

## Notes sur le cours Réseaux ATM

### DIAPO 1 :

Nous commençons les deux dernières séances consacrées aux réseaux de transmission de données avec la solution ATM pour Technique Temporelle Asynchrone en Français.

Comme on le verra, on est un peu à mi-chemin entre les mondes paquets et circuits car pour la première fois, on a défini des solutions qui visent tous les types de trafic et pas un réseau spécialisé pour la voix ou les données.

D'un point de vue de la classification initiale, on est bien dans le mode paquet, mais certains pointeurs seront envoyés en avant vers les solutions circuits !

### DIAPO 2 :

Les opérateurs de télécommunications payent cher le déploiement et la maintenance de leur double solution de réseau « paquet » et de réseaux « circuit ». Des réflexions ont alors été menées pour essayer d'unifier tout cela. On est parti alors des applications et c'est bien comme cela que tous les réseaux télécoms modernes ont vu le jour comme on le verra dans les cours de réseaux à venir et en particulier de réseaux mobiles.

L'idée est alors apparue que les futurs réseaux de télécommunications devraient écouler des trafics de voix, de données mais aussi de vidéos. Cela se traduit comme on l'a expliqué dans le cours d'introduction par des flux avec des caractéristiques de débit très hétérogènes et des besoins en qualité de service antagonistes.

L'autre fait majeur a été la généralisation dans les réseaux de transport de la fibre optique comme support unique de communication c'est à dire avec des débits très importants et peu d'erreurs de transmission.

C'est aussi à ce moment-là que les équipementiers se sont penchés sur la capacité à construire des commutateurs de paquets rapides ; c'est même par cela que l'on a commencé !

Quelles étaient les forces en présence :

- Les solutions de commutation de circuits conviennent très bien pour les sources de trafic à débit constant. C'est à dire essentiellement la parole téléphonique (et encore dès que l'on parle de compression et/ou de suppression des silences ce n'est plus le cas). Pour la vidéo, cela dépend un peu des techniques de codage/décodage. Elles garantissent un débit constant et un délai d'acheminement constant. Cela ne convient pas du tout aux flux à débit variable ! Ce qui a été envisagé, ce sont des solutions circuits mais avec des débits garantis qui dépendent de la source de trafic (on en reparlera dans le cours SDH).
- Les solutions classiques du monde X.25 et FR a contrario sont dédiées à la transmission de données informatiques mais ne conviennent que très mal pour envoyer de la voix téléphonique de façon massive (pas du tout en X.25 ; un petit peu en FR mais avec des trafics de voix très marginaux) ; quant à la vidéo...
- Le monde de l'Internet n'a pas une vision de la maîtrise de la qualité de service ; on fait au mieux (en particulier à l'époque). Le monde de l'Ethernet était vu comme une technologie de réseaux locaux et globalement, on a les mêmes reproches à faire à ce monde là.

C'est donc à cela que s'est attaqué l'ITU-T en vue de la création d'une nouvelle solution universelle de transmission qui permettent de remplacer les doubles infrastructures de réseau de transport par une solution unique ; mais aussi à une solution de réseau d'accès (en particulier dans les cours ADSL et FTTH).

### DIAPO 3 : Pile de protocoles

La pile de protocoles qui a été standardisée est une extension de ce que l'on a vu dans le cours du relayage de trames. La couche physique est commune pour la transmission de l'ensemble des données ainsi que la couche ATM qui fonctionnellement s'occupera de ce que faisait la couche FR/Q.922 dans le réseau Frame Relay.

On voit que l'on sépare le plan de données qui sert à transmettre tous les types d'applications du plan de contrôle qui va gérer les connexions, le routage... pour lequel, on a l'habitude d'avoir des besoins en fiabilité qui ne sont pas les mêmes que ceux du plan de données. C'était déjà le cas du relayage de trames ; là c'est encore plus marqué car c'est par le plan de données que l'on va faire passer la parole téléphonique !

On voit aussi de façon plus marquée apparaître un troisième plan appelé plan de gestion, qui va se préoccuper de la configuration des équipements, du monitoring, de la gestion/prévention des pannes... On est sur des échelles de temps plus grandes à chaque fois : la  $\mu s$  pour le plan de données, la s pour le plan de contrôle, la minute ou l'heure pour le plan de gestion. Le plan de données est surtout pris en charge au niveau matériel, le plan de contrôle par du software et le plan de gestion par des règles (voire des interventions humaines).

(ce découpage est inhérent à la conception des réseaux télécoms depuis 30 ans ; il a repris de son intérêt dans le monde des réseaux informatiques comme on le verra de façon très soutenue dans les cours de 3A).

L'ensemble des plans de données et de contrôle passe par la couche ATM (comme dans le relayage de trames). Au-dessus vont se matérialiser les différences entre les applications. Pour ce faire, des couches intitulées couche d'adaptation à l'ATM ont été créées. Ce sont elles qui vont permettre justement de différencier les mécanismes dont auront besoin les applications !

### DIAPO 4 : Caractéristiques générales d'ATM

Les caractéristiques essentielles de la solution ATM vont être :

- Le choix du mode connecté. Toujours pas de surprise dans les réseaux d'opérateurs. On poursuit la route tracée par X.25 et FR ;
- Très peu de contrôle d'erreur, de flux, de congestion. Là c'est la suite logique du relayage de trames. Les arguments sont les mêmes que ceux que l'on a avancé pour le FR (qualité des supports) mais là c'est encore plus marqué car les qualités de service attendues sur les différents types de flux ne peuvent impliquer sur la couche commune ATM que l'intersection des mécanismes dont on aurait besoin pour chacun d'entre eux... il ne reste pas grand chose !
- C'est le mode paquet qui a été retenu ! Là le choix est vraiment fondateur car ce sera la même réponse qui sera rendue à chaque fois que l'on se posera la question par la suite.

