

Introduction à la problématique des Réseaux avec QoS

-Éléments d'Architecture de l'Internet à QoS -

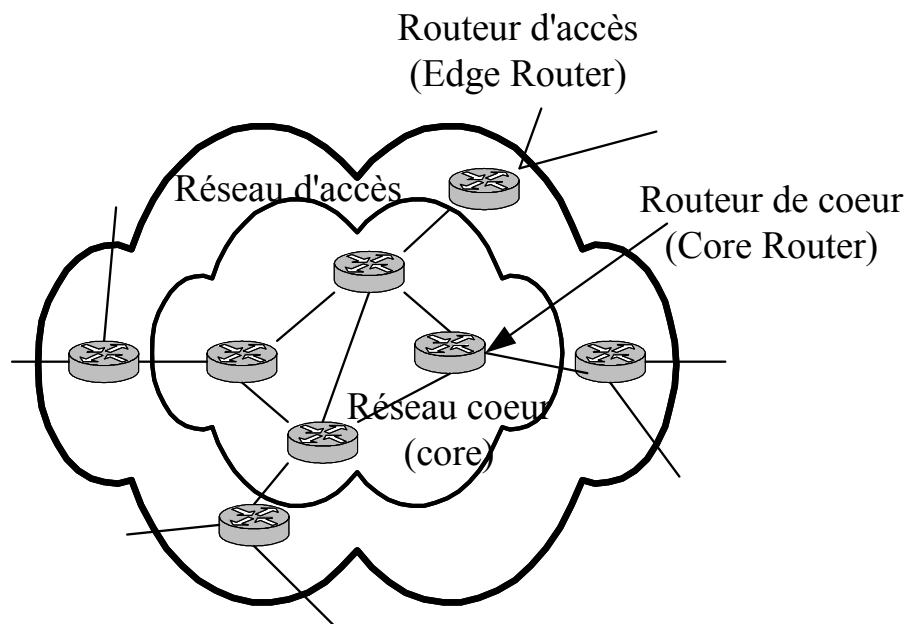
Ingénierie des Réseaux d'Entreprise (Cycle C),
Compléments Réseaux de Transport et Application (Cycle B)

Janvier 2002

Eric Gressier-Soudan

Architecture de Réseau pour la QoS

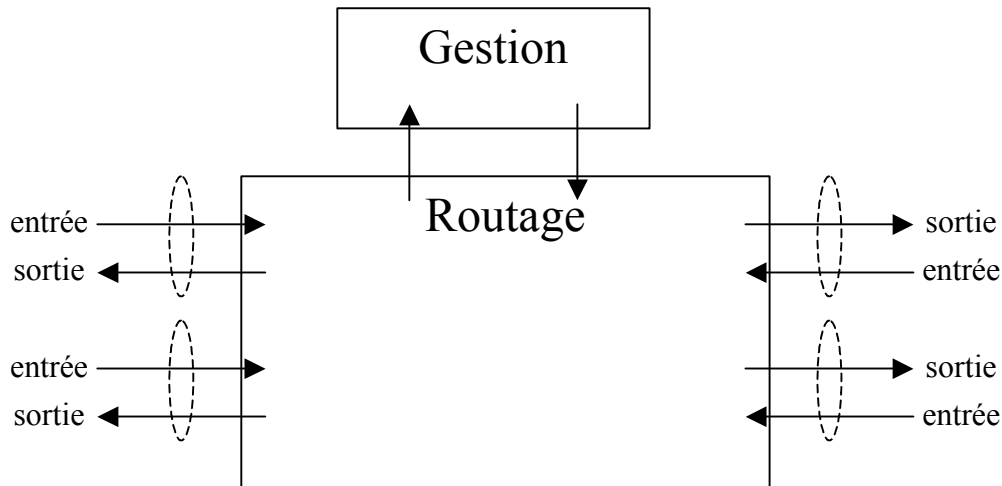
Architecture d'un réseau IP



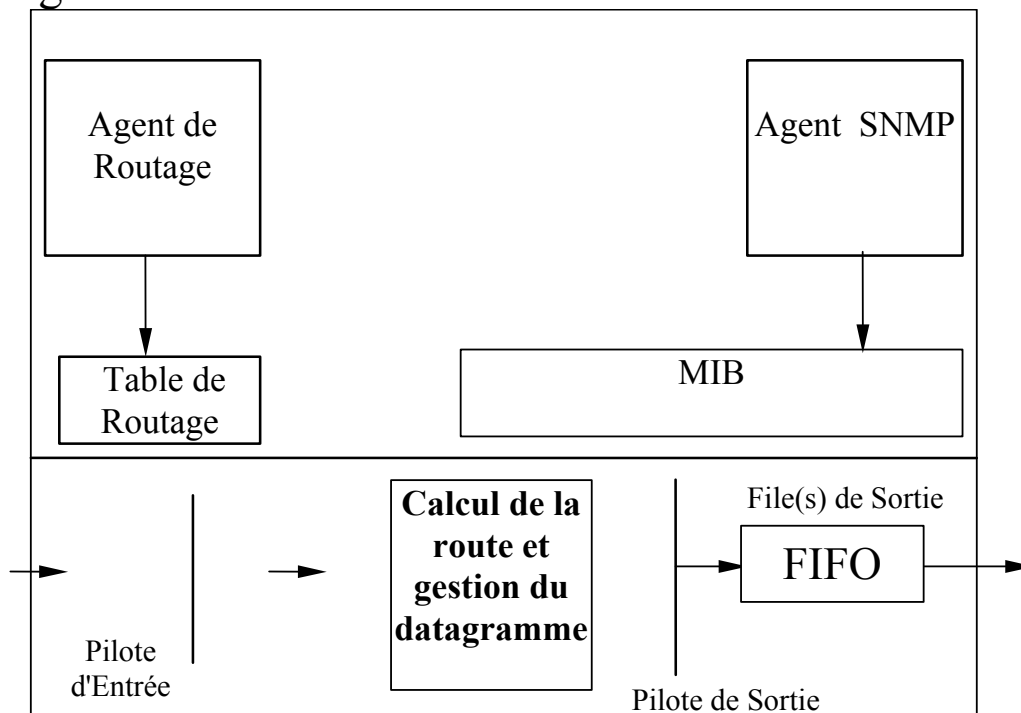
Dans les réseaux à QoS, on distingue deux types de routeurs avec des fonctions différentes :

- les routeurs de cœur, qui font du routage et applique la stratégie de gestion de la QoS décidée par l'opérateur
- les routeurs de bord, qui effectuent l'admission, les opérations de filtrage et de marquage pour la QoS, le lissage de trafic ou la mise en conformité de flux,

Structure d'un Routeur Traditionnel (Best Effort)



