom de librairie> .]
  <identificateur de sorte> | ANY [DEFINED BY <identificateur>]
 | <sélection>

<sélection> ::= <nom> < <sorte>

<constructeur de sorte> ::= <attache> <expression de sorte>
  | <choix> | <énuméré> | <séquence> | <séquence de>
  | <appellation d'entier>

<attache> ::= [[UNIVERSAL | APPLICATION | PRIVATE]
  <expression simple>] [IMPLICIT | EXPLICIT]

<séquence> ::= {SEQUENCE | SET} {[<sorte d'élément>
  {, <sorte d'élément>}*]}

<sorte d'élément> ::= <sorte nommée> [OPTIONAL | DEFAULT
  <expression close>] | COMPONENTS OF <sorte>

<sorte nommée> ::=   [<nom>] <sorte>

<séquence de> ::= {SEQUENCE | SET} [<contrainte de taille>
  | <condition d'intervalle ASN.1>] OF <sorte>

<choix> ::= CHOICE {[<sorte nommée> {, <sorte nommée>}*]}

<énuméré> ::= ENUMERATED {<nombre nommé> {, <nombre nommé>}*}

<nombre nommé> ::=   <valeur nommée> | <nom>

<appellation d'entier> ::= <identificateur>
  {<valeur nommée> {, <valeur nommée>}*}

<valeur nommée> ::= <nom>(<expression simple>)

<sous-intervalle> ::= <sorte>(<condition d'intervalle>)

<condition d'intervalle> ::= <intervalle> {{, | |} <intervalle>}*

<intervalle> ::= <intervalle clos> | <intervalle ouvert>
  | <sous-intervalle contenu> | <contrainte de taille>
  | <composant intérieur> | <composants intérieurs>

<intervalle clos> ::= <valeur inférieure> { : | ..}
  <valeur supérieure>

<valeur inférieure> ::= {<expression close> | MIN} [<]
```

```
<valeur supérieure> ::= [<] {<expression close> | MAX}

<intervalle ouvert> ::= [= | /= | < | > | <= | >=]
  <expression close>

<sous-intervalle contenu> ::= INCLUDES <sorte>

<contrainte de taille> ::= SIZE(<condition d'intervalle>)

<composant intérieur> ::=  {FROM | WITH COMPONENT}
  (<condition d'intervalle>)

<composants intérieurs> ::= WITH COMPONENTS { [...,]
  <contrainte nommée> {, <contrainte nommée>}*}

<contrainte nommée> ::= <nom> [<condition d'intervalle ASN.1>]
  [PRESENT | ABSENT | OPTIONAL]

<condition d'intervalle ASN.1> ::= (<condition d'intervalle>)

<primaire étendu> ::= <synonyme> | <primaire indexé>
  | <primaire de champ> | <primaire de structure>
  | <primaire de choix> | <primaire composé>

<primaire de choix> ::= <identificateur> : <primaire>

<primaire composé> ::= [<qualificatif>] {<valeur séquence>
  | <valeur de séquence> | <valeur d'identificateur d'objet>
  | <valeur réelle>}

<valeur séquence> ::= {[<valeur_nommée> {, <valeur_nommée>}*]}

<valeur_nommée> ::= <nom> <expression>

<valeur de séquence> ::=   {[<expression> {, <expression>}*]}

<valeur d'identificateur d'objet> ::=
  {<composant d'identificateur d'objet>+}

<composant d'identificateur d'objet> ::= <identificateur>
  [(<expression close>)]

<valeur réelle> ::= {<mantisse> , <base> , <exposant>}

<mantisse> ::= <expression>

<base> ::=     <expression simple>

<exposant> ::= <expression>
```

Exemple

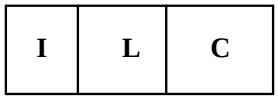
```
-- On considère un protocole de lecture de fichier à distance.
-- L'accès se fait via un nom et un mot de passe. Si la personne
-- qui souhaite se connecter est inconnue ou bien si le système est
-- occupé, un message de refus est renvoyé à cette personne.
-- Si le fichier existe, les données (sous forme de chaîne
-- d'octets) sont envoyées au demandeur, sinon une erreur est
-- signalée.
-- A la fin de transfert, une demande de déconnexion est envoyée et
-- cette demande est acquittée. Les deux messages liés à la phase
-- de déconnexion ne contiennent aucune informations, leur
-- présence suffit au protocole.
-- Les types nécessaires à ce protocole sont décrits en ASN.1 et
-- placés dans un module appelé M_AccesFichier.

     M_AccesFichier DEFINITIONS ::=
     BEGIN
       AccesSimpleFichier ::= CHOICE
```

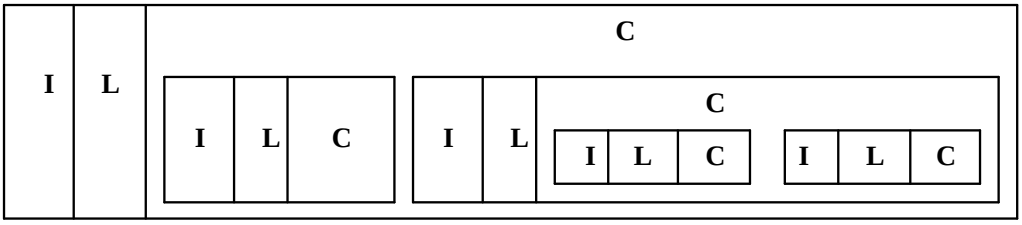
```
    { SeConnecter    PDUConnexion,
      ResultatConnexion    PDUResultatConnexion,
      SeDeconnecter  PDUDeconnexion,
      AckDeconnexion PDUAckConnexion,
      ObtenirFichier PDUDemandeFichier,
      ResultatFichier      PDUResultatFichier} ;
  PDUConnexion ::= SEQUENCE {     nom IA5String,
                          Mot_passe IA5String} ;
  PDUResultatConnexion ::= CHOICE
    { Acceptee IA5String,
      Rejetee INTEGER {AccesInvalide(0),
                  SystemeOccupe(1)}} ;
  PDUDeconnexion ::= NULL ; -- PDU sans informations
  PDUAckConnexion ::= NULL ; -- PDU sans informations
  PDUDemandeFichier ::= IA5String ; -- Nom de fichier
  PDUResultatFichier ::= CHOICE
    { Fichier SEQUENCE {
                  NomFIchier IA5String,
                  ContenuFichier OCTET STRING},
      Erreur INTEGER {
                  NonTrouve(0),
                  FichierOccupe(1)}} ;
END
```

III.4. Règles de codage des informations

La BER (Basic Encoding rules) définit les règles de transfert des types ASN.1. La BER code les informations sous forme d'**éléments de données**. Tous les éléments de données ont une même forme générique. Chaque élément de données est composé de trois champs : l'*identificateur*, la *longueur* et les composants (I, L, C).



Forme primitive



Forme construite

Structure des éléments de données de la BER.

L'identificateur (I) comprend trois parties :

1. La classe (bits 8 et 7) :
   - 00 : type universel,
   - 01 : type d'application,
   - 10 : type spécifique à un contexte,
   - 11 : type privé.

2. La forme (bit 6) : 0 (forme primitive), 1 (forme construite).

Tous les types sont codés sont la forme primitive sauf SEQUENCE, SEQUENCE OF, SET et SET OF qui sont codés sous la forme construite.

3. L'attache (bits 5 à 1 véhiculent l'attache)

La longueur (L) a l'une des trois formes suivantes :

- format court sur un octet,
- format long qui code la longueur sur 127 octets au maximum,
- format indéfini (qui utilise des marques spéciales pour terminer une longueur).

Le contenu (C) contient *0* ou *x* octets qui spécifient le contenu de l'élément.

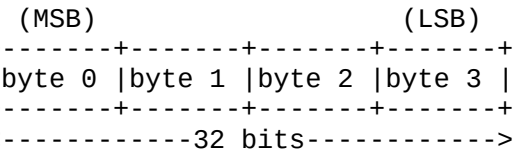
Grâce aux règles de la BER, les données échangées entre processus d'application sont toujours codées de la même manière indépendamment des types d'équipements sur lesquels se trouvent ces processus.

IV. Introduction à XDR

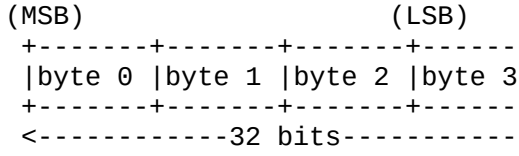
XDR (eXternal Data Representation) est le protocole de représentation de données le plus utilisé dans le monde Internet. Il a été proposé par Sun Microsystems. Il a été publié sous forme de RFC pour la première fois en 1987, ensuite il a été amélioré en 1995 et en 2006 (le RFC 4506 est la dernière version de XDR).

IV.1. Types XDR et leur représentation binaire

```
<Integer> ::= int <identifier>;
```



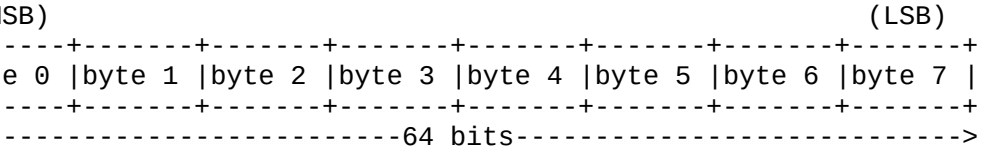

```
<Unsigned integer> ::= unsigned int <identifier>;
```



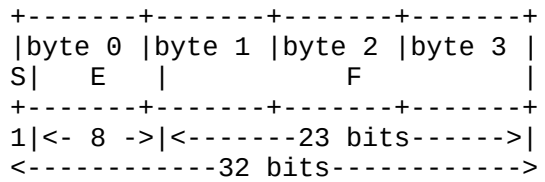

```
<Enumeration> ::= enum { <name-identifier = constant>, ... } <identifier>;

<Boolean> ::= bool <identifier>;

<Hyper Integer> ::=  hyper <identifier>;
<Hyper unsigned> ::=  unsigned hyper <identifier>;
```

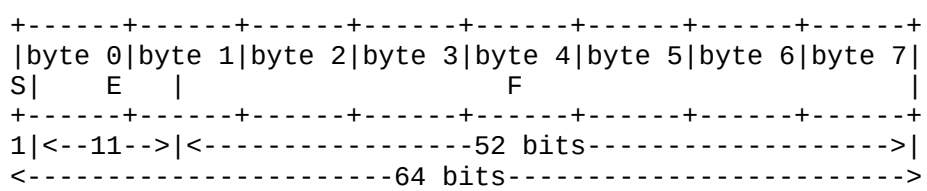


<Floating-Point> ::= **float** <identifier>;

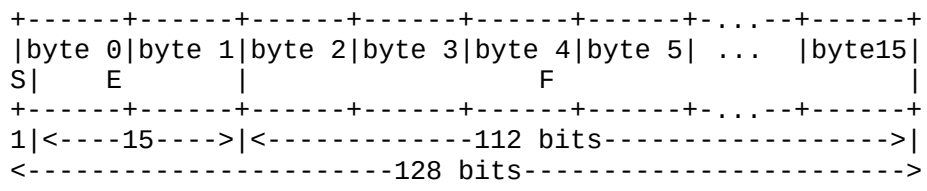


Un nombre flottant est décrit par : (-1)**S * 2**(E) * 1.F
S: signe du nombre, E : exposant et F : partie décimale (mantisse)

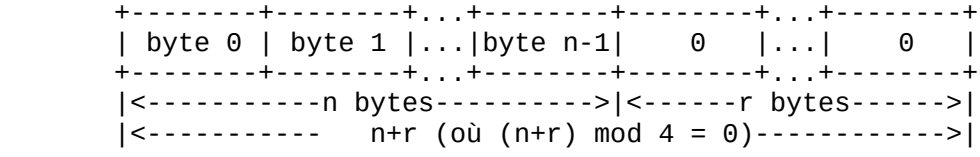
<Double-Precision Floating-Point> ::= **double** <identifier>;



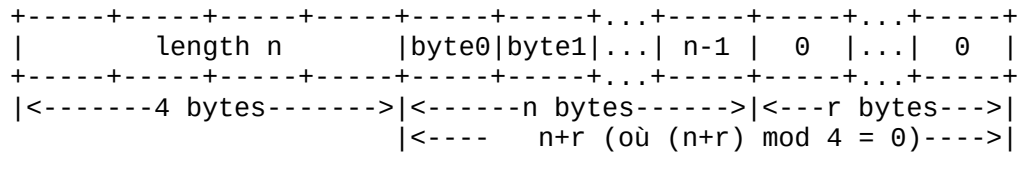
<Quadruple-Precision Floating-Point> ::= **quadruple** <identifier>;



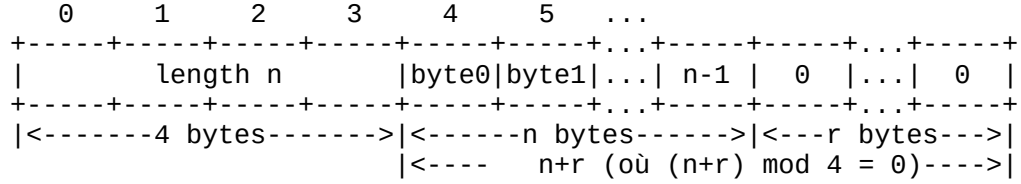
<Fixed-Length Opaque Data> ::= **opaque** <identifier> ‘[’ <n> ‘];;



<Variable-Length Opaque Data> ::= **opaque** <identifier> {‘<’ ‘>’ | ‘<’ <n> ‘>’; };



<String> ::= **string** <object> {‘<’ ‘>’ | ‘<’ <n> ‘>’; };



```
<Fixed-Length Array> ::= <type-name> <identifier> '[' <n> ']';

          +---+---+---+---+---+---+---+---+...+---+---+---+---+
         |   element 0   |   element 1   |...|  element n-1  |
         +---+---+---+---+---+---+---+---+...+---+---+---+---+
         |<--------------------n elements------------------->|


<Variable-Length Array> ::= <type-name> <identifier> {'<' '>' | '<' <n> '>'; };

          0  1  2  3
         +--+--+--+--+--+--+--+--+--+--+--+...+--+--+--+--+
         |     n     | element 0 | element 1 |...|element n-1|
         +--+--+--+--+--+--+--+--+--+--+--+...+--+--+--+--+
         |<-4 bytes->|<-------------n elements------------>|


<Structure> ::= struct {
            <component-declaration-A> ; <component-declaration-B>;
            ... } <identifier>;

         +-------------+-------------+...
         | component A | component B |...
         +-------------+-------------+...

