CHAPITRE 2

ÉTAT DE L'ART

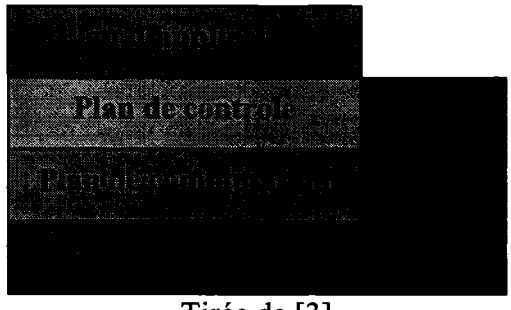
Afin de résoudre la problématique exprimée dans le CHAPITRE 1, qui consiste à faciliter la gestion et la surveillance des mécanismes de QdS dans un réseau hétérogène par le biais d'un outil logiciel, le présent chapitre présente le contexte dans lequel l'outil en question doit être inséré. Les NGN sont abordés en présentant l'approche développée dans le forum multiservices MSF. Par la suite, comme la QdS est un élément indispensable dans les NGN, une introduction technique des différents mécanismes de QdS est fournie. Par ailleurs, étant donné que le projet présenté dans ce mémoire se concentre principalement sur l'aspect « surveillance » des mécanismes de QdS, une section de ce chapitre est consacrée aux différents standards développés pour effectuer de la surveillance de réseaux. Finalement un survol des différents systèmes industriels de surveillance disponibles sur le marché est réalisé afin d'en faire ressortir les lacunes et, du fait même, de faire ressortir la nécessité de développer un système comme QMA.

2.1 Les réseaux de prochaine génération (NGN)

Le forum multiservice (voir http://www.msforum.org) a été fondé en 1998 et est composé principalement de fournisseurs de service et d'équipementiers du domaine des télécommunications. Son principal rôle consiste à développer une architecture ouverte pour les systèmes de commutation multiservice. Il est à noter que le développement réalisé jusqu'à maintenant, a été basé sur une infrastructure réseau pouvant offrir des services multimédia tels la voix ou vidéo sur IP.

Deux architectures ont été développées. L'architecture fonctionnelle présentée dans [3] puis l'architecture physique présentée dans [4]. Il est à noter que l'architecture fonctionnelle a été reconsidérée dans [5] et que les auteurs stipulent dans l'introduction :

« L'architecture de référence s'écarte des entités purement fonctionnelles; tous les éléments de l'architecture peuvent être réalisés en tant que composants physiques séparés; la plupart, sinon tous, sont présentement disponibles en tant que produits commerciaux ». Bien que les auteurs aient reconsidéré l'architecture fonctionnelle, le concept était tout de même intéressant. Cette architecture fonctionnelle est donc présentée dans la Figure 1 et dans le Tableau I. Dans ce dernier, les fonctions qui concernent le système considéré par ce mémoire sont mises en caractères gras.



Tirée de [3]

Figure 1 Plan de l'architecture fonctionnelle des NGN

Tableau I

Plan et fonctions de l'architecture fonctionnelle des NGN

PLAN	FONCTIONS
Plan d'adaptation	a. Formate les données de façon appropriée pour la transmission. b. Peut implémenter certains mécanismes de QdS tel la classification, l'ordonnancement, le *policing* et le lissage.
Plan de commutation	a. Fournit l'interconnexion de base entre ports logiques. b. Transmet les données utilisateur en utilisant des étiquettes. c. Supporte une multiplicité d'éléments de commutation dans le domaine d'un contrôleur.