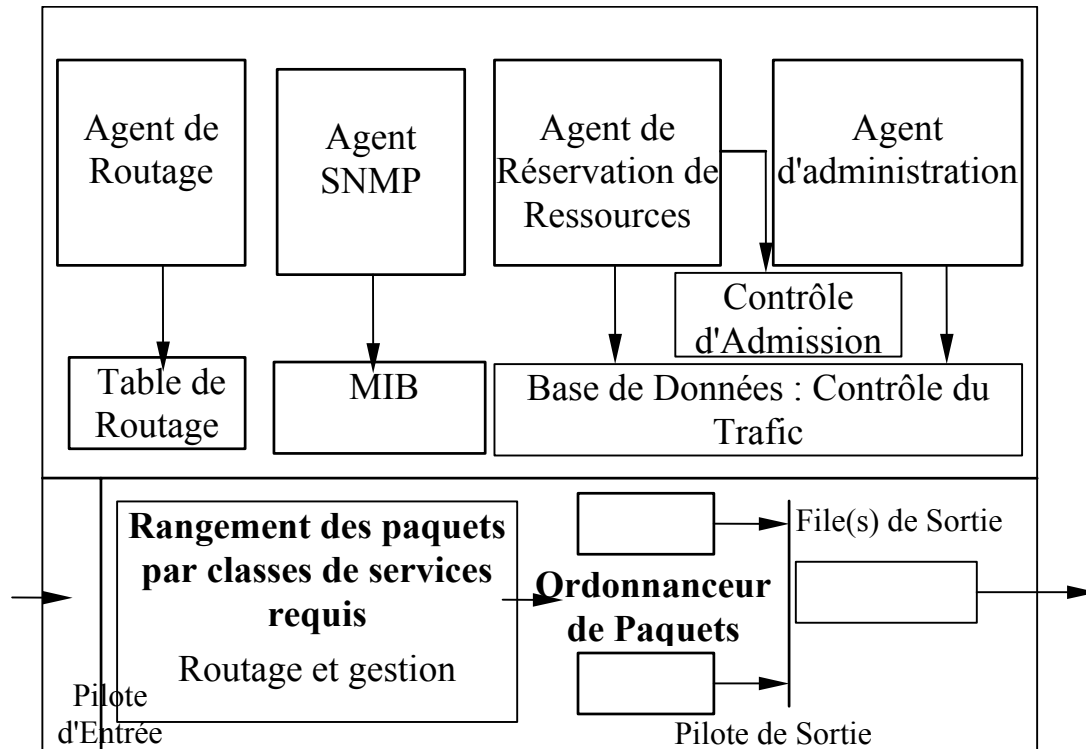
Plan de gestion du routeur



Plan de routage des datagrammes

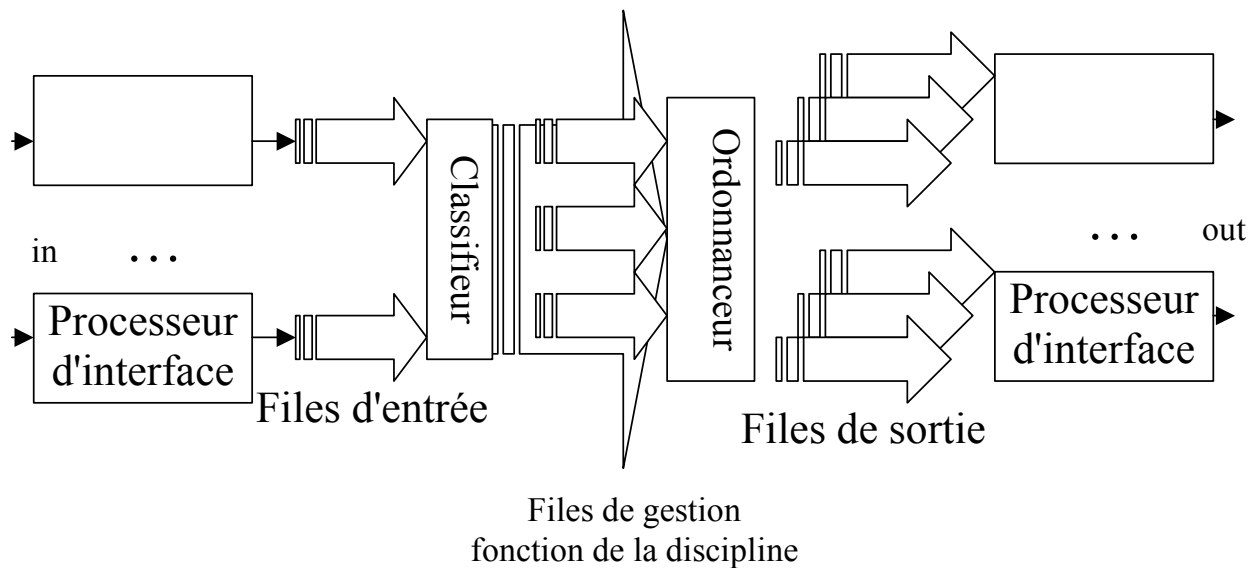
Structure d'un Routeur pour la QoS

Plan de contrôle et de gestion du routeur

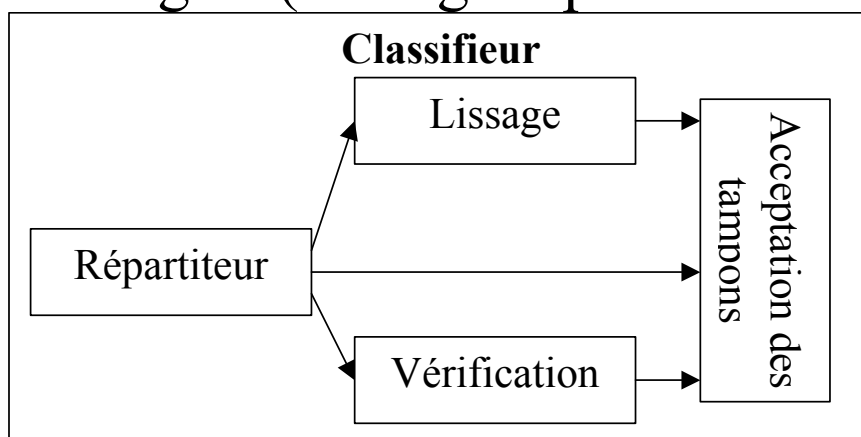


Plan de routage des datagrammes (forwarding)

Fonctions par rapport à la gestion de la QoS



Classifieur : Identification des flots, Lissage du trafic (Shaping/Shaper), Vérification (Policing/Droper) des droits des flots et de leur conformité en fonction de règles (stratégie opérateur réseau)

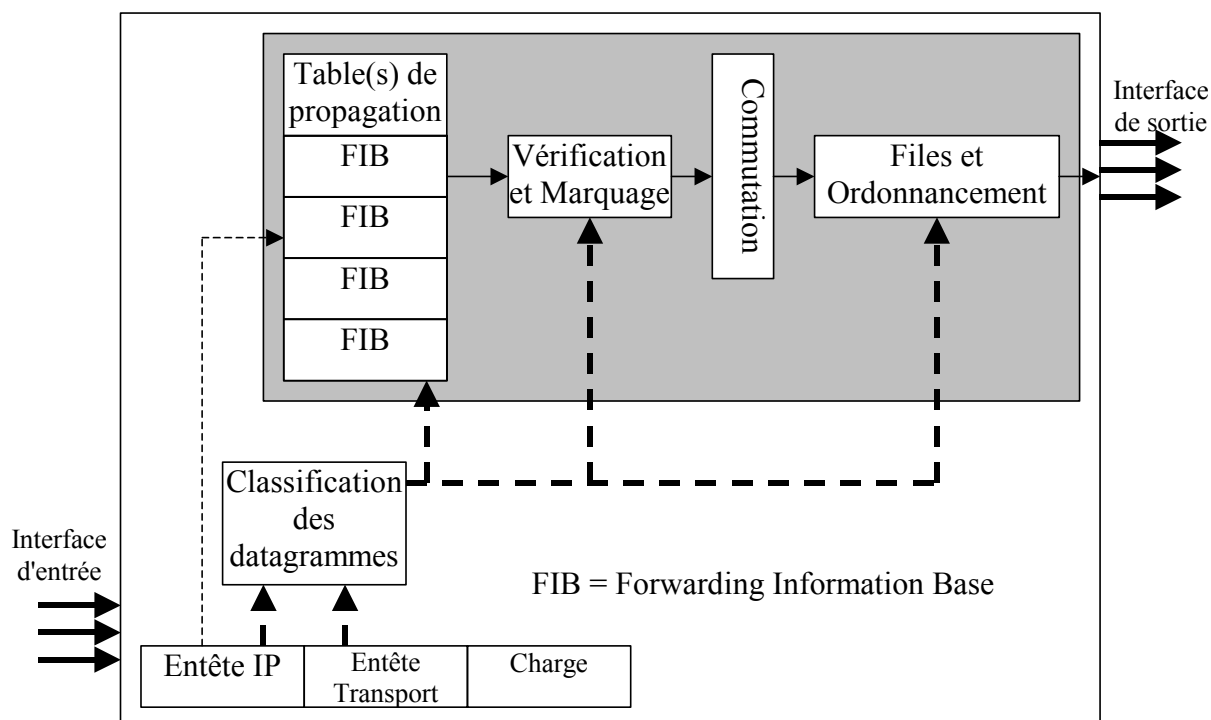


Ordonnanceur : Répartiteur vers les files de sorties fondé sur la stratégie d'allocation des ressources allouées au client en fonction de la QoS requise.

Classifieur (1)

Le Classifieur de datagrammes examine chaque datagramme entrant pour:

- L'identifier en fonction de différents paramètres (@IPS, #portS, @IPD, #portD, protocole, étiquette (ipv6), champ TOS, interface d'entrée encore appelée ingress, information de niveau application¹), permet de retrouver le contexte du flot,
- Vérifier s'il est autorisé à entrer dans ce réseau et dans l'affirmative s'il possède une réservation de ressources (contrat de QoS), si le trafic entrant est conforme au contrat,
- Lisser le trafic pour le rendre conforme éventuellement,
- Rejeter ou marquer le trafic non-conforme de telle façon qu'il soit le premier éliminé en cas de congestion



¹ Par exemple on pourrait pour les échanges Web privilégier les requêtes GET sur les PUSH.