<Discriminated Union> ::= union switch '(' <discriminant-declaration> ')' {
         case <discriminant-value-A>: <arm-declaration-A>;
         case <discriminant-value-B>: <arm-declaration-B>;
         ...
         default: <default-declaration>;
         } <identifier>;

          0   1   2   3
         +---+---+---+---+---+---+---+---+
         |<discriminant> |<implied arm>  |
         +---+---+---+---+---+---+---+---+
         |<---4 bytes--->|

<Void> ::= void;

          ++
          ||               VOID
          ++
         --><-- 0 bytes

<Constant> ::= const <name-identifier> '=' <n>;

<Typedef> ::= typedef <declaration>;

<Optional-Data> ::= <type-name> *<identifier>;
C'est equivalent à la definition suivante :
         union switch (bool xx) {
         case TRUE: <type-name> <element>;
         case FALSE: void;  } <identifier>;
```

IV.2. Exemple d'utilisation de XDR

Une représentation pour le transfert simplifié de fichier

```
const MAXUSERNAME = 32;     /* Longueur maxi du nom d'utilisateur */
const MAXFILELEN = 65535;   /* Taille maxi de fichier */
const MAXNAMELEN = 255;     /* Longueur maxi de nom de fichier */

enum typedecontenu {
        TEXT = 0, /* Fichier ASCII */
        DATA = 1, /* Données brutes */
        EXEC = 2 /* Fichier exécutable */ };

/* Informations additionnelles selon le type de fichier  */
union typefichier switch (typedecontenu genre) {
        case TEXT: void;       /* Pas d'informations supplémentaires */
        case DATA: string createur<MAXNAMELEN>; /* Créateur de données */
        case EXEC: string interpreteur<MAXNAMELEN>;    /*
          interpréteur de programme */  };