Quand on veut faire passer des flux à débit variable, il vaut mieux utiliser le mode paquet. En effet, le mode circuit propose une allocation d'un débit fixe (au moins pour une période donnée) et pour des flux à débit variable, cette adaptation ne peut se faire au même rythme que les fluctuations du trafic. Dans le même ordre d'idées, si l'on prend par exemple un transfert de fichiers, il n'a pas intrinsèquement un débit, il prend ce qu'on lui donne !

Tout cela concourt au choix du mode paquet... avec les difficultés associées qui sont la maîtrise de la qualité de service en particulier différenciée => c'est encore un point fort d'ATM car la réflexion a été menée de façon complète... mais pas définitive comme on le verra par la suite !

Les opérateurs ont aussi pris peur avec le mode circuit sur le réseau d'accès en se posant la question des infrastructures nécessaires pour cela – déploiement de fibre

optique ; comment améliorer la solution cuivre... (cf. cours sur les réseaux d'accès à la fin de la matière).

C'est ce mode paquet qui se cache derrière le A de Asynchrone.

- Un autre point important a été le choix de paquets de taille unique et petits. Il y a plusieurs raisons à cela :
  - Il est plus simple de mettre en œuvre des commutateurs qui travaillent sur des messages de taille unique : gestion de la mémoire, de la matrice de commutation, des files d'attente, fonctionnement slotté avec prise de décision périodique...
  - D'un point de vue des performances, prendre une variance de taille de paquets nulle est la solution qui est optimale (on en reparlera dans le cours de performances de réseau ; toute variabilité dans les longueurs de paquets conduit à des temps d'attente qui vont beaucoup plus fluctuer)
  - Petite taille vs Grande taille : là le débat a été tranché en faveur des paquets de petites tailles en raison du traitement de la voix. Si l'on avait pris des paquets de grande taille, soit on aurait fait beaucoup de padding pour terminer de les remplir soit on aurait mis beaucoup de temps pour les remplir, ce qui conduit à des délais indésirables pour cette application.  
L'inconvénient est l'overhead : chaque paquet de petite taille a son en-tête ce qui est pénalisant pour les transferts volumineux !
- La complexité protocolaire va donc porter sur les couches d'adaptation à l'ATM. Comme il y aura un service minimal rendu par ATM, les fonctionnalités dont les applications auront besoin sont remontées au niveau des extrémités (retransmission des morceaux manquants, absorption des variations de délai...)
- En termes fonctionnels, la couche physique s'occupe des transmissions sur le support de communication, la couche ATM joue le rôle de la surveillance des transmissions et de l'acheminement des données, la couche AAL est une couche de transport !

## DIAPO 5

On aboutit alors pour le plan de données aux piles de protocoles suivantes : l'application (attention comme on le verra par la suite, cela pourra être de vraies applications au sens des réseaux informatiques mais aussi des paquets IP, des trames Ethernet, des trames Frame Relay) est située sur les extrémités au-dessus d'une couche AAL qui se déroule également entre les extrémités. Le réseau de transport est constitué de commutateurs ATM qui fonctionnent en mode paquet et circuit virtuel et qui commutent des paquets ATM.

Je n'ai pas dressé ici la pile de signalisation que l'on verra plus tard dans le cours.

## DIAPO 6

Le format général des paquets ATM est assez simple, encore plus que ce que l'on a vu dans les réseaux précédents.

La dénomination retenue est celle de cellule.

Le combat a fait rage sur le choix de la longueur de la cellule : les américains prônaient des paquets de 64 octets (pour la transmission de données), les européens des paquets de 32 octets (pour la voix). On a tranché à un format de 48 octets de données utiles et de 5 octets d'en-tête...

Sur l'en-tête, on voit apparaître essentiellement le numéro de la connexion, le type de cellule sur 4 bits et un octet de code d'erreur qui ne porte que sur l'en-tête. On est donc encore plus radical qu'en Frame Relay puis la protection contre les erreurs porte uniquement sur l'en-tête. C'est un choix motivé par le fait que si les données transportées contiennent de la voix, la protection contre les erreurs n'a pas d'intérêt !

## DIAPO 7 Principes des AAL

Les couches d'adaptation à l'AAL ont été construites en fonction des applications. Il a fallu classer les applications par groupes en fonction de critères discriminants de façon à ne pas avoir une couche de transport par application. Les critères qui ont été retenus sont : contraintes temporelles O/N ; débit constant O/N ; mode connecté de l'« application » O/N.

Cela donne 4 classes : la première A pour les flux contraints temporellement et à débit constant (voix, visiophonie, certaines applications vidéos), la deuxième B pour les flux contraints temporellement et à débit variable (voix avec suppression de silence, vidéo), la troisième C pour le transfert de données correspondant à un mode connecté (transfert de fichiers, X.25, FR ; en 2020, on pourrait dire les flux IP qui passent par TCP) et la quatrième D pour les flux de données correspondant à un mode non-connecté (transfert de messages ; paquet IP, trames Ethernet ; idem ici ce serait les flux IP qui passent par UDP).

Ce premier effort a été mené très tôt. Nous retiendrons cette première classification car c'est désormais largement la façon avec laquelle pour la création de toute nouvelle solution de réseaux de télécommunications : caractéristiques des applications et des services que l'on veut rendre.

Les fonctionnalités qui seront offertes par la couche AAL dépendront de la classe : vérification de l'intégrité des données, retransmissions, rattrapage de la gigue, segmentation réassemblage...

## DIAPO 8,9 et 10 Couche ATM

Elle est très simple en termes protocolaires, justement pour ne pénaliser aucune application.

De nombreux points communs avec le relayage de trames (et dans une moindre mesure X.25) : mode circuit virtuel avec numéros relatifs, pas de numérotation des cellules.

La particularité est que le numéro de connexion est divisé en deux parties : un identifiant de conduit virtuel et un identifiant de circuit virtuel. L'idée va être de regrouper à l'intérieur d'un plus gros tuyau (le conduit virtuel) plusieurs petits tuyaux (les circuits virtuels). Cela permet de mettre plus facilement en œuvre des mécanismes de qualité de service, en mettant dans un même conduit virtuel des circuits virtuels de même type de service.

