

Réseaux et QoS

Ingénierie des Réseaux d'Entreprise

Décembre 99

Eric Gressier-Soudan

Plan

1. Contexte et concepts généraux

2. Réseaux et la QoS

3. Internet et QoS

Références Bibliographiques

Etude et Implantation d'un service de liaisons temps réel dans une plateforme à objets répartis. A. Arazo. Mémoire d'Ingénieur CNAM. Paris. Novembre 1996.

Open Distributed Processing. G. Blair, J-B. Stefani. Addison Wesley. 1997.

Qualité de Service (QoS) et Contrôle de trafic dans les réseaux IP – Tutoriel. O. Bonaventure. Global Networked Solutions. 22 Octobre 1999.

Integrated Services in the Internet Architecture : an Overview. R. Braden, D. Clark, S. Shenker. RFC1633. July 1994.

Switched, Fast and Gigabit Ethernet, third Edition. R. Breyer, S. Riley. MacMillan Technical Publishing. 1999.

Ipv6, Théorie et Pratique. G. Cizault. O'Reilly. 1998.

Quality of Service : Delivering QoS in the Internet and in Corporate Networks. P. Ferguson, G. Huston. J Wiley. 1998.

Les Communications Multipoints dans les Réseaux Haut Débit Multimédia : Le Multicast en environnement IP sur ATM. O. Fourmaux. Thèse de Doctorat de l'Université P. et M. Curie. 14 Décembre 1998. Paris.

Le Routage dans l'Internet. C. Huitema. Eyrolles. 1995.

Conception d'un Système Distribué temps Réel fondé sur ATM. C. Lizzi. Thèse du CNAM. Décembre 1999.

Deploying IP Multicast in the Enterprise. T. Maufer. Prentice Hall. 1998.

Multimedia : Computing Communications and Applications. R. Steinmetz, K. Nahrstedt. Prentice Hall 1995.

Les hauts débits en Télécoms. C. Servin, S. Ghernaoui-Hélie. InterEditions. 1998.

TCP/IP illustrated, Volume 1 : The protocols. W.R. Stevens. Addison Wesley. 1994.

IPng and the TCP/IP Protocols. S. A. Thomas. J. Wiley 1996.

The Use of RSVP with IETF Integrated Services. J. Wroclawski. RFC2210 September 1997.

CONTEXTE et CONCEPTS GENERAUX

Applications Multimédia (MM)

Applications de Présentation (unidirectionnel):

- Vidéo/Audio à la demande : Films/Musiques, Distribution TV (émissions, reportages...), Télé-surveillance (fabrication ->détection de pièces défectueuses, malades...)
- Courrier/Forum de discussions avec des données MM
- Système d'information (bornes interactives, Web)

Applications Interactives et Multi-participants: Vidéo Conférence, Télé-opération, Jeux en réseau, Café électronique

Applications cumulant les deux profils :

- Travail coopératif : prototypage rapide, entraînement en simulation (réalité virtuelle) , télé-maintenance
- Enseignement à distance

Plusieurs Média sont utilisés simultanément : son, image, image animée, graphiques ... **numérisés avec ou sans compression de données** (JPEG, MPEG-x, H261...) qui transforme un média en diminuant son volume mais en rendant son débit variable (apparition de rafales - burst).

Média

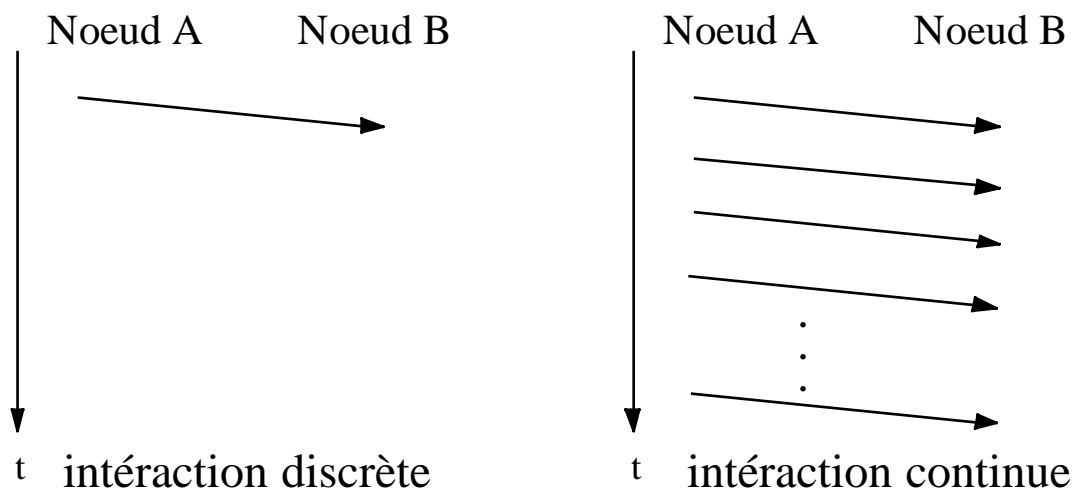
Media continu

Les données correspondent à un flux dont la présentation est assujettie à des contraintes temporelles (temps réel). Le temps séparant l'arrivée de deux données est connu et dépend de leur nature (voix -> 64kb/s soit 8 bits toutes les 125us, pour de la vidéo haute définition -> 200Mb/s soit 25 ± 5 images par s). L'infrastructure de transport doit intégrer tous les types de média.

Media discret

Les média discrets n'ont pas de contraintes temps réel (image, texte, graphique ...).

Modèles d'interactions associés :



Projection sur un bus logiciel orienté objet : Le cas continu est différent du passage de message synchrone ou asynchrone et de l'appel de procédure à distance (modèle Client/Serveur), de l'invocation de méthode d'un objet (Modèle Objets Répartis).

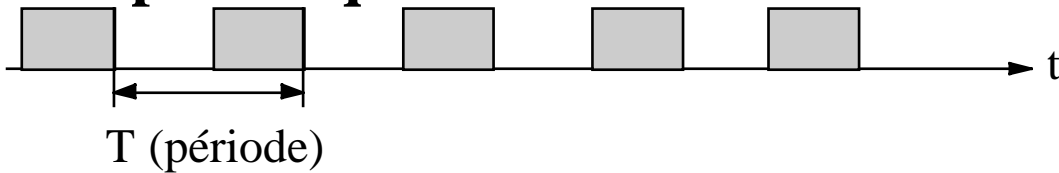
Contraintes sur les média à échanger

Nature des Flots de données échangés:

- Caractéristiques temporelles
 - **flot asynchrone** : appliqué aux média discrets, une donnée n'a pas de contrainte spécifique p/r aux données précédentes, elle atteint le récepteur le plus rapidement possible (politique "best effort" ... éventuellement n'arrive pas)
 - **flot synchrone** : appliqué aux média continus, chaque donnée est séparée de la précédente par un intervalle de temps fixe (séquence vidéo, voix numérisée ..), on dispose d'une **périodicité forte** (propriété CBR pour Constant Bit Rate, flot direct)
 - **flot isochrone** : appliqué aux média continus, chaque donnée est séparée de la précédente par un intervalle de temps moyen encadré par un temps minimum et un temps maximum (trafic sur FDDI est isochrone), le **flot** peut être **sporadique**, **faiblement périodique** ou **apériodique** (propriété VBR pour Variable Bit Rate, flot avec compression)
- Taille des données
 - **Constante**
 - **Variables avec périodicité des tailles**
 - **Complètement irrégulières**

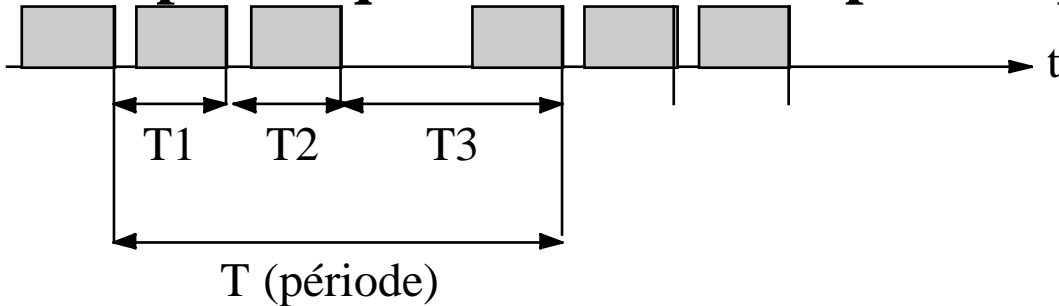
Flots de Données - Caractéristiques temporelles

Flot périodique

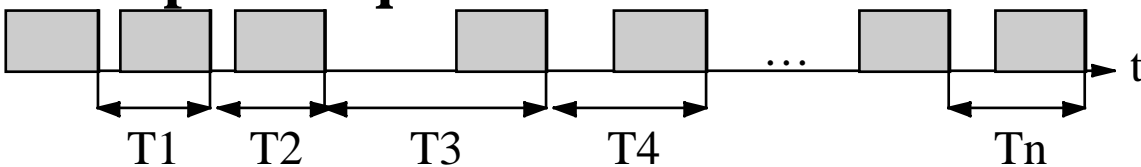


La gigue est nulle

Flot sporadique ou faiblement périodique

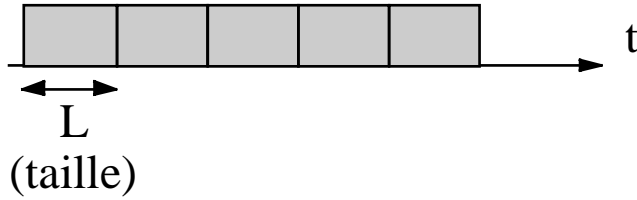


Flot apériodique

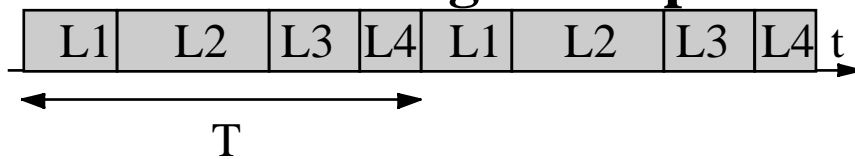


Flots de Données - Volume

Taille constante



Taille avec changement périodique



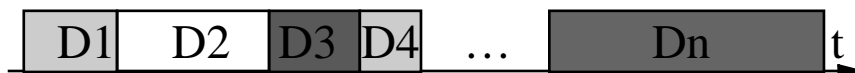
Flot de messages irréguliers



Flots de Données - Continuité

(vue logique)

Flot continu



Flot discret



Quelques Repères sur la numérisation des données audio/vidéo

La numérisation procède par échantillonnage d'un signal analogique.

La fréquence d'échantillonnage dépend de la fréquence du signal analogique ($F_E = 2 * F_S$ d'après Nyquist).

Exemple le plus classique : Numérisation de la voix

Bande passante de la voix humaine 200-3200Hz soit 3000 Hz, le téléphone a choisi 4 KHz, la fréquence d'échantillonnage est donc de 8000Hz, et chaque échantillon est codé sur 8 bits -> 64Kb/s

Pour le son qualité CD, on considère la Bande passante de l'oreille 20Hz-20Khz, la fréquence d'échantillonnage considérée est 44,1 KHz, chaque échantillon est codé sur 16 bits, d'où un débit de 1411200Kb/s (2 voies à cause de la stéréo).

Les images sont représentées par des ensembles de points (pixels) représenté par 1 bit, 8 bits (couleur ou dégradé de gris), ou 24 bits (couleur-brillance). Après, il faut considérer le nombre de pixels par ligne, et le nombre de lignes d'une image. Le poids d'une image sur un écran SVGA de résolution 1024*768 est de 18,874 Mb/s.

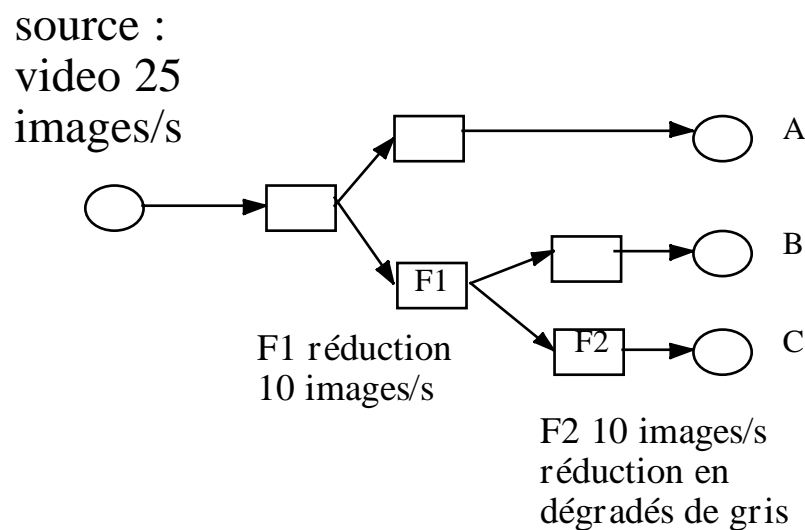
Pour les vidéo, on parle de vitesse de projection exprimée en images (frames) par secondes. La valeur standard varie de 25 à 30 images par secondes. Suivant le format de l'image, on a de l'ordre de 80Mb/s (standard) à 200Mb/s (TV haute définition) suivant la qualité de la vidéo transmise.

Communications Multi-participants

Schémas d'interactions : 1 vers N (broadcast/multicast), N vers 1 (supervision), et M vers N (conversation)

Gestion de groupes de participants à constitution dynamique

Conséquences sur la gestion des contraintes temporelles : les différents participants et les moyens d'acheminement des flux n'ont pas tous les mêmes capacités/caractéristiques.



Conséquences sur la synchronisation : ce n'est pas seulement la synchronisation temps réel, mais contraintes de séquençement des données (ordre local à la source, ordre total, ordre causal...)

Intéractions entre flux de données

Modèle utilisateur : Producteur-Consommateur (Push-Pull)

Flux simple : Les contraintes de temps portent sur les éléments du flot entre-eux (synchronisation intra media)

Flux complexe : Plusieurs flux sont synchronisés les uns par rapport aux autres (synchronisation inter-media)

exemple : la synchronisation son-mouvement de lèvres

Ces contraintes doivent être vérifiées même si les entités participantes sont éloignées et multiples (schéma diffusion).

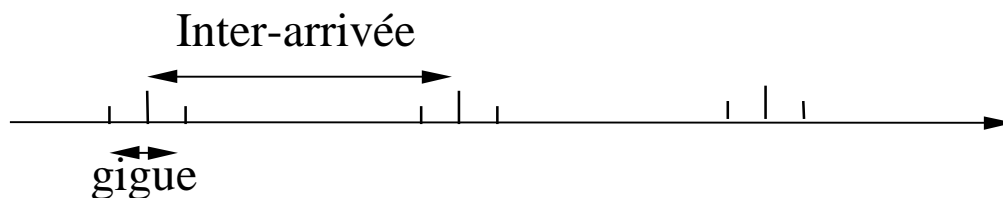
Les contraintes de synchronisation peuvent être dynamiques, c'est à dire connues à l'exécution (interactivité et non prédictibilité, par exemple séquence vidéo lors d'un interview).

Il faut disposer d'un langage de spécification des contraintes de synchronisation temporelles.

Catégories des Contraintes de Qualité de Service

3 principales catégories de contraintes :

- . **Temporalité** : temps de transfert (latence) en ms pour continu et discret, gigue en ms pour continu



- . **Volume** : débit exprimé en b/s pour les interactions discrètes, en unités de données par s pour les interactions continues
- . **Fiabilité** : pour un flux continu taux de perte d'unités de données ou taux d'erreurs bit par unité de données, pour un flux discret taux d'erreurs bit uniquement

Autres catégories possibles : criticité des informations, causalité des échanges, propriétés de sécurité

Les contraintes peuvent s'exprimer de façon déterministe, de façon probabiliste (taux de satisfaction), stochastique (loi de distribution aléatoire).

Classes de qualité de service : QoS garantie, ou QoS "best effort"

Opérations de Gestion de la QoS

- **Spécification de la QoS:** création d'un contrat entre l'application, et l'environnement d'exécution
Négociation de la QoS: en vue d'obtenir un accord entre utilisateur et fournisseur
- **Contrôle d'Admission :** tests qui déterminent si le système est capable de supporter le contrat requis
Réservation de Ressources: pour garantir le contrat accepté
- **Surveillance de la QoS :** surveillance par l'utilisateur du respect des contraintes de QoS qui ont été garantie par le fournisseur, un grain de surveillance doit pouvoir être indiqué 100ms par exemple
Vérification de QoS : respect du contrat de QoS par l'utilisateur
Maintenance de la QoS : actions prises par le fournisseur en cas de défaut constaté sur la QoS garantie
- **Renégociation de QoS :** si la maintenance ne parvient pas à rétablir le niveau de service demandé, l'utilisateur doit pouvoir renégocier son contrat

Contraintes des applications multimédia

En résumé des contraintes spécifiques des applications multimédia distribuées :

Support des schémas d'interaction Multimédia,

Synchronisation temporelle intra-media et inter-media avec contraintes de délai d'acheminement et de débit,

Gestion de Contraintes de Qualité de Service (QoS),

Communications multi-participants.

Besoins :

Modèle de Description d'Application

Modèle de Spécification de QoS

Modèle d'Architecture

Propriétés architecturales des solutions

Vers une approche Systèmes Multimédia Distribués (Multimédia # Temps Réel) :

- . **Partage de Ressources**
- . **Disponibilité - Fiabilité** (Réplication)
- . **Extensibilité - Modularité - Ouverture** (approche Composants, Orienté Objets)
- . **Performance** (contraintes multimédia)
- . **Décentralisation** (structures d'organisation et prises de décisions)
- . Prise en compte du **grand nombre** (machines/utilisateurs/réseaux/...)
- . **Hétérogénéité** (matériels, réseaux, OS, logiciels, langages programmation)
- . **Interopérabilité** (standards)
- . **Contraintes Spécifiques du Multimédia**

Architecture Ouverte

Plateformes Candidates

OMG-CORBA

OSF-DCE

MS-DCOM

SM-Java

Insuffisantes, pas de réponses aux contraintes spécifiques des applications multimédia : proposition d'extensions ou un nouveau modèle

Modèle de Systèmes Répartis : RM-ODP

RM-ODP: Reference Model of Open Distributed Processing

définit

une Architecture des Systèmes Répartis Ouverte
et Hétérogène

Modèle Générique

fondé sur la notion de **Points de vue** (aspects d'analyse et spécification) :

- **Entreprise** (besoins des applications, contraintes opérationnelles et organisationnelles)
- **Information** (modèle des données)
- **Traitements** (modèle des traitements sous forme d'Objets qui interagissent)
- **Ingénierie** (objets, OS, protocoles, liens réseau, contraintes sur l'infrastructure logicielle et matérielle ...)
- **Technologie** (implantation et conformité à la spécification...)

Approche Orientée Objet

Une plateforme de type CORBA pourrait être candidate p/r au Point de Vue de l'Ingénierie et de RM-ODP. Elle serait aussi relativement conforme au Point de Vue des Traitements.

Rappels sur l'Orienté Objet

Objet :

- Etat + Données Internes (inaccessibles de l'extérieur)
- Opération = point d'accès à une fonction exécutable par un objet
- Interface regroupe des opérations, sert à la désignation des Opérations

Signature d'une opération : nom d'opération + paramètres

Un objet n'évolue que par l'utilisation des opérations de son interface.

Propriétés des objets :

- . Encapsulation
- . Héritage
- . Polymorphisme
- . Agrégation

Avantages :

- . Séparation entre spécification et implantation
- . Modularité et Extensibilité
- . Réutilisabilité des composants
- . Adéquation des abstractions : Analyse, Spécification, Conception, Programmation

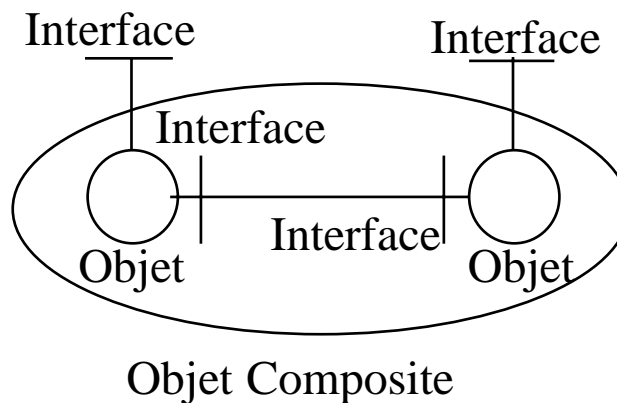
Modèle Objet d'ODP

Modèle objet des langages de programmation
+ plusieurs interfaces par objet
+ notion d'objet composite (réparti)

une interface = une vue abstraite de l'objet (concept de rôle)

exemple : une interface d'accès aux opérations, une interface pour les opérations de gestion

Schématisation d'un objet RM-ODP :

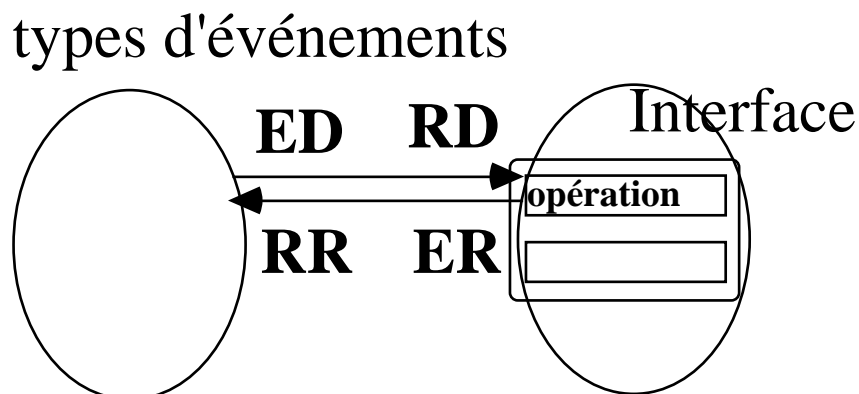


Modèle des Traitements de ODP

Repose sur la notion d'Objets en environnements répartis.