Classifieur (2)

La charge du classifieur est fonction de la complexité de ses traitements, du nombre de flots qui le traverse (si une classification opère par flot et non par classes de services), des règles de gestion de QoS à appliquer. Il faut un compromis entre l'efficacité et la mise en oeuvre de la gestion de QoS.

Le classifieur d'un routeur de cœur peut être plus simple que le classifieur d'un routeur de bord. Celui-ci peut s'appuyer sur le travail fait aux entrées du réseau (plus de lissage ni de vérification à faire).

Remarque : La classification en niveau 7 n'a de sens que pour les passerelles ou les coupes-feux.

Vérification

L'utilisateur peut marquer son trafic (par exemple mettre un niveau de priorité supérieur à celui auquel il a droit pour exploiter plus de bande passante que spécifier dans son contrat).

On peut aussi vérifier la conformité du flot par rapport au débit annoncé.

Cela peut être utile pour limiter la charge d'une machine en cas d'attaque visant à provoquer un déni de service. On peut se servir des routeurs pour participer à la gestion de la sécurité du réseau.

Mécanisme de Lissage du trafic

Le flot de messages peut devenir aléatoire, rafales/saccades avec des données de tailles variables.

Idéalement, il faudrait émettre et transférer des données de taille identique, à un rythme uniforme:

- C'est particulièrement important dans le contexte du multimédia.
- C'est fondamental pour rendre un trafic conforme à l'entrée ou la sortie d'un réseau (frontières) en fonction de sa position de client ou d'opérateur.

Le **lissage du trafic** (traffic shaping) consiste à réguler la vitesse et le **cadencement** des données passant à travers un routeur.

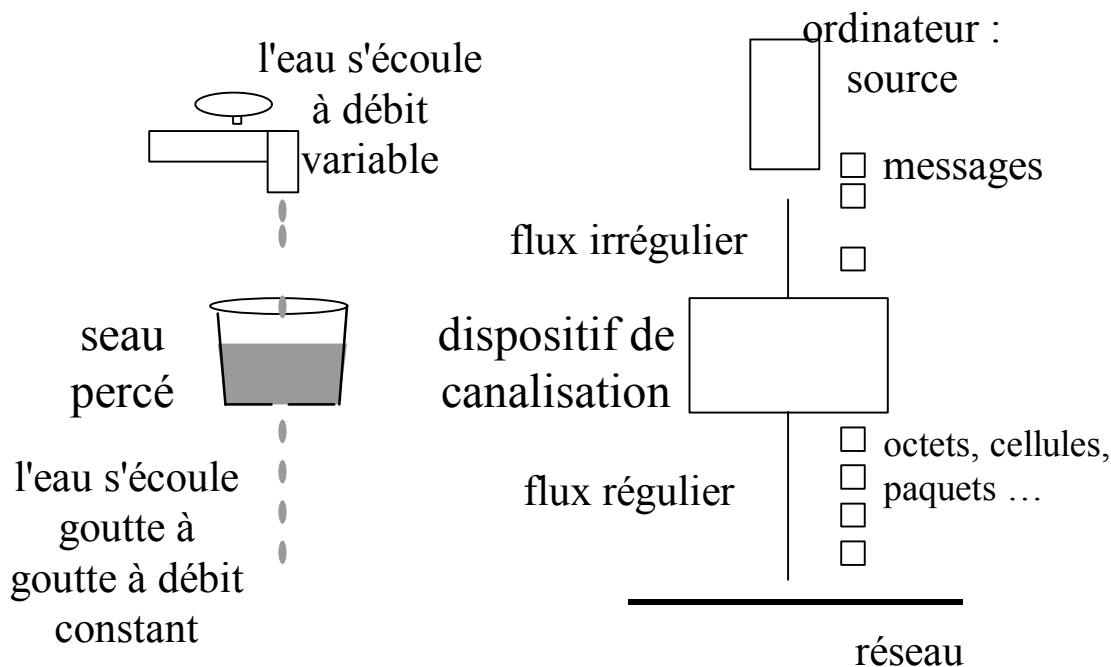
Le lissage de trafic est utilisé habituellement pour le contrôle de débit afin d'éviter la congestion d'un réseau.

Le mécanisme utilisé est un seau à jeton.

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. Le contrôle de flux étant un mécanisme de transport ou de liaison.

Leaky Bucket

Modèle du seau percé



L'émission se fait à une cadence régulière, le dispositif de lissage 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 seau 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 lissage en cas d'avalanche.

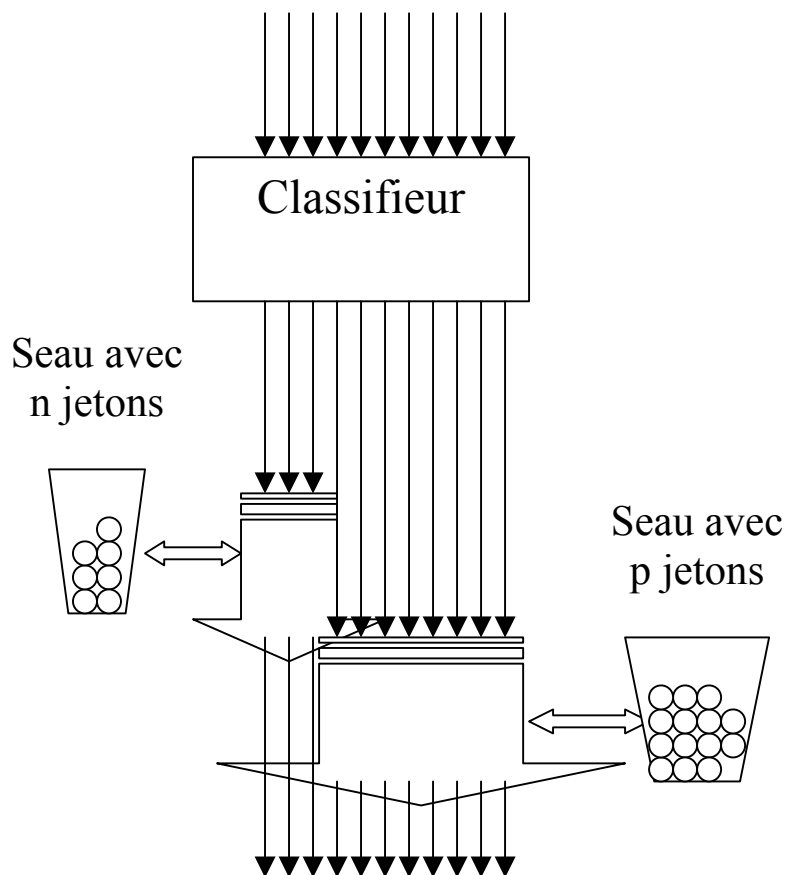
Le modèle du seau percé à jetons fonctionne sur le même principe que le byte counting leaky bucket, excepté que le grain de gestion n'est plus l'octet mais le datagramme. Pour chaque période, le dispositif de lissage dispose de n datagrammes à transmettre au maximum. Il peut en transmettre n en une seule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

Différence avec le "leaky bucket", quand le dispositif est plein, les datagrammes ne sont plus détruits mais rejetés ou marqués éligibles à la destruction en cas de congestion (pas défini dans l'Internet, mais proposé par des constructeurs notamment CISCO en se fondant sur les deux bits non utilisés du champ TOS).

Le lissage peut opérer de façon différente (pas le même nombre de jetons) en fonctions des flux et de leur importance.

Il est possible de combiner plusieurs techniques de lissage (seau à jeton -> uniformise le trafic et les éventuelles rafales, suivi d'un seau percé -> limite le trafic à une valeur seuil prédéfinie)

Exemple d'utilisation du Token Bucket



Deux ensembles de flots subissent un lissage différent.

C'est le même schéma qui sert à vérifier la conformité du trafic soumis au routeur:

- Dans le cas du lissage, on retarde le flux excessif, en supposant qu'il n'excède pas le débit annoncé (attention aux flux sensibles à la latence ou à la gigue comme la voix).
- Dans le cas de la vérification, on élimine le trafic en excès ou on le marque pour élimination lors de congestion (passage en mode best effort).