Cela est illustré sur les diapos 9 et 10.

Une connexion de circuit virtuel se déroule entre deux extrémités du réseau, une connexion de conduit virtuel entre deux nœuds du réseau ou entre une extrémité et un premier nœud. Une connexion de circuit virtuel passera alors par plusieurs conduits virtuels consécutifs.

Deux types de commutateurs ont été prévus :

- Des commutateurs de circuits virtuels qui regardent le couple VPI/VCI appelés Commutateurs ATM
- Des commutateurs de conduits virtuels qui ne traitent que le champ VCI appelés brasseurs.

L'idée était d'avoir le plus souvent des conduits virtuels permanents et des circuits virtuels commutés. Les brasseurs sont donc des équipements plus simples avec en quelque sorte des tables de commutation quasi-permanente. C'était aussi lié au fait que la signalisation ATM est compliquée et que cela a permis des premières solutions avec uniquement des connexions permanentes et des brasseurs.

La différence entre commutateurs (switch) et (brasseur) vient du fait que les *switches* traitent la signalisation ATM et donc permettent d'établir des connexions commutées alors que les brasseurs (*cross-connect* ou *X-connect*) ne font que l'opération de commutation.

## DIAPO 11 – les couches physiques pour l'ATM

L'ATM a été conçu pour fonctionner a priori sur tout type de support physique : fibre bien sûr mais aussi paire de cuivre, câble et même hertzien (satellite en particulier). La plupart de ces supports et en particulier la fibre dans le cœur de réseau est souvent exploitée en mode synchrone (le plus emblématique est la SDH que nous verrons plus tard). ATM peut aussi fonctionner sur des supports fonctionnant en mode asynchrone (fibre sur le réseau d'accès par exemple). On a donc ajouté, pour faire fonctionner l'ensemble, des fonctionnalités sur la partie supérieure de la couche physique qui permettront d'émettre les cellules sur le support de communication. C'est la sous-couche intitulée sous-couche de convergence.

En dessous, on a le support de transmission à proprement parler qui n'est pas du tout lié à ATM.

Sur la partie haute par conséquent de la couche physique on a la formation des messages de la couche physique – ça ce n'est pas ATM, en revanche, il faut décrire la façon avec laquelle on va encapsuler les cellules ATM dans ces messages de niveau physique que dans la terminologie des réseaux télécoms, on appelle « trames ». Dans la suite du cours et même des cours que je vous ferai, je les nommerai « trames de la couche physique » pour éviter la confusion avec les trames Ethernet, HDLC ou Frame Relay.

(attention, c'est un des soucis de vocabulaires qui provient du fait que les réseaux télécoms – circuit – ont développé leur vocabulaire séparément de celui des réseaux informatiques et que le terme de trame n'a pas le même sens dans les deux mondes). Attention, ce mot trame a aussi un sens dans la terminologie de la vidéo et de la télévision, c'est le nom des messages qui sortent des « codecs » (codeur/décodeur). On parle par exemple de trame MPEG par exemple ☺

Pour que le récepteur récupère les cellules, il y a nécessité de les délimiter. C'est ici également que l'on positionne le traitement du champ de contrôle d'erreur HEC (Header Error Control).

#### **DIAPO 12 – Adaptation des débits**

Le principe de fonctionnement va être le suivant. Le support de communication a un certain débit. Dans le monde paquet, quand les « cellules » (ou trames) sont prêtes, on les donne de façon asynchrone à la couche physique. Si de façon instantanée, le débit des cellules ATM est plus grand que le débit du support, on est obligé de stocker les cellules en file d'attente. Pour mettre en œuvre le plan de contrôle, on a aussi des messages à envoyer (tests de connectivité, d'équipements...) – on les envoie sous la forme de cellules qui vont s'intercaler avec les cellules de « donnée » ATM. Le choix qui a été fait est, quand on n'a rien à émettre, d'émettre des cellules vides (le codage ferait aussi un HEC égal à zéro ; l'ensemble passe un mécanisme de scrambling qui permet d'éviter de longues suites de « 0 » qui pose des problèmes de synchronisation).

On a donc expliqué comment on remplissait les espaces entre cellules. Attention, cette technique engendre une légère gigue potentielle, on ne respecte pas exactement l'écart temporel entre les cellules initiales selon le nombre de cellules vides que l'on va être amené à ajouter. On l'appelle « gigue d'insertion » (c'est relativement modeste tout de même).

#### **DIAPO 13 – Protection contre les erreurs**

Le code détecteur d'erreur ne porte que sur l'en-tête. Le mécanisme est un moyen terme entre mécanisme de détection et de correction d'erreur. Le fonctionnement est retracé par l'automate présenté ici : un état où l'on ne fait que détecter et l'autre où l'on détecte et l'on corrige. Attention, les erreurs sont faibles sur la fibre mais elles arrivent en rafale, ce qui explique l'automate.

Dans l'état correction, si l'on reçoit une cellule correcte, on y reste. Si l'on reçoit une cellule en erreur : s'il y a une seule erreur, on la corrige. S'il y en a plusieurs on supprime. Dans tous les cas on passe dans le mode détection.

Dans le mode détection, si on reçoit une cellule en erreur, on la supprime. Sinon on passe dans le mode correction.

#### **DIAPO 14 – Délimitation**

Pour faire de la délimitation, on aurait pu utiliser la technique d'HDLC mais elle est coûteuse (au moins un octet par cellule + transparence). On aurait pu se servir de techniques proposées par la couche physique sous-jacente que l'on verra plus tard mais cette solution aurait été dépendante de la couche PHY.

On a tiré profit du fait que les cellules sont de taille constante et c'est donc le code d'erreur qui va permettre de délimiter les cellules.