Application = ensemble d'objets

Intéraction = Invocation d'Opération, ou Réaction d'Invocation décrite à l'aide d'événements



Chaque événement a :

- un Type,
- un Nom,
- une Valeur

Conformité des Traitements : QoS

Modèle des Traitements :

=> Pouvoir Spécifier des Comportements et des Propriétés

=> Vérifier les Spécifications

Règles de Conformités exprimées sur les événements, donc sur les interfaces et sur les communications entre objets

Exemples :

- règle comportementale : ED et RR sur un objet Client
- règle temporelle : le délai entre RD et ER est Δ

Qualité de Service (QoS):

- taux d'erreur message,
- taux d'erreur bit,
- débit,
- garantie de délai

...

On a bien une notion de **Contrat** associée à une interface.

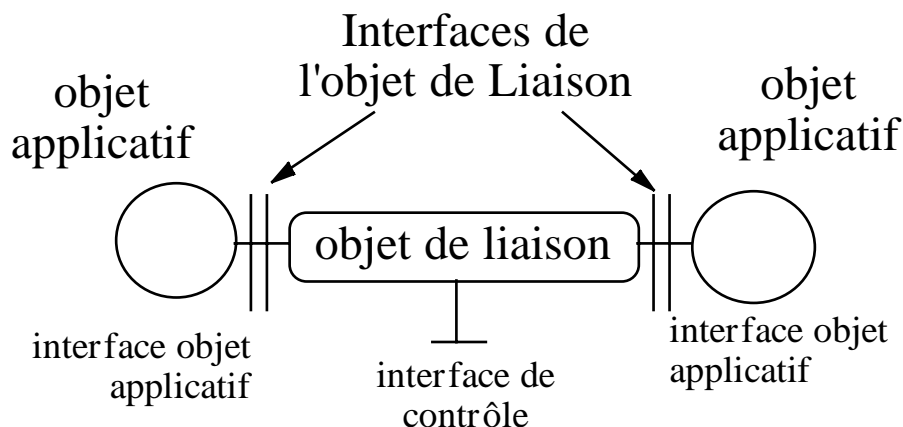
La QoS garantie par un objet implique la possibilité de supporter des propriétés de QoS par l'environnement d'exécution.

Liaisons entre Objets

Une Liaison correspond à l'abstraction du mécanisme qui supporte une interaction entre Objets.

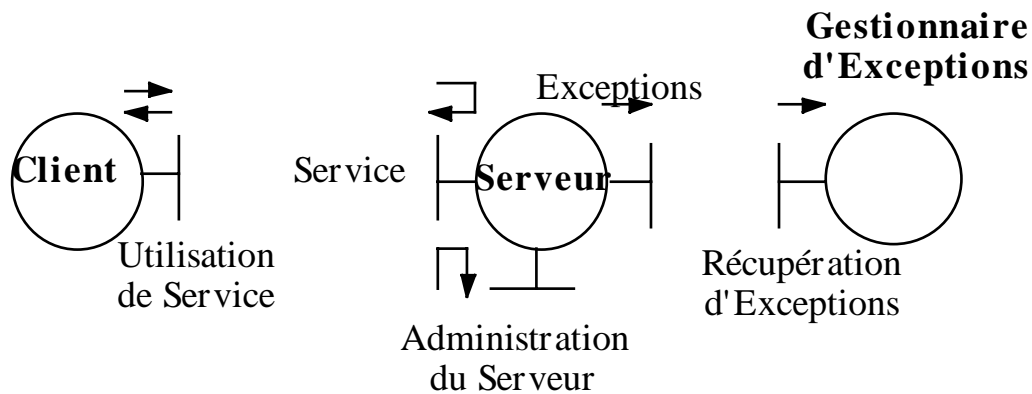
Les contraintes de QoS sur les interactions entre Objets imposent des contraintes de QoS sur les Liaisons.

Les Liaisons deviennent des Objets avec une Interface de Contrôle qui permet de les manipuler.



On voit ici l'extension de la notion de connexion ou d'association qu'on trouvait dans les couches du modèle ISO.

Modèle d'objets de base



correspond aux objets dans CORBA

Il existe des objets particulier qui ont la charge d'ajouter des nouveaux objets à l'environnement. Ces objets particuliers s'appellent des **fabriques à objets** (object factory).

Les objets sont fabriqués à partir de modèles (template) complètement spécifiés y compris le contrat de QoS, c'est un processus d'instanciation.

Extensions Multimédia du Modèle Objet

Interface production et consommation de flux



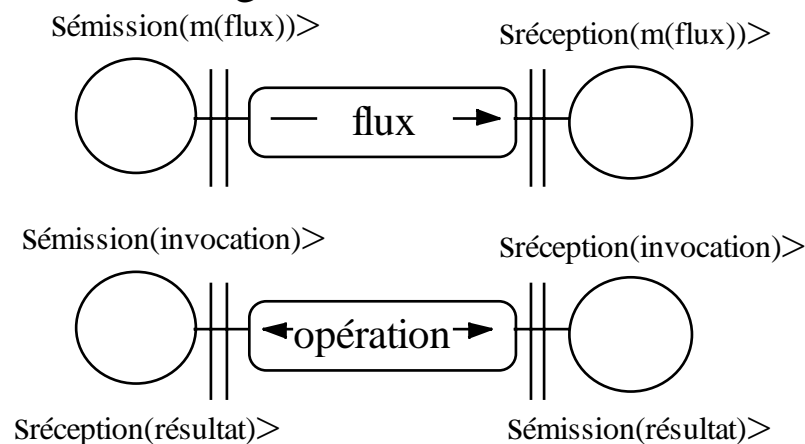
Une interface de type flux supporte des interactions multimédia de type continu, elle est définie en terme d'un ou plusieurs flots de données. Chaque flot est caractérisé par un nom, un type de media géré, et la direction du flot.

Interface signaux ou événements temps réel



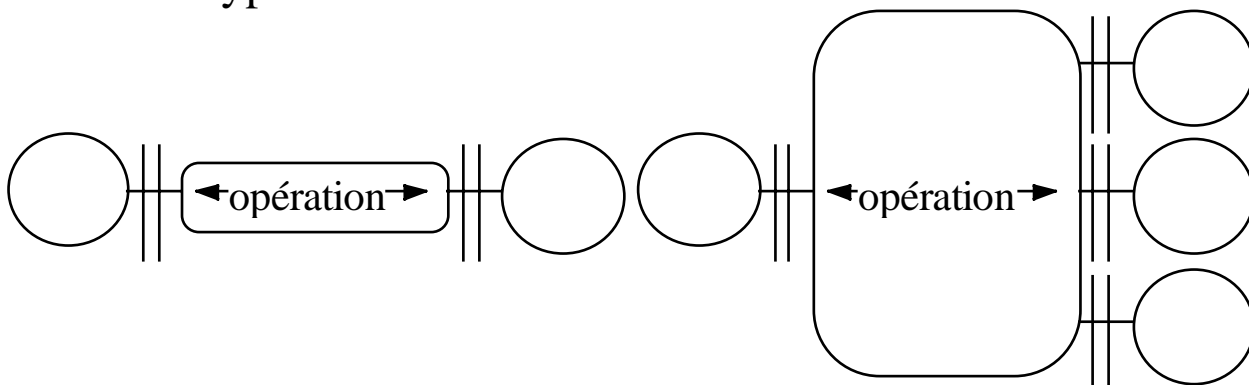
Les interfaces de type signal sont utilisés pour la gestion de la QoS et la synchronisation temps réel. Chaque signal dans un interface est représenté par un nom, le type, sa valeur, et la causalité induite (générateur, consommateur).

Les signaux peuvent être utilisés aussi bien pour des flux que pour des services, dans chaque cas ils représentent l'émission ou la réception d'un message.

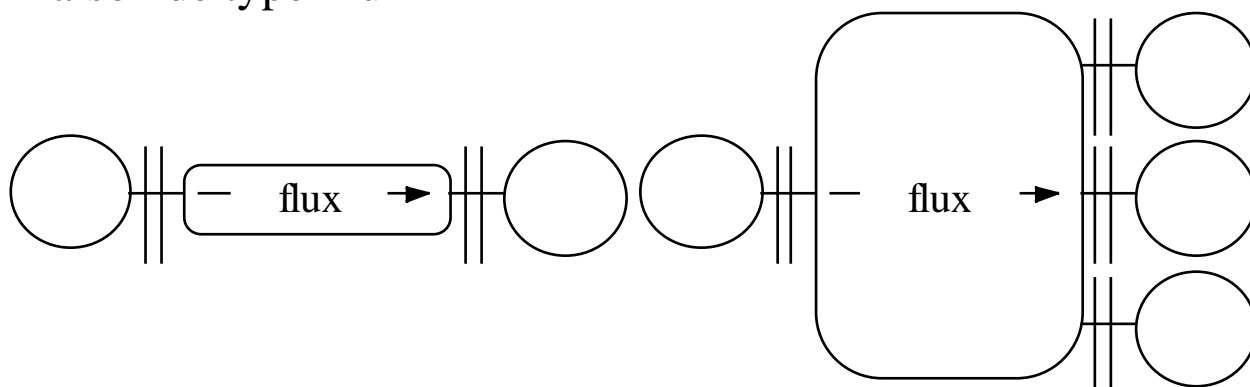


Modèles de Liaisons entre Objets

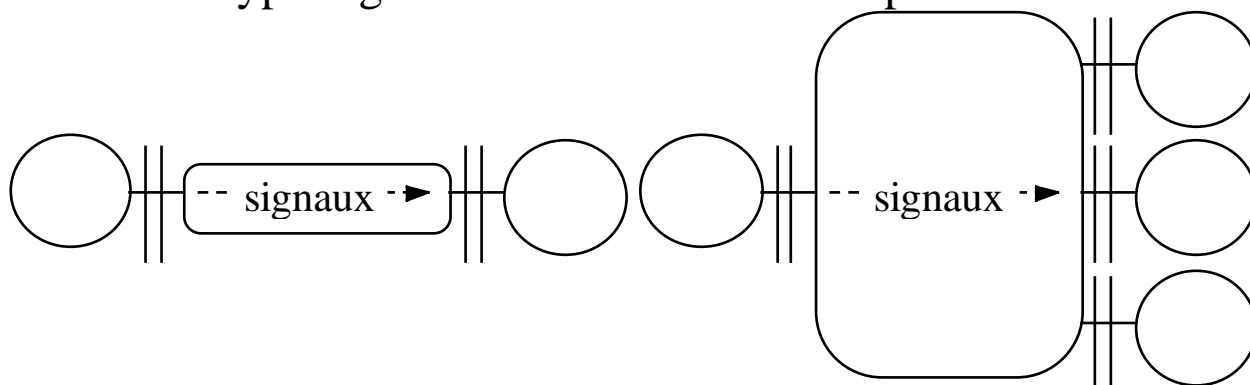
Liaison de type Service



Liaison de type Flux



Liaison de type Signaux ou Evénements Temps Réel



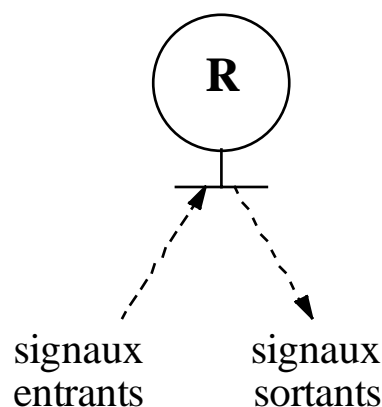
Objets réactifs

Les objets présentés sont des objets asynchrones. Objets ou objets de liaison, ils prennent un certain temps pour exécuter leur traitement.

La spécification des contraintes de QoS temporelle sert à contraindre le comportement des objets, et à édicter des contraintes environnementales sur l'architecture.

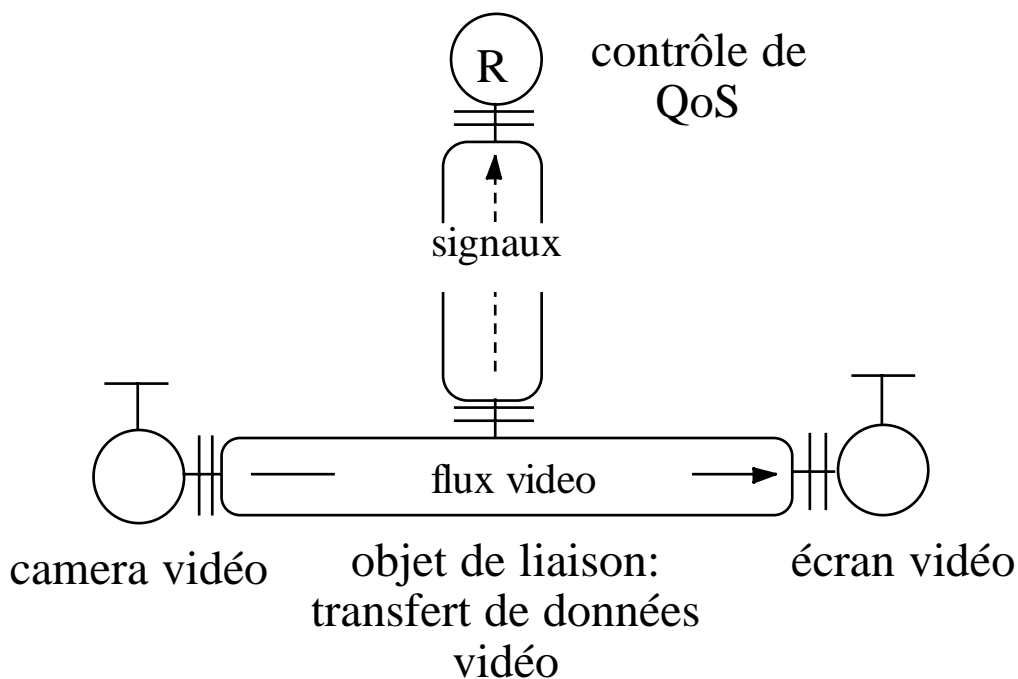
Les Objets Réactifs ont un rôle particulier, ils servent à contrôler le comportement d'objets, en particulier à observer l'évolution de la QoS ou à gérer la synchronisation temps réel. En ce sens, ils reçoivent des signaux, les traitent et envoient des signaux.

Les objets réactifs ont un comportement synchrone. Ils s'exécutent en un temps nul. Ils sont programmés en langage synchrone (LUSTRE, SIGNAL, ESTEREL).



Exemple de schématisation d'une application Multimédia

Exemple de modélisation :



Echange Vidéo avec Moniteur de QoS

Temps Réel Multimédia

Temps Réel Multimédia = QoS + Contrôle

Objets réactifs (à exécution immédiate) => Contrôle

Objets applicatifs ou système (asynchrones contraints par une spec de QoS) => Traitements

Avantages :

- . Les contraintes temporelles sont formalisées explicitement par des équations de QoS
- . Claire séparation entre le contrôle et l'application
- . Portabilité, seule les équations de QoS doivent être recalculées avec un nouvel environnement

Equations de QoS

La spécification d'équations de QoS (ou plutôt d'inégalités) est fondée sur un modèle événementiel (suppose donc un temps discret). Exemples avec expression déterministe :

Latence bornée d'un transfert d'images vidéo :

pour tout n , (n représentant l'occurrence d'un événement) :

$$|\text{date}(\text{réception-image-video},n) - \text{date}(\text{émission-image-video},n)| \leq \text{latence}$$

Gigue d'un transfert d'images vidéo :

pour tout n :

$$\min \leq |\text{date}(\text{réception-image-video},n) - \text{date}(\text{émission-image-video},n)| \leq \max$$

$$\text{gigue} = \max - \min$$

Débit max d'un transfert d'images vidéo :

pour tout n :

$$|\text{date}(\text{réception-image-video},n+k) - \text{date}(\text{réception-image-video},n)| \leq \text{durée}$$

$$\text{débit} = k/\text{durée}, \text{ en images par seconde}$$

Taux de perte d'un transfert vidéo :

pour tout n :

$$(|\text{date}(\text{émission-image-video},n+k) - \text{date}(\text{émission-image-video},n)| \leq d) \text{ et } (|\text{date}(\text{réception-image-video},n+k') - \text{date}(\text{réception-image-video},n)| \leq d) \text{ et } (k' \leq k)$$

$$\text{taux de perte } 1 - (k'/k)$$

Exemple (1)

Dans la réalité un événement est repéré par :

nom-d'une-interface.nom-d'un-signal.causalité avec causalité pouvant prendre les valeurs (ED/ES₁,RD/RS,[ER,RR] conformément au modèle vers p20 du poly)

Exemple du transfert vidéo entre une caméra et un écran

Objets :

Caméra avec l'interface vidéoOut, et écran avec l'interface vidéoIn

L'expression de la contrainte "latence bornée" devient :

pour tout n :

$|date(\text{écran.vidéoIn.RS},n)$

$- date(\text{caméra.vidéoOut.ES},n)| \leq 10\text{ms}$

La spécification de paramètres de QoS est construite en indiquant, des clauses requises (Req) et des clauses fournies(Prov) , avec la relation Req(A) -> Prov(A).

₁ D pour Donnée et S pour Signal, dans le cas des signaux, il n'y a pas de réponse (R)

Exemple (2)

Objet Liaison transfertVideo :

```
// interface avec la caméra
interface flux transfertVideoIn {
    fluxEntrant videoIn (video) ;
}
//interface avec l'écran de restitution
interface flux transfertVideoOut {
    fluxSortant videoOut(video);
}
//interface contrôle de QoS
interface signal qosControle {
    signalOut videoEmis (date) :
    signalIn videoDélivré (date) ;
}
```

Clause Fournie

//transfert tient le 25 images/s avec une latence entre 40 et 60 ms

pour tout n, date(transfertVideoOut.videoOut.ES, n+24)

<= date (transfertVideoOut.videoOut.ES, n) + 1000 ms

et pour tout n, date(transfertVideoOut.videoOut.ES, n)

<= date (transfertVideoIn.videoIn.RS, n) + 60 ms

et pour tout n, date(transfertVideoIn.videoIn.RS, n) + 40 ms

<= date (transfertVideoOut.videoOut.ES, n)

et pour tout n, date(qosControle.videoEmis.ES,n)

= date(transfertVideoIn.videoIn.RS, n)

et pour tout n, date(qosControle.videoDélivré.RS,n)

= date(transfertVideoOut.videoOut.ES, n)

Clause Requisite

// si la caméra envoie 25 images/s

pour tout n, date(transfertVideoIn.videoIn.RS, n+24)

<= date (transfertVideoIn.videoIn.RS, n) + 1000 ms

Point de Vue de l'Ingénierie

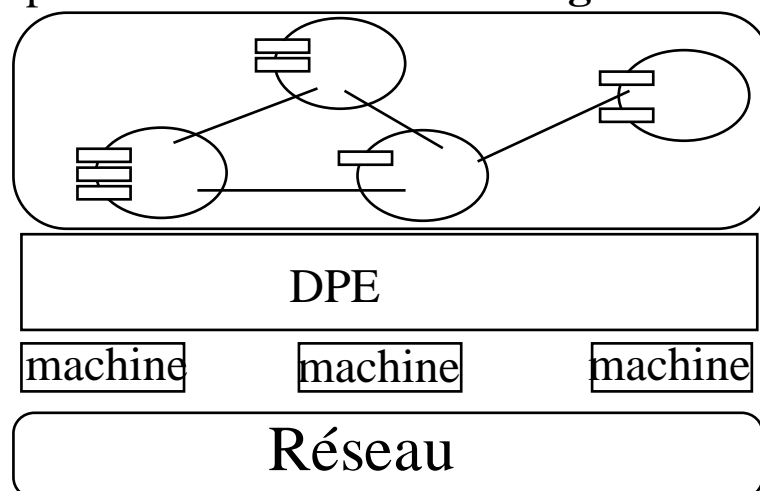
C'est l'ensemble des règles qui décrivent l'implantation des :

- Objets
- Liaisons

Le Projet **TINA** : Telecommunication Information Networking Architecture.

Objectif : Définir l'architecture des systèmes répartis du futur pour le monde des Telecom. C'est une spécialisation de RM-ODP qui vise des applications Multimédia.

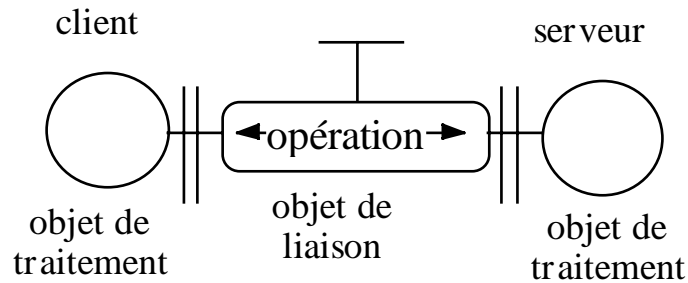
DPE pour **Distributed Processing Environment**



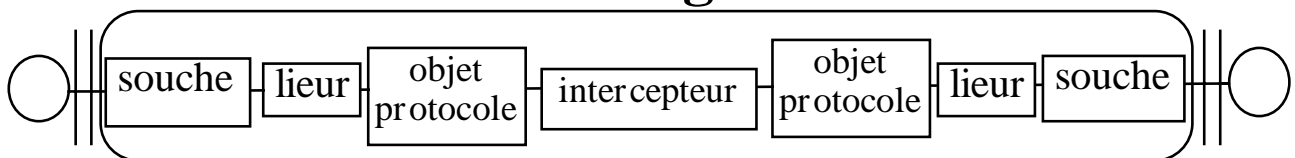
L'angle d'observation du cours est le DPE, et plus particulièrement la QoS pour le Transfert de Données en environnement de type Internet.