Dans les deux cas, il faut connaître les caractéristiques du trafic, et donc le contrat de QoS associé:

- * r , token rate, le débit en octets par seconde
- * b , la profondeur du seau en octets

Conséquence de la classification

Lissage vs Vérification :

On peut mettre ces opérations en entrée comme en sortie de réseau :

- Vérification en entrée plutôt chez un opérateur
- Vérification en sortie plutôt chez un utilisateur, permet de ne pas être en conflit avec l'opérateur
- Lissage en entrée permet de calibrer les flux
- Lissage en sortie permet de rendre son flux conforme au contrat de QoS
- Lissage dans les nœuds intermédiaires : re-lisse le flot victime du slow start/congestion avoidance et participe à l'évitement de la congestion

Que faire du trafic non conforme :

- Destruction des paquets non conformes
- Transmission en mode "best-effort"
- Marquage pour candidat à la destruction

Vérification et Lissage ont un impact sur les connexions et donc sur les applications :

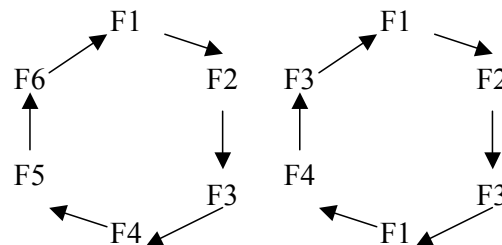
- Lissage ralentit les paquets, compatible avec TCP mais fâcheux pour le multimédia, et pour le téléphone en particulier
- Vérification provoque des retransmissions, ce qui peut mettre à l'épreuve TCP

Ordonnancement des files de sortie (1)

FIFO : simple et classique, on peut avoir un FIFO avec taille fixe, dès que la file est pleine, le trafic excédentaire est éliminé

Priorité : simple, risque de famine des datagrammes moins prioritaires, ressemble à de l'ordonnancement en noyau temps réel, en fait cette stratégie est trop figée et systématique

Round Robin : ordonnancement par tourniquet, trop équitable, on utilise alors **Weighted RR**



Class-Based Queuing : routage par classe (penser à l'ordonnanceur d'un système d'exploitation qui essaie de satisfaire tous les travaux tout en privilégiant certains), c'est une variante plus équitable de la gestion par priorité, c'est bien adapté au champ DSCP de l'entête IP

La bande passante est attribuée par classe (classe = type d'applications), chaque classe bénéficie d'une priorité

Ordonnancement des files de sortie (2)

Weighted Fair Queuing : vise l'équité entre les différents flots en raisonnant en volume d'octets transmis, les plus petits flots ont priorité, (ressemble à l'ordonnancement des processeurs qui cherche à privilégier les travaux en mode interactif sur les travaux batch plus volumineux). WFQ tente d'établir un comportement déterministe dans la prédiction des temps de réponse... adapté à la QoS.

WFQ, c'est du CBQ avec un poids affecté aux classes

WFQ tente de reproduire un comportement de type multiplexage temporel des flux, la tranche temporelle dépend du poids affecté à la classe.

C'est un domaine qui n'est pas très éloigné de l'ordonnancement dans les systèmes d'exploitation.

Politiques de guérison de la congestion

Eviter que toutes les connexions TCP se synchronisent en effectuant le slow start puis le congestion avoidance simultanément.

Mécanismes à ajouter (une seule file) liés au contrôle de congestion :

- Random Early Detection (RED), élimination au hasard de datagrammes dans les files de sortie pour éviter des pertes par rafale sur un flot, et déclencher le contrôle de congestion inopinément.
- Weighted RED, extension qui permet une politique d'élimination multicritère, les paquets de moindre importance sont éliminés d'abord

Mécanismes à ajouter (plusieurs files) liés au contrôle de congestion :

- Longest Queue Drop (LQD), élimination sur la file la plus longue
- Dynamic Queue Length Threshold (DQLT), quand un taux d'occupation est dépassé les datagrammes sont éliminés. La valeur du seuil évolue dynamiquement.

Limitation du trafic d'une source TCP

Cette méthode modifie la valeur du crédit alloué à l'émetteur par le récepteur d'une connexion TCP.

Un routeur diminue le crédit dans un segment TCP qui le traverse. Il rajoutera le crédit utilisé dans les datagrammes qui suivent.

Inconvénients :

- Extraire le crédit nécessite d'examiner l'entête TCP plus en détails, c'est coûteux
- Le routeur maintient un état de la cnx qui le traverse
- Si un segment avec un crédit diminué est perdu, difficile de reconstituer l'état réel.

Cette approche s'appelle TCP rate control.

Quelques URL sur la gestion de trafic en liaison avec l'univers Linux intéressantes (source Joël Berthelin, ingénieur systèmes et réseaux au Département Informatique du Cnam) :

<http://www.docum.org/>

Commentaire de Joël Berthelin : Quelqu'un qui teste et expérimente LinuxTrafficControl avec son analyse, ses réflexions, et surtout :

- des "policing" a lui
- les outils qu'il utilise
- ses outils de gestion de la QoS

Cette page pointe sur une autre plus ciblée sur "Linux Traffic Control" : <http://lartc.org/>, dans laquelle on trouve <http://lartc.org/lartc.pdf>, "Linux Advanced Routing & Traffic Control HOWTO".

à toutes fins utiles, pour générer et afficher des trafics : l'outil iperf

- présentation : <http://dast.nlanr.net/Projects/Iperf/index.html>
- download : <http://dast.nlanr.net/Projects/Iperf/release.html>
- docs : http://dast.nlanr.net/Projects/Iperf/iperfdocs_1.6.3.html#representative

Approche IntServ

RSVP - Resource ReserVation Protocol

Protocole de Réserveation de Ressources Réseau **par flot de transport unidirectionnel**, il est prévu pour IPV4 comme IPV6. Il faut plutôt le voir comme un protocole de signalisation.

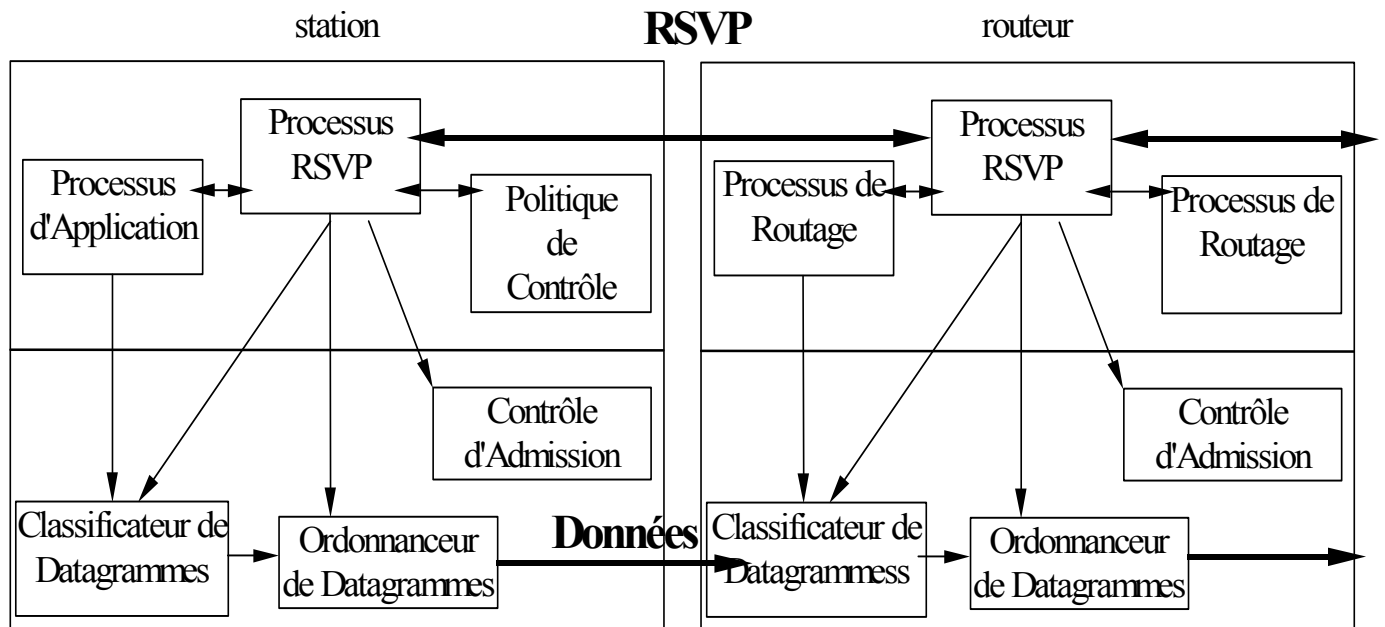
Il s'accompagne d'un modèle de gestion des ressources

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

Architecture pour la signalisation RSVP



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

Remarque : L'API QoS winsock 2 supporte le modèle RSVP.