On a donc de nouveau un automate au niveau du récepteur avec deux états « synchro cellule prise » et « non prise ». Dans l'état synchro non prise, on part à la recherche de code d'erreur. On regarde donc sur une fenêtre coulissante de 5 octets si le 5<sup>ème</sup> octet est le reste de la division par le polynôme générateur des 4 précédents. Si tel n'est pas le cas on continue.

Si c'est oui, on va regarder si 53 octets plus loin le miracle ne se produit pas de nouveau.

La synchro est prise dès que le motif code d'erreur est repéré sur plusieurs (6 de tête ?) cellules consécutives.

Attention en cas d'erreur de transmission, on ne repère pas instantanément la synchro, il faut que cette perte atteigne plusieurs (5 ?) cellules consécutives.

Cette technique est originale et plus robuste que la technique de fanion.

On en reparlera dans les cours de 3A.

#### **DIAPO 15 – Insertion sur le support**

Les cellules ATM vides ou pas sont alors insérées sur le support de communication (l'exemple d'illustration est celui des trames SDH – voir cours numéro 8).

#### **DIAPO 16 et 17 – Couche ATM – déjà évoqués.**

#### **DIAPO 18 et 19 – Format des cellules ATM**

Comme on l'a déjà évoqué, le format des cellules ATM est extrêmement simple. En gros un numéro de connexion un code d'erreur et 48 octets de données.

Le numéro de connexion n'a pas la même taille à l'accès au réseau et entre commutateur. A l'accès un champ de contrôle de flux avait été prévu qui n'a jamais été utilisé.

La seule partie qui va permettre de différencier les cellules sera les 3 bits du Payload Type Identifier et le bit CLP (Cell Loss Priority). La philosophie du bit CLP est la même que le DE du relaiage de trames.

Pour le PTI, on a 3 bits. Le premier indique si l'on a une cellule de gestion ou une cellule de « données » (dans les faits, une cellule du plan de données ou du plan de contrôle).

Les cellules de gestion permettent de poursuivre la fiabilisation du réseau ATM en effectuant des tests de chacune des connexions (on avait déjà évoqué les cellules de la couche physique pour surveiller le support sous-jacent). On va donc effectuer des mesures de performances à l'intérieur de chacune des connexions : de proche en proche et de bout en bout de la connexion de circuit virtuel. Cela permet de mesurer le délai, la perte la gigue du circuit virtuel. A également été standardisé un type de trames de gestion intitulé Fast Resource Management qui permet d'augmenter, diminuer les débits des connexions (cf. partie qualité de service).

Pour les cellules de « données » (ie. plan de données et plan de contrôle), le deuxième bit signifie que la cellule est passée par un nœud congestionné => EFCI (équivalent de FECN).

On note qu'un bit n'est pas utilisé. Il est laissé à la couche AAL qui pourra l'utiliser en prévenant la couche ATM quand elle lui envoie les cellules.

On constate par deux fois ce phénomène qui n'est pas extrêmement propre d'un point de vue réseau : ce bit qui est piloté par l'AAL et le HEC qui est dans l'en-tête de la cellule et qui est traité par la couche Physique ; cela va à l'encontre de la sacro-sainte indépendance entre les couches !

#### **DIAPO 21 :**

Je passe vite sur cette diapo pour dire que l'on a aussi du contrôle à l'intérieur de chaque connexion de conduit virtuel ; qu'il y a des protocoles en plus pour faire la configuration... ce ne sont pas les points marquants d'ATM

#### **DIAPO 22 :**

Si l'on regarde les équipements d'un réseau ATM on repère des interfaces à l'accès au réseau (User to Network Interface), des interfaces entre les équipements du réseau NNI (Network to Network Interface). Les fonctions qui avaient été introduites pour le relayage de trames sont reprises et étendues.

On retrouve les fonctions d'admission de connexion (CAC), de contrôle des paramètres de l'utilisateur UPC, de gestion des ressources (RM), de contrôle des priorités (PC). Une fonction supplémentaire apparaît qui est celle du Traffic Shaping (TS) qui consiste à remettre en forme le trafic. Cela même l'utilisateur peut le faire. Pour un même profil de trafic (débit moyen, maximum...), on va tenter de le lisser pour éviter ce que l'on appelle des grumeaux (bursts) qui pénalisent les performances d'ensemble. C'est bien ATM qui a inventé cela et que vous illustrerez dans les cours de second semestre. Ce n'est que complémentaire par rapport au contrôle des paramètres utilisateurs. Par exemple, dans un commutateur, ce sera en entrée que l'on vérifiera les paramètres et la conformité de trafic, mais ce sera en sortie qu'on le remettra en forme car c'est le multiplexage avec d'autres trafics qui forme les grumeaux (on est bien toujours dans un réseau paquet).

Notons encore que les fonctions d'admission de connexions deviennent encore plus compliquées qu'en relayage de trames car ce ne sera pas forcément le même algorithme que l'on utilisera pour accepter un trafic de voix (qui a des contraintes de gigue) et un transfert de fichiers. On en reparlera plus loin même si une fois encore ces algorithmes ne sont pas normalisés !

#### **DIAPO 23, 24 et 25 : Les couches AAL**

Les couches AAL sont elles-mêmes subdivisées en deux parties : une partie basse intitulée SAR pour Segmentation And Reassembly et une sous-couche haute intitulée Convergence Sublayer. Les fonctions spécifiques sont positionnées dans la sous-couche de convergence.

#### **DIAPO 25 : Des classes de services aux AAL**

Les classes de service avaient été identifiées dès le début de la standardisation ATM ; à chacune devait correspondre une AAL.