Modèle de Communication

Modèle des traitements :

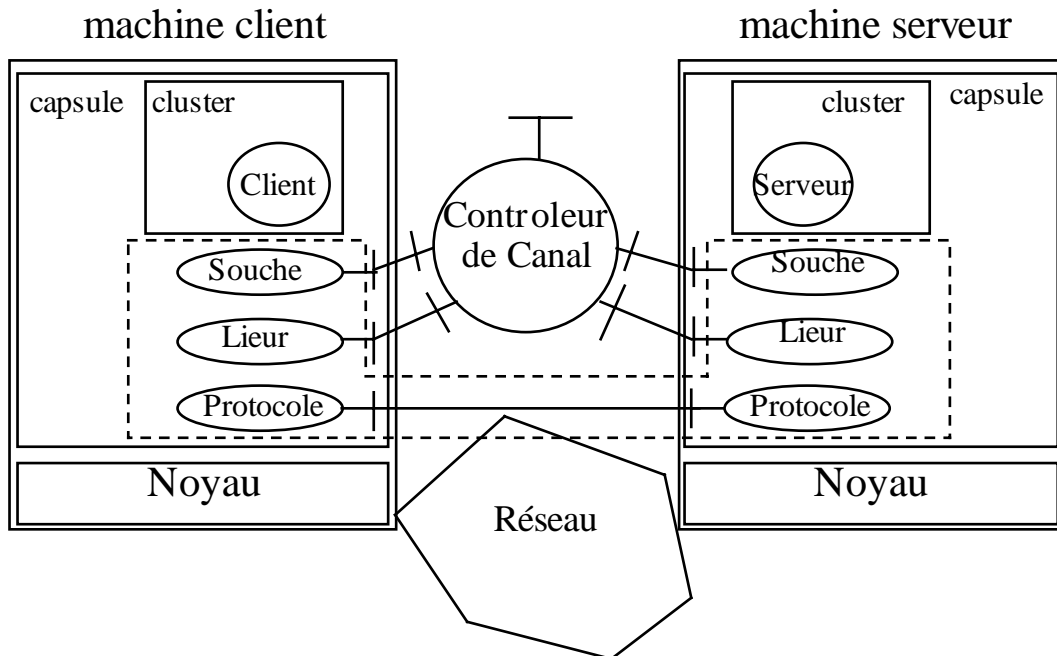


Modèle d'ingénierie :



Canal

intercepteur fonction équivalente à pont CORBA/passerelle
(conversion de protocole)



Systemes d'Exploitation pour le multimédia

Quelques caractéristiques :

Temps Réel : déterminisme logique et temporel
ponctualité plutôt que rapidité
travaille en échéances plutôt qu'en priorités

Problèmes traités :

- . Ordonnancement
- . Gestion Mémoire (allocation en temps borné, zéro-copie)
- . Placement des fichiers sur disque (stratégie d'ordonnancement)
- . Communication interprocessus
- . Architecture flexible, extensible, modulaire
- . Réseau et protocoles de communication temps réel

Objectif : Gérer la QoS en local et en distant

C'est un domaine à part entière qui mériterait une 1/2 valeur ;-) !

RESEAUX STANDARDS

et la

QOS

Support de la QoS

Niveau Physique : Raisonement en Bande Passante, on cherche le haut débit, une fiabilité intrinsèque, et si possible des propriétés temporelles (ISDN -> isochrone, réseaux à haut débit SDH-SONET, technologie ADSL, câble TV)

Niveau Liaison : Raisonement en débit, et en allocation de ressources d'accès par exemple FDDI, Frame Relay, Ethernet 802.1Q/p, on peut considérer ATM ici, le contrôle de flux (LLC, HDLC, PPP ...) entre dans la gestion des ressources

Niveau Réseau : Raisonement en débit à travers plusieurs réseaux, allocation et gestion de ressources (contrôle de congestion, approche de type connexion préférée, gestion des pannes de nœuds), l'incontournable IP et ses extensions (approches IntServ et DiffServ), ATM aussi

Niveau Transport : Raisonement en contraintes temporelles de bout en bout, de processus applicatif à processus applicatif (XTP, ST-II, suite Tenet), allocation de ressources mémoire (tampons) avec mécanisme zéro-copie, reprise sur erreurs suivant le type de trafic

Toutes les couches traversées introduisent des délais : emballage/déballage, gestion de PDU, attente en file (in et out)

Gestion de la QoS dans l'ISO

Exemple Tableau des paramètres de QoS en couche Transport :

- **Débit**
- **Temps de transit**
- Résiliation
- **Taux d'erreur relatif (pdt une période d'observation)**
- **Taux d'erreur total (erreur, pertes et doublons)**
- Temps d'établissement d'une connexion
- Probabilité d'échec d'une ouverture de connexion
- Temps de fermeture d'une connexion
- Probabilité d'échec d'une fermeture de connexion
- Coût
- Contrôle d'accès
- Priorité

Pas adapté ! pas de diffusion, prédominance du contrôle d'erreur, mécanisme de retransmission inadapté aux contraintes temporelles du multimédia, mécanisme de fenêtre inadapté, pas de garantie de QoS temporelle

Protocoles de Liaison - LAN

Token Ring, FDDI, Ethernet 10M-100M-1G
possibilité de Multicast

- Token ring : pb temps d'attente du jeton non borné, il faut utiliser le mécanisme de priorité mais pas d'équité, débit faible (16Mb/s)
- FDDI : 100Mb/s, pb du temps d'attente du jeton mais borné par le TTRT, apte à transmettre du multimédia pourvu que le réseau ne soit pas trop long, ni trop de stations (gigue de traversée) ... réseau d'artère qui tombe en désuétude, sinon version FDDI-2 pour écouler du trafic isochrone
- Ethernet : 100Mb/s et 1Gb/s, non déterministe, toutefois avec 802.1Q/p, possibilité de rendre certains trafics prioritaires dans la traversée des commutateurs ! Pas de garantie sur les délais de transfert. Logique de type "best effort"

802.1Q/p

Norme associée aux VLAN : extension de la trame Ethernet : passe de 1518 o à 1522 o

4 octets ajoutés devant le champ type (VLAN tag):

3 bits 1 b 12 bits
(TR)

Tag Protocol Identifier (8100 pour Ethernet)	USER PRIORITY	CFI	VLAN ID
	TAG Control Information		

2 Octets

2 Octets

8 niveaux de priorité, qui permettent à un commutateur d'écouler un trafic prioritairement à un autre

peut servir à mapper le champ priorité d'un datagramme IP

Protocoles de Liaison - WAN

- **X25** : faible débit (64Kb/s à 2Mb/s), protocole de contrôle de flux et de contrôle d'erreur de commutateur à commutateur (très fiable), pas support explicite de la QoS
- **Frame Relay** : réseau physique de 1,5 à 45Mb/s sous-jacent (52Mb/s avec la proposition HSSI), commutateurs fonctionnent en mode "forwarding", pas de contrôle de flux ni d'erreur entre commutateurs (mieux que X25 pour le multimédia), ébauche de gestion de ressource de QoS par le mécanisme d'indication de congestion (bits BECN et FECN²) et le bit DE³, possibilité d'utiliser le champ TOS d'un datagramme IP (délai, débit, fiabilité, coût) pour définir le bit DE
- **ISDN** : 2* 64Kb/s, réseau d'accès pour l'utilisateur, vidéo en compressé H261.

² BECN = Backward Explicit Congestion Notification, Forward Explicit congestion Notification

³ DE = Discard Eligible

ATM

Réseau à Commutation de Cellules qui a pour objectif de multiplexer différents flots de données en un seul lien qui utilise une technologie de type TDM ou MRF (Multiplexage à Répartition dans le Temps) comme SONET, SDH, PDH.

Couche
d'Adaptation -AAL

S-Couche de convergence
S-Couche SAR
Couche ATM
Couche Physique

Réseau Haut Débit : 155Mb/s sur un lien OC3

En fait la partie réservée aux données applicatives dépend de l'AAL.

Classes de Services et AALs

Les Classes de services considérées dépendent de l'intervalle de temps séparant les cellules, et de la tolérance à la gigue :

Classe	Description	Exemple
CBR (classe A)	Débit constant garanti	Audio/vidéo non compressé
rt-VBR (classe B)	Débit variable : trafic temps réel	Audio/vidéo compressé
nrt-VBR (classe C/D)	Débit variable : trafic non temps réel	Transactionnel
ABR (classe C/D)	Débit Disponible	Interconnexion de réseaux locaux
UBR (classe C/D)	Débit non défini	Données info, support d'IP

Différentes AAL sont proposées :

AAL 1 : pour transmettre les données temps réel à débit constant suivant un mode connecté orienté flot de bits, pas de détection d'erreur mais indication d'erreur (doute émis sur sa nécessité)

AAL 2 : transmission de données temps réel à débit variable, avec préservation des frontières des messages, mode connecté (obsolète)

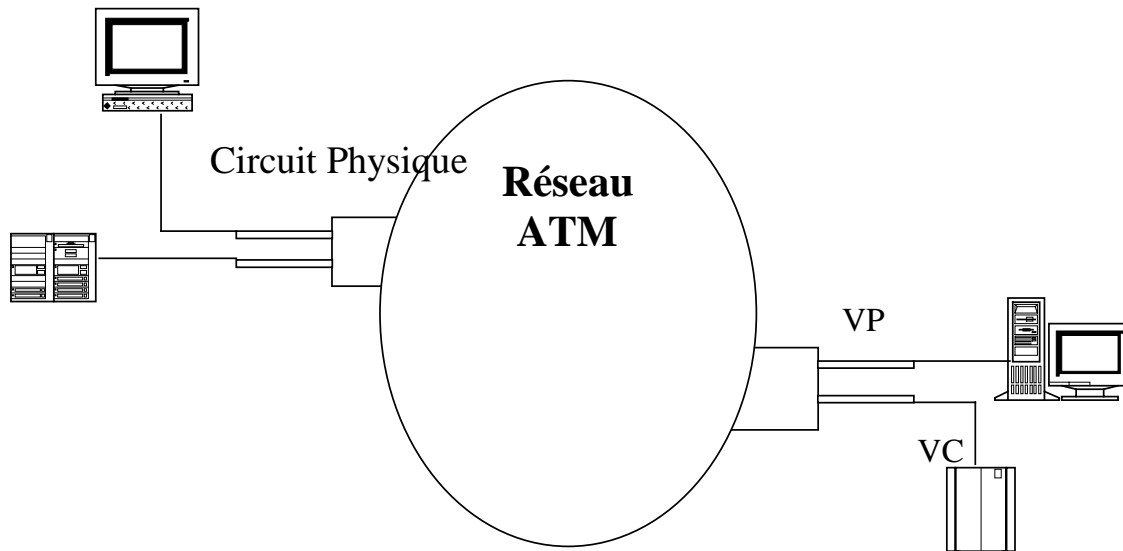
AAL 3/4 : mode flot ou message, supporte le multiplexage sur un VC (en cours d'abandon)

AAL 5 : supporte le point à point et le point à multipoints, connecté ou non, mode message ou mode flot, message de taille max en mode message de 64Ko, pas de contrôle d'erreur de bout en bout (seule AAL qui semble avoir de l'avenir)

AAL 0 : accès direct aux cellules ATM, on peut écrire sa propre AAL

Réseau ATM en mode connecté

Voie virtuelle (VC) et Chemin virtuel (VP)



Cellule ATM : 5 octets d'entête + 48 octets de données soit 53 octets

La commutation opère à partir des identificateurs de VP et de VC : brassage de VP, et commutation de VC.

Fonctions associées à la gestion de trafic :

- Contrôle d'admission,
- Surveillance du trafic utilisateur,
- Contrôle de cellules (lié au bit CLP dans la cellule),
- Cadencement du trafic (GCRA – Generic Cell Rate Algorithm) ...

Modèle de Contrat - QoS ATM

Le réseau ne garantit les contraintes de QoS que si le trafic de la source (utilisateur) respecte le contrat :

Paramètres de Trafic :

- *Peak Cell Rate (PCR)*, cells/s (*débit max*)
- *Substainable Cell Rate (SCR)*, cells/s \leq PCR (*débit moyen*)
- *Maximum Burst Size (MBS)*, cells, (*nombre max de cellules envoyées au débit PCR*)
- *Minimum Cell Rate (MCR)*, cells/s, pour service ABR(*débit minimal garanti*).

Paramètres de QoS demandée :

- *maximum Cell Transfer Delay (maxCTD)*, sec (*délai max*)
- *peak-to-peak Cell Delay Variation (peak-to-peak CDV)*, sec (*gigue max*)
- *Cell Loss Ratio (CLR)*, cells (*taux de perte de cellules max*)

services ATM

attributs	CBR	rt-VBR	nrt-VBR	UBR	ABR
paramètres de Trafic:					
PCR	√	√	√		√
SCR, MBS	n/a	√	√	n/a	n/a
MCR	n/a	n/a	n/a	n/a	√
paramètres de QoS:					
ppCDV	√	√			
maxCTD	√	√	√		
CLR	√	√	√	√	

√ = spécifié , n/a = non applicable

Réservation à l'initiative de l'émetteur.

Projection des caractéristiques de flux de messages applicatifs – paramètres (1)

- Quand tout est constant :

Si I intervalle de temps entre deux requêtes successives (secondes), le débit en requêtes par unités de temps est $1/I$

Si S est la taille d'une requête (bits), le débit soumis est S/I (bits par seconde)

PCR = $S/(I * 48^4 * 8)$; $1/PCR$ = temps séparant l'arrivée de 2 cellules

- Quand les flux sont irréguliers :

Si I_{\min} intervalle de temps min entre deux requêtes successives (secondes), le débit en requêtes par unités de temps est $1/I_{\min}$

Si S_{\max} est la taille max d'une requête (bits), le débit soumis est S_{\max}/I_{\min} (bits par seconde)

$$\mathbf{PCR} = S_{\max}/(I_{\min} * 48 * 8)$$

Pour les services à débit variable, on a besoin de S_{moy} et I_{moy} qui permet d'obtenir **SCR** = $S_{\text{moy}}/(I_{\text{moy}} * 48 * 8)$

N_{raf} Taille max d'une rafale de requêtes de taille S_{\max} sur une période d'observation T pendant laquelle on a évalué $S_{\max} \geq S_{\text{moy}} \geq S_{\min}$ et $T \geq I_{\text{moy}}$.

$$N_{\text{raf}} = \text{partie_ent}[(S_{\text{moy}} - S_{\min}) * \text{partie_ent}[T/I_{\text{moy}}] / (S_{\max} - S_{\min})] = \mathbf{MBS}$$

⁴ Dans l'absolu, il faudrait tenir compte des données de gestion ajoutées qui sont spécifiques à l'AAL utilisée.

Protocoles de Transport Multimédia

- **Cadencement,**
- **Suite Tenet,**
- **XTP,**

Canalisation du trafic

Le flot de messages peut devenir aléatoire, rafales/saccades avec des données de taille variable.

Idéalement, il faudrait émettre et transférer des données de taille identique, à un rythme uniforme. C'est particulièrement important dans le contexte du multimédia.

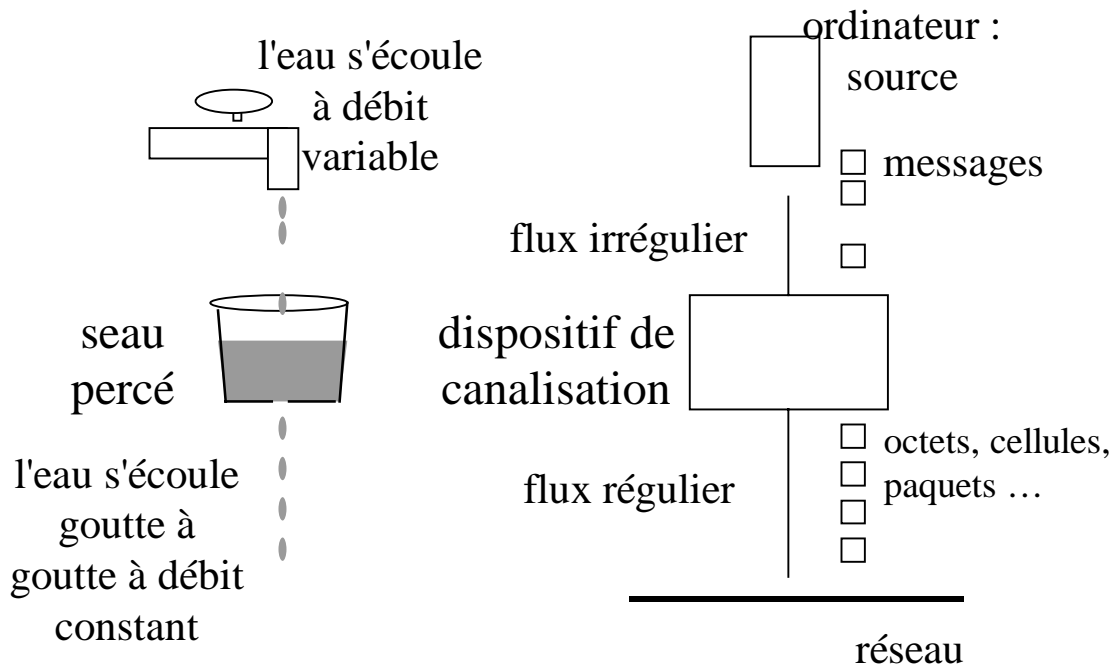
La **canalisation du trafic** consiste à réguler la vitesse et le **cadencement** des données transmises.

Canalisation du trafic (traffic shaping) utilisé pour le contrôle de congestion habituellement.

Attention : Ne pas confondre avec le contrôle de flux (fenêtre glissante) qui consiste à limiter vis à vis du fournisseur le volume de données en transit sur le réseau et chez le récepteur.

Leaky bucket

Modèle du seau percé



L'émission se fait à une cadence régulière, le dispositif de canalisation effectue un tamponnement des messages arrivant à un rythme irrégulier.

L'insertion des PDUs sur le réseau se fait périodiquement (suivant tops d'horloge).

Deux conditions : il faut un flux arrivant, et si le dispositif de canalisation est plein, le surplus est perdu (seau déborde).

Modèle du **seau percé à compte d'octets** (byte counting leaky bucket) : n octets peuvent être transmis entre deux tops d'horloge. Attention, la sortie n'est plus cadencée aussi régulièrement

Token leaky bucket

Le modèle du seau percé est assez rigide. Il faudrait un mécanisme flexible pour pouvoir augmenter le débit en sortie du dispositif de canalisation en cas d'avalanche.

Le modèle du seau percé à jetons fonctionne sur le même principe que le byte counting leaky bucket, excepté que le grain de gestion n'est plus l'octet mais le paquet. Pour chaque période, le dispositif de canalisation dispose de n paquets à transmettre au maximum. Il peut transmettre n paquets en une seule fois !

Différence avec le "leaky bucket", quand le dispositif est plein, les paquets ne sont plus détruits mais rejetés.

TENET - Pile de protocoles

TCP	UDP	RMTP	CMTP	R	R
IP		RTIP		C	T
Liaison de Données				A	C
				P	M
					P

La couche Liaison de Données est supposée capable d'offrir un service à caractère déterministe : FDDI, DQDB, ATM.

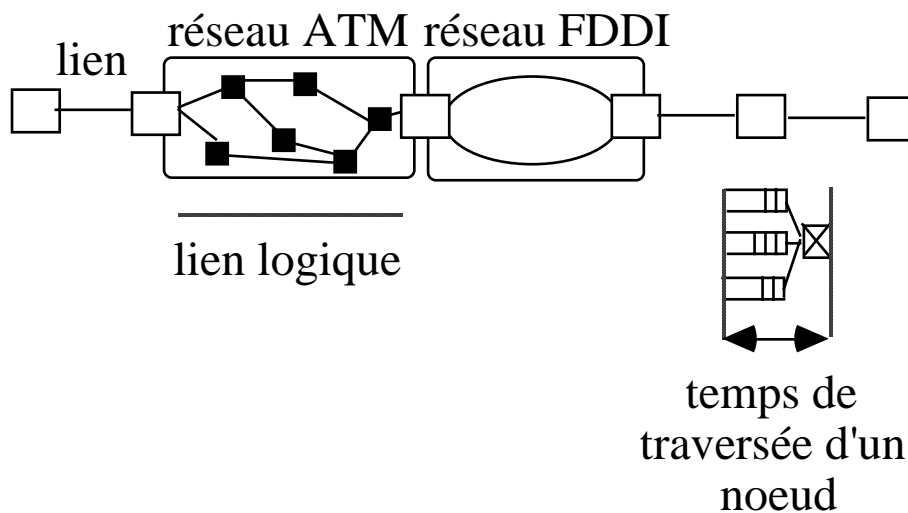
- **RTIP** : Extension de IP pour gérer la notion d'urgence des Datagrammes, et la surveillance du trafic réel ... en cas de pb, il réduit le trafic, il est orienté connexion unidirectionnelle
- **RMTP** : Real time Message Transport Protocol, orienté message
- **CMTP** : Continuous Media Transport Protocol, orienté flot d'octet
- **RCAP** : Real time Chanel Administration Protocol, Réservation de ressources
- **RTCMP** : Real Time Control Management Protocol, Gestion de Ressources

TENET - Hypothèses du modèle

Le réseau est composé d'un ensemble de sous-réseaux interconnectés par des noeuds de routage.

Les sites traversés ont des horloges synchronisées, ou resynchronisées périodiquement.

Les liens de communication entre noeuds sont déterministes. Un lien logique est : un lien physique ou un sous-réseau entier. Le lien logique est donc lui aussi déterministe.



Les files d'attente dans les noeuds intermédiaires gèrent les messages en fonction de leur priorité. Une politique Earliest Deadline First peut être adoptée pour gérer les messages en fonction de leur priorité.

Les liens logiques ont un taux d'erreur négligeable.

TENET – Modèle de Gestion de ressources - Types de Canaux

Un canal entre deux communicants est matérialisé par un chemin fixe entre ces deux entités. C'est une sorte de circuit virtuel.