struct file {/* structure de fichier */
        string nomfichier <MAXNAMELEN>; /* nom de fichier    */
        typefichier type;               /* type de fichier */
        string proprietaire <MAXUSERNAME>; /* propriétaire du fichier */
        opaque contenu <MAXFILELEN>;     /* contenu du fichier */
 };
```

Soit un utilisateur "Jean" qui a un fichier nommé "progmJava" contenant un programme Java erroné qui contient un seul mot "quit". Ce fichier peut être codé par :

OFFSET	Octets en Hexa	ASCII	Commentaires
0	00 00 00 09	....	-- Longueur du nom de fichier = 9
4	73 69 6c 6c	prog	-- Nom du fichier
8	79 70 72 6f	mJav	-- .....
12	67 00 00 00	a...	-- ... complété par 3 octets à 0
16	00 00 00 02	....	-- Type de fichier est EXEC = 2
20	00 00 00 04	....	-- Longueur de l'interpréteur = 4
24	6c 69 73 70	Java	-- nom de l'interpréteur
28	00 00 00 04	....	-- longueur du propriétaire = 4
32	6a 6f 68 6e	Jean	-- nom du propriétaire
36	00 00 00 06	....	-- longueur des données du fichier = 6
40	28 71 75 69	(qui	-- octets de données du fichier
44	74 29 00 00	t)..	-- ... complétés par 2 octets à 0

Exercices

Exercice 1
On souhaite spécifier en ASN.1 les données relatives aux échanges de documents d'une entreprise Y. Les informations échangées concernent le personnel et les véhicules. Une personne est définie par les attributs suivants :
- nom, prénom
- fonction (sous forme de texte)
- le (ou les) véhicule(s) utilisé(s) par cette personne
- situation familiale
- pour une personne avec des enfants, le nombre et l'âge des enfants

Pour chaque véhicule de l'entreprise, on dispose des informations suivantes :
- numéro d'immatriculation
- puissance fiscale
- type de carburant

a) Définir les types ASN.1 correspondant aux données définies précédemment
b) Prendre un exemple de personne et montrer la suite d'octets obtenus par la BER pour cette personne.

Exercice 2
Même question que pour l'exercice 1 mais en utilisant XDR.

Exercice 3
Définir un nombre minimal de types ASN.1 pour décrire les formats de cellules générées en tenant compte des différents protocoles de la couche d'adaptation d'ATM (AAL).

Chapitre 9
Couche Application

I. Introduction

La prise en compte des problèmes d'hétérogénéité et de diversité des applications a conduit l'ISO à définir une architecture unifiée de la couche application spécifiée par la norme ISO 9545. Cette norme met en évidence les relations entre le traitement d'informations et les services de communication OSI. Elle ne propose pas de service, mais définit une structure modulaire et générale de représentation des normes applicatives. En ce sens, elle définit des modèles abstraits supportant les ressources nécessaires à la communication entre applications dans un environnement OSI. Elle précise en particulier :

- la nature des normes de la couche application et leurs relations,
- le cadre de travail dans lequel les protocoles applicatifs doivent être développés,
- les catégories d'objets identifiables nécessaires aux spécifications et opérations des protocoles,
- la liaison entre les activités de traitement et l'information distribuée et les normes de la couche application.

II. Structure générale de la couche application (norme ISO 9545)

II.1. Processus d'application

Une application répartie, c'est-à-dire s'exécutant sur plusieurs sites, est réalisée par un ensemble de processus d'application. Un processus d'application (ou AP : "Application Process") est une représentation abstraite des éléments d'un système ouvert réel qui réalisent le traitement de l'information pour une application particulière.

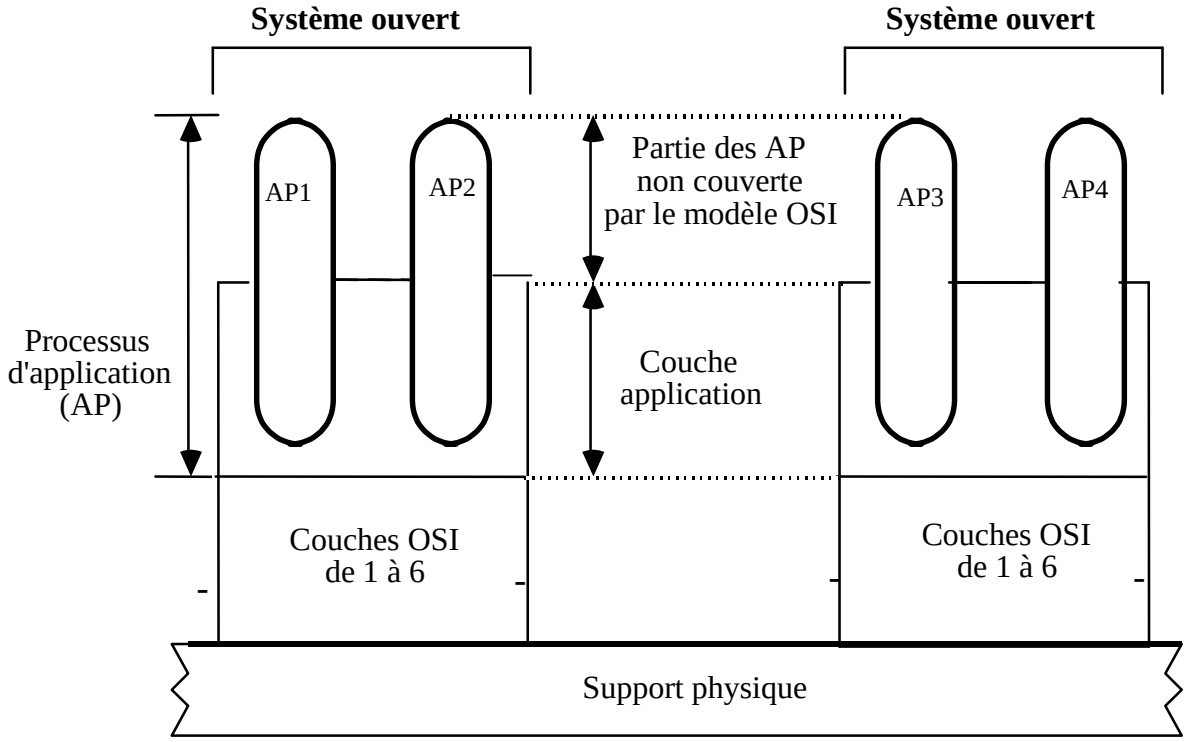
Exemples d'AP

Un opérateur effectuant une réservation de place dans un avion, un programme mettant à jour une base de données, un programme de régulation de vitesse d'un moteur pas à pas, un serveur de fichiers, un processus de supervision d'une cellule de production, … sont tous des exemples de processus d'application.

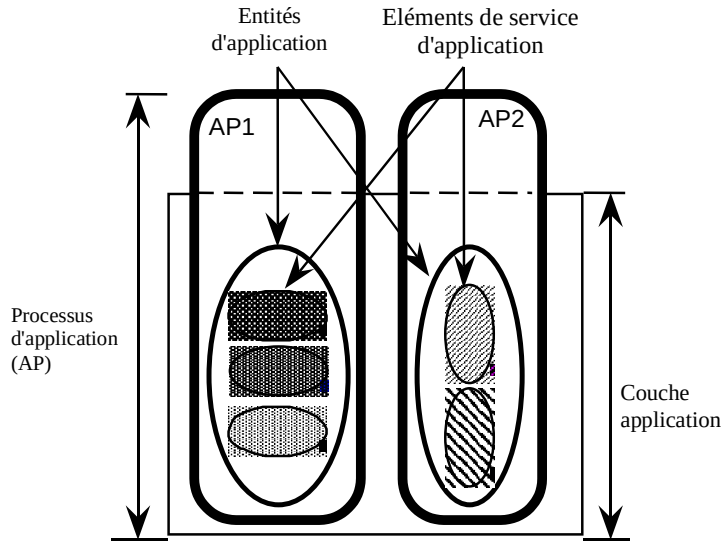
Un AP agit dans deux environnements : l'environnement OSI et l'environnement propre à l'utilisateur.
L'exécution d'un AP est conceptualisée par la notion d'instanciation de processus d'application (ou API : AP Invocation). Les relations établies entre API représentent la coopération entre AP. Un AP est un programme statique, et chaque fois qu'on active ce programme on crée une instanciation de l'AP.

II.2. Entité d'application

Une entité d'application (ou AE : "Application Entity") représente un ensemble de fonctionnalités et/ou de ressources nécessaires aux besoins de communication d'un AP.
Une instanciation d'entité d'application (ou AEI : "Application Entity Invocation") est une utilisation des fonctionnalités/ressources d'une entité d'application pour les activités de communication spécifiques d'une API. Les aspects communication d'une API sont représentés par une ou plusieurs instanciations d'entités d'application.



Place des processus d'application dans l'environnement OSI.



Architecture générale de processus d'application.

II.3. Association d'application

Une association d'application (ou AA) est une relation coopérative entre deux AEI pour l'échange d'informations et de coordination. En fonction des besoins de communication des deux AEI, une ou plusieurs AA peuvent être établies. Ces AA peuvent être utilisées simultanément ou séquentiellement par les deux AEI.

On notera, que dans les niveaux OSI de 1 à 6, on parle de *connexion*, alors que le terme choisi pour désigner une connexion au niveau application est celui *d'association* (en effet, les processus d'application *s'associent* pour réaliser un travail commun).

II.4. Elément de service d'application

L'ensemble des fonctionnalités spécifiques à un domaine d'application est appelé ASE (Application Service Element). En d'autres termes, les services de niveau application sont regroupés, selon leurs fonctionnalités, en ensembles appelés ASE. Chaque ASE est défini par un ensemble de services et un protocole normalisés (exemples d'ASE : MMS, FTAM, ACSE, ...).

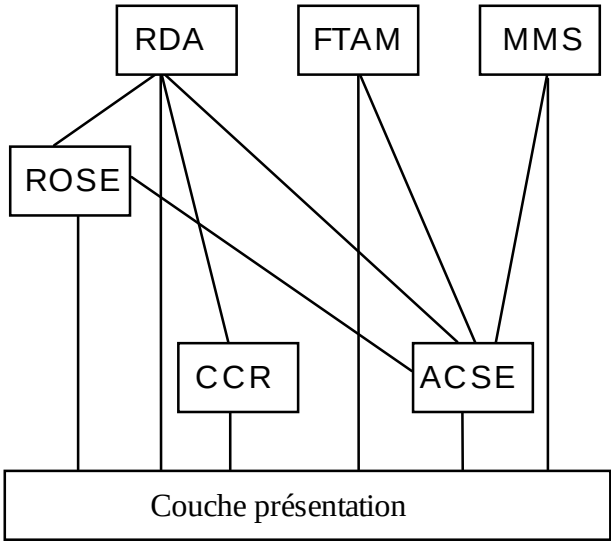
L'ISO préconise la distinction de deux groupes d'ASE : les CASE et SASE.

Les CASE (Common Application Service Elements) sont des ASE de base auxquels beaucoup d'applications font appel. Il y en a quatre : ACSE, CCR, ROSE et RT (voir liste ci-après).

Les SASE (Specific Application Service Elements) sont des ASE qui fournissent des services spécifiques. Le nombre de ASE existants est élevé et d'autres ASE devraient être définis dans le futur. Pour chaque ASE, ce sont deux cents pages en moyenne (voire plus) qu'il faut pour présenter les services et protocoles de cet ASE en détail. Les principaux ASE utilisés seront présentés brièvement par la suite ; il s'agit de :

1) *ACSE* : service de contrôle d'association,
2) *CCR* : service d'engagement, concurrence et reprise,
3) *FTAM* : service de transfert, accès et gestion de fichiers,
4) *RDA* : service d'accès aux bases de données distantes,
5) *TP* : service de traitement transactionnel,
6) *JTM* : service de manipulation et transfert de travaux,
7) *DS* : service d'annuaire (ou répertoire),
8) *ROSE* : service d'opérations distantes,
9) *VT* : service de terminal virtuel,
10) *MMS* : service de spécification de messagerie industrielle,
11) *RT* : service de transfert fiable,
12) *CMIS* : service commun d'informations de gestion,
13) ASE orientés messagerie,
14) ASE orientés documents.

Les ASE ne sont pas complètement indépendants les uns des autres. Par exemple MMS, FTAM et TP utilisent les services d'ACSE pour établir les associations et RDA utilise CCR pour assurer les reprises en cas de panne. Certaines fonctions réalisées par un ASE se retrouvent dans d'autres ASE, par exemple les services de manipulation de fichiers MMS sont ceux définis dans FTAM.

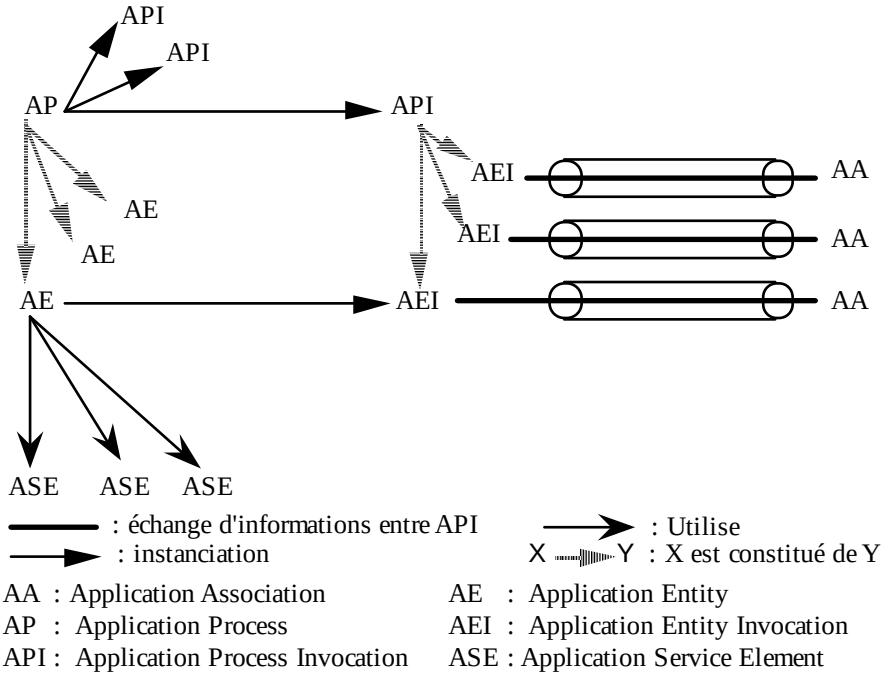


Exemple de coopération entre ASE.

II.5. Contexte d'application

Pour échanger efficacement des données sur une association d'application, la paire d'AEI doit mutuellement reconnaître et suivre un ensemble de règles qui gouverne cet échange. Un accord doit être retenu entre les AEI sur les fonctions à mettre en œuvre sur l'association d'application. L'ensemble des règles agréées est appelé *contexte* d'application de l'AA.

Un contexte d'application englobe l'ensemble des éléments de service, leurs options, leurs conventions de combinaison et de coordination entre eux et avec les services de présentation. Un contexte d'application est négocié lors de l'établissement de l'association d'application.



Schématisation des relations entre AP, API, AE, AEI et AA.

III. ASE Contrôle d'association (ACSE)

III.1. Objectif

L'objectif de ACSE (Association Control Service Element) est la définition de service et la spécification de protocole de contrôle d'associations d'application en mode connecté. ACSE assure l'établissement et la terminaison d'associations d'application, l'identification du contexte d'application applicable aux associations.

III.2. Services de ACSE

- *A-Associate* (service confirmé) : établissement d'une AA ;

- *A-Release* (service confirmé) : pour libérer une AA (cas d'une terminaison normale) ;

- *A-Abort* (service non confirmé) : pour abandonner une AA (demande faite par l'utilisateur) ;

- *A-P-Abort* (service non confirmé) : pour abandonner une AA (demande faite par le fournisseur de service).

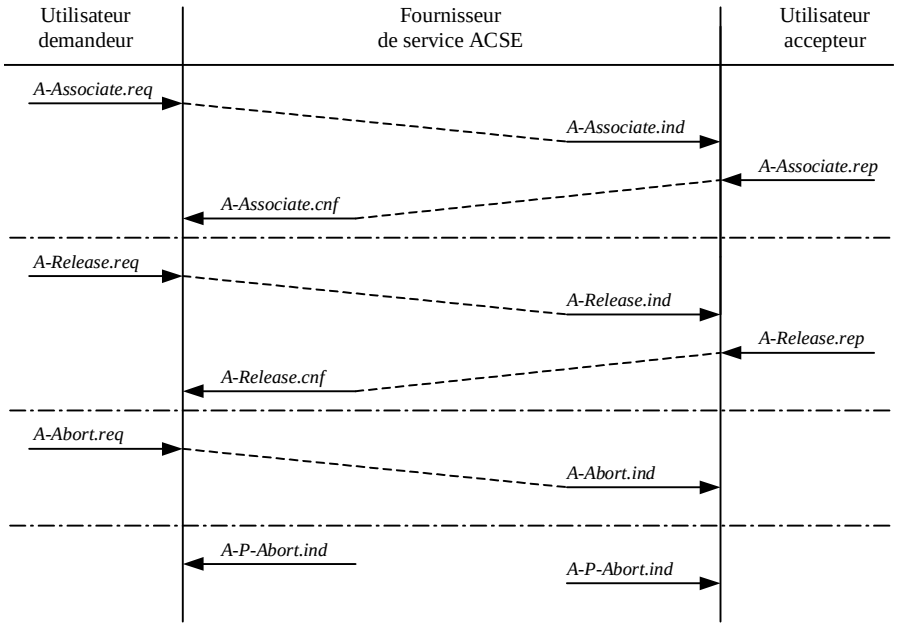


Diagramme d'enchaînement des services de ACSE.

IV. Service d'annuaire (ou service répertoire)

IV.1. Objectif

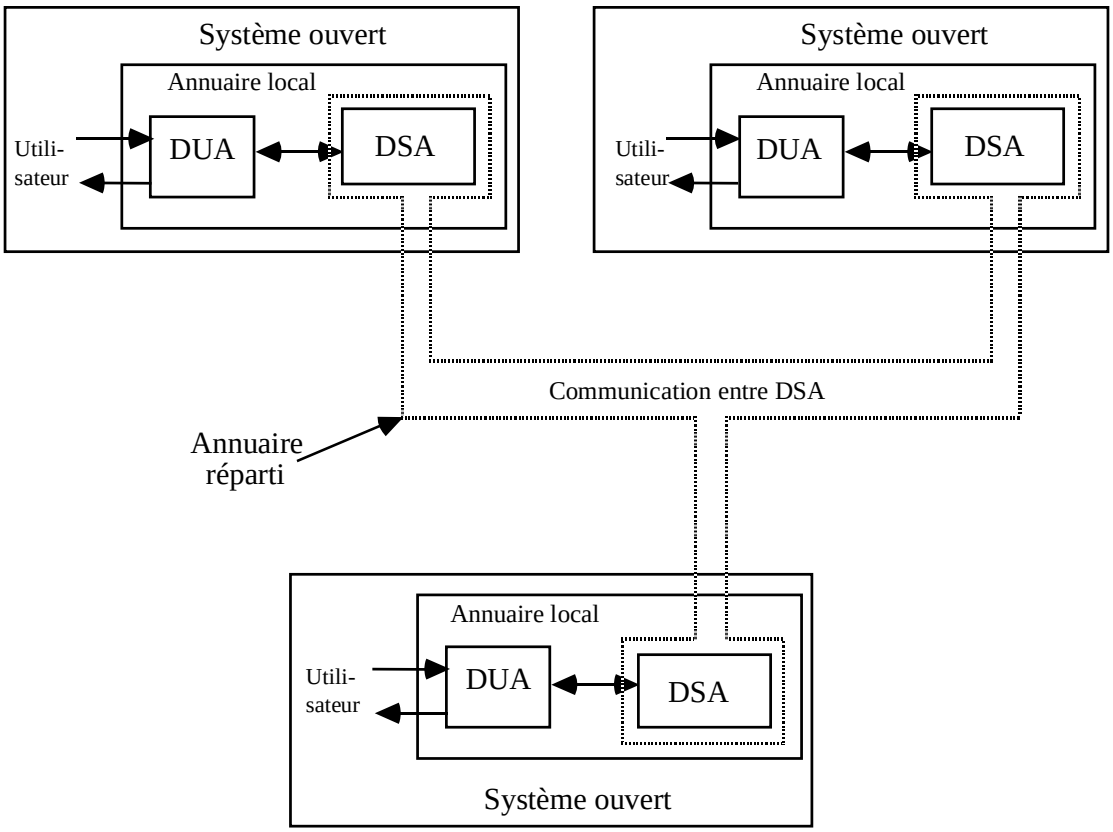
DS (Directory services) offre la possibilité de manipuler des noms mnémotechniques et conviviaux en effectuant une correspondance dynamique entre le nom d'un objet désigné et son adresse réelle.

Le serveur de noms, ou répertoire réseau, est utilisé par des personnes (exploitants réseau) ou par des processus d'application autorisés à manipuler les informations contenues dans la base de données que constitue ce répertoire.

La base de données gérée par les systèmes connectés à un réseau OSI est une collection, structurée en arbre, d'informations relatives à des objets.

A un objet (ou à une entrée du répertoire) correspond une classe d'objet (exemple : pays, organisation, individu, ...) et un ensemble d'attributs typés possédant une ou plusieurs valeurs. Les attributs d'un objet spécifient des accès contrôlés donnant la possibilité de créer des sous-arbres d'information.

L'utilisateur accède aux données de l'annuaire via un DUA (Directory User Agent). Si les données demandées sont disponibles localement, le DUA les retourne à l'utilisateur, sinon le DSA (Directory System Agent) entame une coopération avec les autres DSA pour obtenir l'information demandée. On notera que la base de données de l'annuaire peut être centralisée ou distribuée.



Système d'annuaire réparti.

IV.2. Principaux services de DS

- Lien nom/attribut : met un nom en relation logique avec les informations relatives à un objet (par exemple, les liens <nom, adresse>) ; cette capacité est analogue à celle de l'annuaire téléphonique.

- Lecture : permet d'obtenir la (ou les) valeur(s) de tout ou partie des attributs d'un objet (ou d'une entrée du répertoire).

- Comparaison : permet de vérifier qu'une valeur d'attribut fournie correspond à une valeur attendue pour cet attribut (mot de passe, par exemple).

- Liste : permet de lister les subordonnés immédiats de l'arbre d'information du répertoire pour une entrée particulière.

- Recherche : permet de rechercher dans une partie de l'arbre les entrées qui satisfont les conditions exprimées par le filtre associé à la requête.

- Abandon : permet à un demandeur de service d'abandonner sa requête.

- l'ajout : permet d'ajouter une entrée à l'arbre d'information.

- Suppression : permet de supprimer une entrée dans l'arbre d'information.

- Modification : permet de supprimer, ajouter ou remplacer des attributs ou des valeurs d'attributs.

V. ASE Opérations distantes (ROSE)

V.1. Objectif

ROSE (Remote Operations Service Element) permet aux applications interactives d'exprimer et de réaliser leurs opérations distantes.

Une entité demande à une autre entité l'exécution d'une opération et de lui rendre les résultats de l'exécution (concept du client serveur, mais au niveau d'une opération).

La spécification et la mise en œuvre de protocoles interactifs sont facilitées par le concept d'opérations distantes qui fournit un mécanisme unique de représentation des opérations et de leurs résultats.

V.2. Services de ROSE

- *RO-Invoke* : demande d'exécution d'une opération ;

- *RO-Result* : notifie le succès d'une opération ;

- *RO-Error* : notifie l'erreur d'une opération ;

- *RO-P-Reject* : notifie le refus de l'entité distante d'exécuter une opération ;