Tableau I (suite)

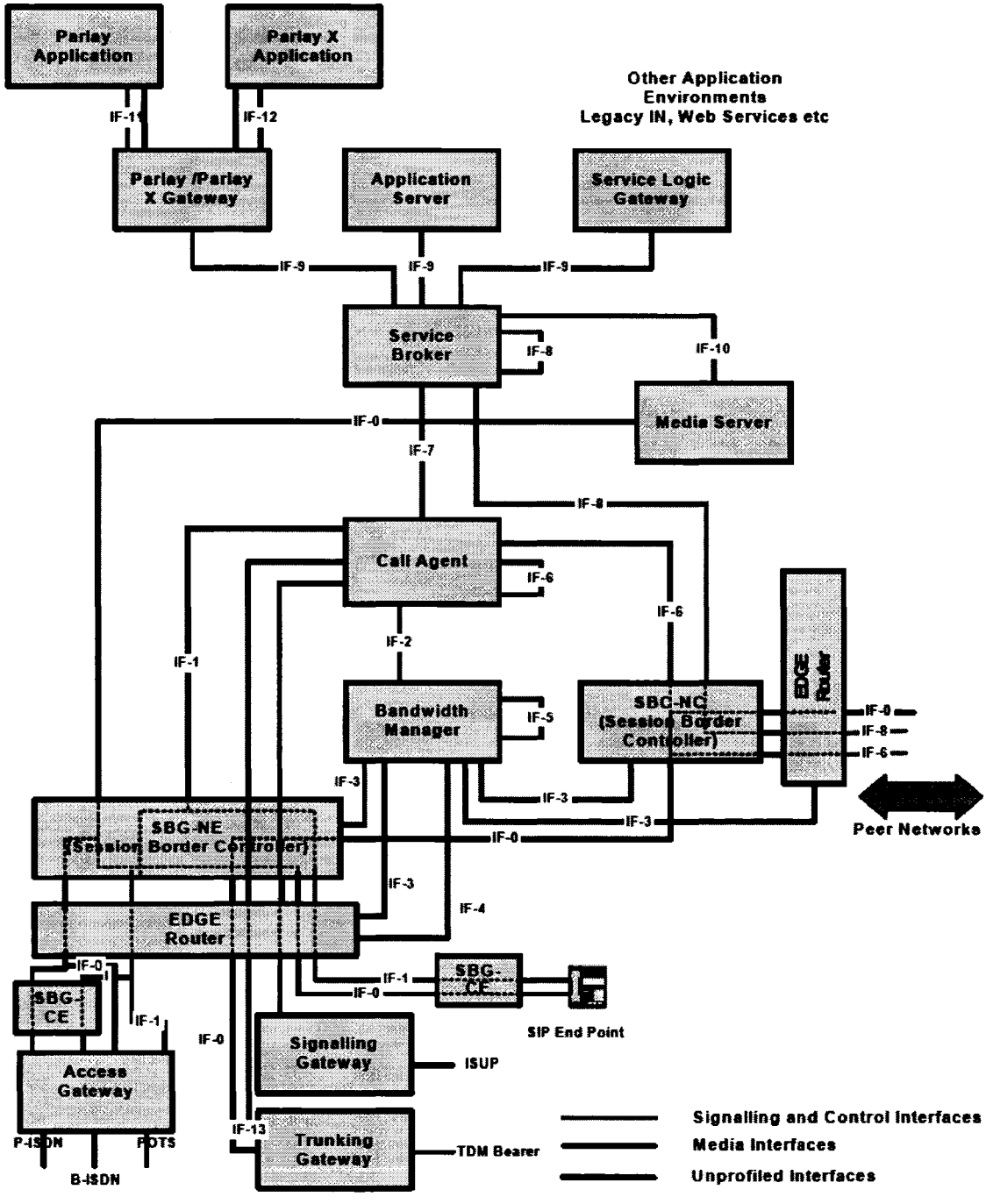
PLAN	FONCTIONS
Plan de commutation (suite)	a. Réplique l'information pour les connexions point à multipoints. b. Fournit une interface au plan d'adaptation. c. Partitionne et partage les ressources d'un commutateur.
Plan de contrôle	a. Achemine le trafic entre les plans d'application, de commutation et d'adaptation et il alloue les ressources du plan de commutation et d'adaptation. b. Contrôle les associations d'étiquettes entre les ports et interfaces via le plan de commutation. c. Commande l'établissement, la modification ou le relâchement des connexions. d. **Assigne les paramètres de QdS** pour chaque connexion ou flot. e. Contrôle les fonctions du plan d'adaptation. f. Fournit une interface au plan d'application. g. Fournit une variété de services accédés par des protocoles de signalisation pour la voix, la vidéo et les données. h. **Effectue le contrôle d'admission et de l'ingénierie de trafic.** i. **Fournit des statistiques et des alarmes.** j. Reçoit l'information de signalisation et l'achemine aux autres entités du plan de contrôle s'il y a lieu. k. Négocie les connexions et les paramètres d'adaptation tel le débit et le type de Codec.
Plan d'application	a. Fournit des services de messagerie tel le courriel et la boîte vocale. b. Fournit des services locaux de signalisation tels l'appel en attente, le transfert d'appel, etc. c. Fournit des services de traitement d'information tel le traitement des cartes de crédit et autres.