Le modèle TENET suppose deux types de Canaux Temps Réel :

- Un canal TR est dit déterministe si le délai d'acheminement de bout en bout est garanti dans tous les cas.
- Un canal TR est dit statistique si le délai d'acheminement de bout en bout est assuré avec une probabilité supérieure à une valeur seuil

TENET - Spécification QoS

Paramètres de QoS spécifiés par l'utilisateur :

- Délai d'Acheminement
- Gigue
- Débit
- Fiabilité (taux d'erreur)

TENET - Délai d'acheminement

Délai d'acheminement (émission->délivrance au récepteur), la fiabilité de l'information n'est pas garantie ... une donnée peut arriver à l'heure mais être erronée, attention !

Soit $D_{c,m}$ le délai associé à un message m sur un canal c

Délai Déterministe :

$D_{c,Max}$ délai max $D_{c,Max}$ spécifié par l'utilisateur

pour tout m : $D_{c,m} \leq D_{c,Max}$

Délai Statistique :

$Z_{c,min}$ seuil de respect du délai max $D_{c,Max}$ spécifié par l'utilisateur

pour tout m : $\text{Prob}(D_{c,m} \leq D_{c,Max}) \geq Z_{c,min}$

Certaines applications de transmission d'image, de son ou d'animation tolèrent des erreurs mais nécessitent le respect des délais d'acheminement pour les messages corrects.

TENET - Gigue

Gigue, écart toléré par rapport à un temps de référence

Soit D_c un délai idéal pour l'acheminement de données

Gigue Déterministe :

$J_{c,Max}$ écart max p/r à D_c , spécifié par l'utilisateur

pour tout m : $J_{c,m} = | D_{c,m} - D_c | \leq J_{c,Max}$

Gigue Statistique :

$U_{c,min}$ seuil de respect de $J_{c,Max}$, spécifié par l'utilisateur

pour tout m : $\text{Prob}(J_{c,m} \leq J_{c,Max}) \geq U_{c,min}$

TENET - Débit

Le Débit est contraint par la charge du service de communication, et le taux d'erreur de transmission⁵

soit TH_c le débit courant du canal de communication c

Débit Déterministe :

soit $TH_{c,min}$ le débit min spécifié par l'utilisateur

pour tout c : $TH_c \geq TH_{c,min}$

Débit Probabiliste :

$V_{c,min}$ seuil de respect de TH_{min} spécifié par l'utilisateur

pour tout c : $\text{Prob}(TH_c \geq TH_{c,min}) \geq V_{c,min}$

⁵ Les erreurs de transmission impliquent des retransmissions qui pénalisent le réseau

TENET - Fiabilité

Fiabilité, elle concerne le taux d'erreur admis, certaines applications ont des contraintes d'échéance et de fiabilité absolue : embarqué militaire par ex.

$W_{c,min}$ seuil de succès min souhaité par l'utilisateur

$$\text{Prob}(\text{message délivré correct}) \geq W_{c,min}$$

Dans la fiabilité on ne comptabilise pas les messages qui sont hors délai et hors gigue (pourtant éliminés), ils sont comptés ailleurs.

TENET - paramètres de calcul pour la réservation de ressources

Sur un intervalle d'observation I :

- $X_{\min,c}$ délai d'inter-arrivée min des messages sur le canal c (taux d'arrivée $1/X_{\min,c}$ correspond au Lambda des files d'attentes)
- $X_{\text{ave},c}$ délai moyen d'inter-arrivée des messages sur le canal c
- $t_{c,n}$ temps maximum pour servir un message au noeud n sur le canal c (taux $1/t_{c,n}$ correspond au Mu des files d'attentes)
- $d_{c,n}$ délai limite pour traverser le noeud n sur le canal c (correspond à une échéance globale)
- $d_{c,n}^m$ délai de traversée du noeud n sur le canal c par le message m (correspond à une échéance message)
- $S_{\max,c}$ taille max d'un message sur le canal c

TENET - Tests des Ressources (1)

Un canal ne peut exister que s'il y a assez de ressources, ceci est fait à l'aide de tests au moment de l'établissement du canal, à la traversée de chaque noeud.

Tests Intermédiaires (à l'établissement ou en mode établi) :

Puissance CPU pour un Canal Déterministe :

pour un canal c , le taux d'utilisation de la CPU d'un noeud est $t_{c,n}/X_{\min,c}$, (règle classique du Lambda/Mu en files d'attentes), ce qui doit être vrai pour chaque noeud d'où la contrainte à vérifier

$$n (t_{c,n}/X_{\min,c}) < 1$$

autres Tests :

Puissance CPU pour un Canal Statistique

Test de Délai Limite

Test d'allocation de tampons pour un Canal Déterministe

Test d'allocation de tampons pour un Canal Statistique

Test de Contrôle de Gigue

TENET - Tests des Ressources (2)

Tests Destinataires :

Test D pour un canal quelconque :

$$D_{c,Max} \geq \sum_{n=1}^N d_{c,n}^m$$

pour $m =$ demande d'ouverture lors de la réservation de ressources à la création d'un canal

On calcule le $d_{c,n}$ de chaque noeud par :

$$d_{c,n} = 1/N * \left[D_{c,Max} - \sum_{i=1}^N d_{c,i}^m \right] + d_{c,n}^m$$

où $N =$ tous les noeuds traversés sauf le noeud courant de numéro n (ici le destinataire pour ce test)

Test Z pour un canal statistique : Calcul de la probabilité de dépassement du délai d'acheminement $D_{c,Max}$ ou D_c

TENET - RCAP

Service de Réserveation de ressources ou de Contrôle d'Admission.

Chaque site dispose d'un daemon RCAP, le demandeur fournit la suite des noeuds intermédiaires que doit traverser le canal temps réel à établir.

Cette réserveation de ressources peut emprunter des connexions TCP entre daemon RCAP.

La réserveation de ressources faite par RCAP sert pour RTIP

Format d'un message de réserveation de ressources :



HR : Header Record

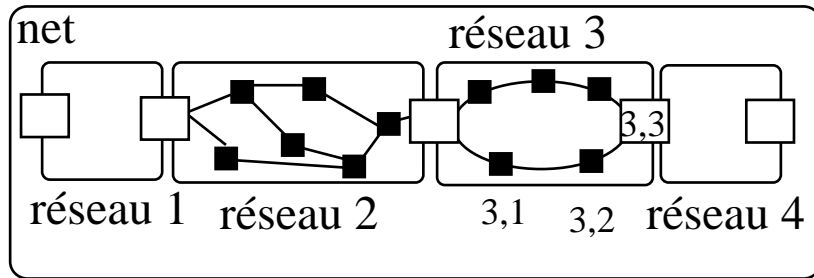
NSR : Network Subheader Record

ER : Establishment Record

NSR -> début d'une description de ressources pour un sous-réseau

ER -> paramètres de chaque noeud du sous réseau qui représentent les ressources locales affectées au canal, les ressources sont réservées si elles sont disponibles au passage du message

TENET - Exemple de réservation



Message "establish_request" :

réseau 3, noeud 3,2 (ajoute $ER_{3,2}$):

HR	NSR	ER	ER	NSR	ER	ER
	net	réseau 1	réseau 2	réseau 3	3,1	3,2

réseau 3, noeud 3,3 (ajoute $ER_{3,3}$):

HR	NSR	ER	ER	NSR	ER	ER	ER
	net	réseau 1	réseau 2	réseau 3	3,1	3,2	3,3

$ER_{réseau 1}$ et $ER_{réseau 2}$ récapitulent les ressources déjà utilisées pour le canal temps réel dans les réseaux 1 et 2.

réseau 3, noeud 3,3 - fin du réseau 3 (synthèse) :

HR	NSR	ER	ER	ER
	net	réseau 1	réseau 2	réseau 3

réseau 3, noeud 3,3 - début du réseau 4 :

HR	NSR	ER	ER	ER	NSR
	net	réseau 1	réseau 2	réseau 3	réseau 4

TENET - PDU RCAP

établissement par **establish_request** : **réponse positive** par **establish_accept** les paramètres liés aux tests D et Z sont connus

réponse négative par **establish_denied** qui provoque la libération des ressources réservées

fermeture par **close_request** (initiateur), propagation par **close_request_forward** (source) ou **close_request_reverse** (destinataire)

interrogation sur l'état des ressources de tous les noeuds d'un canal temps réel par **status_request** et réponse par **status_report**

TENET - CMTP

Transfert de Données temps réel isochrone

Trafic de type continu : vidéoconférence, son haute fidélité... ex : une caméra transmet vers une carte vidéo sur un PC

Deux types de transferts :

- flot de messages : l'utilisateur produit un message de longueur variable pdt un intervalle de longueur fixe (qui donne la période)
- flot d'octets : l'utilisateur spécifie la quantité d'octets émise par période sous la forme d'un min et d'un max

Les données peuvent contenir des informations de synchronisation.

TENET - RMTP

Transfert de Données temps réel asynchrone

Traffic de type sporadique

Ca semble être un TCP allégé : sans contrôle de flux et sans contrôle d'erreur

XTP – Xpress Transport Protocol

Hypothèses de conception : taux d'erreur sur les liens de transmission faible, intégration en une seule couche des couches réseau et transport, intégrable dans du silicium, destiné aux communications temps réel.

Types de services supportés :

- Connexion
- Transaction
- Datagramme
- Datagramme avec acquit
- Flot isochrone
- Transfert en rafale

Communications en groupe de diffusion supportées.

Toutefois, pas de protocole de gestion de ressources, ni de QoS.

Autres travaux :

- HeiTS : Heidelberg Transport System, destiné au multimédia, conçu avec IBM
- METS : Multimedia Enhanced Transport Service, destiné au Multimédia, conçu par l'université de Lancaster, fonctionne au-dessus d'ATM, et gère une QoS statistique.

QoS et Internet

La gestion de la QoS temporelle dans les protocoles Internet classiques tient d'une politique "au mieux" ou "best effort".

- Rappels
 - IP
 - Multicast
 - UDP
 - TCP

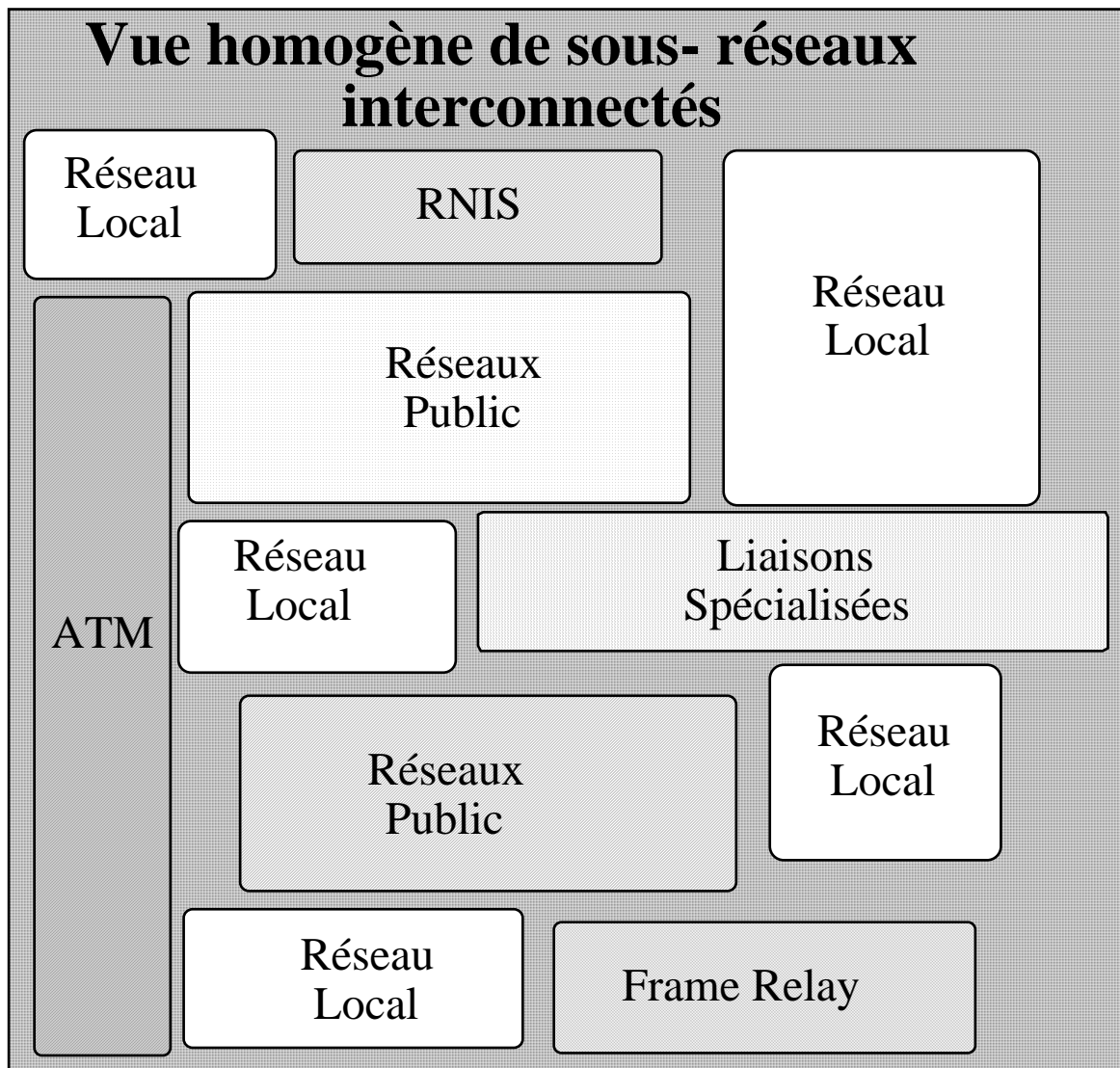
- Approche QoS IntServ

IP

IP : Fédération et Interconnexion de liaisons de données (sous-réseaux)

Hétérogénéité : Fournisseurs de services d'Interconnexion, Diamètre des liaisons de données, Modes d'Adressage

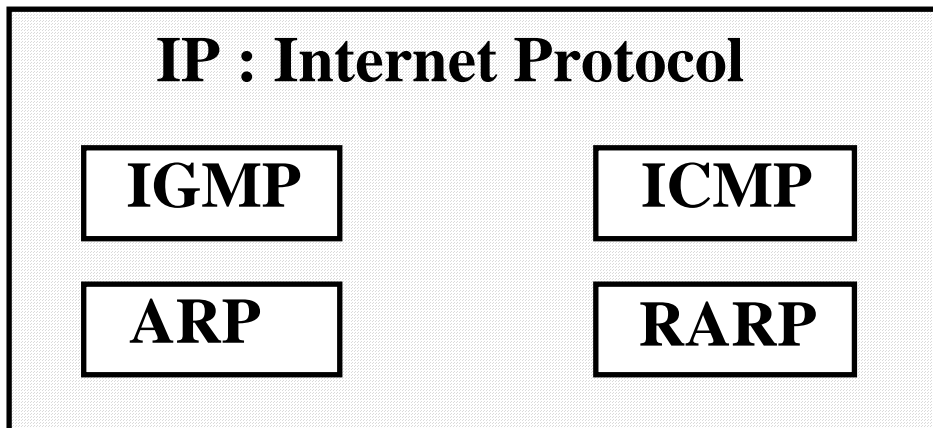
Un seul Réseau de transfert !



- **Homogénéisation** : Adressage et Adaptation de la transmission à la liaison traversée
- **Routage** : intra-domaine et inter-domaine
- **Contrôle de congestion** : Gestion des Ressources de l'ensemble du réseau fédérateur

Architecture d'IP

IP V4 :



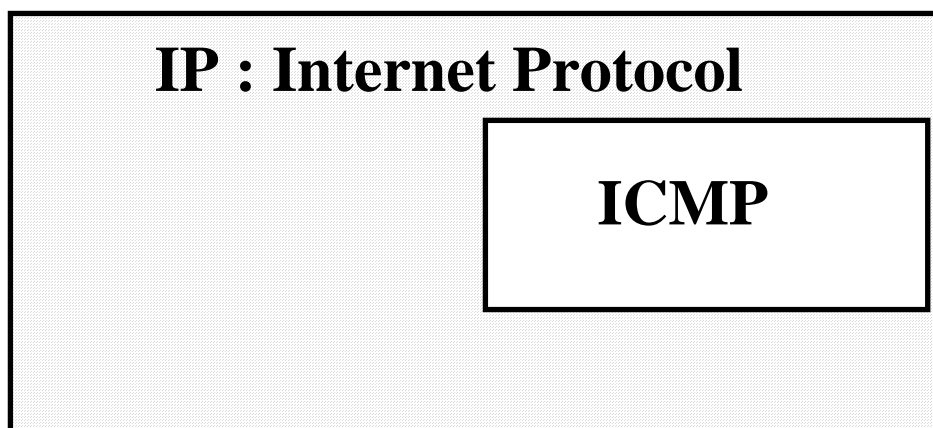
ICMP : Internet Control Message Protocol

ARP : Address Resolution Protocol

RARP : Reverse Address Resolution Protocol

IGMP : Internet Group Management Protocol (Multicast IP)

IP V6 :



L' Internet Protocol (IP)

* Communications dans le mode minimal : **DATAGRAM** (mode non connecté, paquets **non Acquittés**)

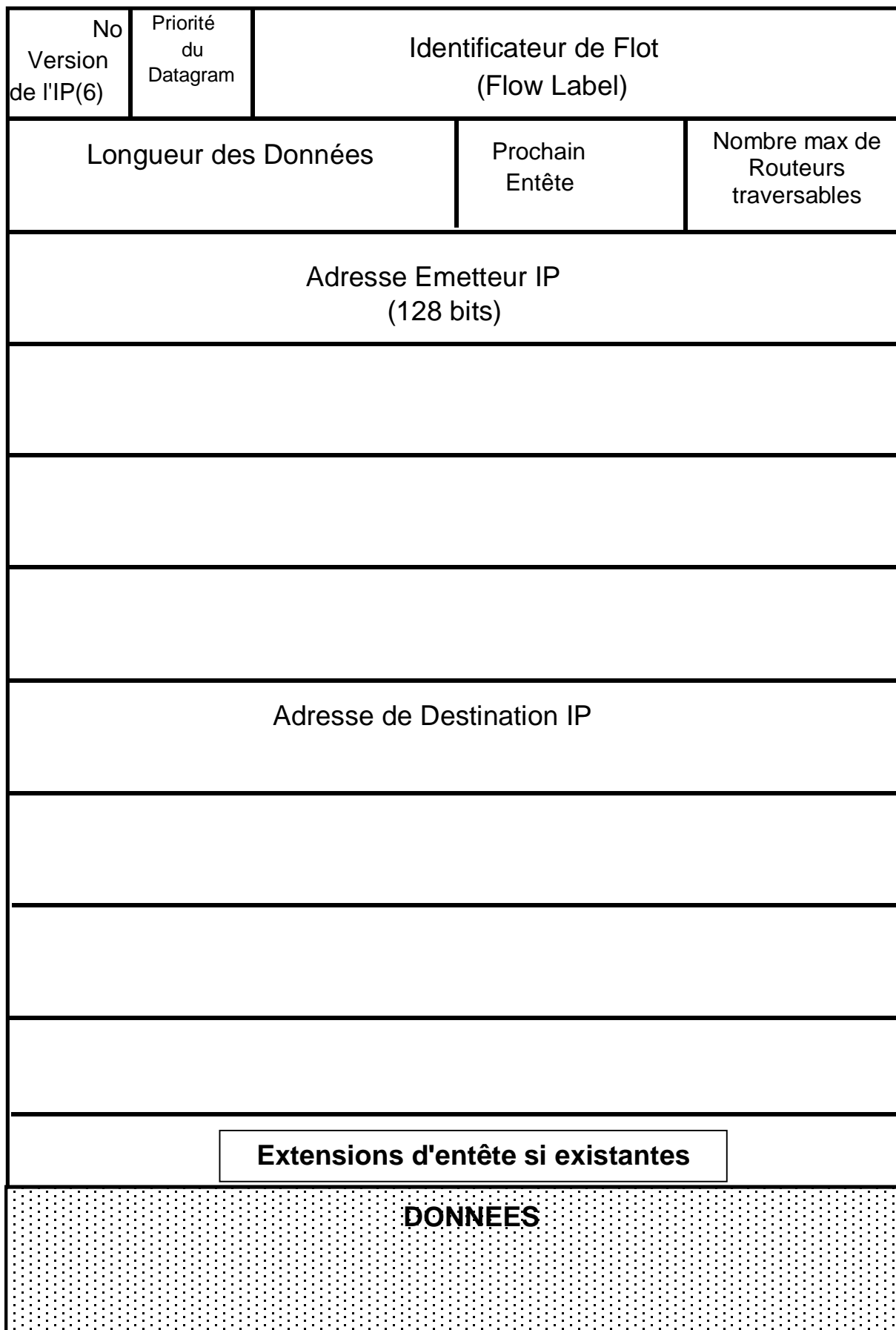
=> la détection des messages erronés ou perdus et leur réémission sont à la charge de l'émetteur des messages (couche Transport).