- *RO-U-Reject* : notifie le refus de l'usager distant d'exécuter une opération.

VI. ASE de transfert fiable ou RT

VI.1. Objectif

RT (Reliable Transfer) assure le transfert fiable de données et permet de minimiser la quantité d'informations à retransmettre lors d'une reprise après anomalie ou panne.

VI.2. Services de RT

- *RT-Open* : établir une association avec un autre utilisateur de RT ;

- *RT-Close* : fermer une association ;

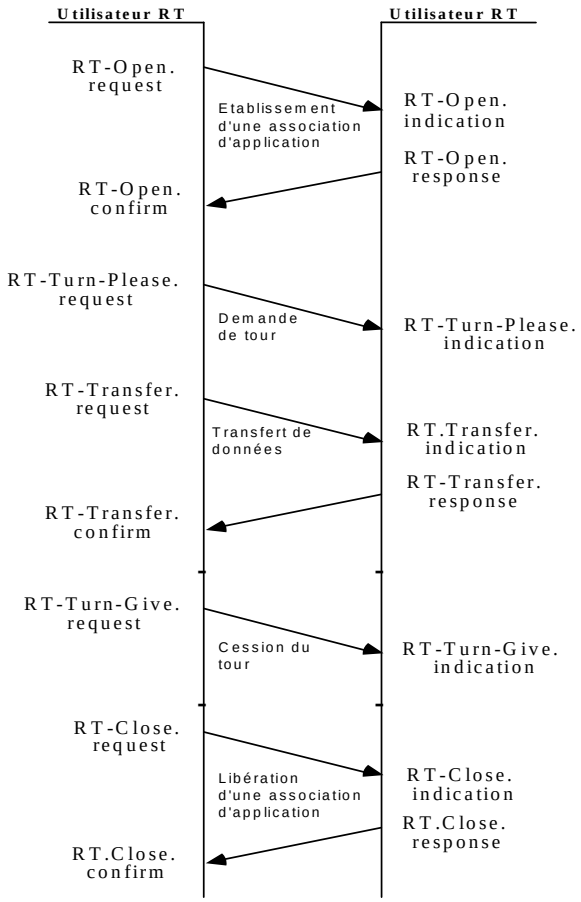
- *RT-Transfer* : demander le transfert d'une unité de données ;

- *RT-Turn-Please* : demander à son correspondant le tour ; seul l'utilisateur qui possède le tour peut émettre des unités de données ;

- *RT-Turn-Give* : abandonner son tour au profit de son correspondant ;

- *RT-P-Abort* : permet au fournisseur de service en cas d'anomalie de signaler à l'utilisateur de RT que l'association ne peut plus être maintenue ;

- *RT-U-Abort* : permet à l'utilisateur d'abandonner une association.



Exemple d'utilisation des primitives de RT.

VII. ASE Transfert, accès et gestion de fichiers (FTAM)

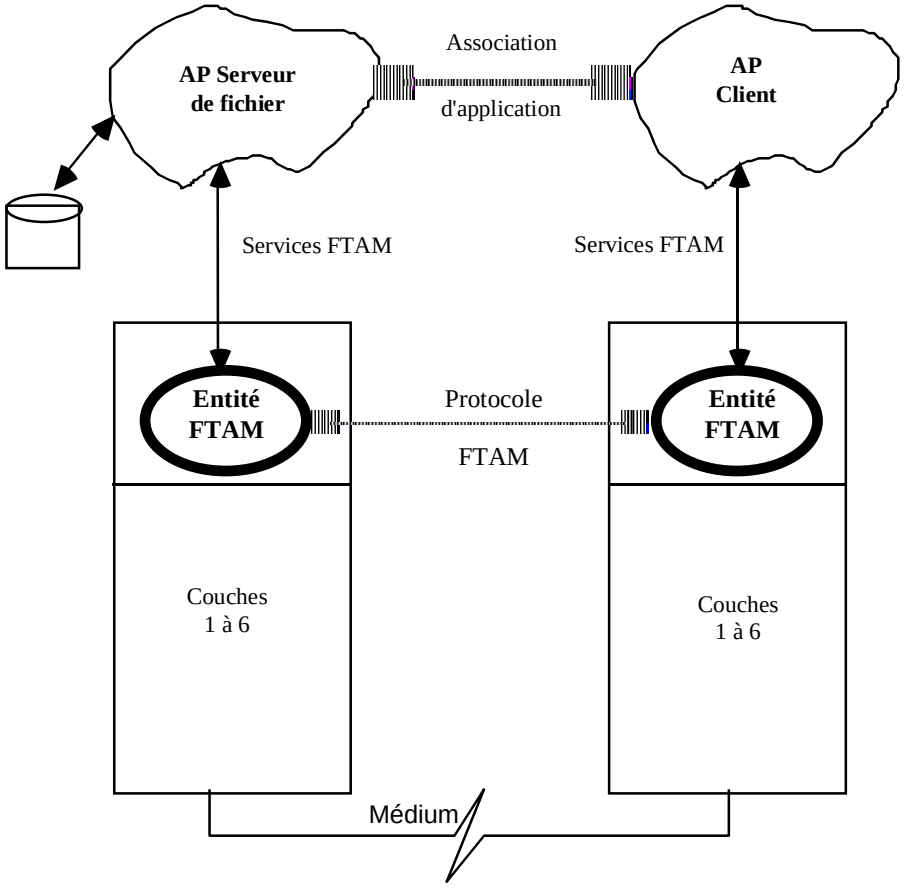
VII.1. Objectif

FTAM (File Transfer Access and Management) assure la gestion des échanges de fichiers entre systèmes ouverts et la manipulation de fichiers avec possibilité d'accès à tout ou une partie d'un fichier.

FTAM permet le transfert partiel ou complet d'un fichier à destination, ou à partir, d'un site distant. Il offre des services de contrôle d'erreurs, de reprise, des mécanismes de contrôle d'accès et de sécurité, d'accès sélectif à des fichiers à structure hiérarchique, de transformation de ces structures, de lecture et de modifications des attributs de fichiers.

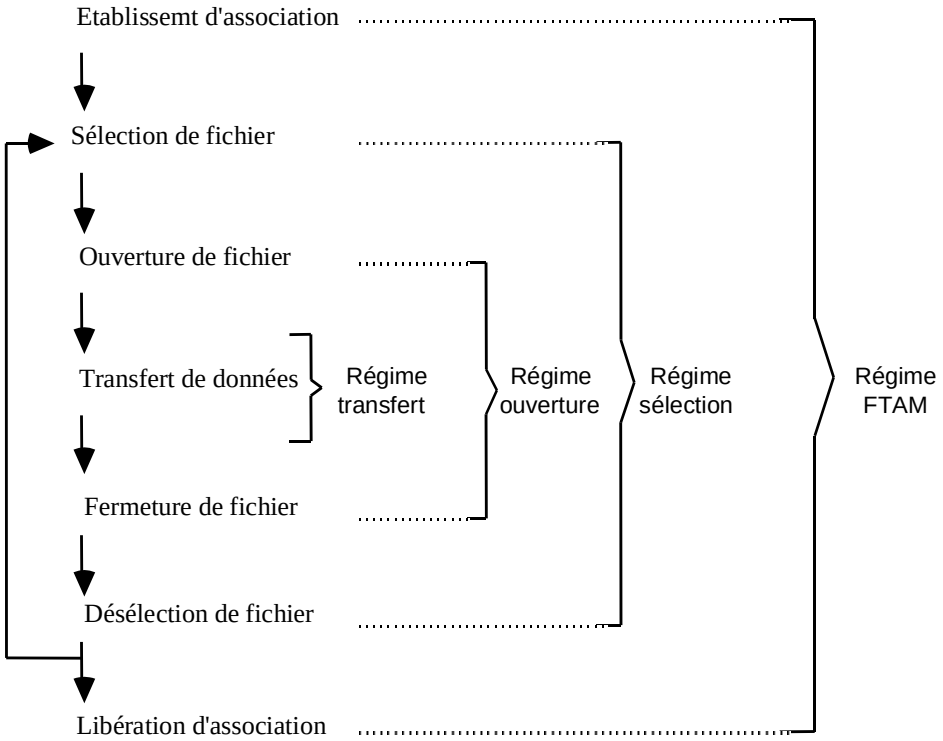
Pour manipuler un fichier indépendamment des choix d'implantation (fichier organisé en séquentiel, direct, séquentiel indexé, … sur bande, sur disque, …), FTAM utilise la notion de système de fichiers virtuels (virtual filestore en anglais) permettant de traiter des représentations abstraites de fichiers seulement.

Chaque fichier est décrit à l'aide d'une liste d'attributs : nom de fichier, opérations autorisées sur le fichier (lecture, écriture, destruction,…), date de création, identité du créateur, les contraintes d'accès… et surtout les structures arborescentes d'accès aux informations du fichier (dites FADU : File Access Data Units). Les FADU décrivent les données d'un fichier sous forme d'une arborescence où les feuilles contiennent les données effectives du fichier.



VII.2. Régimes de FTAM

Les régimes FTAM permettent de préciser les opérations qui peuvent être invoquées pendant une phase donnée d'utilisation de FTAM.



Phases de manipulation de fichier.

VII.3. Unités fonctionnelles de FTAM

Unité fonctionnelle *Noyau*

- établissement de régime FTAM,
- terminaison de régime FTAM,
- sélection de fichier,
- désélection de fichier.

Unité fonctionnelle *Lecture*

- ouverture de fichier,
- fermeture de fichier,
- demande de lecture de données,
- transfert de données,
- fin de régime de transfert de données,
- fin de transfert,
- annulation de transfert de données.

Unité fonctionnelle *Ecriture*

- ouverture de fichier,
- fermeture de fichier,
- demande d'écriture de données,
- transfert de données,
- fin de régime de transfert de données,
- fin de transfert,
- annulation de transfert de données.

Unité fonctionnelle *Accès au fichier*
- localisation des unités de données (c'est-à-dire des enregistrements),
- effacement des unités de données.

Unité fonctionnelle *Gestion réduite de fichiers*
- création de fichier,
- destruction de fichier,
- lecture des attributs de fichier.

Unité fonctionnelle *Gestion étendue de fichiers*
- modification des attributs de fichier.

Unité fonctionnelle *Groupement*
- début de groupe,
- fin de groupe.

Unité fonctionnelle *Verrouillage de FADU*
- verrouillage de FADU.

Unité fonctionnelle *Reprise*
- reprise de régime,
- pose de points de reprise,
- annulation de transfert de données (sur erreurs irrécupérables).

Unité fonctionnelle *Redémarrage de transfert de données*
- redémarrage de transfert de données,
- pose de points de reprise,
- annulation de transfert de données (sur erreurs irrécupérables).

VII.4. Dépendance des couches présentation et session

L'unité fonctionnelle *Noyau* FTAM nécessite l'emploi de :
- l'unité fonctionnelle *Noyau* de la couche présentation,
- l'unité fonctionnelle *Noyau* de la couche session.

L'unité fonctionnelle Reprise FTAM nécessite l'emploi de :
- l'unité fonctionnelle *synchronisation mineure* de la couche présentation,
- l'unité fonctionnelle *synchronisation mineure* de la couche session.

L'unité fonctionnelle Redémarrage FTAM nécessite l'emploi de :

- les unités fonctionnelles *synchronisation mineure* et r*esynchronisation* de la couche présentation,
- les unités fonctionnelles *synchronisation mineure* et *resynchronisation* de la couche session.

VIII. ASE Engagement, Concurrence et Reprise (CCR)

VIII.1. Objectif

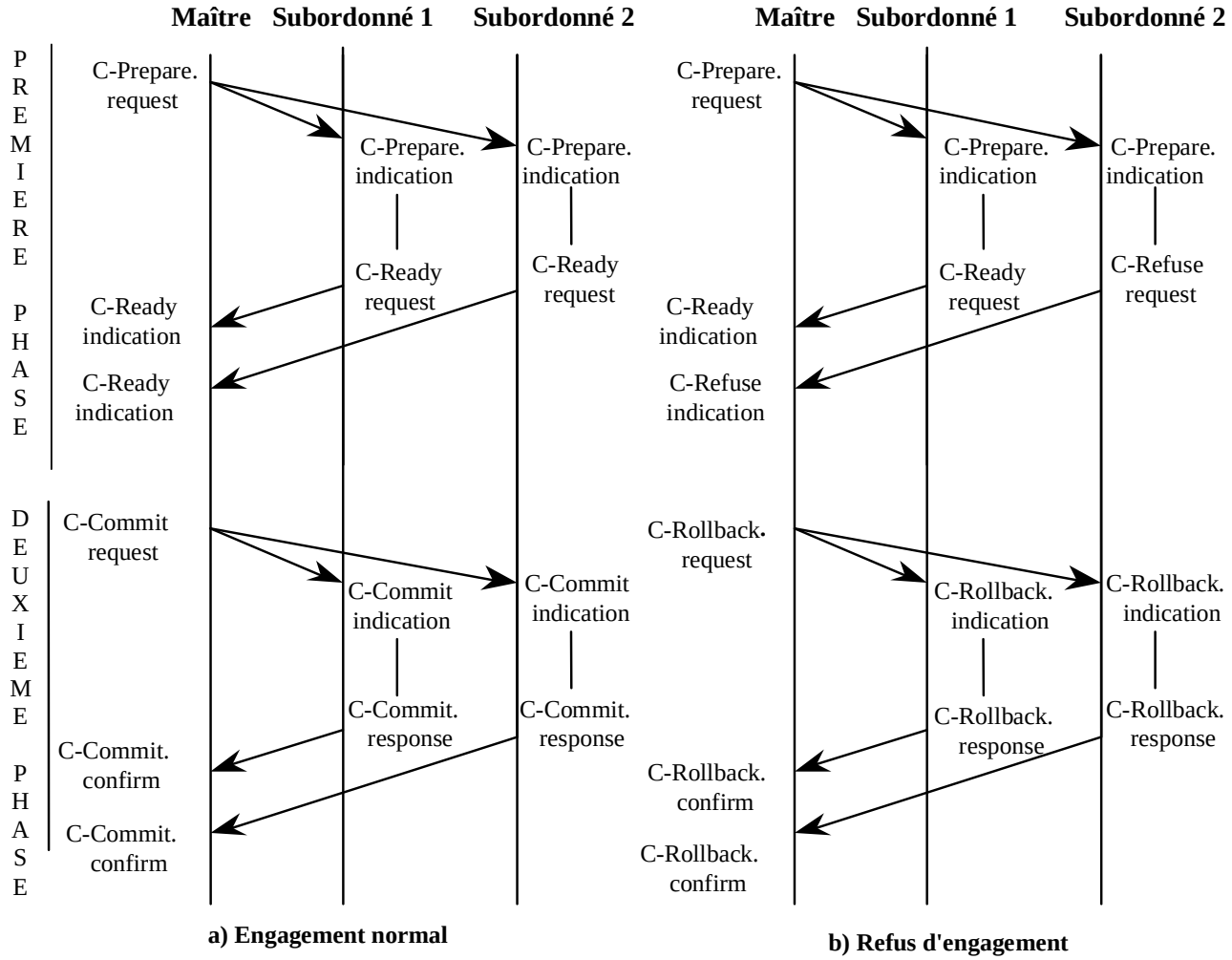
CCR = Commitment, Concurrency and Recovery = Engagement Concurrence et reprise

Le service CCR fournit des services permettant de commencer et de conclure des échanges de protocole et l'activité liée à chaque AA de sorte que la séquence entière apparaisse atomique aux autres applications même en cas de pannes.

Une opération atomique est effectuée par plusieurs sites du réseau. Le contrôle de la validation de l'opération atomique s'effectue en deux phases sous la responsabilité d'un site appelé site *maître* (les autres sites sont appelés sites *subordonnés*).

VIII.2. Services de CCR

- *C-Begin* (confirmé) : initialiser une action atomique ;
- *C-Ready* (non confirmé): appelé par un subordonné pour indiquer au maître qu'il est prêt à accepter l'engagement ;
- *C-Refuse* (non confirmé): appelé par un subordonné pour indiquer au maître qu'il refuse l'engagement ;
- *C-Prepare* (non confirmé): opération utilisée par le maître pour demander aux subordonnés s'ils acceptent ou refusent l'engagement ;
- *C-Commit* (confirmé): opération utilisée par le maître pour demander aux subordonnés ayant accepté l'engagement de valider leurs opérations ;
- *C-Rollback* (confirmé): opération utilisée par le maître pour demander aux subordonnés d'annuler leur travail lié à l'opération atomique ;
- *C-Restart* (confirmé): opération utilisée par un maître ou un subordonné pour réinitialiser l'action atomique.



Exemples d'utilisation des primitives de CCR

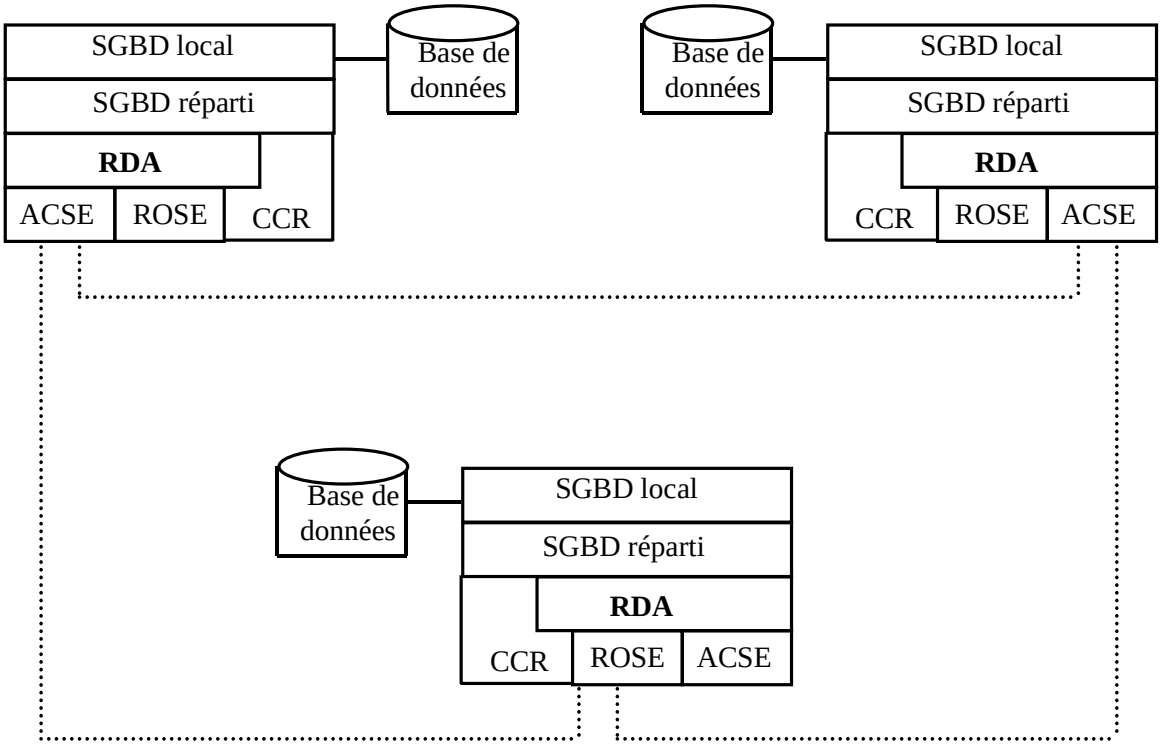
IX. ASE Accès aux bases de données distantes (RDA)

IX.1. Objectif

Le service RDA (Remote Database Access) permet l'accès aux bases de données réparties. RDA prend en charge les accès simples ou transactionnels à des bases de données. Une transaction RDA comprend plusieurs opérations sur une ou plusieurs bases de données. Une transaction RDA est toujours exécutée de manière atomique.

Pour accéder à une base de données centralisée ou à une partie d'une base de données répartie, plusieurs composants doivent être considérés :

- tout d'abord le SGBD local ;
- le SGBD réparti pour accepter et traiter les opérations en provenance ou à destination de sites distants ;
- ACSE : pour l'établissement d'AA entre entités impliquées dans le partage de BD ;
- CCR : pour assurer l'accès cohérent à la BD ;
- ROSE : pour assurer l'exécution des opérations réparties sur les BD.



Exemple de système réparti utilisant le service RDA.

IX.2. Services de RDA

- Primitives de gestion de dialogue
  - + *R-BeginDialogue* : établir un dialogue,
  - + *R-EndDialogue* : terminer un dialogue,
  - + *R-SuspendDialogue* : suspendre un dialogue,
  - + *R-ResumeDialogue* : reprendre un dialogue.

- Primitives de gestion des transactions
  - + *R-BeginTransaction* : commencer une nouvelle transaction,
  - + *R-Commit* : indiquer la phase d'engagement de la transaction,
  - + *R-Rollback* : indiquer que la transaction en cours doit se terminer par un retour-arrière.

- Primitives de contrôle des opérations
  - + *R-Cancel* : annuler toutes ou partie des opérations RDA,
  - + *R-Status* : obtenir le statut d'une ou de plusieurs opérations RDA.

- Primitives de gestion des ressources
  - + *R-Open* : identifier une ressource de données,
  - + *R-Close* : mettre fin à la disponibilité d'une ressource de données.

- Primitives de langage de base de données
  - + *R-ExecuteDBL* : demander l'exécution d'une déclaration DBL (DataBase Language),

+ *R-DefineDBL* : définir une commande DBL, côté client (ou stocker une commande DBL, côté serveur),
+ *R-InvokeDBL* : exécuter une commande DBL stockée,
+ *R-DropDBL* : annuler une ou plusieurs commandes DBL.

X. ASE Traitement transactionnel (TP)

X. Objectif

Le service TP (Transactional Processing) fournit un cadre pour le traitement transactionnel et la coordination des ressources sises sur des systèmes différents.

Les transactions prises en compte ont une propriété multiple : ACID (Atomicité, Cohérence, Isolation, Durabilité).

- Atomicité : toute les opérations d'une unité de tâche doivent être effectuées, soit qu'aucune ne l'est.

- Cohérence : l'ensemble des opérations d'une unité de tâche doit être terminé correctement (c'est-à-dire, conformément à ses spécifications) et les ressources protégées doivent être laissées dans un état cohérent.

- Isolation : l'ensemble des opérations d'une unité de tâche doit s'exécuter sans interférence avec des actions extérieures.

- Durabilité : les effets des opérations d'une unité de tâche ne doivent pas être altérés par une défaillance applicative ou de communication.

X.2. Unités fonctionnelles de TP

- Unité fonctionnelle *Dialogue*
  + début de dialogue TP,
  + fin de dialogue TP,
  + coupure par le fournisseur TP,
  + coupure par l'utilisateur TP,
  + erreur signalée par l'utilisateur.

- Unité fonctionnelle *Contrôle non partagé*
  + demande de contrôle TP,
  + contrôle TP accordé.

- Unité fonctionnelle *Synchronisation*
  + synchronisation TP,
  + synchronisation contrôle accordé TP.

- Unité fonctionnelle *Engagement*
  + engagement TP,
  + rapport de décisions heuristiques TP,
  + actions exécutées TP,
  + engagement réalisé TP,

+ demande de préparation TP,
+ retour arrière TP,
+ retour arrière réalisé TP,
+ fin de dialogue différé TP,
+ contrôle différé accordé TP.

- Unité fonctionnelle *Transaction non enchaînée*
  + transaction TP suivante différée,
  + transaction TP non chaînée,
  + début de transaction TP.

XI. ASE Manipulation et transfert de travaux (JTM)

XI.1. Objectif

JTM (Job Transfer and Management) définit les services nécessaires à la soumission et au transfert du traitement (un job) à effectuer et à la récupération des résultats sur un site.

XI.2. Classes de service JTM

- *Classe de base :*
  + Initialisation de tâche JTM : créer une spécification de tâche pour un mouvement de document,
  + Initialisation de manipulation de tâche JTM,
  + Mise à disposition JTM : transmettre un document à un collectionneur ou à un agent d'exécution,
  + Signal de fin JTM : signaler la fin d'une activité,
  + Etat JTM : obtenir l'état d'une activité,
  + Annulation JTM : annuler une activité,
  + Arrêt JTM : terminer une activité,
  + Message JTM : gérer un message de l'utilisateur.

- *Classe complète :*
  + initialisation de manipulations d'enregistrements de contrôle de transfert JTM,
  + initialisation de manipulation de compte rendu JTM,
  + question JTM : obtenir une liste de noms de documents,
  + suspension JTM : demander la suspension temporaire d'une activité,
  + libération JTM : annuler la suspension d'activité.

XII. ASE Terminal virtuel (VT)

XII.1. Objectif

VT (Virtual Terminal) définit de façon abstraite les services visibles des terminaux réels pour assurer le dialogue entre utilisateurs OSI sans se soucier des caractéristiques physiques des terminaux (nombre de lignes, signification des caractères de contrôle, ...).

VT intervient, entre autres, dans les applications interactives basées sur la transmission et la manipulation d'images graphiques. Le transfert et la manipulation de données se font dans un environnement de terminal virtuel défini par un ensemble de paramètres négociés entre partenaires.

Durant la phase de négociation des caractéristiques d'un environnement de terminal virtuel, on utilise des profils représentant des ensembles de paramètres prédéfinis qui représentent les caractéristiques de VT courants.

XII.2. Services de VT

Trois groupes de services VT :

- *G1 : Noyau*
  - + établissement d'association VT,
  - + terminaison d'association VT,
  - + transfert de données VT,
  - + contrôle de remise VT,
  - + gestion de jeton VT.

G2 : Négociation de changement de profil

+ négociation de changement de profil VT.

G3 : Négociation de plusieurs interactions

+ négociation de plusieurs interactions VT.

XIII. ASE Spécification de messagerie industrielle (MMS)