C'est là que la standardisation a patiné :

- Entre l'AAL3 et l'AAL4, la différence est ténue => on les a fusionnées en une AAL3/4 extrêmement sophistiquée et compliquée à mettre en œuvre avec des cartes chères
- L'AAL-1 est aussi assez compliquée. Les cartes fournies étaient très chères

- L'AAL-2 n'a jamais été terminée suite à des dissensions dans la standardisation ;
- Devant la lenteur de la standardisation, une AAL supplémentaire a été créée, l'AAL-5 avec comme objectif de permettre de transmettre des paquets IP au-dessus d'ATM et aussi de prendre en charge la signalisation.

La conclusion de cet épisode est que l'on n'a pas une AAL-5 mais des AAL-5 et c'est même la seule qui soit réellement opérationnelle (à ce moment-là de la standardisation). Les mécanismes nécessaires pour les applications sont donc présents dans l'AAL-5 qui sert un peu à tout.

Pour des raisons de concisions c'est donc la seule que je finis par présenter !

Cette AAL-5 n'a pas que des défauts. Elle a le mérite de la simplicité.

#### **DIAPO 27 : AAL-5**

Par exemple la partie basse de segmentation et de réassemblage ne fait que découper en tronçons de 48 octets. Aucun contrôle supplémentaire n'est présent. Il faut simplement qu'on lui fournisse des morceaux qui aient une longueur multiple de 48 octets.

Quand elle donne ces tronçons de 48 octets à la couche ATM, elle lui précise si le morceau en question est le dernier du message initial ou pas. Cette information sera contenue dans le troisième bit du PTI vu précédemment.

La sous-couche de convergence est un peu plus complexe. Elle est elle-même redécoupée en deux sous-couches intitulées : CPCS comme Common Part Convergence Sublayer qui est utilisée par tout le monde et la sous-couche SSCS comme Service Specific Convergence Sublayer qui elle dépend de l'application.

On voit donc que la complexité initiale des AAL se retrouve là... ou pas.

La sous-couche CPCS fait du contrôle d'erreur sur 4 octets, de l'alignement sur un multiple de 48 octets et donc du padding. Elle dira combien d'octets de padding ont été ajoutés.

Le choix du mécanisme de contrôle d'erreur n'est pas anodin. On aurait pu le mettre au niveau de la couche SAR, mais cela aurait coûté plus cher en overhead, 1 octet à minima pour 48 octets, ce qui est beaucoup pour des paquets IP par exemple. Qui plus est, cela n'aurait pas non plus permis de détecter les pertes de cellules qui ne sont pas numérotées.

Cette fois on peut faire les deux à la fois !

#### **DIAPO 28 et 29 : Exemple de SSCS, l'AAL de signalisation ou S-AAL**

Le principe que l'on vient de décrire sur l'AAL-5 est que l'AAL-5 permet simplement de détecter et de supprimer les messages en erreur ou incomplets. Si l'on veut de la fiabilité supplémentaire, on l'ajoutera dans la partie SSCS ; c'est le cas pour l'AAL de signalisation. On a pour cela inventé le protocole SSCOP qui permet à l'instar de HDLC de faire de la fiabilisation en numérotant les messages, en effectuant du contrôle de flux et en faisant du polling pour savoir ce qui a été reçu correctement.

Attention, au-dessus on trouve le protocole d'envoi de messages de signalisation. Il fonctionne un peu sur le même principe que Q.933 mais on l'a vidé de la partie de fiabilisation qui est descendue au niveau de l'AAL-5.

Les noms qui sont présents proviennent du réseau téléphonique et l'on reverra les protocoles qui en sont les inspirateurs : Q.2931 à l'accès au réseau provient du Q.931 que nous reverrons dans le cours 9 (RNIS) – c'est le protocole de signalisation téléphonique par excellence. On l'a simplement enrichi pour qu'il puisse contenir les descripteurs de trafics et les besoins en qualité de service. Dans le cœur de réseaux (entre commutateurs), on utilise B-ISUP qui est le descendant du protocole du réseau téléphonique ISUP (ISDN User Part) que l'on verra dans le cours 8.

Ce sont des protocoles simples : demande de connexion, sonnerie, décroché, raccroché...

C'est ce qui est illustré sur la diapo 29.

### **DIAPO 30 : Adressage ATM**

De la même façon que l'on avait parlé de la création du FR-Forum, l'ATM-Forum s'est créé pour accélérer les travaux de l'ITU-T. Il a aussi travaillé avec l'IEEE et l'IETF pour transporter des paquets IP ou des trames Ethernet.

De nombreux formats d'adresses ont été proposés. J'en ai présenté ici. Ce sont des adresses assez longues semi-structurées.

### **DIAPO 31 et 32 : Le routage en ATM – PNNI**

La transition est bonne puisque l'on vient de parler d'adressage. On va parler maintenant de routage.

Là encore, il y a quelques pépites en ATM qui seront largement reprises dans les cours de l'année prochaine.

Dans le routage, il y a plusieurs aspects comme vous l'avez déjà vu dans les cours TCP/IP et Réseaux Locaux.

Il y a en effet, le protocole de routage tout d'abord qui permet de donner aux entités qui calculent les routes d'avoir des informations sur l'état du réseau ; les algorithmes de routage à proprement parler qui permettent de calculer les routes ; l'acheminement et dans le cas du mode circuit virtuel (où l'on ne peut plus parler de routage) l'aiguillage où l'on ne fait que commuter. Dans le cas du mode datagramme les deux dernières étapes sont confondues (cf. Ethernet et IP).

Dans ce qui va suivre, certaines notions sont assez proches de celles que vous avez vu dans les cours de TCP/IP. Par exemple, le choix du routage s'est porté sur un protocole à état de liaison plutôt que sur un protocole à vecteur de distance qui ne permet pas d'avoir une idée précise de la route empruntée et donc de sa qualité.

Le deuxième point est que l'ambition d'ATM était d'avoir une solution universelle et donc la seule solution consistait à agréger les informations pour ne pas avoir à tout connaître du réseau à tout moment.

Là où en revanche ATM se dégage largement c'est sur la notion de routage à QoS, c'est à dire, en ATM, l'objectif est que la route qui sera choisie satisfasse la qualité de service attendue et que la procédure de signalisation rende effective la réservation de ressources.