Tableau I (suite)

PLAN	FONCTIONS
Plan d'application (suite)	d. Fournit des services d'adressage IP et de noms de domaine tel DHCP, DNS et autres.
Plan de gestion	a. Effectue la gestion des fautes. b. **Effectue la gestion des configurations.** c. Effectue la gestion des comptes. d. **Effectue la gestion des performances.** e. Effectue la gestion de la sécurité.

Tiré de [3]

La Figure 2 présente la seconde version de l'architecture de référence des réseaux de nouvelle génération selon le forum multiservice. Les interfaces et les composantes de cette architecture sont respectivement présentées dans le Tableau II et le Tableau III.



Tirée de [5]

Figure 2 Architecture de référence des NGN selon MSF

Tableau II

Description des interfaces de l'architecture de référence des NGN

Interface	Description	Protocoles supportés
IF-0	Interface média	RTP
IF-1	*User Agent* ou *Media Gateway* vers *Call Agent*	SIP, MGCP, H.248
IF-2	*Call Agent* vers *Bandwidth Manager*	SIP, NRCP
IF-3	*Bandwidth Manager* vers *Edge Router* ou *Session Border Controller*	H.248, COPS-PR
IF-4	*Bandwidth Manager* vers *Core/Edge Router*	Sous étude
IF-5	*Bandwidth Manager* vers *Bandwidth Manager*	NRCP
IF-6	*Call Agent* vers *Call Agent*	SIP, SIP-T
IF-7	*Call Agent* vers *Service Broker*	SIP
IF-8	*Service Broker* vers *Service Broker*	SIP
IF-9	*Service Broker* vers *Application Server*	SIP
IF-10	*Service Broker* vers *Media Server*	SIP
IF-11	*Parlay Gateway* vers *Parlay Application*	PARLAY API
IF-12	*ParlayX Gateway* vers *ParlayX Application*	PARLAYX API
IF-13	*Trunking Gateway* vers *Call Agent*	MGCP, H.248

Tiré de [5]

Il est intéressant de constater, dans le Tableau II, que toutes les interfaces supportent des protocoles déjà existant excepté l'interface entre le *Bandwidth Manager* et les équipements réseau. Cette problématique peut être temporairement résolue en établissant une connexion sécurisée *Secure Shell* (SSH) [6] entre le *Bandwidth Manager* et l'équipement réseau en question. Une fois la connexion établie, le *Bandwidth Manager*

peut envoyer une série de commandes aux équipements en question. Il va de soi qu'il est absolument nécessaire que les équipements supportent SSH.