XIII.1. Objectif

MMS (Manufacturing Message Specification) décrit les fonctions de base d'un système réparti utilisé pour le contrôle/commande de processus manufacturiers (installations industrielles).

MMS utilise des objets abstraits dits VMD (Virtual Manufacturing Device). Chaque équipement réel (robot, automate programmable, machine à commande numérique, etc.) est modélisé par un VMD.

XIII.2. Services de MMS

MMS offre 85 services :

- services de gestion de VMD (connaître l'état d'un, son contenu, …),

- services de gestion de domaines (chargement et déchargement de programmes),

- services de gestion de tâches (lancer, arrêter, … une tâche à distance),

- services d'accès aux variables (lecture et écriture de variables distantes),

- services de gestion de sémaphores (contrôle d'accès aux ressources partagées),

- services de gestion d'événements (notification et utilisation des occurrences des événements pour la synchronisation des tâches),

- services d'E/S physiques à partir de périphériques (capteur, clavier, …),

- services de gestion de journaux,

- services d'accès aux fichiers (lire des fichiers à distance, …).

XIV. ASE orientés messagerie et échange de documents

XIV.1 ASE Transfert de messages

Il assure le transfert fiable de messages en mode stockage/renvoi. Il permet de remettre un message à un ou plusieurs destinataires avant une date donnée.

XIV.2. ASE Classement et de recherche de document

Il fournit les services d'accès à un système de stockage de données de grande capacité. Il ne traite pas le contenu des documents qu'il manipule. Il offre des services analogues à FTAM, mais il diffère de FTAM en ce sens qu'il traite la manipulation de groupes de documents et des relations entre documents, en plus de la manipulation de documents isolés.

XIV.3. ASE Transfert, accès et manipulation de documents (DTAM)

DTAM (Document Transfer, Access and Management) est destiné au transfert de documents (télex, autres). Ce service est destiné à des applications vidéotex. Les services de DTAM permettent d'éditer des données à distance, d'entrer des données à distance en utilisant des prompteurs, de produire automatiquement des tables de matières, des index, des glossaires, …

XIV.4. ASE d'impression

Cet ASE permet à différents utilisateurs de partager des équipements d'impression d'images de haute performance. Les documents à imprimer sont représentés dans un format spécifique : ODIF, SGLL, Postscript, Interpress, SPDL, …

Exercices

Exercice 1
Soit une application de type SGBD réparti dans laquelle un utilisateur met à jour une relation répartie sur trois sites. Ecrire à l'aide de CCR, les appels de services nécessaires pour cette application, en supposant que :
- La première tentative de validation échoue car un des sites refuse de valider.
- La deuxième tentative de validation échoue car la réponse (Ready) d'un des trois sites se perd.
- La troisième tentative de validation réussit.

Exercice 2

Proposer des automates d'états finis pour représenter la communication entre processus d'application qui coopèrent selon le protocole CCR. Différentes hypothèses de fonctionnement peuvent être envisagées (il faut traiter les situations en commençant par le cas le plus simple).

Exercice 3
On considère une application de type client/serveur où les deux processus d'application (le client et le serveur) utilisent ACSE pour établir des connexions. L'échange de données entre les deux processus se fait sans ASE normalisé. Le client se charge de demander l'établissement de la connexion. Ensuite le processus client envoie un message contenant une question et le processus serveur renvoie un autre message contenant la réponse. On suppose que le mode connecté est utilisé au niveau des couches 7 à 2 et qu'il n'y a pas d'erreurs de communication. On vous demande de déterminer le nombre de primitives de services et le nombre de trames échangées sur le support physique pour permettre au client de poser une question et obtenir une réponse.

Exercice 4
Etude de l'architecture générale de construction de RTSE à partir de RPC sous TCP/IP.

Exercice 5
On considère une application client/serveur où le client invoque des opérations sur une BD. On suppose que la longueur maximale Q octets pour une question et R octets pour une réponse. Le mode connecté est utilisé à tous les niveaux. Calculer le rapport entre l'information utile et le total des informations transmises sur le réseau. Selon le type de réseau utilisé les calculs sont différents. Se limiter à un réseau de type FDDI.

Exercice 6
On se place au niveau d'une couche I quelconque (qui peut être la couche application, session transport, etc.). Un échange de données entre deux entités distantes de la couche I (pour transférer un fichier, par exemple) nécessite des transmissions de trames au niveau physique. Expliquer les principaux facteurs qui permettent de déterminer le nombre de trames. Donnez un exemple de formule de calcul du nombre de trames avec des hypothèses à définir selon votre choix.

Exercice 7

Pourquoi distingue-t-on les CASE et les SASE dans la couche application OSI ? Quel(s) lien(s) existe-t-il entre ces deux types d'éléments de service d'application ?.
Dans certaines couches OSI, la notion d'unité fonctionnelle est introduite. A quoi correspond cette notion ? Qu'apporte-t-elle de plus pour les réseaux ?

Chapitre 10
Administration de réseaux

I. Introduction

La gestion (ou administration) de réseau est un instrument essentiel de planification, d'organisation et de gestion de l'informatique dans l'entreprise.

L'administrateur a besoin :

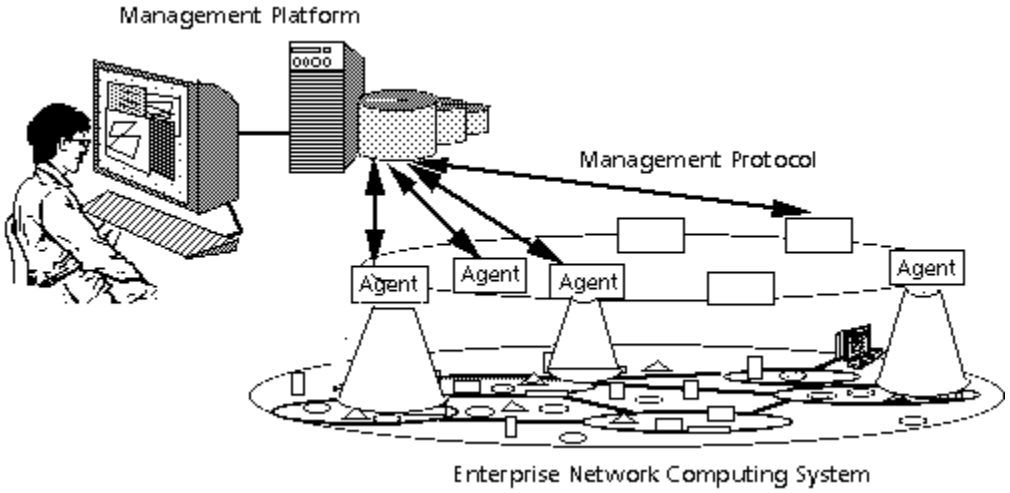
- de données en temps réel pour agir,
- planifier, optimiser, gérer l'évolution du réseau,
- faire des statistiques, comptabilité, …
- contrôler les accès aux ressources du réseau,
- garantir la sécurité,
- etc.

Pour l'administrateur de réseau il s'agit de : trouver un compromis judicieux entre la nature des services offerts aux utilisateurs, la qualité de ces services et les moyens à mettre en œuvre pour assurer le succès de cette politique de qualité.

Les objectifs de la gestion de réseau peuvent être :

- à court terme : exécution d'opérations d'exploitation au quotidien (contrôle de pannes, des utilisateurs, des accès, etc.)
- à moyen terme : coordination, déploiement et mise à niveau du réseau.
- à long terme : planification de l'évolution du réseau (estimation des coûts, etc.)

La fonction d'administration de réseau doit être transparente aux utilisateurs.



Architecture générale de la gestion de réseau

II. Fonctions de gestion de réseau définies par l'ISO

La meilleure façon de garantir une mise en œuvre convenable et une certaine pérennité pour administrer tous les éléments du réseau consiste à adopter une plate-forme d'administration basée sur des standards internationaux. On peut utiliser des plates-formes conformes aux normes OSI ou aux standards TCP/IP. Aujourd'hui le marché est surtout dominé par les outils compatibles aux standards TCP/IP.

Dans le cadre de la normalisation internationale, l'ISO a défini cinq classes de fonctions d'administration de réseaux :

1- Gestion des configurations

- topologie des réseaux (les lignes, les routeurs, les répéteurs, …)
- noms et adresses des stations et des abonnés,
- paramètres de fonctionnement de chaque couche,
- …

2- Gestion des performances

- Mesurer :
  + la fréquence des messages,
  + les tailles des messages,
  + le nombre d'abonnés actifs,
  + …
- Analyser les mesures
  + détecter des points de trafic (heures de pointe),
  + définir le niveau de qualité du service (temps de réponse, taux d'erreur, coût),
  + …
- Optimiser la gestion de trafic
  + modifier les paramètres de gestion des couches,
  + anticiper sur les situations de congestion.

3- Gestion de la sécurité

- Gestion des accès
  + autorisations,
  + identification, …
- Cryptographie (sécurité des informations échangées)
  + ne pas modifier un message pendant son transfert (c'est le risque actif),
  + seuls les destinataires autorisés peuvent utiliser le message(c'est le risque passif),
  + le récepteur ne pourra pas nier avoir reçu un message.

4- Gestion des fautes et anomalies

- détection de fautes, pannes, anomalies,
- traitement,
- prévention de pannes.

5- Gestion de la comptabilité

- taux d'utilisation des équipements (routeurs, lignes, etc.)
- facturation,
- estimation et maîtrise des coûts de communication.

III. Produits d'administration/gestion de réseaux

III.1. Principaux concurrents (pour les réseaux locaux)

- NetWare (Novell) : 60% du marché, une norme de fait,
- Vines (Bayan),
- Lan manger (Microsoft : perçu comme le standard du futur,
- Lan Sever (IBM) : meilleure solution pour relier un réseau de PC à un site central IBM.

III.2. Critère de choix et comparaison des gestionnaires de réseau

- environnements (OS/2, Unix, …) ;
- notoriété du constructeur (Microsoft, …) ;
- présence commerciale, qualité du service après-vente, … ;
- rapidité de traitement ;
- ouverture vers d'autres réseaux et systèmes ;
- qualité de la gestion des erreurs ;
- facilité d'utilisation et de maintenance.

Constructeur et produits	Type de réseaux	Gestion config.	Analyse de trafic	Analyse de protocoles	Gestion d'alarmes	Gestion des coûts	Gestion de la sécurité
Lan Tools Lan map Lan trafic	Netware	■	■				
Farallon Comp. PhoneNet TraficWatch	AppleTalk	■	■				
Cheyenne Soft. Monitrix	Netware	■					
Network Gen. Sniffer	Anneau à jeton et Ethernet	■		■			
Excellan Lan Analyzer	802.3, OSI, TCP/IP, IPX, DecNet		■	■	■		
Hewlett-Pack. HP4972A Lan robe	802.3	■	■	■	■		
DEC LTM			■		■		
Kinetics Lan ranger	Mac Ethernet		■	■			
Blue lance LT Auditor						■	
Lan Services Lan Trail Lan Shadow						■	■
TRW NM 2000	TCP/IP		■	■	■	■	

Exemples outils d'administration de réseaux locaux
(Source :: Télécoms Magazine)

Problèmes :

- Il y a beaucoup d'outils sur le marché (ce qui complique le choix).
- La plupart des outils sont propriétaires.
- La plupart des outils ne répondent pas à tous les besoins des utilisateurs.

IV. Protocoles et services de gestion de réseau

Un système d'administration de réseau se décompose en quatre types d'éléments :

- les équipements administrés,
- une station d'administration,
- une base des informations de gestion associé à chaque équipement administré (MIB : Management Information Base),
- le protocole de gestion de réseau qui permet les échanges entre les équipements administrés et la station d'administration.

Les deux protocoles les plus utilisés pour la gestion de réseau sont :

- CMIP (Common Management Information Protocol) : dans le monde OSI. Le service associé à CMIP s'appelle CMISE (Common Management Information Service Element) ;

- SNMP (Simple Network Management Protocol) : dans le monde TCP/IP.

Services de CMISE (services OSI)

- *M-INITIALIZE* : Etablissement d'association ;

- *M-TERMINATE* : Terminaison d'association ;

- *M-ABORT* : avortement d'association ;

- *M-CREATE* : Création d'objet de gestion de réseau ;

- *M-DELETE* : destruction d'objet de gestion de réseau ;

- *M-GET* : obtention d'information sur des objets de gestion de réseau ;

- *M-CANCEL-GET* : avortement d'une opération M-GET ;

- *M-SET* : modification des attributs d'un objet de gestion de réseau ;

- *M-ACTION* : invocation d'une opération spécifique ;

- *M-EVENT-REPORT :* rapport d'événements sur une opération de gestion de réseau.

Principaux services associés à SNMP (TCP/IP)

- *GET* : obtention d'information sur des objets de gestion de réseau ;

- *SET* : modification des attributs d'un objet de gestion de réseau ;

- *TRAP :* notification d'événement (alarme) à un agent de gestion de réseau. ;

Exemples d'objets de la MIB dans TCP/IP

- Objets « System »
  + *sysDescr* : nom complet et identification de la version du hardware, du logiciel, de l'OS
  + *sysUpTime* : temps écoulé depuis la dernière réinitialisation de la station d'administration du réseau

- Objets « Interfaces »
  + *ifSpeed* : estimation de la largeur de bande courante bits/s
  + *ifPhysAddress* : adresse du niveau inférieur à IP

- Objets « Address translation »
  + *atTable* : table de traduction entre adresse réseau et adresse physique

- Objets « IP »
  + *ipInReceives* : nombre total de datagrammes reçus avec ou sans erreurs
  + *ipInHdrErrors* : nombre de datagrammes rejetés à cause d'erreurs
  + *ipOutNoRoutes* : nombre de datagrammes rejetés à cause de destination invalide
  + *ipFragOKs* : nombre de datagrammes fragmentés

Objets « ICMP »

+ *tcmpInMsgs* : nombre de message ICMP reçus
+ *tcmpInEchos* : nombre de messages « Echo request » reçus

Objets « TCP »

+ *tcpRtoMax* : valeur maximale utilisée pour les retransmissions
+ *tcpRtoMin* : valeur minimale utilisée pour les retransmissions
+ *tcpMaxCOnn* : nombre maximum de connexions simultanées