Types de Réreservation

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

- **service garanti** (pour trafic avec contraintes TR équivalent ATM-CBR ou RT-VBR),
- **service avec contrôle de charge** (best effort amélioré équivalent nRT-VBR ou ATM-ABR) dont la définition est très floue en fait.
- le **Best-Effort** classique existe toujours !

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 (Traffic Specification)**:

- * r, débit moyen ($O_{\text{datagramme IP/s}}$) 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 spécification de taux de perte ni de latence (celle-ci est évaluée pendant le parcours du chemin avec le message PATH).

Le récepteur spécifie le trafic qu'il veut réserver, **Rspec (Resource Specification)** :

- 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 lissage du flux : tente de remettre le flot de données en conformité avec le contrat : utilisation d'un "token bucket" pour la régulation de débit et de buffers

Pas de Fragmentation possible !!!!

Tspec, Rspec d'un flot de données "Qos Charge Contrôlée"

Classe de service QoS Charge Contrôlée :

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

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 nœuds offrant ce type de QoS ont la charge d'éviter toute interférence entre flots.

Le **Rspec** est identique à celui du service QoS garantie.

Dans la suite, nous ne parlerons plus de ce modèle de réservation, car il ne présente pas d'intérêt particulier.

Modèles de Réserveation

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).

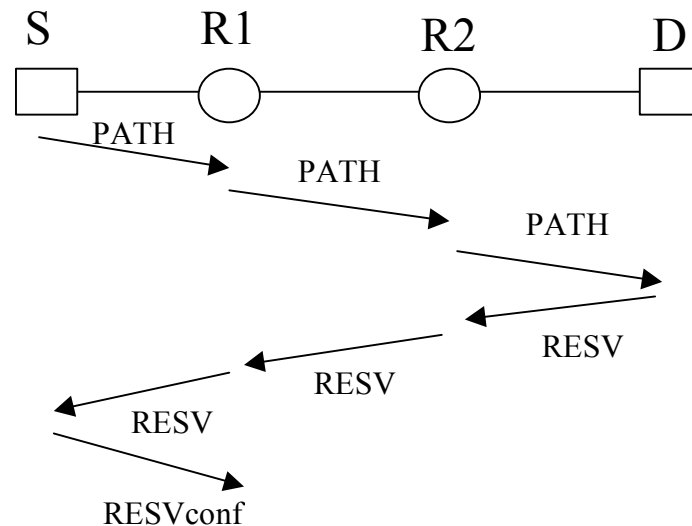
Sélection Emetteur	Mode de Réserveation	
	Distinct	Partagé
Explicite	FF	SE
Ouvert		WF

- Filtre Fixe (FF – fixed filter) - les ressources sont réservées pour le flot uniquement -> unicast et multicast
- Partage Explicite (SE – shared explicit) - 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 provient de plusieurs émetteurs, les flots du même type partagent les mêmes ressources -> multicast, particulièrement intéressant pour le multicast sur arbre partagé.

Principe de RSVP pour la réservation

Le chemin (unicast ou multicast) est établi par l'émetteur, et la réservation effective des ressources nécessaires est effectué 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 relâchés.

L'instabilité du chemin due à la nature du routage à datagramme de l'Internet est un problème pour RSVP.

Contenu d'une requête PATH (1)

Les parties essentielles d'un message PATH, du point de vue de la réservation, sont les parties Adspec et Sender_Tspec. Sinon, un message PATH contient un paramètre Sender_Template (Adresse IP et port de la source), et un paramètre Session (Adresse IP et port de la destination, ainsi que le protocole).

PHOP (Previous Hop, adresse du nœud prédécesseur), cette information est conservée par le routeur sous forme d'information d'état pour mémoriser le prochain routeur sur le chemin de retour lors de la réservation.

SENDER_TEMPLATE,

SENDER_TSPEC (trafic généré par la source non modifié par les nœuds traversés),

- * r, débit moyen ($O_{\text{datagramme IP/s}}$) 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 spécification de taux de perte ni de latence (celle-ci est évaluée pendant le parcours du chemin avec le message PATH).

Contenu d'une requête PATH (2)

ADSPEC (passé au contrôle d'admission local, il représente, au nœud courant, un résumé/somme des ressources disponible en terme de débit et de délai sur le chemin de donné, l'initiateur d'une réservation sur un chemin y insère ses propres infos de capacité), des bits indicateurs :

Partie générale :

- **break bit** indique l'existence d'un routeur qui ne supporte pas RSVP ou l'approche IntServ sur le chemin du flot de données,
- **number of IS hops**, nombre de noeuds implantant l'approche IntServ,
- **available path bandwidth**(octets/s), donne une estimation de la bande passante disponible, la règle de composition de cette information est le minimum entre la bande passante déterminée localement au routeur courant, et la bande passante estimée par ses prédécesseurs sur le chemin et contenue dans ce champ reçu de l'Adspec,
- **Minimum path latency**(μ s arrondi à la centaine de μ s la plus proche), suivant le même principe que pour la bande passante disponible, on dispose d'une estimation au fur et à mesure du délai d'acheminement minimum, ce paramètre tient compte des délais sur les liaisons, des temps de traitement, il n'inclut pas les temps d'attente dans les files (variables par nature),
- **composed MTU** (octets), le MTU en cours d'évaluation d'un chemin, l'objectif est de fournir cette information au récepteur qui ne peut la récupérer par les mécanismes classiques proposés par IP².

² Les mécanismes habituels permettent à l'émetteur, et uniquement à celui-ci, de découvrir le MTU minimum via le "MTU path discovery"

Contenu d'une requête PATH (3)

Partie pour le service QoS garantie :

- **break bit** indique l'existence d'un routeur qui ne supporte pas le service QoS garantie sur le chemin du flot de données,
- **composed Ctot**, sert à quantifier le temps de retard total pris par un datagramme proportionnellement au trafic applicatif à travers tous les routeurs du chemin,
- **composed Dtot**, indique la variation maximale du temps total de transit à travers tous les routeurs du chemin,
- **composed Csum**, sert à quantifier le temps de retard total pris par un datagramme proportionnellement au trafic applicatif à travers tous les routeurs du chemin depuis le dernier routeur (ou le plus proche vis à vis du routeur courant sur le chemin de remontée) qui a re-lissé le trafic, effectivement le dernier routeur ayant fait un lissage a rendu le trafic conforme.
- **composed Dsum**, indique la variation totale maximale du temps de transit à travers les routeurs du chemin depuis le dernier routeur (ou le plus proche vis à vis du routeur courant sur le chemin de remontée) qui a re-lissé le trafic.

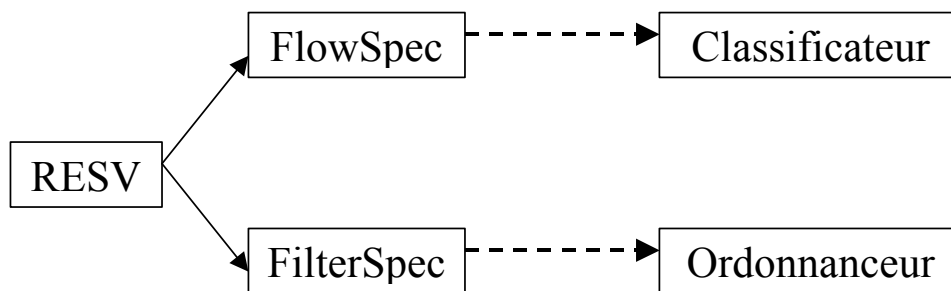
Partie pour le service charge contrôlée : l'information significative est le "break bit" qui indique l'existence d'un routeur qui ne supporte pas le service QoS charge contrôlée sur le chemin du flot de données. Ce service est beaucoup plus simple à mettre en œuvre.

La partie Adspec est optionnelle dans un message PATH. Le premier message PATH qui ouvre un chemin pour un émetteur doit contenir un Adspec. Les messages PATH suivants rafraîchissent le chemin, ils ne sont pas obligés d'en contenir.

Contenu d'un message RESV

Il contient :

- L'ensemble des critères définissant la QoS retenue (FlowSpec) : Rspec + Tspec
- La description du flot (FilterSpec)



Les FlowSpec et FilterSpec sont conservées comme information d'état du flot dans les routeurs.

Réservation pour le service QoS garantie

Le modèle théorique avec QoS garantie fournit un moyen d'évaluer une borne du délai de bout en bout.