- Adressage Internet et Routage entre Réseaux
- Conversions d'Adresses (@IP<->08:00:20:06:4b:8e) et adaptation à la liaison traversée
- Fragmentation/Réassemblage, Adaptation de la taille des messages soumis par la couche Transport suivant les possibilités offertes par la couche Liaison.
- Encapsulation/Désencapsulation par rapport à la couche Transport

0	4	8	16	19	24	31
No Version de l'IP(4)	Longueur de l'entête (nb de mots de 32 bits)	Façon dont doit être géré le datagram TOS - type of service	Longueur du Datagram, entête comprise (nb d'octets)			
No Id -> unique pour tous les fragments d'un même Datagram			flags (2bits): .fragmenté .dernier	Offset du fragment p/r au Datagram Original (unit en nb de blk de 8 o)		
Temps restant à séjourner dans l'Internet TTL	Protocole de Niveau Supérieur qui utilise IP		Contrôle d'erreurs sur l'entête			
Adresse Emetteur IP						
Adresse de Destination IP						
Options : pour tests ou debug					Padding: Octets à 0 pour que l'entête *32 bits	
DONNEES						

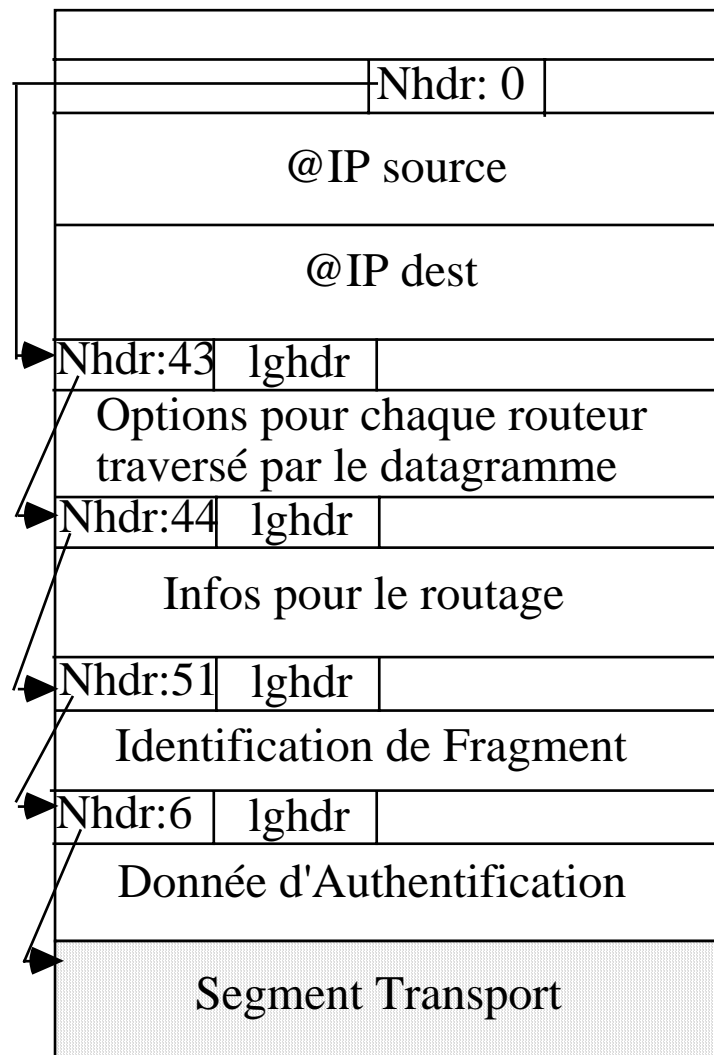
TTL : Time To Live, est exprimé en nombre de machines restant à traverser, décrétementé de 1 par chaque routeur franchi

0 4 8 16 24 31



Utilisation du "Prochain Entête" en IPV6

Il est conseillé d'organiser les entêtes d'une certaine façon qui traduit l'ordre des traitements faits par un routeur lors de la commutation d'un datagramme :



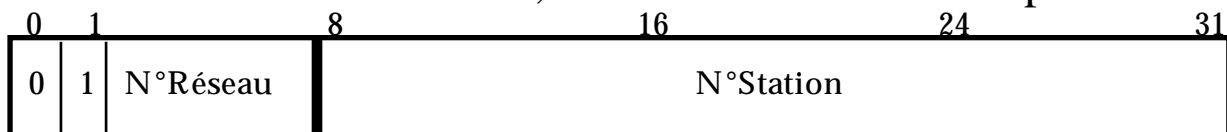
Le champ Nhdr contient le numéro d'extension d'entête du prochain champ (0-options de gestion du TTL, 43-routage ...), la dernière extension contient le numéro du protocole transporté dans le champ Nhdr (même fonction que le champ "Protocole de niveau Supérieur du data gramme IPV4).

Adressage Internet IP V4 (32 bits)

Adresses Uniques Universelles :

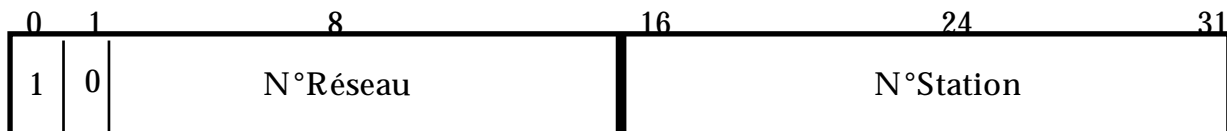
A . B . C . D
(N°Réseau, N°station)

Classe A : Peu de Réseaux, de nombreuses Stations par Réseau



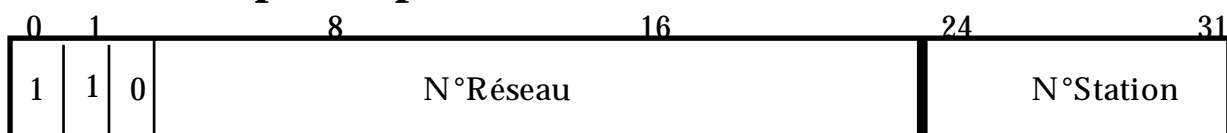
N°de Réseau : 1-126, **127** adresse de **rebouclage en local**

Classe B :



N°de Réseau : 128.1 - 191.254

Classe C : Beaucoup de Réseaux, Peu de Stations par Réseau
La classe la plus répandue

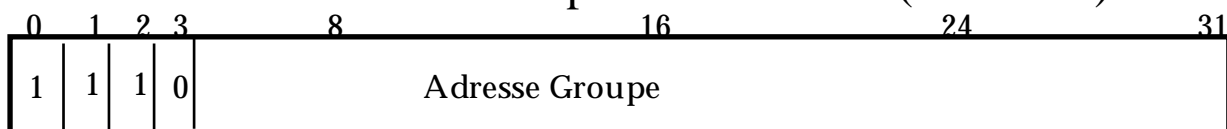


N°de Réseau : 192.0.1 - 223.255.254

N°de Station : 1 - 254

Broadcast : 255 dans le champ **N° de Station**

Classe D : Adresses de Groupes de Diffusion (Multicast)



N°de Réseau : 225.0.0.0 - 239.255.255.255 (224.0.0.x réservée pour les protocoles de routage)

Adresses IP v6-IPng(128 bits)

Les nouvelles adresses sont sur 128 bits, la notation est donnée par groupe de 16 bits :

0108:0000:0000:0000:0008:0800:200C:417A

qui peut être simplifiée en :

0108:0:0:0:8:0800:200C:417A

ou encore en :

0108::8:800:200C:417A

(pas plus d'un seul "::" dans une adresse)

Exemples :

adresse locale à un sous-réseau (ne sort pas) :

8 bits	n bits	m bits	p bits
11111110	0	n° ss-réseau	n° station

adresse dépendante d'un fournisseur :

3 bits	n bits	m bits	p bits	125-(n+m+p) bits
010	n° fourn	n° adhérent	n° ss-réseau	n° station

convergence IPv4 - IPng :

80 bits	16 bits	32 bits
000.....000		adresse IP v4

Un exemple de Ping en IPV6 :

```
ping6 5F0D:E900:80DF:E000:0001:0060:3E0B:3010
```

```
PING 5F0D:E900:80DF:E000:0001:0060:3E0B:3010: 56 data bytes
```

```
64 bytes from 5f0d:e900:80df:e000:1:60:3e0b:3010: icmp_seq=0 time=43.1 ms
```

```
64 bytes from 5f0d:e900:80df:e000:1:60:3e0b:3010: icmp_seq=1 time=40.0 ms
```

```
64 bytes from 5f0d:e900:80df:e000:1:60:3e0b:3010: icmp_seq=2 time=44.2 ms
```

```
64 bytes from 5f0d:e900:80df:e000:1:60:3e0b:3010: icmp_seq=3 time=43.7 ms
```

```
64 bytes from 5f0d:e900:80df:e000:1:60:3e0b:3010: icmp_seq=4 time=38.9 ms
```

```
64 bytes from 5f0d:e900:80df:e000:1:60:3e0b:3010: icmp_seq=5 time=41.2 ms
```

```
64 bytes from 5f0d:e900:80df:e000:1:60:3e0b:3010: icmp_seq=6 time=39.1 ms
```

```
64 bytes from 5f0d:e900:80df:e000:1:60:3e0b:3010: icmp_seq=7 time=42.1 ms
```

```
- --- 5F0D:E900:80DF:E000:0001:0060:3E0B:3010 ping statistics ---
```

```
9 packets transmitted, 9 packets received, 0% packet loss
```

```
round-trip min/avg/max = 38.9/41.3/44.2 ms
```

source :

Stéphane Bortzmeyer à l'Institut Pasteur

Gestion du Datagramme - TOS (1)

Ce champ n'est pas géré par tous les algorithmes de routage (seulement OSPF et RIP V2).

0	1	2	3	4	5	6	7
Priorité		Type de Service					
		D	T	R	C		

La priorité influe sur la gestion des files d'attente des datagrammes vers une liaison de données. Celui qui a la préséance la plus élevée est transmis en premier :

000 : normal

001 : prioritaire

010 : immédiat

011 : urgent

La priorité de 0 à 7 permet de marquer l'importance du datagramme.

Gestion du datagramme (2)

Le champ "Type de service" est lié à la métrique utilisée par le routage :

- bit **T** : Débit (Througput/Bandwith) -> demande le plus grand débit
- bit **R** : Fiabilité (Reliability/Error Rate) -> demande le plus faible taux d'erreur
- bit **C** : Coût minimal (Cost) -> demande un coût miniamal
- bit **D** : Délais courts (Delay) -> demande le plus court délai (évite les satellites)

Combinaison de bits possible. Ce champ n'est interprété que par certains les routeurs qui ont des algorithmes de routage de nouvelle génération.

Gestion du datagramme (3)

En IPV6, on utilise la priorité du datagramme, on a deux types de trafics, donc deux types de datagrammes:

- La source, d'un flot de données supporté par des datagrammes sujets au contrôle de congestion (0-7), peut diminuer son débit si elle est avertie d'une congestion.
- La source d'un flot de données supporté par des datagrammes non sujets au contrôle de congestion (8-15) ne peut diminuer son débit si elle est avertie d'une congestion (conférences audio ou vidéo par exemple).
- Pas de priorité relative d'un type de trafic sur l'autre.
- Plus la priorité est élevée, plus le datagramme a de chances d'être conservé en cas de congestion

Exemple d'utilisation de la priorité par un routeur dans le cas d'un trafic temps réel audio:

Certaines techniques de numérisation du son tolèrent les pertes de messages pourvues qu'elles ne soient pas successives.

L'émetteur peut alors marquer certains datagrammes avec priority 8 et d'autres avec 9.

Ainsi, un routeur pourra éliminer les datagrammes marqués 8 de préférence aux autres en cas de congestion.

Identification de Flux de données ou de Canal

Un flot d'information entre deux entités est marqué/identifié par le champ "Flow label" et par l'adresse IP de la source.

Ce champ peut servir, au niveau d'un routeur, à optimiser des traitements : un émetteur pour un canal particulier indique toujours les mêmes options et ses datagrammes nécessitent toujours les mêmes traitements.

La marque peut servir de clef dans une table de routage.

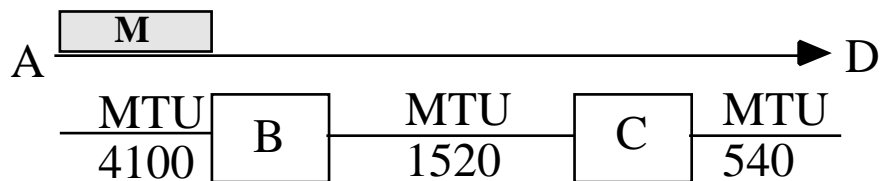
Ce champ peut être utilisé pour RSVP et RTP vus plus loin.

IPV4 : Fragmentation/Réassemblage des datagrammes

Il y a fragmentation quand un segment (unité de données de la couche Transport Internet) traverse des liaisons dont les sections sont plus petites.

La taille des données sur la liaison (MTU) est variable⁶ : 1500o (Ethernet), 1492o (IEEE802.3), 4464o(Token-Ring 4Mb/s), 17914o (Token-Ring 16Mb/s), 4352o(FDDI), 576o(X25), 296o (PPP) ...

20o Entête IP + 4000o Données



Un fragment a une taille multiple de 8 octets sauf le dernier.

Envoi par A d'un datagramme de 4020o

A->B: M Lg= 4020, DF=0, MF=0, position=0

Fragmentation sur B (1 datagramme = plusieurs paquets)

B->C: f1 Lg= 1520, DF=0, MF=1, position=0 (paquet 1)
 f2 Lg= 1520, DF=0, MF=1, position=1500 (paquet 2)
 f3 Lg= 1020, DF=0, MF=0, position=3000 (paquet 3)

puis Fragmentation sur C

C->D: f11 Lg= 520, DF=0, MF=1, position=0 (paquet 1)
 f12 Lg= 520, DF=0, MF=1, position=500 (paquet 2)
 f13 Lg= 520, DF=0, MF=1, position=1000 (paquet 3)
 f21 Lg= 520, DF=0, MF=1, position=1500 (paquet 4)
 f22 Lg= 520, DF=0, MF=1, position=2000 (paquet 5)
 f23 Lg= 520, DF=0, MF=1, position=2500 (paquet 6)
 f31 Lg= 520, DF=0, MF=1, position=3000 (paquet 7)
 f32 Lg= 520, DF=0, MF=0, position=3500 (paquet 8)

Assemblage sur le récepteur D. Attention, la fragmentation est pénalisante, elle induit du retard dans la traversée des routeurs, on préfère aujourd'hui se caler sur le plus petit MTU du chemin de données.

Fragmentation à la source en IPV6, la spécification des fragments utilise un entête "Fragment" (44) qui permet de reproduire ce qu'on avait en IPV4.

⁶ Source R. Stevens dans TCP/IP Illustrated V1

Encapsulation IP

Le champ "protocole de niveau supérieur" (8 bits) dans l'entête IPV4 indique à quel protocole est destiné le datagramme.

à titre indicatif :

- 1 : ICMP
- 2 : IGMP
- 4 : IP dans IP (encapsulation)
- 6 : TCP (Transmission Control Protocol)
- 8 : EGP (Exterior Gateway [=routeur] Protocol)
- 17 : UDP (User Datagram Protocol)
- 89 : OSPF

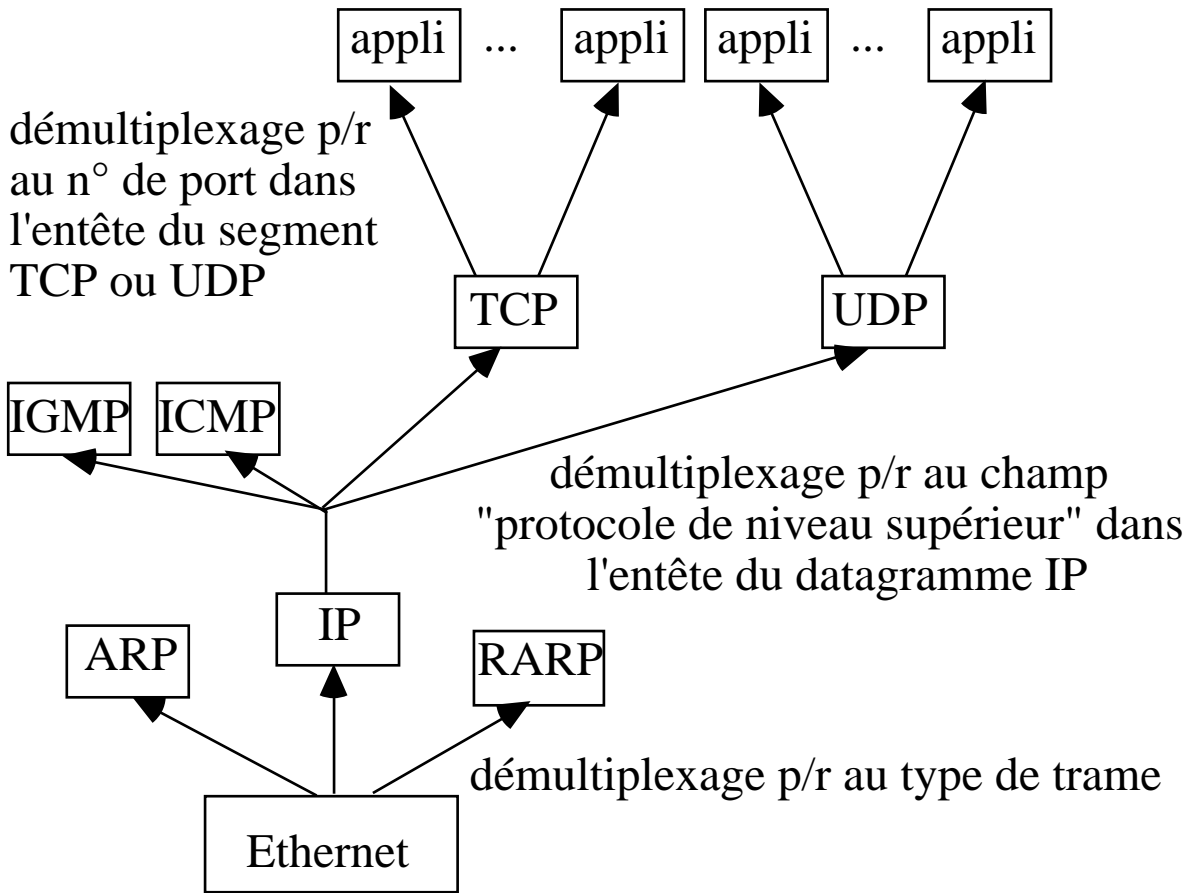
En IPV6, c'est le champ NextHeader qui joue ce rôle (celui de la dernière extension ou de l'entête standard si pas d'extension):

- 4 : IP dans IP (encapsulation)
- 6 : TCP (Transmission Control Protocol)
- 17 : UDP (User Datagram Protocol)
- 46 : Resource Reservation Protocol
- 58 : ICMP
- 59 : No Next Header

Son rôle est en fait beaucoup plus important:

- 43 : Routing Header
- 44 : Fragment Header

De la trame à l'utilisateur en LAN



Implantation de IP V4

La couche IP n'examine pas le datagramme reçu champ par champ. Ca ne gênerait pas pour une station de travail, mais pour un routeur, ça serait inefficace.

L'implantation est optimisée pour les traitements les plus fréquents, en particulier, pour les datagrammes sans options.

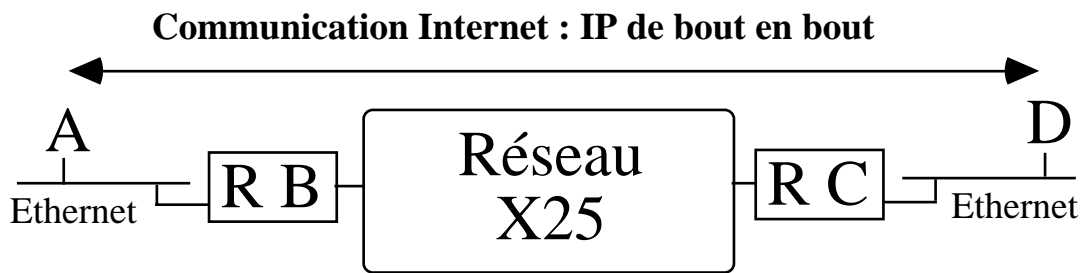
Certains routeurs peuvent atteindre un taux de commutation de 2 Gb/s ...(en 97)

C'est pour cela que l'utilisation des options tombe en désuétude.

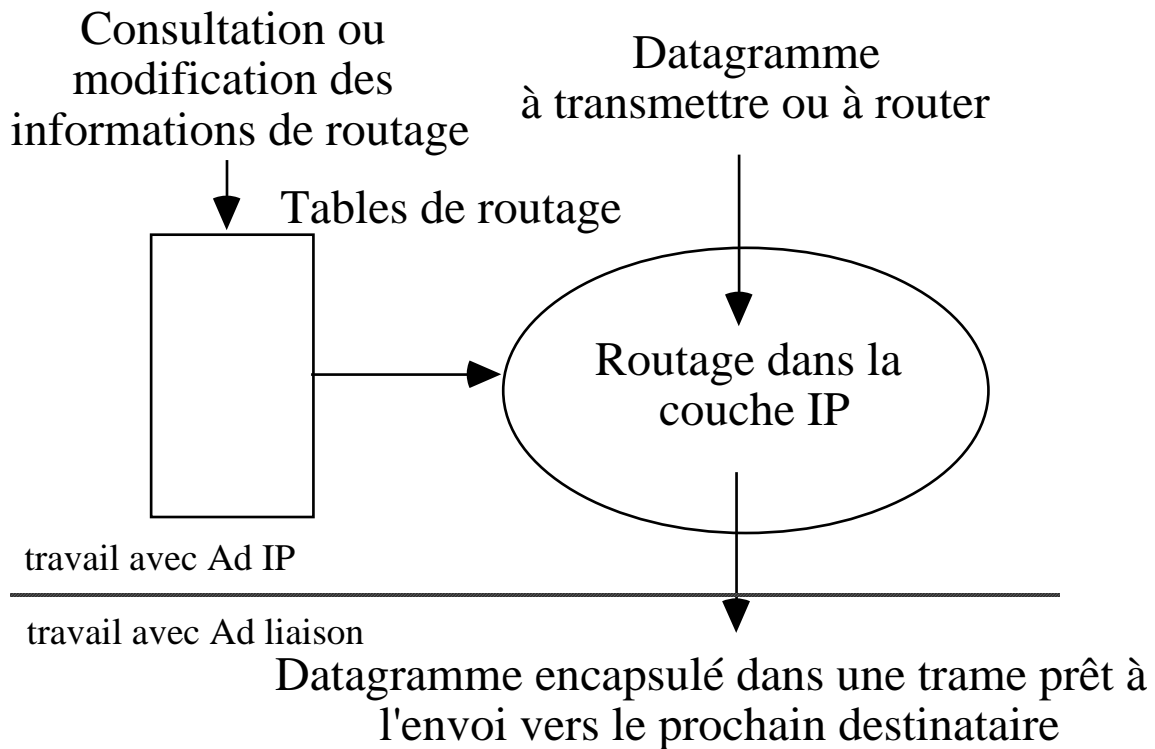
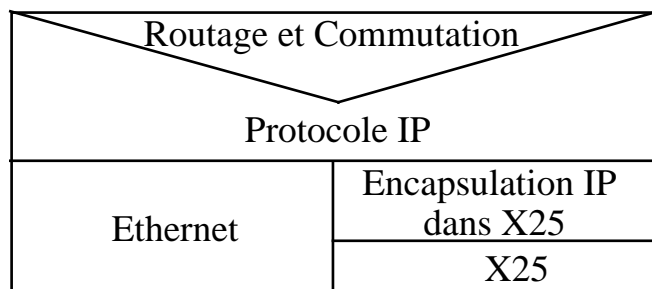
Pour faire du routage depuis la source, il faut nécessairement utiliser le champ option ...