+ *tcpOutSegs* : nombre total de segments envoyés

Objets « UDP »

+ *udpInDatagramms* : nombre de datagrammes reçus
+ *udpNoPorts* : nombre de datagrammes reçus pour lesquels il n'y avait pas d'application au port de destination

Chapitre 11
Introduction à IPv6

I. Introduction

Le protocole IP a été conçu, il y a plus d'une vingtaine d'années, pour connecter des millions d'ordinateurs. Depuis quelques années, IP est victime de son succès et ne permet plus de répondre à la demande de connexion de milliards de machines informatisées dont disposeront les internautes de demain.

La nouvelle génération de IP, IPng (next generation), ou IPv6 va offrir de nouvelles capacités d'adressage, des options de sécurité, et bien d'autres fonctionnalités qui vont faciliter les interconnexions globales.

IPv6 a été recommandé par les responsables de la nouvelle génération du protocole Internet de l'IETF (Internet Engineering Task Force) au cours du meeting de l'IETF de juillet 1994 à Toronto et définitivement spécifié par la recommandation RFC 1752.

Il est important de souligner que les changements des protocoles comme TCP ou IP affectent fatalement les applications existantes. En conséquence, de tels changements doivent être effectués avec précaution et seulement quand ils deviennent nécessaires.

II. Classes d'adresses IPv4

Parmi les différentes insuffisantes que l'on pourrait attribuer à IPv4, c'est essentiellement sa politique des gestion des adresses qui est la plus cruciale et qui a motivé le passage à IPv6.

Une adresse IPv4 est représentée sur 4 octets, soit une capacité, en théorie, de coder un peu plus de quatre milliards d'adresses.

C'est le modèle de structuration des adresses qui est à l'origine des limites de IPv4. En effet, le champ adresse est composé de trois champs : classe d'adresse, numéro (préfixe) de réseau et numéro de hôte. On parle de classes A, B, C et D. Les adresses IP ne sont pas allouées de manière efficace : il y a eu du gaspillage. Par exemple, un réseau qui contient 256 stations nécessite des adresses de classe B qui monopolise 2^{16} adresses au lieu de 2^8 (s'il avait 255 stations, un réseau de classe C aurait suffit, sans aucun gaspillage).

Le second problème de IPv4 est lié à la taille des tables de routage. Les routeurs fédérateurs doivent maintenir des tables de routage nécessitant des tailles mémoire énormes. Malheureusement, le problème du routage ne peut pas être résolu simplement en augmentant la taille mémoire allouée aux tables de routage. Il faut aussi tenir compte du temps de traitement des tables qui peut influer considérablement sur le temps de routage (donc sur le temps de transit des paquets).

Deux solutions ont été recommandées pour faire face aux insuffisances de IPv4 tout en gardant la même longueur pour les adresses :

- Subnetting (création de sous-réseaux) en décomposant le champ *Numéro Réseau* en deux, *Préfixe Réseau* et *Préfixe Sous-réseau*
- CDIR (Classless Inter-Domain Routing) : routage sans notion de classe ce qui permet une allocation d'adresses plus efficace.

Même avec ces solutions, les adresses IP vont manquer (prévision entre 2000 et 2018). Par conséquent, le futur d'Internet passe par une reconsidération fondamentale de l'adressage dans le protocole IP.

III. Principales extensions apportées par IPv6

IPv6 a été conçu comme une évolution de IPv4 et non comme un changement radical du protocole IP. Par conséquent, les applications fonctionnant sous IPv4 devraient fonctionner normalement sous IPv6.

Les changements entre IPv4 et IPv6 peuvent être classées de la manière suivante :

- capacités d'adressage et de routage étendues : la taille des adresses passe de 32 à 128 bits ;
- introduction d'un nouveau type d'adressage appelé *anycast* permettant d'identifier un groupe de machines et un paquet envoyé à une adresse *anycast* est délivré à une des machines appartenant au groupe désigné par l'adresse ;
- simplification du format de l'entête de paquet : pour réduire le coût de traitement des entêtes, certains champs ont été supprimés et d'autres sont rendus optionnels ;
- possibilités de définition de la qualité de service demandée par certains types d'applications (les applications temps réel notamment) ;
- capacités d'authentification et de confidentialité.

IV. Système d'adressage

IV.1. Types d'adresses

IPv6 définit trois types d'adresses :

- *Unicast* : une adresse pour chaque interface (équipement). Un paquet envoyé à une adresse *unicast* est délivré à une seule interface.
- *Anycast* : une adresse désigne un groupe d'interfaces. Un paquet envoyé à une adresse *anycast* est délivré à une des interfaces identifiées par l'adresse *anycast*.
- *Multicast* : une adresse désigne un groupe d'interfaces. Un paquet envoyé à une adresse *multicast* est délivré à toutes les interfaces identifiées par l'adresse *multicast.*

Une interface (c'est-à-dire un équipement, un ordinateur, …) peut avoir plusieurs adresses de types éventuellement différents (*unicast*, *anycast* et *multicast*).

Il n'y a pas d'adresses *Broadcast*, comme dans IPv4, car leur fonction est réalisée par les adresses *multicast*. De plus, la fonction *broadcast* est pénalisante, car elle nécessite un certain traitement pour chaque nœud, même si celui-ci va ignorer le paquet diffusé en *broadcast*. Le *multicast* cible certains nœuds seulement, ce qui est plus économique.

L'adressage IPv6 permet de regrouper les adresses hiérarchiquement, par réseau, par fournisseur d'accès Internet, géographiquement, par société, etc. De tels regroupements devraient permettre de diminuer la taille des tables de routage et d'accélérer le traitement au niveau des routeurs.

Le type spécifique d'une adresse est indiqué par les premiers bits de cette adresse. Par exemple, les préfixes suivants sont définis :

- Adresses globales de fournisseurs d'accès : préfixe = 010

- Adresses globales géographiques : préfixe = 100
- Adresses *unicast* sur lien local : préfixe = 1111 1110 10
- Adresses *multicast* : préfixe = 1111 1111

Actuellement, près de 85% de tout l'espace d'adressage reste disponible pour le futur.

Les adresses IP peuvent être écrites de trois manières :

- une forme hexadécimale complète : **X :X :X :X :X :X :X** où chaque **X** représente une valeur sur 16 bits ;
- une forme hexadécimale abrégée qui ressemble à la forme précédente mais dans laquelle les valeurs **X** égales à 0 sont condensées comme dans l'exemple suivant (attention l'abréviation :: ne peut apparaître qu'une seule fois dans une adresse) :
  1 :0 :0 :0 :0 :0 :0 :15 s'écrit en forme condensée **1 ::15 ;**
- une forme permettant le rapprochement entre adresses IPv4 et adresses IPv6 qui s'écrit sous la forme : **X :X :X :X :X :X :d.d.d.d** où chaque **X** représente une valeur sur 16 bits et chaque **d** représente une valeur sur 8 bits. Par exemple au lieu d'écrire l'adresse IPv4 **0 :0 :0 :0 :0 :0 :194 :12 :5 :01** avec des zéros on l'écrit de la manière suivante ::**194.12.5.01**

Une adresse IPv6 qui contient une adresse IPv4 commence par une série de 96 bits à zéro.

0 :0 :0 :0 :0 :0 :0 :0 (ou ::) est appelée adresse non spécifiée. Elle ne doit être assignée à aucun nœud et ne peut être utilisée comme adresse de destination.

0 :0 :0 :0 :0 :0 :0 :1 (ou ::**1**) est appelée adresse de *loopback* (bouclage) et peut être utilisée par nœud pour s'envoyer un paquet à lui-même. Cette adresse est l'équivalent de l'adresse **127.0.0.1** dans IPv4.

IV.2. Plan d'adressage

a) Adresses globales d'ensemble unicast

Plusieurs manières de hiérarchiser les adresses IP ont été proposées. La dernière proposée à l'IETF est dite "Aggregatable Global Unicast Address Format" ou plan d'adressage agrégé. Ce plan hiérarchise une adresse IP de la manière suivante :

010	TLA (13 bits)	NLA (32 bits)	SLA (16 bits)	Id Interface (64 bits)

Plan d'adressage agrégé.

- un champ égal à 010 (pour indiquer une adresse *unicast*)
- *TLA (Top Level Aggregator)* : les TLA identifient les grands opérateurs internationaux,
- *NLA (Next Level Aggregation)* : les NLA identifient les opérateurs intermédiaires échangeant leur interconnectivité en des points d'interconnexion. NLA constitue un identificateur de site (ou domaine),
- *SLA (Site Level Aggregator)* : permet de hiérarchiser le plan d'adressage de site (définir les sous-réseaux),
- identificateur d'interface.

b) Adresses unicast de lien local

Ces adresses sont destinées à l'utilisation sur un lien unique pour des tâches telles que la découverte des voisins ou lorsqu'il n'y a pas de routeur. Leur utilisation est donc restreinte à un lien (par exemple, l'ensemble des machines reliées par un réseau Ethernet). Les routeurs ne doivent pas transmettre les paquets contenant ce type d'adresses.

1111111010	54 bits à zéro	Id d'interface (64 bits)

Adresse lien local.

c) Adresses unicast de site local

Ces adresses sont destinées à l'utilisation sur un site unique sans l'utilisation d'un préfixe global. Par exemple, un site non encore connecté à Internet peut utiliser ces adresses, ce qui lui évitera de demander un préfixe de réseau. C'est en quelque sorte des adresses IP privées. Les routeurs ne doivent pas transmettre des paquets avec ce type d'adresse en dehors du site concerné. Plusieurs sous-réseaux peuvent être identifiés dans un site.

1111 1110 10	38 bits à zéro	Id sous-réseau (16 bits)	Id d'interface (64 bits)

Adresse site local.

d) Adresses anycast

Un paquet destiné à une adresse *anycast* (donc à un ensemble d'interfaces) est délivré à l'interface la plus proche ayant cette adresse selon la mesure de distance (nombre de routeurs à traverser, temps de transmission, etc.) du protocole de routage utilisé.

Les adresses *anycast* sont syntaxiquement indistinguables des adresses *unicast*. Lorsqu'une adresse *unicast* est attribuée à plus d'une interface, elle devient une adresse *anycast* et le nœud auquel cette adresse est attribuée doit être configuré pour savoir qu'il s'agit d'une adresse *anycast*.

Un usage prévu pour les adresses *anycast* est l'identification des groupes des routeurs appartenant à une entreprise fournissant un accès à Internet, ce qui permet de banaliser l'accès aux routeurs de cette entreprise. L'expérience de l'utilisation large des adresses *anycast* reste pour le moment assez limitée.

Préfixe de sous-réseau (n bits)	128 - n bits à zéro

Adresse *anycast* pour les routeurs de sous-réseaux.

e) Adresses multicast

Une adresse *multicast* identifie un groupe de nœuds (interfaces). Un même nœud peut appartenir à plusieurs groupes *multicast*.

1111 1111	Flag (4 bits)	Scope (4 bits)	Identificateur de groupe (112 bits)

Adresse *multicast*.

- *Flag* (drapeau) : contient 0000 pour une adresse permanente (qui est affectée par une autorité compétente de l'IETF) et 0001 pour une adresse temporaire. Par exemple, une adresse multicast est allouée de manière temporaire à un ensemble de participants le temps d'une téléconférence.

- *Scope* (champ d'action de l'adresse) = 0 : réservé, 1 : champ d'action défini par le nœud local, 2 : champ d'action défini sur le lien local, 5 : champ d'action défini sur le site local, 8 : champ d'action défini sur l'organisation locale, E : champ d'action global, toutes les autres valeurs ne sont pas encore assignées. Le champ *Scope* permet de garantir le confinement des paquets dans une zone déterminée et éviter ainsi que des paquets associés par exemple à une téléconférence se dispersent sur tout le réseau mondial.

IV.3. Auto-configuration des adresses

Pour s'adapter à des évolutions d'interconnexion pour plusieurs décennies, IPv6 est conçu pour faciliter la configuration automatique des adresses.

Les capacités d'auto-configuration sont importantes, que l'allocation d'adresses soit géographique ou basée sur des fournisseurs d'accès. Il peut être, par moment, nécessaire de renuméroter les adresses des machines d'une organisation (suite à une délocalisation de l'entreprise, à un changement de fournisseur d'accès, etc.)

L'auto-configuration utilise le protocole ND (Neighbor Discovery) de découverte de voisin. Dans un scénario typique, un hôte débute le processus d'auto-configuration par s'auto-attribuer une adresse de lien local pour un usage temporaire. Une fois que cette adresse est formée, l'hôte envoie un message ND vers cette adresse pour s'assurer qu'elle est bien unique. Si aucun message ICMP ne revient en retour, cela signifie que l'adresse est bien unique. Dans le cas contraire, l'hôte doit essayer une autre adresse. Utilisant cette nouvelle adresse de lien local comme adresse source, l'hôte envoie une requête ND de sollicitation de routeur. La sollicitation est envoyée en utilisant le service *multicast*. Les routeurs répondent aux requêtes de sollicitation par un paquet qui contient l'intervalle des adresses valides pour le sous-réseau. Les routeurs envoient également des annonces (paquets contenant les intervalles d'adresses valides) de manière périodique aux groupes *multicast* locaux sans avoir reçu de sollicitation. Cela permet aux routeurs de contrôler si les hôtes utilisent ou non l'auto-configuration.