Cette borne est :

$$[(b-M)/R * (p-R)/(p-r)] + ((M+C_{tot})/R) + D_{tot} \text{ si } p > R \geq r,$$

et

$$((M+C_{tot})/R) + D_{tot} \text{ si } r \leq p \leq R,$$

où

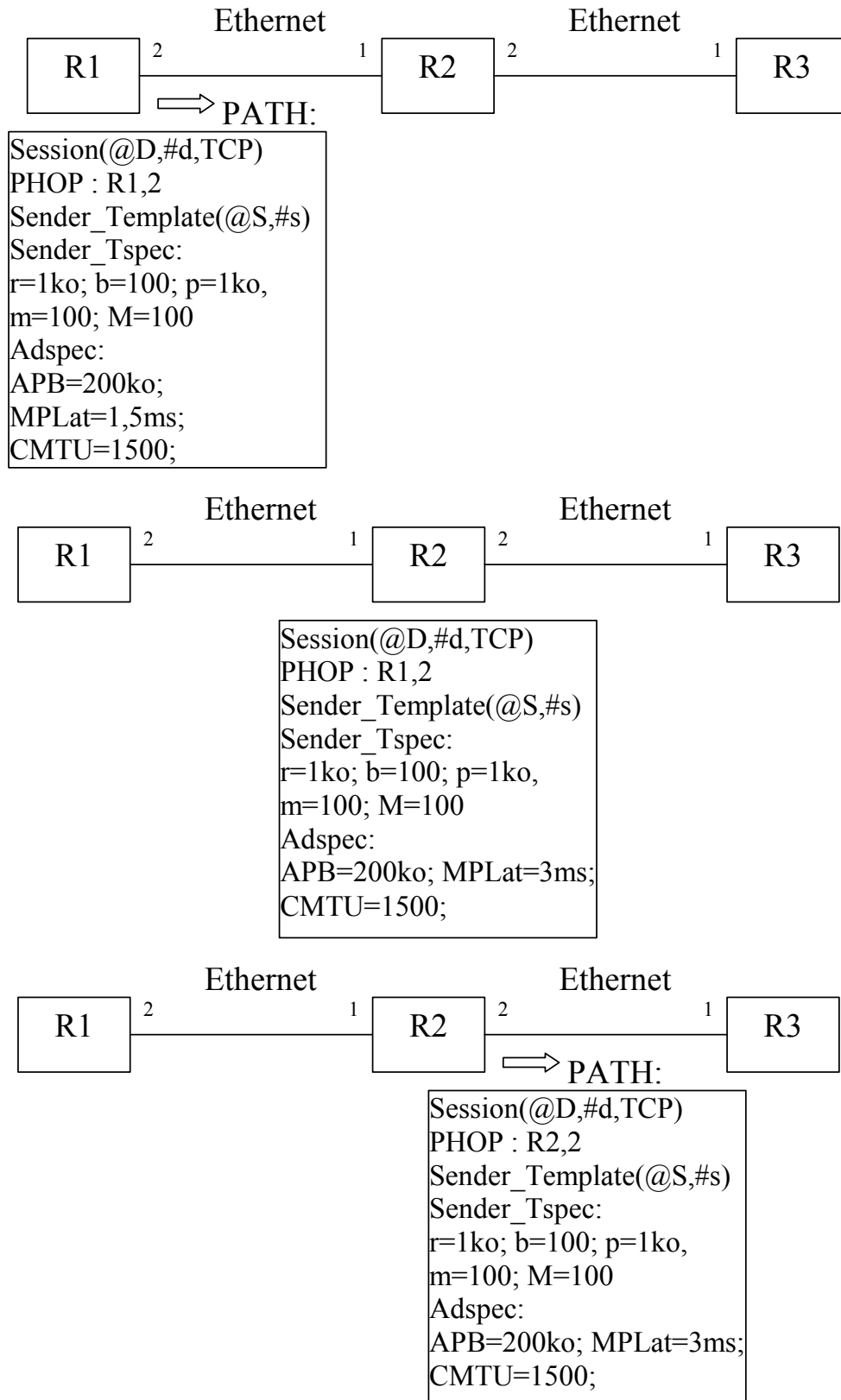
b, r sont des paramètres du token bucket, p, M sont issus du Tspec,

R est issu du Rspec,

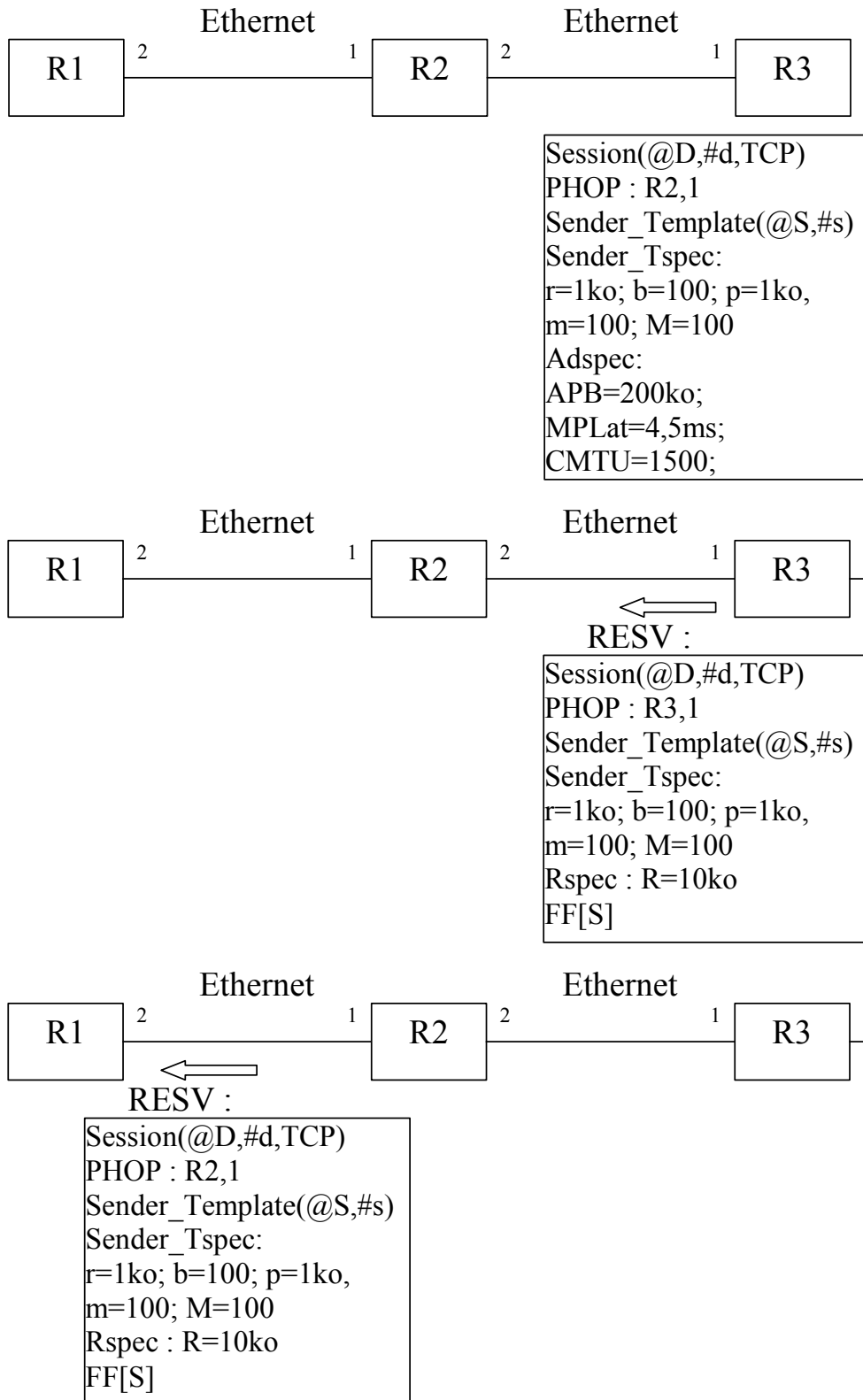
Ctot, et Dtot proviennent de l'Adspec).

La conséquence sous-jacente à cette approche est que si les ressources mobilisées tout le long du chemin d'un flot de données requérant une QoS garantie sont suffisantes, alors le débit de l'application pourra être supporté, les délais seront respectés et il n'y aura **pas de pertes de datagrammes dues à des congestions** aussi longtemps que la source respecte sa spécification de trafic, et, tant qu'il n'y a pas de panne ou de changement de route.