Attention, la notion de routage à Qualité de Service est souvent ambiguë. Assez régulièrement, on se contente de choisir une route qui semble convenir à un certain trafic... mais si on ne réserve pas les ressources, il n'y a pas de raison que la garantie soit au rendez-vous ! Une route qui semblera bonne pourra le sembler pour beaucoup de monde et si on ne réserve pas les ressources, il n'y a aucune raison que les performances soient au rendez-vous !

Dans la partie acheminement, on va avoir plusieurs morceaux. Les tables de routage sont remplies, les demandes de connexion arrivent. On a alors le choix de faire du routage « bond par bond » ou du routage par la source. Attention, ce que l'on entend par source ici, ce sera le premier commutateur. Attention encore, cela ne correspond pas non plus à du routage centralisé, chaque commutateur d'extrémité a effectué ses calculs. L'intérêt de la technique du routage par la source, c'est d'éviter des boucles de routage. Cela a l'inconvénient d'alourdir les messages de signalisation (B-ISUP) qui contiennent la route complète.

Attention comme la décision est prise par la source, il va falloir que l'on s'assure que les ressources sont bien là au fur et à mesure que la demande de connexion progresse, ce sera l'objet de la fonction GCAC (comme Generic) qui se déroule sur le premier commutateur et qui est assez conservatrice (on prend de la marge sur les ressources du chemin) et la fonction

CAC qui ne se déroule que sur un commutateur donné et qui donne un accord simplement local.

Comme quand la demande de connexion arrive sur un nœud, il se peut que les ressources ne soient plus disponibles (ou il y a eu une panne), on ajoute une possibilité de faire du back-tracking intitulé crankback en ATM.

### **DIAPO 33 : Protocole PNNI, un protocole à état de liaison**

Chaque commutateur construit sa base de données topologique au vu des informations qu'il récupère dans des messages de type HELLO. On utilise le principe de l'inondation pour que tous les nœuds aient bien les connaissances.

Si cela fonctionne à petite échelle, on n'a aucune chance que cela puisse passer à l'échelle. Pour se faire, on va structurer hiérarchiquement le réseau. 13 octets sont disponibles pour cette hiérarchisation qui donne donc 104 niveaux hiérarchiques possibles.

J'ai ici représenté un exemple avec 3 niveaux hiérarchiques.

### **DIAPO 34, 35 et 36 : Vision hiérarchisée du réseau**

Voici une version hiérarchique du réseau précédent. Les nœuds et les liens physiques sont au niveau le plus bas. Dans les niveaux supérieurs, ce n'est plus qu'une vue logique et agrégée de ce qui est en-dessous. On voit apparaître un nœud en noir qui est le leader. C'est lui qui transmettra les informations agrégées au niveau supérieur et qui distribuera les informations qu'il a collectées au niveau supérieur à l'ensemble des nœuds du niveau inférieur.

Les nœuds logiques d'un même niveau possèdent les mêmes informations. Les nœuds logiques sont implantés dans des nœuds physiques.

On illustre la vision qu'un nœud aura du réseau sur la diapo 36

### **DIAPO 37 : Mais quelles informations distribuer ?**

C'est là que c'est compliqué. Chaque élément d'un chemin aura des caractéristiques intrinsèques, que l'on appelle des attributs dans la terminologie ATM – débit maximal du lien, débit disponible.

On va alors pouvoir calculer des informations sur les chemins tels que les taux de perte, la gigue, le délai quand on emprunte le chemin. Les calculs ne sont pas forcément simples ; on fait des hypothèses simplificatrices. Si le délai moyen est bien la somme des délais moyens sur les différents éléments qui constituent un chemin, c'est plus délicat pour la gigue ou la perte ! Les hypothèses d'indépendance entre les différents liens sont difficilement maîtrisables !

### **DIAPO 38 : Qu'est-ce que cela donne lors l'on agrège ?**

L'agrégation permet de réduire les tables de routage (cf. cours TCP/IP) mais ce n'est pas tout. Agréger, c'est simplifier. L'ATM-Forum a fourni des aides pour cela. On peut par exemple représenter un sous-réseau par ses points d'entrée et de sortie ; on met un lien logique entre ces points que l'on va pondérer en disant : entre tel point d'entrée et de sortie, je peux fournir tel débit, tel délai, tel taux de perte, telle gigue...

On peut de façon plus simple le représenter par la forme du milieu où l'on rajoute un nœud logique central. Attention, cela permet de calculer moins de choses car on dira on possède un chemin qui est la concaténation des deux chemins du point d'entrée vers le nœud central puis du nœud central vers le nœud de sortie et en faisant une combinaison des paramètres on a les caractéristiques de ce chemin.

On peut même enrichir les informations en donnant des informations supplémentaires sur un lien physique qui existerait !

### DIAPO 39 : remarques

Les calculs sont loin d'être triviaux. En effet, dans un réseau, on a plusieurs chemins entre des points d'entrée et de sortie. Ce n'est pas du tout simple d'y arriver ; il faut regarder les qualités des différents chemins : on n'additionne pas les débits sur les différents chemins...

Encore une fois, plus on a de paramètres à calculer, plus c'est compliqué et plus ce sera lourd en termes de signalisation associée.

Qui plus est, il faut trouver le compromis entre qualité de l'information et coût de la signalisation associée.

Remarquons enfin que le standard ne fournit pas les algorithmes mais simplement le protocole PNNI qui sera utilisé pour distribuer les informations.

### DIAPO 40 : Calcul des routes

Maintenant que l'on a les informations, il faut calculer les routes. Dans les réseaux de type IP, on peut avoir plusieurs métriques disponibles (nombre de sauts, débit...) mais on utilise une seule métrique à la fois. On utilise alors l'algorithme de Dijkstra.