IV.4. Mobilité dans IPv6

La mobilité dans IP s'adresse aussi bien à la mobilité des ordinateurs de bureau portables, qu'aux équipements enfouis (embarqués) dans les voitures, avions, etc.

Pour faciliter la mobilité des équipements, IPv6 offre la possibilité à un équipement de maintenir une connexion avec son adresse de base (adresse mère), tout en se déplaçant. Avant de partir en déplacement, les utilisateurs pourront demander à leur routeur de détourner leur trafic vers une adresse externe au sous-réseau. L'adresse externe est recalculée pour chaque sous-réseau externe visité. Cela permet de ne pas toucher aux entrées de DNS (Domain Name Service) pour retrouver les objets, en cas de mobilité.

Un mobile est toujours identifié par son adresse principale (appelée aussi adresse mère). Tant que le mobile se trouve dans son sous-réseau d'origine (sous-réseau mère), les paquets qui lui sont destinés sont délivrés en utilisant les mécanismes de routage conventionnels (c'est-à-dire en utilisant le préfixe de réseau). Lorsque le mobile est rattaché à un sous-réseau étranger, il devient joignable par une ou plusieurs adresses temporaires, en plus de son adresse mère. Les adresses temporaires sont obtenues par le mécanisme d'auto-configuration. La liaison entre une adresse mère et une adresse temporaire est appelée association.

Lorsqu'un mobile envoie un paquet alors qu'il se trouve hors de son sous-réseau mère, il positionne généralement comme adresse source une de ses adresses temporaires et ajoute dans une option destination son adresse principale.

Pour supporter la mobilité, il est nécessaire de disposer de structures de données qui servent à maintenir les associations des mobiles.

L'environnement informatique d'un mobile est différent de celui des environnements informatiques habituels. En particulier, dans beaucoup de cas, les mobiles sont connectés au réseau sans fil, ce qui les rend particulièrement vulnérables aux écoutes et aux différentes attaques. Il est parfois nécessaire de cacher la position d'un mobile.

V. Format de paquet et fonctionnalités de IPv6

Les paquets IPv6 ont la forme générale suivante :

Entête IPv6 (40 octets)	Extension de l'entête (0 ou n octets)	PDU – niveau transport

.**Format de paquet IPv6.**

L'entête IPv6 a la forme suivante :

Version (4 bits)	Priorité (4 bits)	Indentificateur de flux (24 bits)	
Longueur de données (16 bits)		Entête suivant (8 bits)	Nombre de sauts (8 bits)
Adresse source (16 octets)			
Adresse Destination (16 octets)			

Format de l'entête IPv6.

- *Version* : égale à 6 pour IPv6,
- *Priorité* : valeur de la priorité du paquet,
- *Indentificateur de flux* : utilisé pour marquer les paquets pour lesquels un traitement spécial doit être fait par les routeurs,
- L*ongueur de données :* longueur du reste du paquet (extension de l'entête et données de niveau transport),
- *Entête suivant* : identifie le type de l'entête qui suit l'entête IPv6,
- *Nombre de sauts* : nombre de sauts (de routeurs) restant pour le paquet avant destruction.

On notera qu'il n'y a plus de bits de contrôle ("check sum") de l'entête du paquet comme dans le cas de IPv4. La raison est que les réseaux physiques sont de meilleure qualité aujourd'hui, ils vérifient eux-mêmes les erreurs de transmission sur les trames qui contiennent les paquets. Par conséquent, supprimer le contrôle des erreurs sur l'entête diminue le temps de calcul des paquets par les nœuds intermédiaires. Pour se prémunir contre des paquets routés par erreur (erreur non détectée par le réseau physique) un contrôle doit être fait au niveau transport. La solution retenue actuellement consiste à effectuer un contrôle (en utilisant des bits de contrôle calculés par la source et contrôlés par la destination) par la couche transport.

V.1. Le champ Priorité

Deux classes de trafics peuvent être générés par les applications : un trafic sujet à contrôle de congestion et un trafic non sujet à contrôle de flux. Le champ *Priorité* permet de fixer la priorité de transmission du paquet. En cas de congestion du réseau (saturation des tampons de réception de routeur), les paquets moins prioritaires sont écartés. Il y a 6 catégories de trafic avec contrôle de congestion classées par ordre de priorité décroissant :

- Trafic de contrôle Internet (priorité 7) : c'est le trafic le plus important distribuer.
- Trafic interactif (priorité 6) : utilisé pour les sessions interactives comme Telnet, X-window, etc. Le délai de communication doit être minimisé pour ce type de trafic,
- Trafic de masse assisté (priorité 4) : un exemple des applications qui génère ce type de trafic (sporadique, mais important en volume) est FTP ou NFS. C'est un type de trafic où l'utilisateur attend la fin du transfert pour poursuive son travail
- Trafic de données non assisté (priorité 2) : un exemple des applications qui génèrent ce type de trafic est le courrier électronique. C'est un type de trafic où l'utilisateur n'attend pas la fin de transfert pour continuer son travail.
- Trafic de remplissage (priorité 1) : trafic pour applications en tâche de fond, comme les news ;
- Trafic non caractérisé (priorité 0) : aucune information n'est connue sur le type de trafic.

Le trafic sans contrôle de flux est un trafic pour lequel un débit constant de données, un délai de distribution constant ou non sont désirés. Les exemples d'applications générant ce type de trafic rentrent dans le domaine de l'audio et de la vidéo en temps réel. Huit niveaux de priorités sont définis pour cette classe de trafic (du trafic le plus facile à détruire jusqu'au trafic le plus difficile à détruire). Par exemple, pour le trafic basse fidélité (conversations téléphoniques, par exemple), la perte de quelques paquets perturbe la communication. Par contre, la perte de plusieurs paquets pour le transfert d'un signal vidéo haute fidélité ne perturbe pas énormément la communication. Les valeurs de priorités associées à ces types de trafics sont supérieures à 7.

Il n'y a pas de relation de priorité entre trafics appartenant aux deux classes. La priorité s'applique à l'intérieur d'une même classe de trafic.

V.2. Le champ Identificateur de flux

L'échange entre deux équipements, pour réaliser une tâche donnée (par exemple, l'envoi d'un film, une téléconférence, etc.) est modélisé par un flux de données ayant certaines caractéristiques. Un flux est identifié par une adresse source et un numéro de flux. Les caractéristiques d'un flux conditionnent le routage des paquets correspondant à ce flux. Un traitement spécial doit être déclaré pour chaque flux de paquets. Les traitements associés aux flux particuliers sont généralement définis au moyen des extensions de l'entête.

Les routeurs doivent mémoriser les numéros des Identificateurs de flux qui les traversent pour servir, le plus possible, de la même manière les paquets associés à un même flux.

Le champ *Priorité* et *Identificateur de flux* devraient être pris en compte par les routeurs pour garantir une certaine qualité de service selon les besoins des applications.

V.3. Nombre de sauts

Ce nombre est appelé durée de vie (ou Time to Live) dans IPv4. Il est décrémenté par chaque routeur que le paquet traverse. Quand la valeur atteint 0, le paquet est rejeté avec l'émission d'un message ICMP vers la source. La valeur initiale de ce nombre n'est pas encore fixée, mais certaines implantations prennent la valeur 64.

V.4. Extensions

Plusieurs options d'extension peuvent être intégrées à un paquet. Chaque extension commence par un champ qui indique l'emplacement de l'extension suivante. Le champ *Entête suivant* dans l'entête IPv6 indique l'existante ou non des extensions. Chaque extension est définie par un entête. Le RFC 1883 recommande l'ordre suivant pour les extensions :

- option *Proche-en-Proche,*
- option *Destination* (traité par les routeurs),
- option *Routage,*
- option *Fragmentation,*
- option *Authentification,*
- option *Sécurité,*
- option *Destination* (informations optionnelles examinées par le nœud de destination).

a) Option Proche-en-proche

La partie optionnelle *Proche-en-proche* (hop-by-hop en anglais) de l'entête transporte, quand elle est présente, les informations qui doivent être examinées par chaque routeur le long du chemin vers la destination. Ce champ comporte une ou plusieurs définitions. Chaque définition contient : le type d'option (8 bits), la longueur (8 bits) du champ *Données* de l'option et les *Données* de l'option. Le type d'option désigne l'un des cas suivants :

- ignorer l'option et continuer le traitement de l'entête,
- détruire le paquet,
- détruire le paquet et envoyer un message ICMP d'inaccessibilité à la source,
- détruire le paquet et envoyer un message ICMP d'inaccessibilité à la source, si l'adresse de destination n'est pas une adresse *multicast.*

Les définitions d'options sont en cours d'étude.

b) Option Routage

Cette extension permet d'imposer à un paquet une route (éventuellement) différente de celle offerte par les politiques de routage présentes dans le réseau. Le routage par la source peut être utilisé à des fins de sécurité ou pour accroître les performances (temps de transit) pour garantir certaines exigences en terme de qualité de service.

L'entête de routage contient la liste d'un ou de plusieurs nœuds intermédiaires à traverser avant d'arriver au nœud de destination. Le format de l'entête de routage est le suivant :

Entête suivant (8 bits)	Longueur Entête (8 bits)	Type de routage (8 bits)	Segments restants (8 bits)
Données de type spécifique			

Format général de l'extension de routage.

- *Entête suivant :* identifie l'entête qui suit immédiatement,
- *Longueur entête* : indique la longueur de l'entête en multiple de 8 octets,
- *Type de routage :* indique la variante de routage. Si un routeur ne connaît pas la variante de routage, il détruit le paquet,
- *Segments restants* : nombre de nœuds intermédiaires explicitement spécifiés à visiter avant d'atteindre la destination finale (ce nombre est décrémenté par chaque nœud traversé).

Pour le moment, seul l'entête de type 0 est défini, il a la forme suivante :

Entête suivant (8 bits)	Longueur Entête (8 bits)	00000000	Segments restants (8 bits)
Réservé (8 bits)	Carte bits strict/souple (24 bits)		
Adresse 1 (16 octets)			
….			
Adresse n (16 octets)			

Format de l'extension de routage de type 0.

Les bits du champ *Carte bits strict/souple* doivent être considérés de gauche à droite. Chaque bit correspond à un saut dans le routage. Si le bit considéré est égal à 1 (strict), cela signifie que le prochain routeur doit être un voisin directement accessible à partir du nœud actuel. Si le bit considéré est égal à 0 (souple), il n'est pas nécessaire que la prochaine destination soit voisine de nœud actuel.

Quand le routage utilisé est de type 0, la source ne place pas l'adresse de destination finale dans l'entête IPv6. Dans ce cas, l'adresse de destination est la dernière adresse listée dans l'entête de routage (adresse n dans le format précédent) et l'entête IPv6 contient alors l'adresse du premier nœud à traverser. L'entête de routage ne sera pas examiné tant que le paquet n'a pas atteint le nœud spécifié dans l'entête IPv6. Quand le paquet atteint le nœud spécifié comme destination dans l'entête IPv6, le contenu du paquet et son entête de routage sont actualisés, cela consiste à placer la prochaine adresse à visiter dans l'entête IPv6 et à décrémenter le champ *Segments restants*.

c) Option Fragmentation

Avec IPv4, la fragmentation peut s'effectuer à n'importe quel endroit du chemin entre la source et la destination d'un paquet. Avec IPv6, la fragmentation, devrait seulement être accomplie par les nœuds source et les routeurs. En utilisant un algorithme de découverte de chemin, une source peut déterminer la plus petite valeur des MTU (Maximum Transmission Unit) supportée par chaque réseau se trouvant entre cette source et la destination. Cela permet à la source d'effectuer, de manière plus efficace, la fragmentation à la source et de réduire, par conséquent, le travail des routeurs intermédiaires.

Le format de l'entête de fragmentation est le suivant :

Entête suivant (8 bits)	Réservé (8 bits)	Décalage Fragment (13 bits)	Réservé (2 bits)	M (1 bit)
Identification (32 bits				

Format de l'extension de fragmentation.

- *Décalage Fragment* : indique l'emplacement (en multiples de 8 octets) où se trouvent les données dans le paquet original,
- *M* : = 0 (c'est le dernier fragment), = 1 (d'autres fragments vont suivre),
- *Identification* : identifie de manière unique le paquet original entre une source et une destination.