IPV6 est affiné pour satisfaire l'objectif de performance! Et pourtant il utilise abondamment les extensions d'entêtes !!!

Exemple de Routeur



Architecture du routeur B



Actuellement, X25 serait plutôt encapsulé dans IP.

Multicast IP

IGMP- Internet Group Management Protocol

plutôt que "Management", on aurait envie de "Membership"

Applications :

- . visio/video conférence sur l'Internet.
- . mode "push" sur le Web
- . gestion du routage
- . synchronisation d'horloge (avec NTP)

Efficacité multicast p/r au broadcast : seuls les membres du groupe sont atteints -> économie de bande passante

NB : Pour limiter la portée des datagrammes multicast, on joue sur le TTL, par exemple un TTL de 1 ne fait pas dépasser le premier routeur d'un réseau.

IGMPv1 : protocole de gestion de groupe basé sur deux types de requête : Interrogation-Rapport

IGMPv2 : plus efficace que v1 pour la gestion d'appartenance à un groupe, incorpore un protocole de routage (DVRMP), remplace de fait IGMPv1 (mais compatibilité conservée)

IGMPv3 en cours de conception

Protocole entièrement repris et intégré dans IPv6

Adresses Classe D

adresses unicast -> adresses un seul site (classes A,B,C)

adresses broadcast -> tous les sites d'un réseau (adresse unicast avec le champ station tout à 1)

adresses multicast -> un groupe de stations, adresses de classe D

dans la classe D:

- . les adresses 224.0.0.x et 224.0.1.x sont réservées pour des protocoles de routage ou autres protocoles de service
- . les adresses 239.0.0 à 239.255.255.255 servent pour des adresses multicast privées, elles sont non routables à l'extérieur des entreprises

multicast sur réseau local (avec adresses MAC des LANs 802.x):

On utilise les adresses multicast des LANs.

On extrait 23 bits de l'adresse IP classe D (sur les 32 - 4, soit 28 disponibles) qui correspondent à la fin de l'adresse

```

                11 1111 1111 2222 2222 2233
0123  4567 8  901 2345 6789 0123 4567 8901
1110 [^^^^ ^][^^^  ^^^^  ^^^^  ^^^^  ^^^^  ^^^^]
      < 5b >  <                23b                >

```

On ajoute ces 23 bits à la fin d'une adresse MAC préfixée par 0x01-00-5E, l'adresse (OUI pour Organization Unit Identifier) attribuée par l'IEEE avec le bit multicast à 1, le bit adresse universelle à 0, suivi d'un bit à 0 :

```

[01-00-5E] 0[^^^  ^^^^  ^^^^  ^^^^  ^^^^  ^^^^]
<                48b                >

```

Rappels sur les adresses Ethernet et leur représentation :

représentation dans le standard :

ML[OUI 22b][24b alloués par constructeur]
ordre des bits tels qu'ils sont émis sur le support

M = 1 adresse de diffusion

L = 0 adresse universelle (OUI attribuée IEEE)

OUI = partie telle que 2 constructeurs ont des id différents

reste = identification d'un coupleur tel que 2 coupleurs d'un même constructeur n'ont pas le même id

représentation dans les documents de l'Internet :

	8	7	6	5	4	3	2	1
octet 1			0				L	M
octet 2				U				
octet 3						I		
octet 4								
octet 5								
octet 6								

Entête des trames Multicast IETF (OUI 0x00-00-5E)

01-00-5E	8	7	6	5	4	3	2	1
octet 1	0	0	0	0	0	0	0	1
octet 2	0	0	0	0	0	0	0	0
octet 3	0	1	0	1	1	1	1	0
octet 4	0							
octet 5								
octet 6								

Exemple :

@IP : 224.128.64.32 0xE0.80.40.20

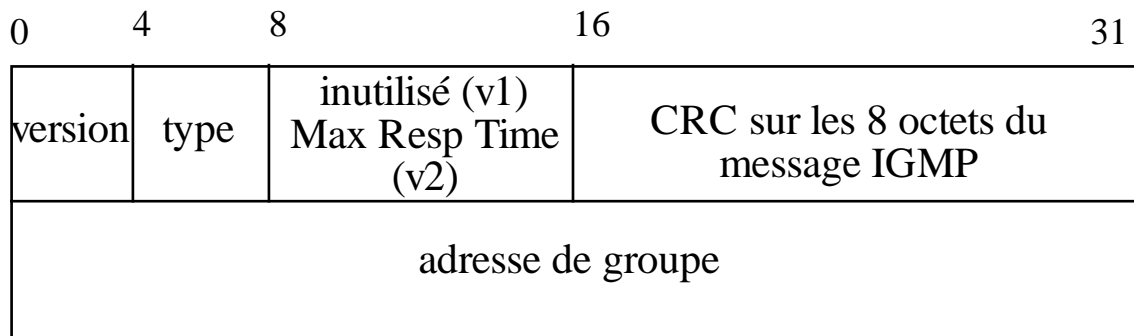
Le deuxième octet : 0x80 s'écrit 1000 0000 en binaire, on ne prend que les 5 bits les plus à gauche pour l'adresse multicast Ethernet

ce qui donne

#Eth : 01:00:5E:00:40:20

Eléments de protocole IGMPv2 (1)

Un message IGMP est encapsulé dans un datagramme IP avec pour champ protocole de niveau supérieur valant 2



Types:

Interrogation sur l'appartenance au groupe/Membership Query (valeur 1)

Appartenance au groupe/Membership Report (valeur 2 pour v1 ou 6 pour v2):

Quitte le groupe/Leave Group(extension IGMP V2)

Eléments de protocole IGMPv2 (2)

- . **Election du gestionnaire de groupes local** : Routeur multicast qui a la plus petite adresse IP d'un LAN, Election par écoute et auto-élimination, il maintient la liste des groupes actifs sur le LAN
- . **L'entrée dans un groupe** est signalée par un Report sur l'adresse de groupe, dès que le routeur associé a enregistré l'arrivée, il reçoit le trafic associé. La latence d'entrée représente le temps entre le premier Report effectué concernant un groupe et le premier Query du gestionnaire de groupe, il représente le temps d'attachement à l'arbre de routage associé au groupe définit par l'adresse de classe D.
- . **Validation abonnements** : Périodiquement (60s), le gestionnaire de groupe local interroge les sites (ou plutôt le LAN classe D 224.0.1 avec TTL de 1) par un multicast de type Query, les stations répondent par Report, une seule station du groupe suffit à maintenir l'abonnement (multicast de la réponse donc entendu par les autres membres du LAN), si pas de réponse, le routeur ne propage plus les informations sur ce groupe

Définitions Générales pour le routage multicast

Routage Multicast : gestion du routage de datagramme d'une source vers des destinataires

Arbre de Distribution : Ensemble des routeurs et sous-réseaux qui permettent aux membres d'un groupe de recevoir les informations d'une source. Il y a un arbre de distribution par source.

Chemin de Montée : Meilleur chemin d'un routeur vers la source d'un arbre de distribution

Chemin de Descente : Tout chemin qui n'est pas un chemin de montée, et qui mène aux destinataires accessibles depuis le routeur considéré

Propagation de Multicast : Action de recopier un datagramme diffusé d'un chemin de montée vers un chemin de descente

Approche "Broadcast-and-Prune"

Objectif élaguer les branches de l'arbre qui supportent un trafic sans destinataires actifs, pour cela on se sert des diffusions pour observer l'état du groupe.

Aux extrêmités, les bases de données IGMP sur les routeurs gestionnaires de groupes donnent les groupes actifs et inactifs. Quand un groupe est devenu inactif, le routeur envoie un message sur le chemin en montée qui demande de le faire disparaître de l'arbre de distribution.

Technique qui consomme de la bande passante puisqu'il faut transmettre pour gérer l'arbre de distribution. Elle est adaptée aux petits réseaux.

Arbres partagés

Un centre qui n'est pas la source du flot de données distribue le trafic vers les destinataires.

Un centre gère plusieurs groupes, et les données de tous ces groupes sont distribuées suivant le même arbre indépendamment des sources. La source ne fait pas nécessairement partie du groupe.

Cette technique nécessite qu'un destinataire indique explicitement qu'il rejoint un groupe.

Technique plus extensible que les arbres de distribution classiques depuis une source. Par contre le centre reste un goulot d'étranglement.

Caractéristiques Générales du routage multicast

Techniques de Routage :

- . Core-Based Trees (CBT)
- . Distance-Vector Multicast Routing Protocol (DVRMP) utilisé pour le M-Bone
- . Multicast extension to OSPF (M-OSPF)
- . Protocol Independent Multicast - Dense Mode (PIM-DM)
- . Protocol Independent Multicast - Sparse Mode (PIM-SM)

Multicast Internet

l'Internet Multicast est mis en oeuvre à travers le M-Bone

Il y a 3 types d'adresses :

- . Les adresses permanentes,
- . Les adresses privées,
- . Il y a des adresses pour des groupes créés de façon transitoire qui sont donc allouées dynamiquement en utilisant le protocole SAP (Session Advertising Protocol) du M-Bone.

Cartes et informations diverses :

en Europe : <http://www.dante.net/mbone/>

en France : <http://www.urec.cnrs.fr/fmbone/>

dans le monde : <http://www.mbone.com/>

UDP/TCP

Relations Applications/Transport

- UDP :
 - Client/Serveur en LAN
 - Multimedia en LAN /WAN
 - Multicast
 - TFTP, RTP, NFS, OSPF, RIP, SNMP, VoIP
 - ...

- TCP :
 - Transfert de données fiable (fichiers, terminal virtuel ...)
 - Client/Serveur en WAN
 - Unicast
 - DNS, Telnet, FTP, HTTP, SMTP, NNTP, NFS, BGP, LDAP ...

User Datagram Protocol - UDP

Protocole de Transport :

- sans connexion,
- sans acquits,
- ne conserve pas l'ordre des messages,
- sans contrôle de flux,
- préserve la notion d'enregistrement.

=> possibilités de :

- pertes de messages,
- duplication des messages,
- déséquencelement,
- émetteur trop rapide p/r au récepteur

**Protocole rapide,
Fiabilité suffisante sur un réseau local
pas trop chargé.**

Contrôle d'erreur activable ! Généralement activé même en LAN aujourd'hui.

Protocole qui n'a pas été modifié depuis sa conception (1980).

Transmission Control Protocol - TCP

- Orienté Flot d'octets

ne préserve pas la notion d'enregistrement,
séquencement des octets garanti.

- Mécanisme de Circuit Virtuel : notion de connexion, en Full-Duplex

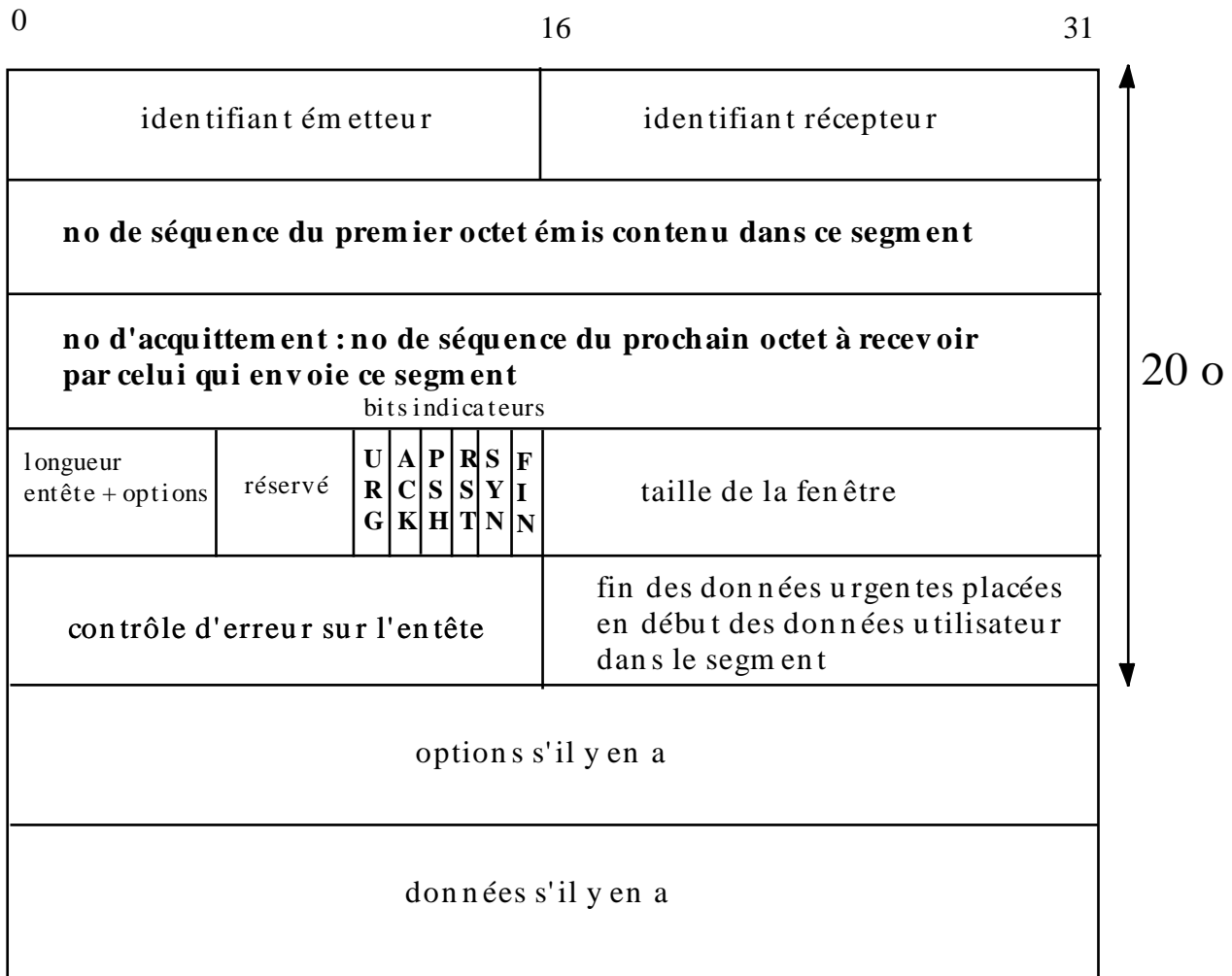
Acquits Positifs avec Retransmissions en cas d'erreurs,
Contrôle de flux
Pas de duplications des messages possibles,
Données urgentes,
Informé des ruptures de connexions.

- Contrôle d'Erreurs,

Pas adapté au multimédia et aux réseaux haut débit actuels.

Protocole qui a été modifié pour améliorer ses performances.

Segment TCP



Les bits indicateurs, s'ils sont positionnés, informent sur la nature du segment :

- . **"SYN"** initialisation d'une connexion, dans ce cas le numéro de séquence porté indique le numéro du premier octet du flot de données, un segment contenant un SYN consomme un octet dans le flot d'octets de données, le numéro i du premier numéro de séquence est déterminé aléatoirement

- . **"ACK"** acquiescement des octets -> numéro envoyé - 1

- . **"RST"** réinitialisation de connexion

- . **"URG"** données urgentes contenues dans le segment

- . **"PSH"** délivrer les données au plus tôt au récepteur dès qu'elles sont correctement reçues

- . **"FIN"** plus aucune donnée ne sera envoyée par celui qui a fait FIN.

Champ option : type (1 octet), longueur totale du champ option (1 octet), infos associées à l'option (variable)

Contrôle de flux TCP

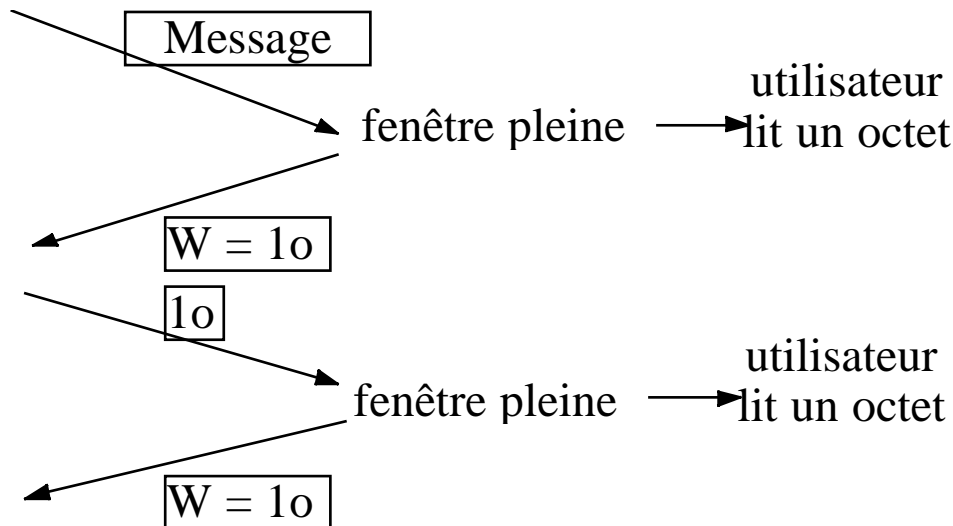
Mécanisme de fenêtre d'octets : Chaque extrémité indique une taille de fenêtre, champ sur 16 bits (soit 65 535 o).

Le récepteur envoie son crédit en fonction des octets de données que retire son utilisateur.

La valeur de la taille de la fenêtre est indiquée dans chaque segment (Principe de crédit plutôt que de fenêtre. C'est pour mieux résister aux pertes de segments, en effet, si un émetteur ne reçoit pas son crédit, il ne reste bloqué que jusqu'au message suivant).

Silly Window Syndrome

Problème qui se produit quand on transmet de grands blocs côté émetteur, et que le récepteur ne lit ses données que par très petits bouts.



Solution :

Le récepteur ne donne que des tailles de fenêtre dont la valeur est au moins supérieure à la moitié de la taille maximum de la fenêtre.

L'émetteur respecte les conditions suivantes pour envoyer ses segments :

- segment de longueur max possible (taille max fenêtre)
- segment de taille $>$ moitié de la taille max de la fenêtre
- pas obligatoire d'attendre un ack en réponse à l'envoi d'un segment pour transmettre à nouveau

Transfert fiable et Fast Retransmit

TCP offre un transfert fiable au-dessus d'un réseau à datagramme

Deux techniques pour les retransmissions :

- Go Back N

A chaque segment émis est associé une temporisation, si elle arrive à échéance, on retransmet tous les segments émis depuis le segment dont la temporisation a expiré. L'évaluation de la valeur de cette temporisation est importante, elle est fondée sur le délai de propagation A/R

- Fast Retransmit and fast recovery

Dès qu'un récepteur reçoit un segment hors séquence, il envoie un ACK avec le no du prochain octet à recevoir qui lui est correct, et sauve le segment reçu en attendant de combler le flot de données avec le segment manquant.

Un émetteur qui reçoit plusieurs fois le même ACK (duplicated ACK) suspecte qu'il y a eu une perte de segment. Au bout de 3 ACKs identiques, il ré-émet le segment manquant

Pb de dimensionnement avec les réseaux haut débit

capacité d'une connexion (*bit* ou en *octet* si on / par 8)
 = débit nominal (b/s) * délai de propagation⁷ A/R

Type de Réseau	nominal b/s	nominal o/s	RTT (ms)	capacité (o)
Ethernet 10Mb/s	10Mb/s	1,25Mo/s	3ms	3,750ko
Ethernet 100Mb/S	100Mb/s	12,5Mo/s	3ms	37,5ko
Ethernet 1Gb/s	1Gb/s	125Mo/s	3ms	375ko
multiplex T1 cont	1,544Mb/s	0,193Mo/s	60ms	11,58ko
multiplex T1 sat	1,544Mb/S	0,193Mo/s	500ms	95,5ko
multiplex T3 cont	44,736Mb/s	5,592Mo/s	60ms	335,52ko

La capacité d'une connexion est à comparer avec la taille de la fenêtre.

La taille de la fenêtre (65,5ko) est trop petite pour certains réseaux. La taille de la fenêtre peut être augmentée avec l'option "window scale" qui permet de définir une taille de fenêtre sur 32 bits (type = 3, lg = 3, valeur).

Le champ option contient un facteur multiplicatif (valeur). Une valeur de 2 dans le champ option "window scale" donne une valeur de $65535 * 2^2$, soit 256 Ko. Cette option s'utilise à l'ouverture de cnx, nécessairement par les deux entités.