Là ça se gâte car il faut trouver un chemin qui satisfasse des contraintes (attributs) et qui minimise potentiellement plusieurs métriques. C'est un problème d'optimisation multi-critère. On ne sait pas s'il y a une solution unique et il est NP-complet.

En général, on choisit parmi les chemins qui satisfont les contraintes, celui qui minimise une première métrique. En cas d'égalité, on va sur une deuxième métrique...

### DIAPO 41 : Calcul des tables

On se rend compte alors que la route peut dépendre de la demande de connexion. Une première solution pourrait être de lancer le calcul des routes à la demande de connexion. Ce n'est pas viable. Les solutions vont alors consister à précalculer un certain nombre de routes, d'en choisir une quand la demande de connexion survient et si aucune ne convient à lancer l'algorithme (l'autre solution aurait pu consister à ne pas accepter la connexion)

### DIAPO 42 : Acheminement

On passe maintenant à l'étape suivante de l'acheminement. La demande de connexion part de la source. Le premier commutateur regarde sa table de routage et récupère une route qu'il met dans le message de demande de connexion, en précisant le chemin complet dans le présent exemple. La demande de connexion suit cette route et chaque commutateur vérifie qu'il peut toujours accepter cette demande de connexion. On met à jour la table de commutation en fonction.

### DIAPO 43 : Acheminement – version agrégée

Attention, on met un peu à mal la notion de routage par la source dans la version avec agrégation. Le route indique à gros grain A-B-C puis dit que pour atteindre B on passe par A.1-A.2 puis que dans A.1 pour aller à A.2 on passe précisément par A.1.1 et A.1.2

Par conséquent cela signifie qu'à l'entrée dans chaque sous-réseau, on précise la route !

### DIAPO 44 : Crankback

Si un nœud ne peut pas accepter la connexion ou qu'il y a un lien en panne, 3 solutions se présentent :

- On refuse la connexion
- On repart à la source et on recommence
- On repart du dernier nœud qui a fait des calculs (qui a regardé sa table de routage) et on repart de là.

C'est cette dernière solution qui a été retenue. La deuxième solution n'est pas satisfaisante car la deuxième meilleure route peut passer par le même tronçon car on n'a pas la route complète !

Bon c'est un peu anecdotique. En revanche, le couplage routage QoS et protocole de réservation de ressources est très bien pensé... mais terriblement compliqué !

**DIAPO 45 :** On va terminer par la boîte à outil des outils de qualité de service qui a été mise en place pour ATM.

**DIAPO 46 :** On rappelle que l'on était parti du fait que c'était le rôle du découpage en classe et en AAL qui devait faire une grande partie du travail.

Les AAL n'ayant pas tenu leurs promesses, on a découplé la qualité de service de l'AAL ; tout passant plus ou moins par l'AAL-5.

C'est le rôle des contrats de trafic.

Attention l'ATM-Forum et l'ITU-T n'ont pas retenu le même vocabulaire.

Les classes de service de l'ATM-Forum sont

- CBR = Constant Bit Rate
- VBR = Variable Bit Rate (temps réel et non temps réel)
- ABR = Available Bit Rate (une variante Guaranteed Frame Rate où la garantie de QoS porte sur les messages et pas les cellules)
- UBR = Unspecified Bit Rate

Bon on n'est pas loin des 4 classes initiales toutefois.

### DIAPO 47 : Et voilà des garanties qui seront apportées aux différents trafics

Pour CBR, du délai, du débit, de la gigue, de la perte, on ne peut pas lui demander de réduire leur débit.

Pour VBR, c'est un peu pareil. Quand ce n'est pas temps réel, pas de garantie de gigue.

Ces deux types de trafic ne peuvent pas réduire leur débit !

C'est la voix et la vidéo.

Pour ABR, on va garantir de la perte et un débit minimal. Ces trafics seront soumis au contrôle de congestion.

Pour UBR, tu ne dis rien, tu n'as rien et en plus tu subis le contrôle de congestion.

(l'ITU-T était vent debout contre ce type de trafic).

L'idée est en bas : le trafic CBR est parfaitement connu. Le trafic VBR fluctue dans le temps et l'objectif est de répartir ce qui reste entre les flux ABR en leur garantissant un minimum. Mais cela n'est que sur un lien donc c'est plus compliqué sur un réseau.

### DIAPO 48 : Paramètres de qualité de service

Un point très important en ATM a été de séparer les descripteurs de trafic qui vont permettre à l'utilisateur de caractériser son trafic à l'entrée du réseau ATM et les besoins en qualité de service qui vont caractériser ce dont il a besoin.

L'autre point important c'est que les paramètres dépendent du trafic. C'est donc différent du FR où les valeurs des paramètres dépendent de la connexion, ici ce sont les paramètres.

Comme descripteur, on va trouver

- le débit crête qui est un débit maximum et qui est là pour tout le monde (sauf UBR qui ne dit rien).
- Le débit minimum pour le seul ABR
- Le débit moyen à long terme pour le trafic VBR

- La gigue à l'accès au réseau (cf. diapo suivante)
- Le nombre maximal de cellules que l'on peut envoyer au débit crête (VBR). On n'a pas réussi à se mettre d'accord sur un paramètre de sporadicité qui caractérise le comportement moyen de la connexion et son comportement instantané.

Les paramètres de qualité de service attendu sont découpés en deux : des paramètres négociables et des paramètres qui ne le sont pas : Cell Error Rate et SECBR caractérise les erreurs de transmissions, et Cell Misinsertion Rate, les cellules qui sont envoyées dans une mauvaise connexion. Les paramètres négociables sont la gigue introduite par le réseau, le délai et la perte. Les MaxCTD sont vérifiés en regardant les pires cas de trafic.

#### **DIAPO 49 : Gigue à l'accès**

En entrée du réseau, on a déjà de la gigue qui est due au multiplexage des différents flux et la gigue d'insertion précédemment évoquée. Attention, un flux VBR est lui-même intrinsèquement gigué par définition.