Exemple de fonctionnement de la réservation sur un chemin unicast (1)

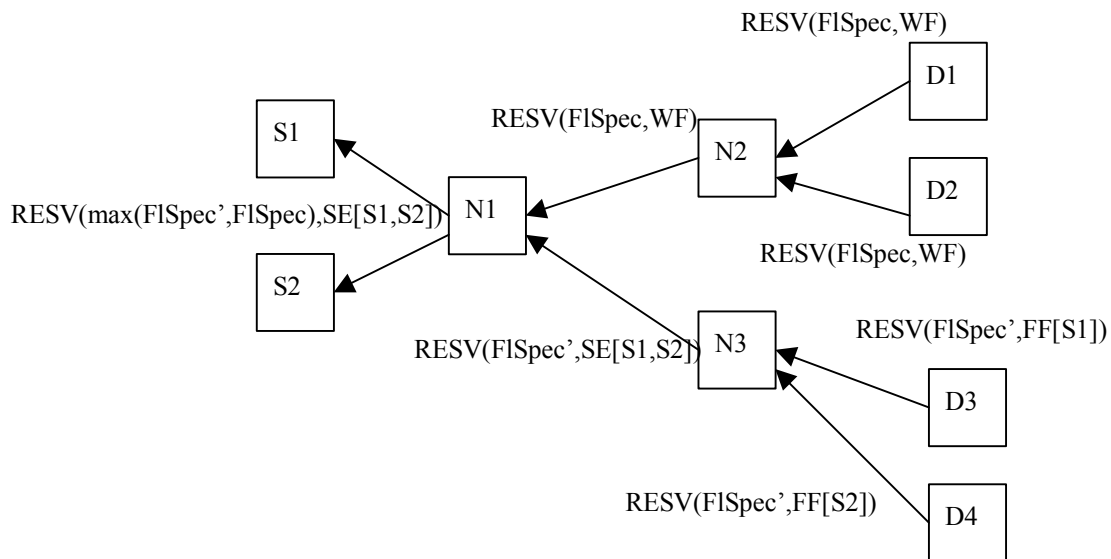


Exemple de fonctionnement de la réservation sur un chemin unicast (2)



Aggrégation des demandes avec filtre sur arbre multicast partagé

Pour un flot de Transport donné (ici se superpose à un arbre couvrant multicast, plutôt un arbre partagé)



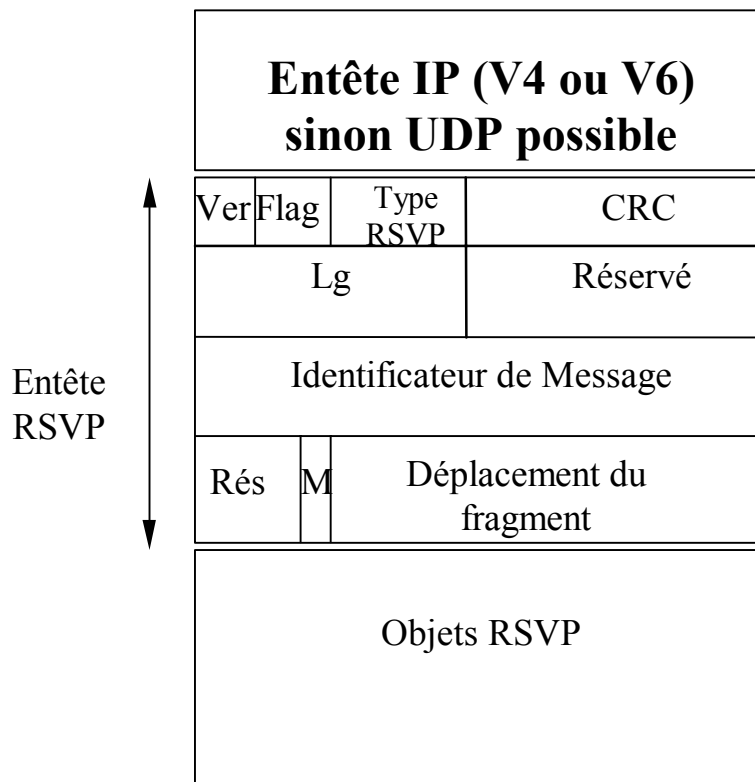
Structure de données de l'API Winsock 2 pour la spécification de QoS

```
typedef struct _QualityOfService
{
    FLOWSPEC    SendingFlowspec; /*flow spec for data sending*/
    FLOWSPEC    ReceivingFlowspec; /*flow spec for data recvg*/
    WSABUF      ProviderSpecific; /*provider specific stuff*/
} QOS;
```

Le Flowspec (FLOWSPEC), contient bien les paramètres d'un Receiver_Tspec, d'un Rspec d'un message RESV et le type de service requis :

```
typedef struct _flowspec
{
    int32      TokenRate;           /* r, In Bytes/sec      */
    int32      TokenBucketSize;     /* b, In Bytes          */
    int32      PeakBandwidth;       /* p, In Bytes/sec      */
    int32      Latency;             /* R, In microsec       */
    int32      DelayVariation;      /* S, In microsec       */
    SVCETYPE   ServiceType;         /* Service Type :       */
                                   /* BEST EFFORT          */
                                   /* CONTROLLED LOAD      */
                                   /* GARANTEED            */
    int32      MaxSduSize;          /* M, In Bytes          */
    int32      MinimumPolicedSize;  /* m, In Bytes          */
} FLOWSPEC;
```

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

Subnet Bandwidth Manager

IntServ sur Réseau Local ! Il n'y a pas toujours des routeurs entre deux hôtes communicants mais des commutateurs de réseaux locaux ou des ponts.

L'approche RSVP peut s'appliquer au niveau 2. Ceci amène au concept SBM (Subnet Bandwidth Manager). Un ensemble de SBM se coordonne pour offrir les QoS supportées par le modèle IntServ.

Contraintes :

- Commutateur supportant 802.1Q/p (sinon pas de priorité donc travail pas de sens)
- Commutateur intelligent, obligation d'avoir tous les composants d'un routeur IntServ dédiés au niveau 2

On retrouve le comportement de RSVP : une réservation des ressources par flot, avec des messages PATH et RESV. La difficulté cette fois réside dans l'intégration de l'approche RSVP dans les niveaux 2 et 3 à la fois. Pour le chemin de retour, on réserve en fonction du nœud précédent l'adresse Mac ou l'adresse IP (au sens ou inclusif si on traverse un routeur).

L'architecture d'un commutateur est alors très proche de celle d'un routeur capable de supporter l'approche IntServ.

Conclusion sur IntServ

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.

Remarque : Deux Modèles sont concurrents : ATM et Internet-RSVP, **Concurrence ou Complémentarité ?** une fusion technologique via l'approche MPLS ?

B. L'approche intégrée IntServ semble trop complexe, et difficile à appliquer sur l'Internet dans son état actuel... applicable éventuellement en intranet.

L'IETF propose une approche différenciée, DiffServ, plus simple à mettre en œuvre et plus efficace, en particulier au niveau des routeurs.

IntServ ou DiffServ ? DiffServ !!!

Nouvelles Perspectives pour RSVP : MPLS

MPLS, c'est d'une certaine façon, deux idées : commutation et SBM, et une technologie de départ, ATM, mises en oeuvre de façon indépendante, entre la couche 2 et la couche 3.

MPLS RSVP-TE

(<ftp://ftp.rfc-editor.org/in-notes/rfc3209.txt>)

RSVP est utilisé pour déterminer les chemins MPLS (Label Switched Paths), les labels de chemin de commutation sont associés à des flots RSVP.

On se sert de RSVP pour établir un chemin entre deux commutateurs MPLS. La réservation de ressource pour faire de la QoS n'est pas obligatoire, quand elle s'effectue, c'est le message de retour qui déclenche la réservation de ressource.

MPLS très brièvement

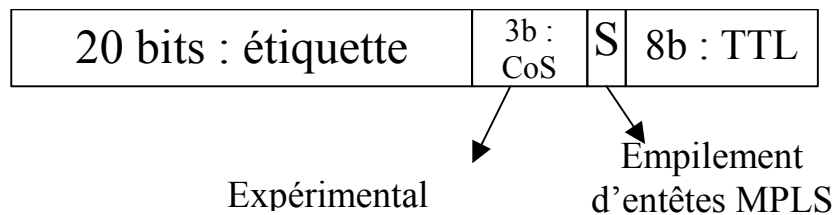
MPLS : MultiProtocol Label Switching

Technique de Commutation entre le niveau 2 et 3.