Tableau III

Descriptions des composantes de l'architecture de référence des NGN

Composantes	Descriptions
Signaling Gateway	Il sert de médiateur entre la signalisation SS#7 du PSTN et le *Call Agent*.
Trunking Gateway	Fournit le transcodage du média entre le réseau TDM externe et le réseau *Internet Protocol* (IP) du fournisseur de service. Il est sous le contrôle du *Call Agent* via MGCP et H.248.
Access Gateway	Supporte les téléphones standards POTS afin qu'ils puissent communiquer dans un réseau IP. Il se situe habituellement chez le client, soit à l'extérieur du réseau du fournisseur de service.
Edge Router	Il achemine le trafic IP dans la dorsale du fournisseur de service. Sous la charge du *Bandwidth Manager*, il est responsable d'effectuer le contrôle d'admission et le *policing*.
Session Border Gateway – Network Edge (SBG-NE)	Il fournit certaines fonctions de bordure telle la translation d'adresse (NAT), la détection et la prévention d'intrusion et il gère également l'association entre les flots RTP et la signalisation.
Session Border Gateway – Network Core (SBG-NC)	Il agit d'une façon similaire au SBG-NE mais il est déployé plutôt à l'interconnexion entre deux fournisseurs de service. Il effectue entre autre du NAT afin de pouvoir mieux cacher la topologie réseau d'un fournisseur.

Tableau III (suite)

Composantes	Descriptions
Session Border Gateway – Customer Edge (SBG-CE)	Il agit d'une façon similaire au SBG-NE mais il est situé chez le client afin de permettre une complémentarité.
Call Agent	Il gère les sessions (établissement, relâchement, etc…) et génère des rapports pour la facturation pour toutes les sessions sous son contrôle. Il demande les services du *Service Broker* et conserve un registre des abonnés de façon statique ou dynamique. Dans le cas où le registre est dynamique, le *Call Agent* doit pouvoir découvrir les autres *Call Agent* où un de ses abonnés s'est connecté.
Bandwidth Manager	**Il gère la QdS dans le réseau.** Il s'occupe d'allouer et de retirer la bande passante à certains flots en plus de gérer l'accès à cette bande passante. Il communique avec les *Edge Router* afin de leur imposer des politiques de service.
Media Server	Il fournit des fonctions aux autres composants réseaux. Ces fonctions peuvent être entre autre de jouer des annonces publicitaires, de détecter et de générer une tonalité, de traiter les fax, d'effectuer du mixage audio, pour les appels conférence par exemple, ainsi que d'autres fonctions.
Service Broker	Il gère l'interaction entre différentes applications, de technologies différentes, au sein d'une simple session.
Application Server	Il fournit l'exécution d'un ou plusieurs services et est orchestré par le *Service Broker*.

Tableau III (suite)