Ici on ne parle pas de la gigue introduite par le réseau qui est liée au multiplexage introduit par le mode paquet. Attention, même avec des flux purement CBR, on a de la gigue car les nombre de flux multiplexés varient d'un commutateur à l'autre !

#### **DIAPO 50 : Comment mesurer la gigue**

On peut mesurer la gigue en deux points de mesure. La différence entre les délais séparant les passages en ces deux points constitue la gigue. Malheureusement, si l'on regarde bien les cellules ATM, on se rend compte qu'on ne peut pas donner les heures de passage au point de mesure 1. On ne peut réserver cette technique qu'au plan de gestion qui va justement le faire en estampillant temporellement les cellules.

La deuxième façon va consister à ne le faire qu'en un point. On surveille un débit. On regarde les écarts successifs par rapport à un échéancier théorique qui avance au rythme de l'inverse du débit que l'on surveille.

#### **DIAPO 51 : Problèmes liés à la gigue**

Les trafics fortement contraints temporellement tels que la voix et dans une moindre mesure sont sensibles à la gigue. Dans un réseau paquet, on peut absorber une partie de la gigue au niveau du récepteur en bufferisant les données au fur et à mesure de leur arrivées en espérant que les données arriveront ensuite suffisamment tôt pour qu'il n'y ait pas de famine. Il faut aussi dimensionner le buffer de réception. Les paramètres de délai initial et de taille de buffer sont des résultats d'étude de performances. Notons que ce délai initial est forcément beaucoup plus faible pour de la voix que pour du streaming.

#### **DIAPO 52 – Generic Cell Rate Algorithm**

Le contrôle des paramètres utilisateurs est apparu clairement avec le relayage de trames dans lequel on surveillait le débit sur une plage de temps. En revanche, il n'y avait pas de garanties de gigue, ni de surveillance de la gigue à l'accès. Là c'est un peu plus compliqué car on veut regarder aussi l'espacement entre les cellules à l'accès.

C'est à cet effet que l'algorithme GCRA a été standardisé. Deux configurations se présentent : soit la cellule est en retard ou à l'heure et c'est bon pour elle (en fait l'utilisateur utilise potentiellement moins que ce qu'il a demandé) ; soit elle est un peu en avance et dans ce cas, il ne faut pas qu'elle le soit trop ! Dans ce dernier cas, c'est que le contrat de trafic n'est pas respecté et donc le réseau peut la refuser... (cf. FR).

Attention ce n'est pas un mécanisme de Traffic Shaping, si elle n'est qu'un peu en avance, on ne la retarde pas car en sortie du commutateur, elle va de nouveau subir de la gigue.

**DIAPO 53 :** Attention la mise à jour de l'échéancier théorique permet d'éviter le phénomène qui est figuré sur la figure. En effet une source qui n'émettrait pas à son débit accumulerait du droit à émettre qui se traduira par des salves de trafic trop longues qui seraient très pénalisantes pour le réseau.

Un mécanisme un peu semblable a été obtenu à l'aide d'une file d'attente. C'est le mécanisme de Leaky Bucket qui est une file d'attente à débit de sortie constant (le débit négocié) et une capacité limitée qui mesure l'accumulation de cellules légèrement en avance. Attention, c'est de la même façon une version algorithmique et pas une vraie file d'attente. On ne fait pas attendre les cellules mais on compte à tout moment combien on en a.

En fait GCRA et une version fluide du Leaky Bucket (on a un nombre réel de cellules dans la file et pas un nombre entier) sont équivalentes. Quand la file déborde, c'est que le trafic n'est pas conforme.

Vous reparlerez de ces mécanismes dans les cours de QoS au S8 et au S9.

#### **DIAPO 54 – Boîte à outils des mécanismes de qualité de service**

ATM a défini de très nombreux mécanismes de qualité de service qui sont enrichis par le routage QoS, la réservation de ressources, UPC déjà vu précédemment.

On va ajouter à l'échelle de temps du délai A/R des cellules, on peut mettre en place des mécanismes de qualité de service.

Ce sera par exemple pour les flux VBR, la possibilité de demander du débit en plus. Plusieurs variantes ont été prévus : IT et DT. Soit on envoie la requête, on attend le retour et on envoie. On peut aussi envoyer la requête et les données juste en suivant avec le risque que cela ne passe pas.

Pour le trafic ABR, c'est le contraire, le flux n'a pas vraiment de débit et il prend ce qu'on lui donne. La question est de savoir comment faire pour éviter des injonctions contradictoires selon l'état des différents liens. Le protocole consiste à partir de la destination et d'envoyer un message de contrôle pour voir ce qu'on peut lui promettre en tenant compte de l'état courant du débit disponible et des autres connexions ABR en cours. En remontant vers la source, on prend à chaque fois le minimum entre ce qu'on peut lui donner sur le lien courant et ce qui s'est passé sur les liens précédents. Le principe d'allocation est AIMD.

Pour ce qui est du contrôle de congestion, il ne concerne que les trafics ABR et UBR. C'est un peu la même chose qu'en FR. On a juste ajouté des messages pour que le nœud congestionné demande de réduire le débit.

A l'échelle des cellules ATM, on a le contrôle des paramètres utilisateurs. On peut lisser le trafic. On peut supprimer les cellules non vitales (trafic ABR), suspendre les ressources si le trafic n'est pas conforme.

C'est aussi le moment où les commutateurs ont commencé à implanter des mécanismes d'ordonnancement qui ne soient pas premier arrivé premier servi. Le choix porté sur des files d'attente séparées par flux ou par classe de service. C'est plutôt ce type de solution qui a été retenu car plus simple et permettant plus facilement le passage à l'échelle ! Les ordonnanceurs peuvent faire simplement des priorités absolues CBR > VBR > ABR > UBR.

On peut faire de l'ordonnancement Weighted Round Robin aussi en mettant des poids plus élevés pour les trafics plus prioritaires.