⁷ au niveau transport, ne pas confondre avec le niveau physique, ce délai de propagation A/R est spécifique à chaque connexion de Transport, il fait l'objet d'une évaluation périodique par la couche Transport

Vieux Paquets

Combien de temps pour épuiser l'espace des n° de séquence TCP :

Type de Réseau	nominal b/s	nominal o/s	délai d'épuisement
Ethernet 10Mb/s	10Mb/s	1,25Mo/s	57mn
Ethernet 100Mb/S	100Mb/s	12,5Mo/s	5,7mn
Ethernet 1Gb/s	1Gb/s	125Mo/s	34s
multiplex T1 cont	1,544Mb/s	0,193Mo/s	46mn
multiplex T3 cont	44,736Mb/s	5,592Mo/s	12mn48s
ATM	155,52Mb/s	19,44Mo/s	3mn40s

Quelle valeur de MSL faut-il pour qu'un vieux paquet ne soit pas accepté comme un paquet correct du flot, 30s, 1mn, **2 mn** ?

Slow Start – Contrôle de Flux

Mécanisme qui participe au contrôle de congestion de l'Internet. Le contrôle de congestion est géré à deux niveaux réseau et transport, mais plutôt au niveau Transport. Principe au niveau automate de Transport : Quand l'émetteur détecte une congestion, il diminue le débit des données qu'il soumet.

Détection ? Quand une temporisation arrive à échéance pour un segment envoyé (pas d'ACK ou ACK en retard p/r à la tempo d'où l'importance du réglage des temporisations qui par ailleurs évoluent dynamiquement), en effet une erreur de transmission est maintenant trop peu fréquente.

Que fait-il ? Il maintient une **fenêtre de congestion**, au départ taille max fenêtre de congestion est toujours bornée par taille max d'un segment (MSS, information envoyée à l'ouverture de cx). En fait, la taille des données émises est le minimum de la taille courante de la fenêtre de congestion, et du crédit alloué par le récepteur (combinaison capacité réseau et capacité récepteur).

Quand il reçoit un ACK avant expiration de tempo associée à son 1er segment émis, il augmente la taille de sa fenêtre de congestion de 1 segment. Et il envoie 2 segments. Quand les 2 segments seront acquittés, il pourra encore augmenter sa fenêtre de congestion (de 2 segments)... augmentation exponentielle de la taille de la fenêtre ! Attention, il n'envoie jamais plus que la taille du crédit spécifié par le récepteur.

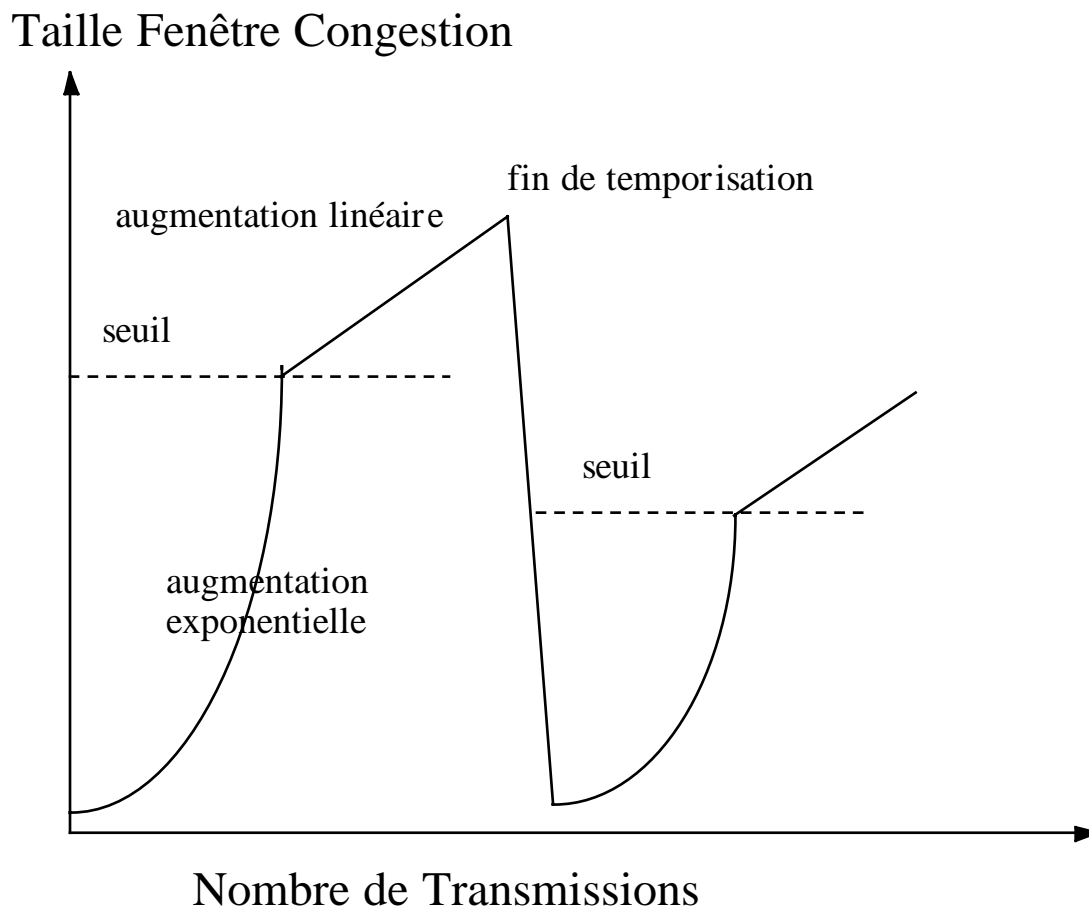
A un certain moment, la capacité maximale du réseau est atteinte, ce qui se traduit par "un segment est perdu"... l'ACK ne revient pas (fin de temporisation)! Il y a congestion !! Idem si ACK dupliqué ! La fenêtre de congestion est trop large. Idem à la réception d'un message ICMP SOURCE_QUENCH

Un deuxième mécanisme entre en jeu, le contrôle de congestion. Le seuil de congestion est initialisé à la moitié de la fenêtre de congestion au moment où la congestion a été détectée, et la fenêtre de congestion est ré-initialisée à 1 segment.

Maintenant, on recommence comme à la phase précédente, quand la fenêtre de congestion atteint le seuil de congestion, elle ne progresse que de 1 segment à la fois.

La détection d'une nouvelle congestion divise à nouveau la fenêtre de congestion de moitié.

Evolution de la taille de fenêtre de congestion



Diamètre d'une connexion TCP

Connaître le diamètre permet d'éviter la fragmentation en cours de route.

Le diamètre de la connexion correspond au MTU du plus petit lien rencontré sur une connexion. Quand une connexion TCP est ouverte, TCP utilise le paramètre MSS fourni par l'autre entité ou le MTU de l'interface de sortie.

Les datagrammes sur cette connexion ont le bit DF à 1 (Don't fragment). Un routeur qui doit fragmenter, élimine le datagramme et génère un message ICMP "can't fragment". Suivant la version d'ICMP, la taille du MTU avec le prochain noeud peut être indiquée.

Les prochains datagrammes envoyés sont plus petits. Toutefois, comme les routes sont multiples, et qu'elles changent, TCP tente périodiquement (toutes les 10 mn recommandé) d'augmenter la taille des segments.

Small is beautiful ?

Supposons qu'on envoie 8ko à travers un réseau de 4 routeurs reliés chacun par un multiplex E1 (2,048Mb/s).

1ère solution : 2 datagrammes de 4096o
 $(4096+40)*8 / 2,048 \text{ Mb/s} = 19,6 \text{ ms}$ par datagramme
soit 98.36 ms au total ($19,6 * (4+1)$)

2ème solution : 16 datagrammes de 512o
 $(512 + 40) * 8 / 2,048 = 2.15 \text{ ms}$ par datagramme
soit 41ms au total ($2,15 * (4 + 15)$)

Ce résultat va à l'encontre des idées reçues qui considèrent qu'il vaut mieux envoyer de gros datagrammes pour rentabiliser l'effort de gestion protocolaire.

En fait, le raisonnement est faussé car on ne tient pas compte des temps de commutation dans les routeurs traversés, ce qui a une influence malgré les avancées technologiques.

Approche IntServ

RSVP - Resource ReserVation Protocol

Protocole de Réserveation de Ressources Réseau, équivalent de RCAP dans TENET, il est prévu pour IPV4 comme IPV6.

Il s'accompagne d'un modèle de gestion des ressources

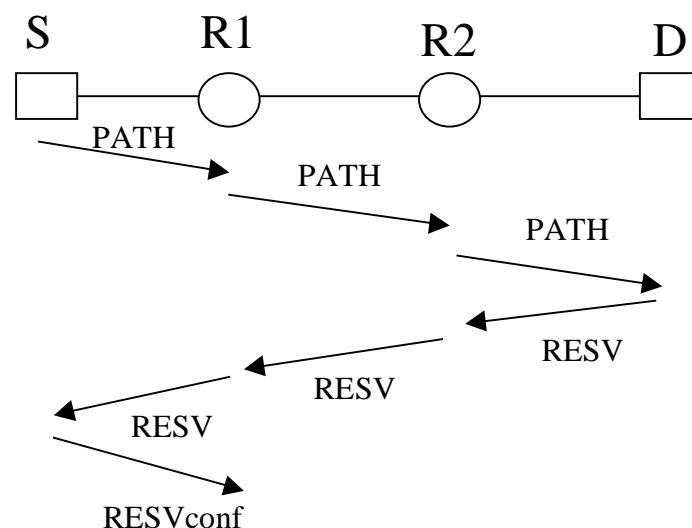
RFC : 1363 (92), puis 2205,2210, 2211, 2212, 2215, 2216 (97)

Il repose sur deux concepts clefs :

- les flots de données (d'un émetteur vers un ou plusieurs récepteurs) unidirectionnels
- les réservations

Un flot est identifié par l'adresse de destination (classe D quand multicast), un no de port de destination, et un protocole.

Le chemin (unicast ou multicast) est établi par l'émetteur, et la réservation effective des ressources nécessaires est effectuée par le(s) récepteurs). L'émetteur n'est pas nécessairement dans le groupe en cas d'adresse multicast.



Les messages de réservation sont émis périodiquement par les récepteurs . Ils participent au maintien d'un état logique du flot. Quand ils ne passent plus le chemin et les ressources associées sont relâchées.

Modèle de Contrat - QoS IETF

"Integrated Services model" (IS), deux modèles :

- **service garanti** (pour trafic avec contraintes TR équivalent ATM-CBR),
- **service avec contrôle de charge** (best effort amélioré équivalent ATM-ABR).

Paramètres de Contrôle d'un noeud:

- **NON_IS_HOP** indique si un noeud est capable ou non de gérer de la QoS
- **NUMBER_OF_IS_HOPS** : nombre de noeuds capables de gérer de la QoS
- **AVAILABLE_PATH_BANDWIDTH** : estimation locale de la bande passante disponible sur le chemin du flot de données (o/s)
- **MINIMUM_PATH_LATENCY** : estimation du délai introduit par la traversée du noeud
- **PATH_MTU** : MTU estimé pour le flot, communiqué au récepteur, sans RSVP seul l'émetteur peut lancer une découverte de MTU
- **TOKEN_BUCKET_TSPEC** : paramètres de trafic :
 - * r, token rate (O_{de datagramme} IP/s) 1 à 10¹² o/s
 - * b, profondeur du seau (o) 1 à 250*10⁹ o
 - * p, débit crête (O_{de datagramme} IP/s) 1 à 10¹² o/s
 - * m, taille minimum d'une unité de donnée traitée, comprend toutes données et toutes les entêtes, et sert à l'allocation de ressources, une unité de donnée inférieure à cette taille est traitée comme si elle était de taille m
 - * M, taille maximum d'un datagramme (o), les datagrammes de taille > à M sont déclarés non conformes

Tspec d'un flot de données "Qos Charge Controlée"

Classe de service QoS Charge Controlée :

L'utilisateur spécifie le trafic qu'il soumet, **Tspec** (cf slide précédent) :

* r, débit ($O_{\text{datagramme IP/s}}$) 1 à 10^{12} o/s

* b, profondeur de la file (o) 1 à $250 \cdot 10^9$ o

* p, débit crête ($O_{\text{datagramme IP/s}}$) 1 à 10^{12} o/s

* m, taille minimum d'une unité de donnée traitée,

* M, taille maximum d'un paquet (o)

pas de contraintes sur le délai et le taux de perte

Pas de spécification de taux de perte ni de latence.

Suppose que le réseau n'est pas en surcharge, et qu'il écoule globalement le trafic qui lui est soumis, les noeuds réservent suffisamment de ressources pour écouler ce trafic. Les paquets de taille $> \text{PATH_MTU}$ sont éliminés (pas de fragmentation autorisée).

Les noeuds gérant ce type de QoS ont la charge d'éviter toute interférence entre flots.

Tspec, Rspec d'un flot de données "QoS Garantie"

La classe de service "QoS garantie" vise à minimiser le temps d'acheminement des données (borne max), et à ne pas perdre de données à cause de surcharges.

L'émetteur spécifie le trafic qu'il soumet, **Tspec** identique au cas précédent.

Le récepteur spécifie le trafic qu'il veut réserver, **Rspec** :

- **R** ($R > r$ du **Tspec**), débit
- **S** écart entre le délai d'acheminement calculé par la réservation, et le délai souhaité par l'émetteur (micro-sec)

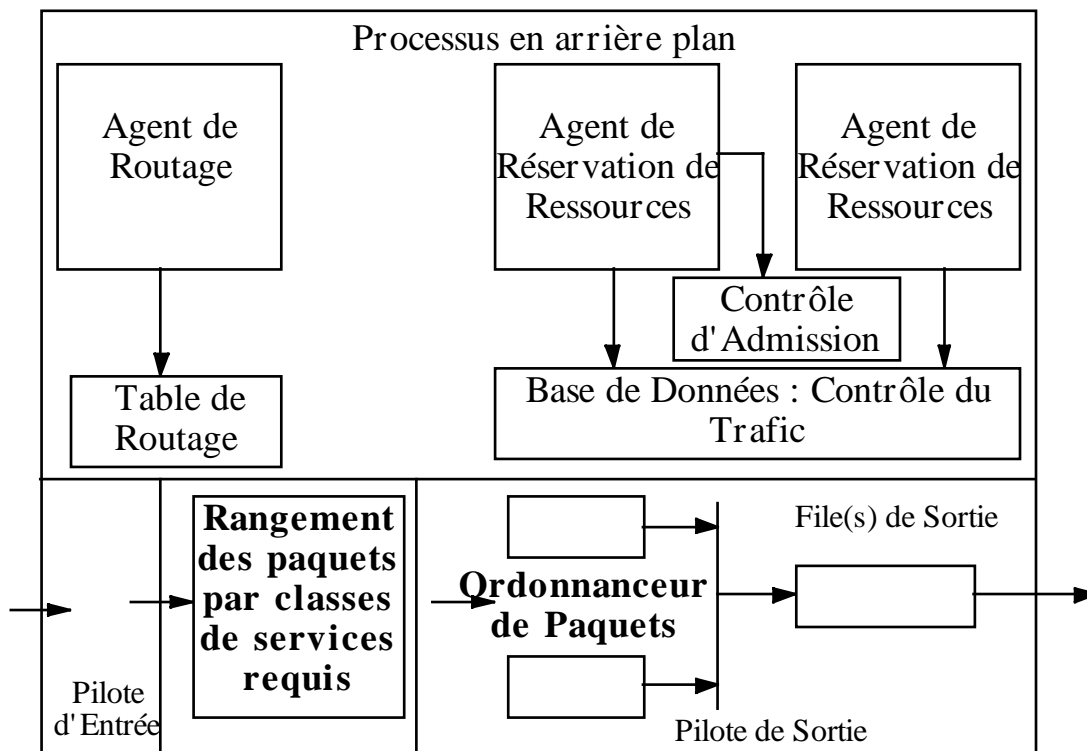
Deux types de politiques pour gérer le trafic :

- simple : comparaison des caractéristiques du flot avec le contrat dans **Tspec**
- avec façonnage du flux : tente de remettre le flot de données en conformité avec le contrat :

 utilisation d'un "token bucket", d'une régulation de débit et de buffers

Pas de Fragmentation possible

Modèle d'Implantation de l'IS - Gestion du Trafic



Problèmes à résoudre :

- choix des paquets à éliminer en cas de surcharge
- gestion de statistiques de coûts

Modèles de Réservation

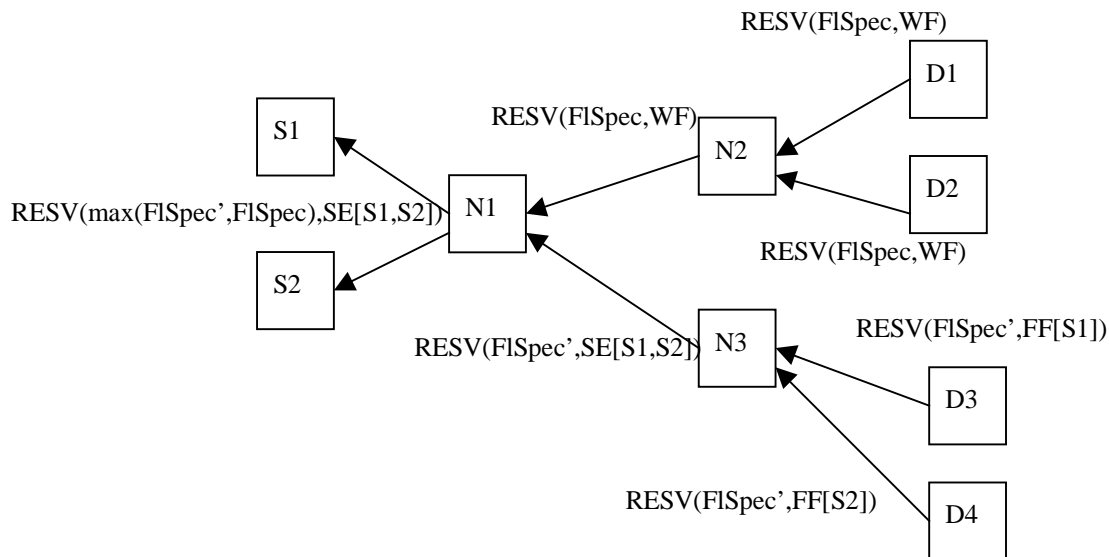
Les réservations de ressources sont faites à l'initiative des récepteurs. Il va falloir définir une politique d'intégration des différentes réservations, notion de "style de réservation" :

Les styles de réservations dépendent de deux options, l'une par le récepteur (mode distinct, mode partagé), l'autre par l'émetteur (mode explicite, mode ouvert).

Sélection Emetteur	Mode de Réservation	
	Distinct	Partagé
Explicite	FF	SE
Ouvert		WF

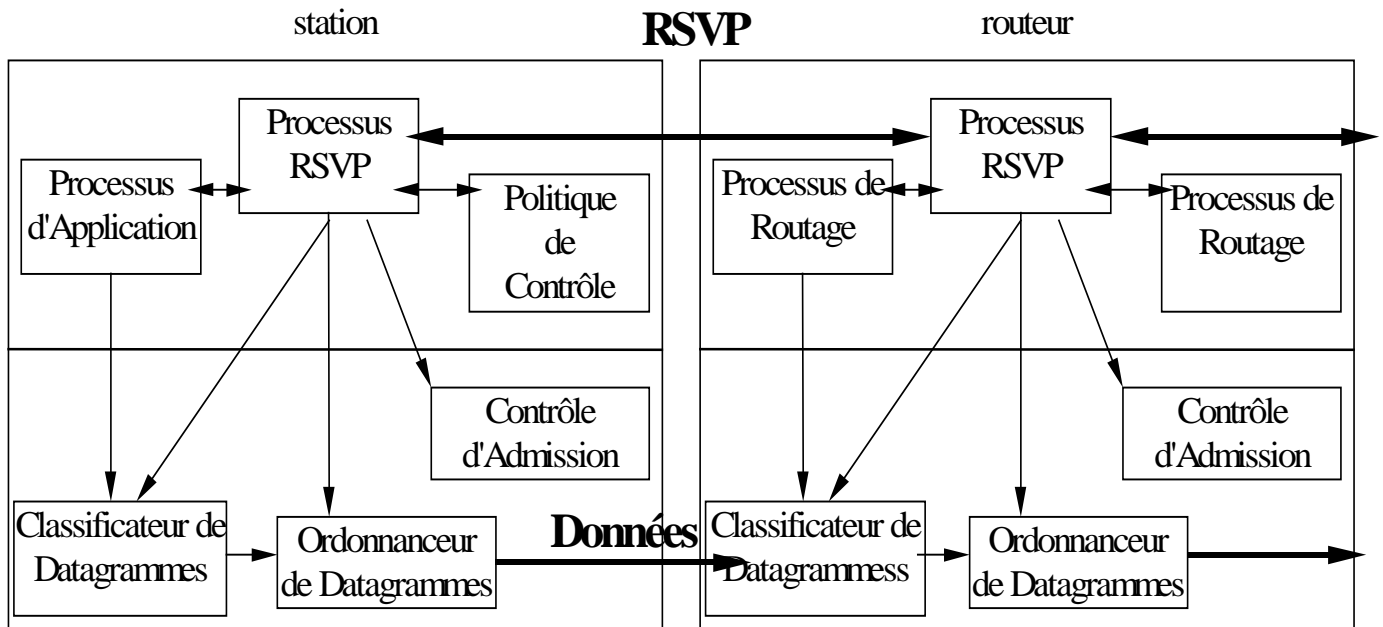
- Filtre Fixe (FF) - les ressources sont réservées pour le flot uniquement -> unicast et multicast
- Partage Explicite (SE) - les ressources sont partagées entre plusieurs flots qui proviennent de plusieurs émetteurs identifiés -> multicast
- Filtre Ouvert (WF - wildcard filter) - les ressources sont réservées pour un type de flot qui proviennent de plusieurs émetteurs, les flots du même type partagent les mêmes ressources -> multicast

Aggrégation des demandes en multicast avec filtre



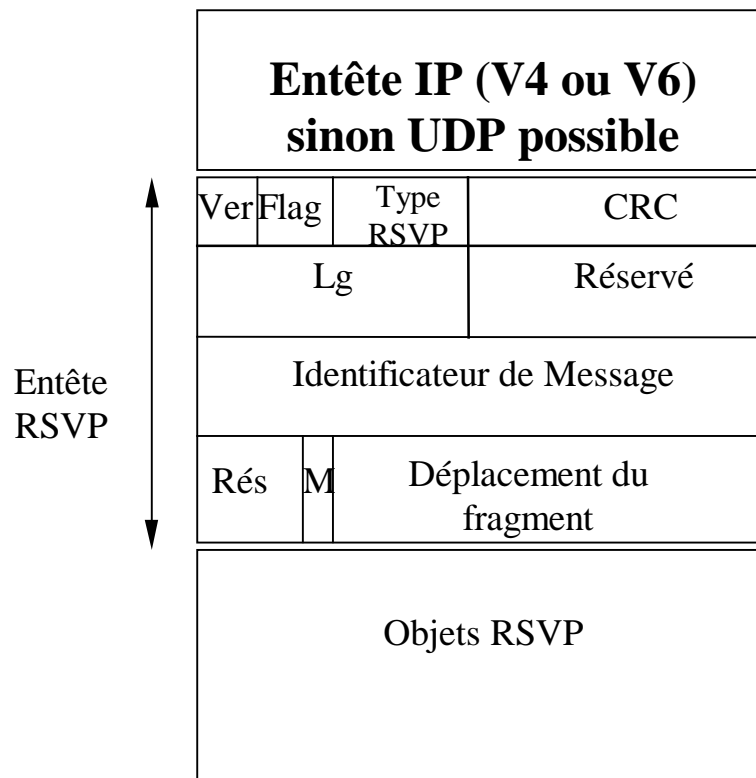
Architecture RSVP

Architecture qui instancie le modèle d'implantation ci-dessus



Interactions avec l'application via une bibliothèque qui masque l'API utilisée et qui dépend de l'OS.