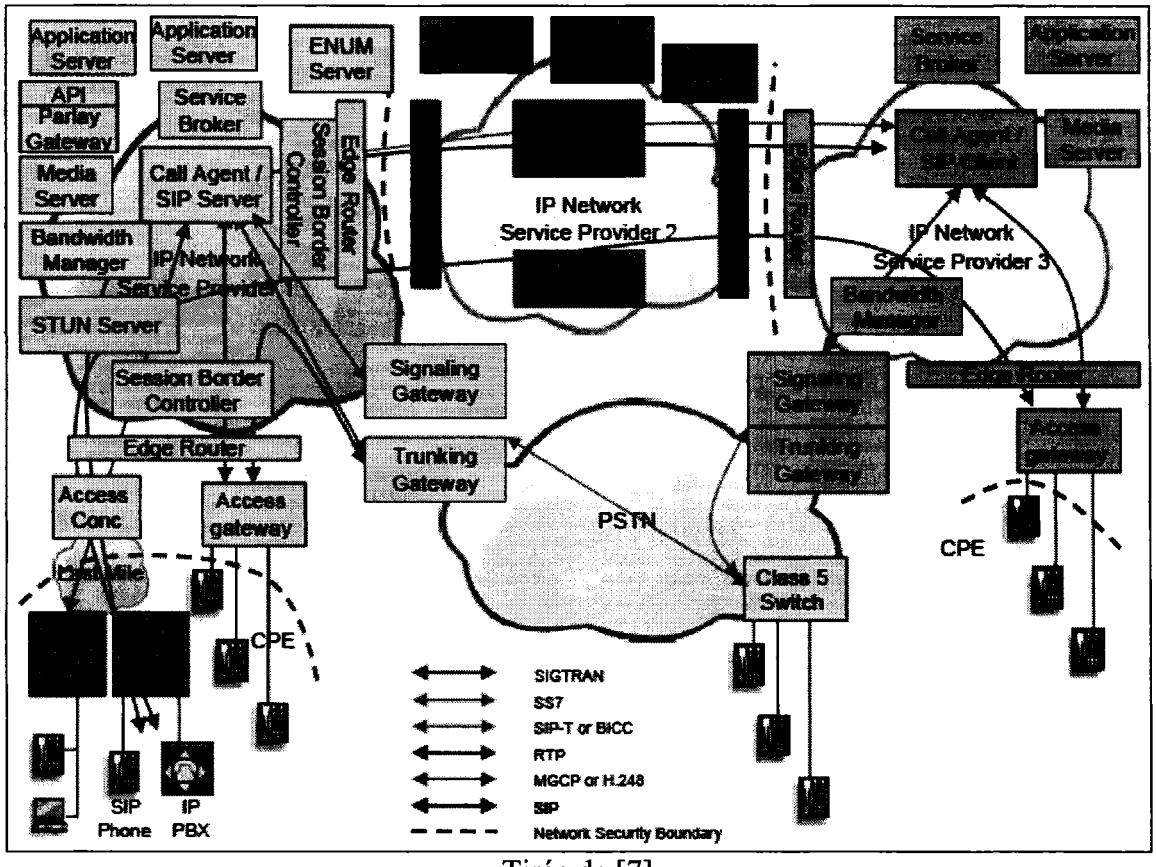
Composantes	Descriptions
Service Logic Gateway	Il permet à une application non-SIP de demander accès aux ressources réseau par le biais du *Service Broker.*
Parlay/ParlayX Gateway[1]	Il s'agit d'un type de *Service Logic Gateway* qui exporte les API Parlay/ParlayX vers les applications.
Parlay/ParlayX Application	Fournit la logique et l'exécution des services et est engagé via le *Parlay/ParlayX Gateway* sous l'orchestration du *Service Broker.*

Tiré de [5]

La Figure 3 présente un scénario de test qui permet de mieux situer les diverses composantes présentées dans la Figure 2. On peut entre autre voir qu'il y a trois fournisseurs de services et que deux d'entre eux ont une connexion au réseau *Public Switched Telephone Network* (PSTN) via un *Signaling* et un *Trunking Gateway.* Par la suite, on remarque que tous les domaines IP distincts sont séparés par un routeur de bordure (ou *Edge Router*). Par conséquent, chaque domaine est en mesure d'effectuer le contrôle d'admission et d'appliquer les politiques de QdS qu'il a définies. Notez que le fournisseur de service 1 fournit un accès IP ainsi qu'un accès aux téléphones *Plain Old Telephone Service* (POTS). Le fournisseur de service 2 est plutôt utilisé comme transporteur d'information entre deux fournisseurs. Quant au fournisseur de service 3, il fournit un accès local à des téléphones POTS uniquement, mais permet l'usage de la signalisation *Session Initiation Protocol* SIP jusqu'au serveur SIP. Les trois fournisseurs de service ont un *Bandwidth Manager* au sein de leur réseau afin de pouvoir gérer la QdS selon les requêtes obtenues. Il agira principalement sur les routeurs de bordure afin de permettre l'admission au réseau des nouvelles connexions multi-médias.

---

[1] Brièvement, il s'agit d'une « API, indépendante aux technologies, qui permet le développement d'applications qui opèrent dans un réseau multiservice. Parlay fournit une interface sécuritaire, mesurable et facturable et a été largement déployé dans les réseaux de télécommunications » (http://www.parlay.org/en/index.asp).



Tirée de [7]

Figure 3 Réseau de VoIP de prochaine génération

Étant donné que la qualité de service est un élément indissociable des NGN, la section qui suit présente une introduction technique aux différents mécanismes de QdS.