d) Option Destination

Des informations supplémentaires peuvent être rajoutées dans un paquet et n'ont de sens que pour le nœud de destination. Cette extension n'est pas encore clairement définie.

VI. Mécanismes de sécurité

La communauté Internet a développé des mécanismes de sécurité spécifiques aux applications, par exemple, pour le courrier électronique (Privacy Enhanced Mail), l'administration de réseaux (SNMPv2 security), l'accès au web (secure HTTP), etc. Malgré cela, les utilisateurs continuent à craindre pour leurs données. En intégrant des mécanismes de sécurité au niveau IP, on renforce davantage la sécurité de l'accès de ou vers les organisations.

La sécurité au niveau IP englobe l'authentification et la confidentialité. Les mécanismes d'authentification assurent qu'un paquet reçu a bien été émis par la partie identifiée comme source dans l'entête. Les mécanismes de confidentialité permettent aux nœuds qui communiquent de crypter leurs données, pour éviter l'espionnage par une tierce partie.

Le concept clé de la sécurité dans IPv6 est celui d'association de sécurité. Une association de sécurité est une relation unidirectionnelle entre un émetteur et un récepteur. Une association de sécurité est définie par une adresse de destination et un *Indice de paramètres de sécurité* (SPI, security parameters index). Généralement les paramètres qui définissent une association de sécurité sont :

- un algorithme d'authentification et un algorithme de modes utilisés avec l'authentification IP,
- une ou plusieurs clés utilisées par l'algorithme d'authentification et l'entête d'authentification,
- un algorithme de chiffrement, un algorithme de mode et des transformations utilisées avec l'entête ESP (Encapsulating Security Payload),
- une ou plusieurs clés utilisées par l'algorithme de chiffrement et l'ESP,
- la présence ou l'absence d'un champ d'initialisation ou de synchronisation d'un vecteur de chiffrement pour l'algorithme de chiffrement,
- la durée de vie de la clé ou temps au bout duquel un changement de clé devrait intervenir,
- la durée de vie de l'association de sécurité,
- la ou les adresses source de l'association de sécurité (l'adresse source doit être générique si plusieurs émetteurs partagent la même association de sécurité avec le récepteur),
- un niveau de sensibilité (secret, top secret, etc.).

VI.1. Authentification

L'entête d'authentification fournit un mécanisme pour l'authentification de paquets. L'émetteur calcule, avec une clé secrète, une signature numérique et l'émet avec le paquet. Le récepteur récupère la signature, la déchiffre, avec une clé secrète. Si le résultat du déchiffrement est bon, le paquet est authentifié. Le mécanisme d'authentification ne permet pas une tierce personne d'usurper l'identité d'un émetteur. Par contre une tierce personne peut intercepter les paquets et les lire.

Le format de l'entête d'authentification est le suivant :

Entête suivant (8 bits)	Longueur entête (8 bits)	Réservé (16 bits)
Indice de paramètres de sécurité (32 bits) SPI		
Données d'authentification (nombre variable de mots de 32 bits		

Format de l'entête d'authentification.

- *Entête suivant :* identifie l'entête qui suit immédiatement,
- *Longueur entête :* longueur du champ *Données d'authentification* en mots de 32 bits,
- *Indice paramètres de sécurité* : identifie l'association de sécurité,
- *Données d'authentification* : nombre variable de mots de 32 bits pour l'authentification.

Le contenu du champ *Données d'authentification* dépend de l'algorithme d'authentification spécifié. L'algorithme de génération de signature utilisé par défaut est MD5 (Message Digest 5). L'algorithme MD5 est exécuté sur un paquet IP plus une clé privée appartenant à la source, puis les données sont insérées dans le paquet. A la destination, le même calcul est exécuté sur le paquet et la clé privée et le résultat est comparé au résultat reçu dans le paquet. Le service d'authentification peut être utilisé de plusieurs manières.

VI.2. Confidentialité des données

Le mécanisme ESP (Encapsulating Security Payload) est utilisé pour assurer la confidentialité et l'intégrité des données. Selon les besoins, ce mécanisme peut être utilisé pour crypter un paquet complet (on parle de ESP mode tunnel) ou un segment de la couche transport (on parle de ESP mode transport).

Pour ESP mode transport, l'entête ESP est inséré dans le paquet IP immédiatement avant l'entête de la couche transport (TCP, UDP, ICMP), comme le montre le format suivant :

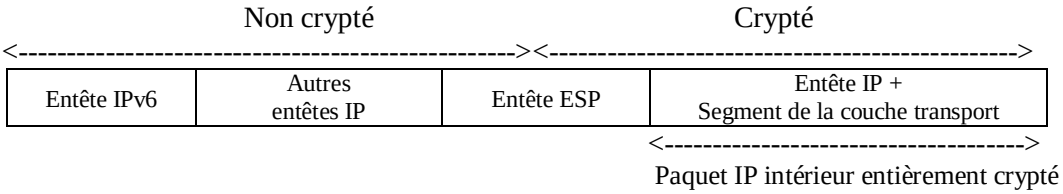
ESP en mode transport.

A la source du paquet, la deuxième partie de l'ESP et le segment de la couche transport sont cryptés. A la réception, un déchiffrement est effectué pour retrouver le segment d'origine. Le mécanisme de ESP mode transport assure la confidentialité pour toutes les applications.

Pour ESP mode tunnel, tout le paquet est crypté. Ainsi, l'ESP est placé devant le paquet, puis le paquet et une partie de l'ESP sont cryptés. Comme les routeurs intermédiaires ne peuvent pas déchiffrer les paquets IP qui les traversent, il est donc impossible de transmettre un paquet totalement crypté seul. Il est nécessaire d'encapsuler le paquet crypté avec une nouvelle entête IP contenant les informations nécessaires (et lisibles) pour le routage.

Le mécanisme ESP tunnel est adapté pour n'importe quelle sorte de porte de sécurité protégeant un réseau. On peut implanter, par exemple, ce mécanisme au niveau d'un firewall seulement pour alléger le travail de déchiffrement effectué par les hôtes à l'intérieur d'un réseau protégé par le firewall.

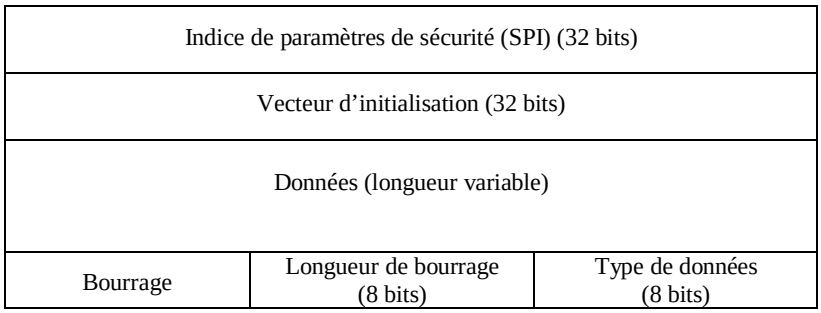
Le format de paquet avec ESP mode tunnel est le suivant :



Format de ESP en mode tunnel.

Toutes les implantations du mécanisme ESP doivent utiliser la méthode de chiffrement DES-CBC (Data Encryption Standard – Cipher Block Chaining).

La figure suivante montre le format de l'entête ESP et les données :

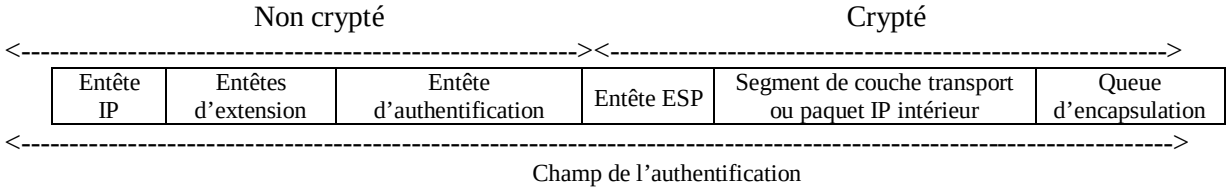


Format de l'entête ESP.

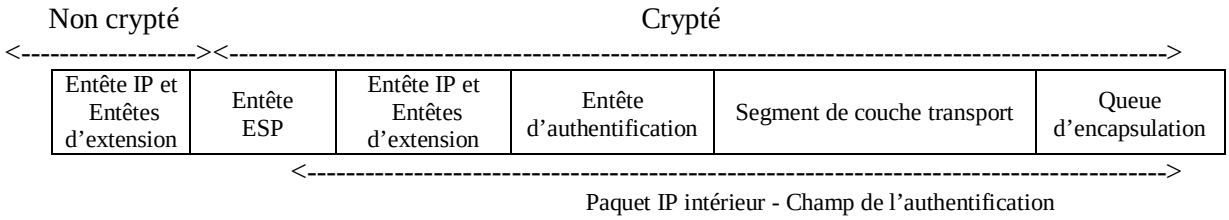
- *Vecteur d'initialisation* : entrées de l'algorithme CBC d'une longueur multiple de 32 bits,
- *Données* : contient les données à crypter,
- *Bourrage* : informations rajoutées pour aligner la taille des champs *Longueur de bourrage* et *Type de données* à 64 bits ,
- *Longueur de bourrage* : taille du bourrage non crypté,
- *Type de données :* type du protocole du champ de données (IP, TCP, …).

VI.3. Authentification et confidentialité

Les deux mécanismes de sécurité peuvent être combinés afin de transmettre des paquets avec un maximum de sécurité. On peut appliquer le chiffrement avant ou après l'authentification donnant ainsi les deux formats suivants :



Chiffrement avant authentification.



Authentification avant chiffrement.

Lorsque le chiffrement est effectué avant l'authentification, la totalité du paquet reçu est authentifié sur les parties cryptées et non cryptées. Cette possibilité s'applique aux deux modes de ESP (mode tunnel et mode transport).

Lorsque l'authentification est effectuée avant le chiffrement, l'entête d'authentification est placé dans le paquet intérieur. Le paquet intérieur est à la fois authentifié et protégé par le mécanisme de chiffrement. Cette possibilité s'applique au ESP mode tunnel seulement.

L'utilisation de l'authentification avant le chiffrement semble être préférable pour diverses raisons. Notamment, comme l'entête d'authentification est protégé, il est donc impossible d'intercepter le paquet et d'altérer le contenu de l'entête d'authentification, sans que cette opération soit détectée.

VI.4. Gestion de clés

Il existe deux principales approches de gestion des clés de chiffrement sur un réseau. La première consiste à gérer manuellement les clés : chacun crée ses clés et les détruit selon ses besoins. Cette approche qui est simple n'est pas envisageable à long terme du fait de l'expansion que connaissent les réseaux. D'où une deuxième approche qui repose sur une gestion automatique des clés. Plusieurs protocoles ont été proposés à ce sujet. La proposition dominante au sein des l'IETF est celle du protocole ISAKMP (Internet Security Association Key Management Protocol). Ce protocole crée, modifie et détruit automatiquement les clés.

VII. Routage

Les principes des protocoles de routage n'ont pas changé avec IPv6. Les travaux ont consisté en l'adaptation des protocoles existants au format des adresses. Ces protocoles profitent des propriétés maintenant incluses dans IPv6 comme l'authentification ou le *multicast*. Comme dans IPv4, on distingue le routage interne et le routage externe.

VII.1. Routage interne

Les protocoles dits de routage interne permettent une configuration automatique des tables de routage. Les routeurs découvrent automatiquement la topologie du réseau et déterminent le plus court chemin pour atteindre un réseau distant. En plus protocoles propres aux constructeurs de routeurs, il existe deux protocoles conçus par l'IETF : RIPng et OSPng.

RIPng est très proche de RIP utilisé dans IPv4. C'est un protocole de la famille "distant vector". Dans ce protocole les routeurs s'échangent périodiquement leurs tables de routage. A la réception d'une table de routage, un routeur met à jour sa table sur la base des nouvelles données reçues. Si un routeur tombe en panne ou si une ligne tombe en panne, les autres routeurs ne recevant plus d'informations de ce routeur suppriment l'entrée correspondante à ce routeur de leur table de routage.

Le deuxième protocoles, OSPF (Open Shortest Path First), fait partie des protocoles dits "plus court chemin". Il est plus efficace que le premier, mais il est difficile à mettre en œuvre. Ce protocole est fondé sur les principes suivants :

- inondation fiable du réseau qui permet à chacun des routeurs de posséder une copie des configurations de tous les autres routeurs et peuvent alors calculer le plus court chemin entre deux points du réseau,
- pour éviter le recalcul fréquent de toutes les tables de routage, OSPF offre la possibilité de découper le réseau en aires. Une aire principale (appelée backbone) doit pouvoir relier toutes les autres aires. Les modifications de tables de routage se limitent, le plus possible, à des aires particulières.

VII.2. Routage externe

Le terme externe vient du fait qu'il s'agit d'un échange de tables de routage entre deux domaines d'administration distincts, généralement entre un client et un fournisseur, un fournisseur et son transporteur international ou entre fournisseurs et transporteurs internationaux.

En IPv4, la notion de domaine d'administration est représentée par un numéro de système autonome (AS : Autonomous System). Il n'est pas clair que cette notion soit utile en IPv6 puisque dans un plan d'adressage hiérarchique, le préfixe peut jouer une notion équivalente au numéro AS.

Avec un protocole de routage externe, il ne s'agit pas de trouver la topologie du réseau, mais d'échanger des informations d'accessibilité explicite entre routeurs pour le faire. Toute annonce du réseau par un domaine implique qu'il accepte de router les paquets vers cette destination.

Le protocole retenu pour IPv6 est BGP-4+ identique à BGP-4 utilisé dans IPv4.

VIII. Incidences de IPv6 sur les protocoles de transport et les applications

L'un des pré-requis à la définition des fonctionnalités de IPv6 était de ne pas toucher aux protocoles de niveau transport. Si on change à la fois les protocoles de niveau réseau et les protocoles de niveaux transport, les utilisateurs seront très hostiles à la migration vers IPv6. Par exemple TCP est utilisé par beaucoup d'applications et l'absence de modifications de ce protocole facilitera le passage de IPv4 à IPv6. Malgré cela, quelques modifications doivent être apportées à UDP et TCP, notamment pour gérer les erreurs sur les entêtes de paquets. Le protocole ICMP devra être modifié un peu aussi.

Dans les applications, on définit des structures de données contenant des adresses IP, pour utiliser des sockets, par exemple. Comme la taille des adresses a changé, il faut changer le code des applications lié aux adresses. Dans le monde Unix, des modifications doivent être apportées notamment au fichier `sys/socket.h` pour redéfinir les structures de données (`AF_INET6, in6_addr`, ...) afin adapter l'interface de sockets à IPv6.

Quelques ouvrages sur les réseaux

1. Tanenbaum A., *Réseaux : architectures, protocoles et applications*. InterEditions.

2. Pujolle G., Les réseaux. Eyrolles.

3. Rifflet J.M., *La communication sous Unix*. Mc Graw Hill.

4. Comer D., TCP/IP : *architectures, protocoles et applications*. InterEditions.

5. Mammeri Z. et Thomesse J.P., *Réseaux locaux : couche physique et couche liaison de données – Normes ISO 8802 - Normes IEEE 802*. Editions TEKNEA,