Format d'un message RSVP :



Les messages RSVP sont gérés comme ceux du protocole ICMP, ils sont dans la charge utile de datagrammes IP.

Types de Messages :

- PATH (Emetteur vers Récepteur(s)) message de chemin
- RESV (Récepteur vers Emetteur) message de réservation
- PATHERR (Récepteur vers Emetteur) indication d'erreur sur le traitement du chemin vers récepteur
- RESVERR (Récepteur vers Emetteur) indication d'erreur lors de la réservation de ressources
- PATHTEAR (Emetteur ou noeuds vers noeuds suivants du chemin et récepteur(s)) abandon du flot
- RESVTEAR (Récepteur(s) ou noeuds vers noeuds précédent du chemin et émetteur) abandon du flot

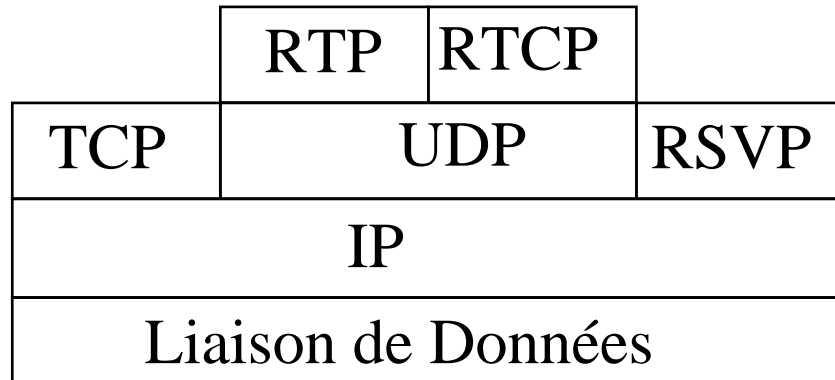
Principaux Objets RSVP

Format général d'un objet RSVP :

Longueur de l'objet	Numéro de Classe	Type de Classe
Contenu de l'objet		

No	Objet	Typ	Description
9	FLOWSPEC	1	flowspec requiert délai borné
		2	flowspec requiert QoS
		3	flowspec requiert QoS garantie
		254	flowspec de plusieurs flots non mélangés
10	FILTER_SPEC	1	spec filtre sur flot pour réseau de type IPV4
		2	spec filtre de type IPV6 utilisant le port source
		3	spec filtre de type IPV6 utilisant l'étiquette de flot
11	SENDER_TEMPLATE	1	description de flot par émetteur pour réseau type IPV4
		2	description de flot par émetteur pour réseau type IPV6
12	SENDER_TSPEC	1	description de trafic généré par l'émetteur
13	ADSPEC	1	déclaration d'info par l'émetteur, et par les nœuds traversés

Pile de protocoles Temps Réel IP



Aucune hypothèse sur la couche Liaison de Données, excepté le fait que certaines liaisons peuvent ne pas être satisfaisantes.

RTP : Real Time Protocol

RTCP : Real Time Control Protocol

RSVP : Resource Reservation Protocol, équivalent de RCAP dans TENET, il s'accompagne d'un modèle de gestion des ressources

applications visées : audioconférence, visioconférence donc de type **isochrone**

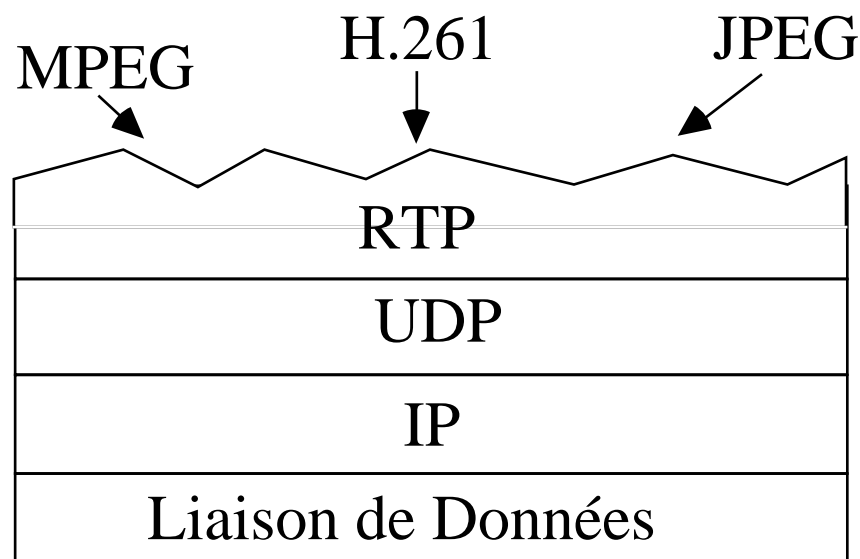
RTP utilise le port 5004

Hypothèses de Conception de RTP

Les flux de données vidéo, son, image tolèrent des pertes de messages mais pas des dépassements d'échéances.

RTP se **combine** avec des protocoles de plus haut niveau spécialisés pour un type de média.

Exemple d'utilisation de RTP avec des techniques de compression vidéo :

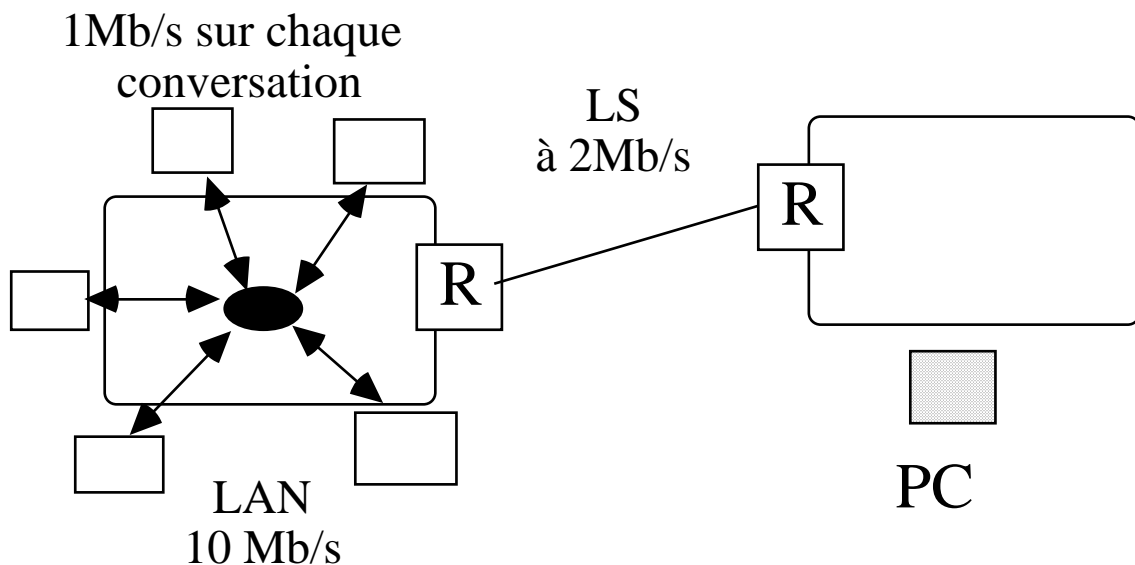


A partir du flux de données, le récepteur doit pouvoir resynchroniser les informations pour les restituer : à travers RTP, la source doit pouvoir mettre des estampilles temporelles.

La notion de conférence implique des flux de données en diffusion et en mode conversation.

Pb à résoudre

Vidéoconférence entre des stations sur un réseau local, un PC sur un autre réseau distant veut rejoindre la conférence :

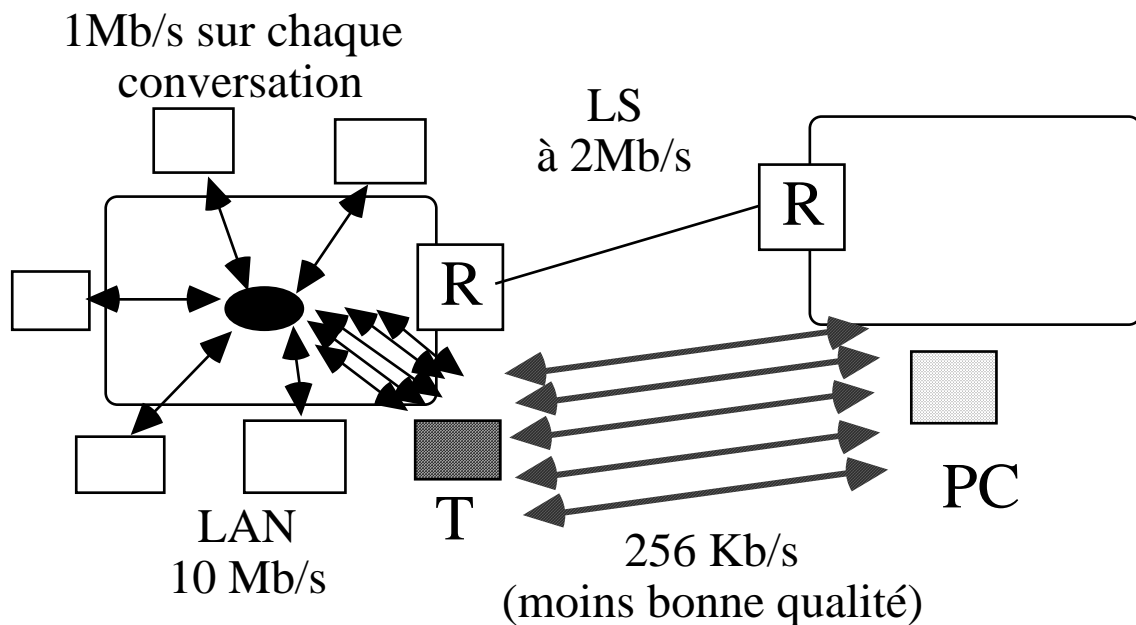


Trafic total sur le LAN de 5Mb/s ne peut pas passer sur la LS.

Solution : Utilisation de Translateurs et de Mixeurs

Translateurs

Le translateur est une sorte de convertisseur capable de modifier un flot de données (isochrone) en un flot de moins bonne qualité :

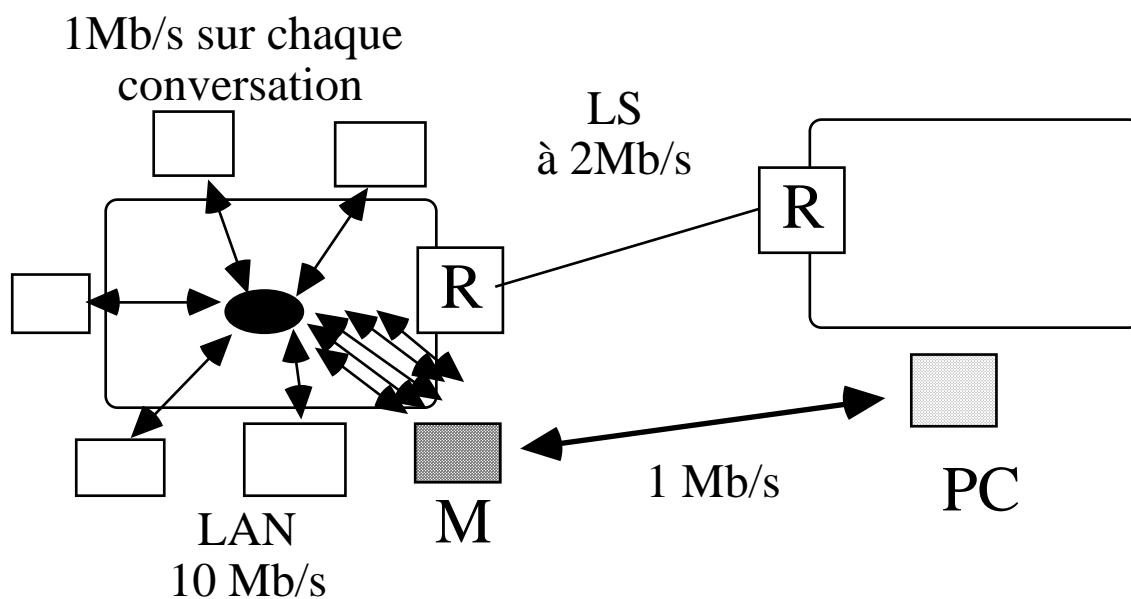


En entrée, le Translateur accepte les flots de 1 Mb/s, et les convertit en flots de 256 Kb/s

La vidéoconférence peut maintenant atteindre le PC.

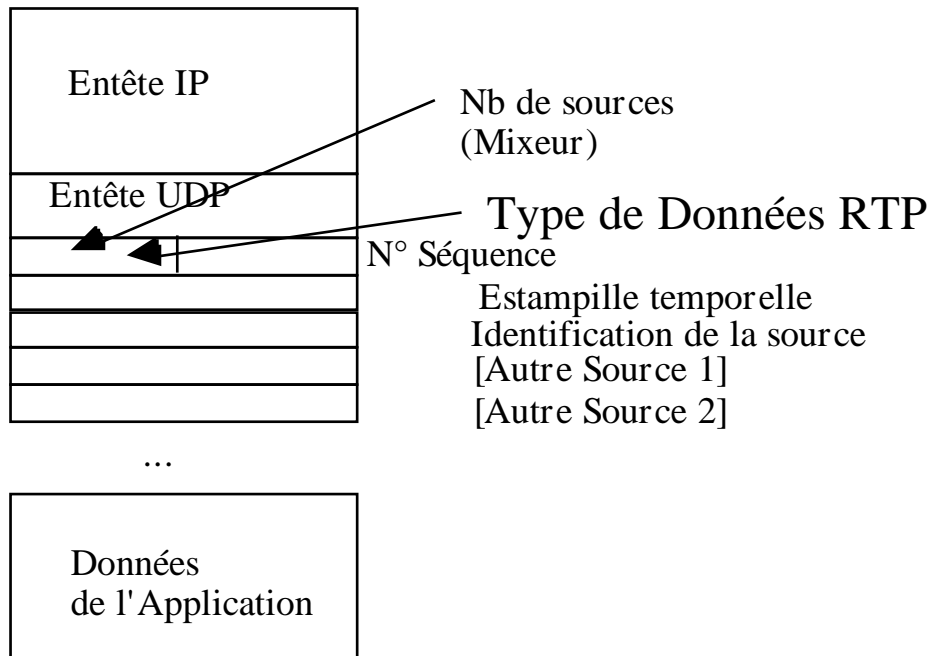
Mixeurs

Les Mixeurs ont un objectif équivalent à celui des Translateurs sauf qu'ils combinent les flots.



Cette technique est plus adaptée à des flots de données audio.

Message RTP

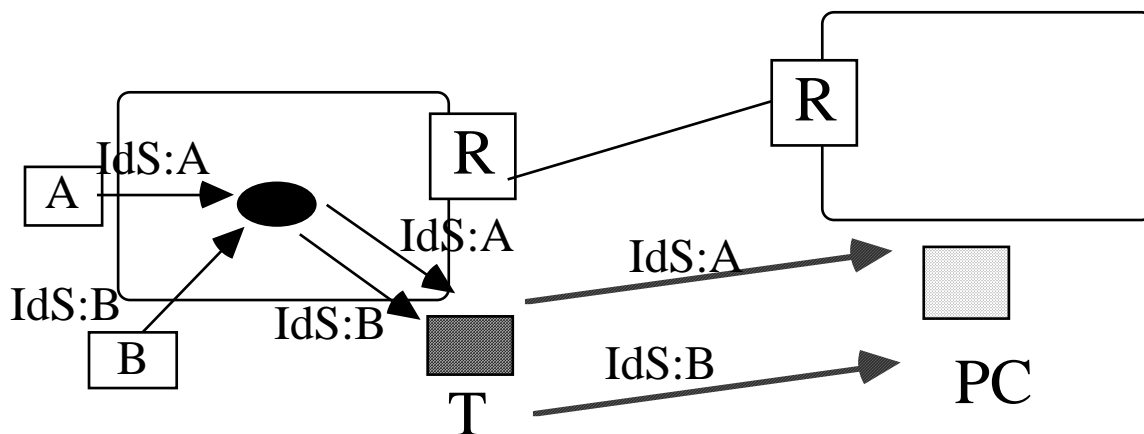


La source est le premier émetteur du message, il détermine le numéro de séquence, l'estampille temporelle (date).

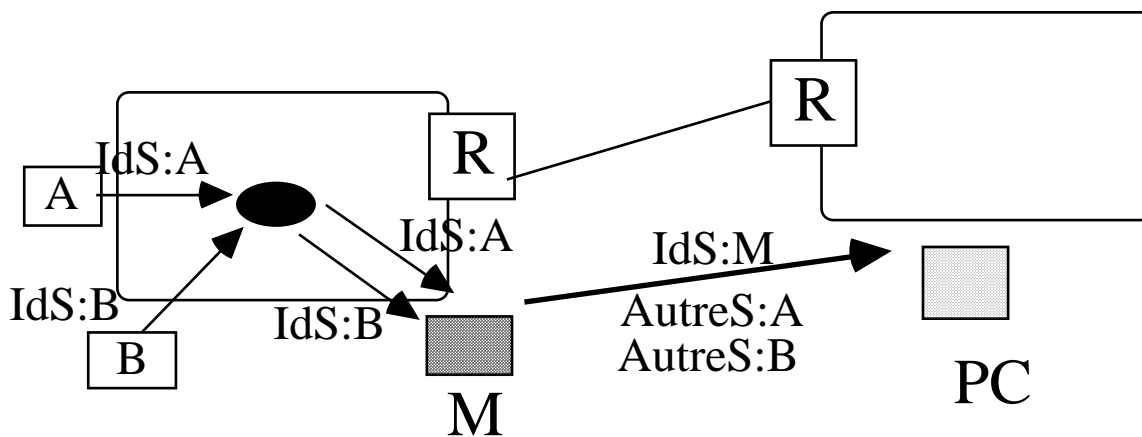
Les translateurs préservent l'identification de la source tandis que les mixeurs la modifient.

Identification de sources

Translateurs



Mixeurs



Protocole RTCP

Accompagne le protocole RTP, correspond au port 5005.

RTP => flot de données

RTCP => flot de contrôle

Les messages RTCP sont envoyés en diffusion sur un groupe "multicast".

Permet d'échanger des "rapports d'activité", 5 types de messages :

- **Rapport Emetteur** : l'émetteur envoie périodiquement aux récepteurs ce qu'ils auraient du recevoir, un émetteur peut être l'initiateur de la conférence, mais aussi un participant : estampilles temporelles (temps absolu émetteur, date RTP), nb de messages RTP, nb d'octets de données transmis, délai depuis le dernier rapport émetteur, délai écoulé depuis le dernier rapport récepteur...
- **Rapport Récepteur** : pendant du rapport récepteur pour un site qui ne fait que recevoir
- **Description de la Source**
- **Fin de participation**
- **Type spécifique**, dépend de l'application

ST2 – Internet Stream Protocol V2

Protocole de nouvelle génération pour le multimédia. Même niveau que IP. Date du milieu des années 90. Expérimental.

Les flots multicast sont supportés par des connections unidirectionnelles.

La version ST2+ (IETF 1995) s'accompagne d'un protocole de gestion de ressources pour les communications multicast à la RSVP.

Dans ST2, la modification de l'arbre qui soutient la diffusion multicast est à l'initiative de la source. Un participant qui veut rejoindre le groupe doit contacter la source. Ce mode de fonctionnement disparaît dans ST2+

Conclusion

A. L'échange de données multimédia fait apparaître de nouvelles contraintes sur les réseaux. Mais c'est un besoin réel pour les applications classiques.

Il faut offrir des réseaux avec des garanties temporelles.

Deux Modèles : ATM et Internet-RSVP

Concurrence ou Complémentarité ?

B. L'approche intégrée IntServ semble trop complexe, et difficile à appliquer sur l'Internet déjà déployé.

L'IETF propose une approche différenciée, DiffServ, qui vise à améliorer la technologie déjà déployée, en particulier au niveau des routeurs.

IntServ ou DiffServ ?