Procède par marquage par insertion d'une nouvelle entête devant l'entête IP et après l'entête de liaison.

S'applique à IP, mais d'autres protocoles peuvent subir la même mécanique !

Structure d'une entête MPLS de 32 bits :



Les routeurs à la frontière marquent les datagrammes, et les routeurs d'artère routent en fonction du marquage MPLS.

Il faut comparer l'entête MPLS à une étiquette de CV. La gestion opérée par un nœud est identique. La traversée d'un nœud se voit attribuer un nouvel entête en fonction du chemin de sortie emprunté.

Au-dessus d'ATM, les champs VPI/VCI sont utilisés comme entête MPLS.

QoS : Approche DiffServ (Internet 2)

Difficultés de l'approche IntServ

Modèle de réservation simule la notion de circuit virtuel, c'est contradictoire avec le modèle Internet qui est fondamentalement un réseau à datagrammes, donc difficile à gérer.

RSVP gère une réservation par flot, cela représente beaucoup de travail au niveau du classifieur, donc de la surcharge. C'est une approche peu extensive.

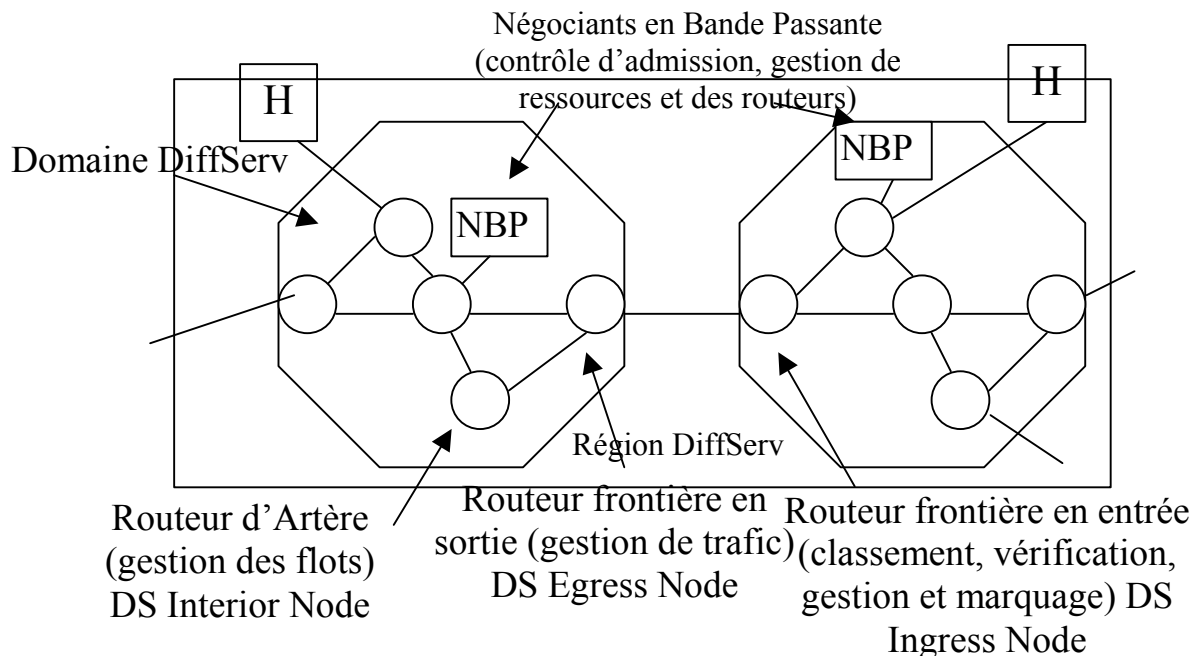
Déploiement difficile de l'architecture IntServ qui nécessite de reconstruire l'Internet en changeant toute l'infrastructure ... tous les nœuds doivent être "IS".

Protocole compliqué, consommateur de ressources (émission périodique de requêtes PATH et de réponses RESV). Difficile d'estimer la bande passante consommée et la bande passante allouable.

Approche DiffServ : Reprendre l'architecture existante (routeurs et protocoles), et la simplifier. Ne plus raisonner en réservation de ressources par flot mais en classes de services qui permettent d'agréger les traitements pour un ensemble de flots.

Architecture d'un réseau DiffServ

RFCs 2474 et 2475



Le champ TOS dont Priority de l'entête Ipv4 est transformé en octet DSCP (Differentiated Services Code Point).

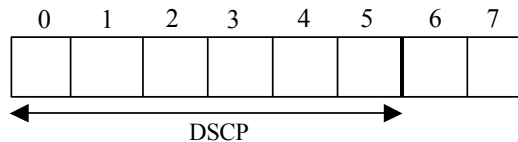
L'administrateur doit classer les besoins des applications réseau (profils réseau). Les applications aident alors le marquage des datagrammes p/r à ces classes qui servent aux routeurs.

Classes de Service de l'IETF pour DiffServ :

- Très haute vitesse (Expedited Forwarding), faibles latence, gigue et taux de perte (aussi qualifié de Premium)
- Vitesse garantie (Assured Forwarding) équivalent à charge contrôlée de IntServ, et qui se subdivise en sous-classes (Or, Argent, Bronze) plus ou moins sensibles à l'élimination de datagrammes en cas de congestion
- Défaut (best-effort), datagrammes qui sont les premières victimes des éliminations en cas de congestion

Champ DSCP et Classification

DSCP, Differentiated Service Code Point : DS Field suivi de 2 bits initilisés (CU, Currently Unused)



Chaque valeur du champ DSCP peut définir une classe ou une sous-classe.

L'espace des valeurs de DSCP est découpé en 3 sous-espaces :

xxxxx0 : 32 valeurs assignées par l'IANA

xxxx11 : usage privé ou expérimental

xxxx01 : usage privé mais semble destiné à l'extension du premier sous-espace

Certaines valeurs ont un rôle bien répertorié :

000000 : best effort, PHB (Per Hop Behavior) par défaut

110000 et 111000 : servent pour les informations de service de l'Internet dont le routage.

Les 3 premiers bits peuvent correspondre aux bits de priorité du champ TOS de l'entête Ipv4, on a alors le format xxx000, mais ceci n'a rien d'obligatoire.

Classes de Service et PHB

Expedited Forwarding (EF) ou Premium, correspond au DSCP 101110, PHB traitement accéléré

Assured Forwarding (AF) ou Olympique, donne différents CodePoints note Afxy : CCCDD0

Précédence à l'élimination en cas de congestion (DD)	Classe 1 (CCC = 001) (or)	Class 2 (CCC = 010) (argent)	Classe 3 (CCC = 011) (bronze)	Classe 4 (CCC = 100)
Faible	AF11=001010	AF21=010010	AF31=011010	AF41=100010
Moyenne	AF12=001100	AF22=010100	AF32=011100	AF42=100100
Forte	AF13=001110	AF23=010110	AF33=011110	AF43=100110

Pour chaque CodePoint, un PHB est défini, en particulier : la classe 1 représente la classe or, la 2 la classe argent, et la 3, la classe bronze.

La RFC 2597, Assured Forwarding PHB Group donne plus de details.

Marquage

Le champ DSCP est considéré comme une étiquette ou une marque qui permet de retrouver un mode de traitement associé à un datagramme portant cette marque, ce mode de traitement ayant été préalablement implanté dans le routeur. La RFC 2474, précise même que cela peut être un index dans une table.

Si un datagramme porte un champ DS qui n'est pas défini dans les tables du routeur, il est traité en PHB Défaut.

Plus globalement, un routeur, en fait la partie classifieur, peut modifier le champ DSCP d'un datagramme. En général, cela est fait à l'entrée d'un domaine DiffServ et à la sortie.

Perspectives

La qualité de service pour le multimédia est un alibi ! Car aujourd'hui, ce ne sont pas ces applications qui sont les plus nombreuses.

Ce pb est plus général et concerne aussi bien les entreprises que les particuliers:

- Qualité de Service entre opérateurs
- Qualité de Service entre l'entreprise et l'opérateur
- Qualité de Service entre le fournisseur d'accès et l'opérateur
- Qualité de Service entre l'utilisateur et son fournisseur d'accès

Le cœur du problème porte sur le contrat de Qualité de Service : SLA, Service Level Agreement dans la littérature, sur sa formalisation, et sur la vérification de son respect par les deux parties