2.2 Introduction technique aux mécanismes de QdS

Lorsqu'on parle de QdS, il est essentiel de parler d'abord de l'usager puisque c'est lui qui subit les effets de la bonne ou de la mauvaise qualité du service. Tel que spécifié précédemment dans le CHAPITRE 1, les principaux critères de QdS sont le délai, la variation de délai et la perte de paquets. Dépendamment de l'application utilisée par l'usager, les critères de QdS seront différents. En effet, pour des applications temps réel

comme la voix ou la vidéo, les critères sont beaucoup plus stricts que pour une application de transfert de fichiers comme le *File Transfer Protocol* (FTP) ou la navigation sur le Web par exemple. Ceci vient du fait que certaines applications ont un profil de trafic plus élastique. C'est d'ailleurs le cas avec la majorité des applications qui utilisent TCP comme protocole de transport. Certaines autres applications ont un profil de trafic plutôt constant. C'est d'ailleurs le cas de la voix sur IP (VoIP).

Pour certaines applications, il est plutôt difficile de définir, de façon précise, des paramètres de QdS adéquats. Dans le cas de la VoIP, la perception de la QdS est subjective et varie selon l'interlocuteur. Dans [8], les auteurs se sont basés sur la recommandation P.85 de l'*Internationnal Telecommunication Union, Telecommunication Standardization Sector* (ITU-T) afin de déterminer, de façon plus objective, les paramètres de QdS pouvant représenter, au mieux, l'appréciation subjective d'un groupe test d'interlocuteurs. De façon générale, on peut considérer qu'une communication de voix sur IP doit subir moins de 150 msec de délais de bout-en-bout, une variation de délai maximale de 20 msec ainsi qu'un pourcentage de pertes moyen de 0.25%.

Afin de répondre aux critères de QdS imposés par les usagers, certains mécanismes doivent être mis en place dans le réseau où chaque nœud doit être considéré. Cette section présente donc la perspective des nœuds en décrivant les services et mécanismes de QdS pouvant être adoptés.

2.2.1 Les services IntServ et DiffServ

Cette section présente les deux types de services pouvant être déployés dans un réseau IP afin d'offrir une garantie de services aux différents flots qui y circulent. Dans un premier temps, *Integrated Services* (IntServ) sera présenté puis *Differentiated Services* (DiffServ).

2.2.1.1 IntServ

Le principe fondamental de ce type de service consiste à réserver une bande passante pour chaque flot individuel qui requiert une garantie de service [9]. Ainsi, tout le long du chemin emprunté par le flot, une partie des ressources de chaque équipement est allouée à ce flot. Il va s'en dire qu'un mécanisme de contrôle d'admission est absolument nécessaire afin que les ressources réservées pour un flot soient utilisées uniquement par celui-ci. De plus, la réservation de la bande passante nécessite inévitablement l'usage d'un protocole de réservation de ressources tel que *Resource Reservation Protocol* (RSVP) [10]. Un équipement réseau qui implémente RSVP doit conserver en mémoire l'état de chaque réservation et transmettre périodiquement des messages de rafraîchissement afin de maintenir ces états actifs. Notez que RSVP est conçu pour fonctionner uniquement dans des routeurs mais qu'un autre protocole nommé *Subnetwork Bandwidth Manager* (SBM), proposé dans [11], modélisé dans [12] et amélioré dans [13], a été développé afin de permettre une extension de RSVP dans les réseaux locaux (LAN). Il est intéressant de noter qu'en utilisant un tel type de service, le réseau ressemble de plus en plus à un réseau à commutation de circuits puisque le service est garanti pour un flot donné et que ce dernier a une portion des ressources réservées à son usage uniquement.

À première vue, ce type de service semble adapté et efficace mais il en est tout autrement. En fait, il nécessite une réservation de ressources par flots ce qui n'est pas souhaitable pour un déploiement à grande échelle. En effet, pour un équipement au cœur du réseau par lequel peut transiter des milliers de conversations téléphoniques par exemple, la réservation de ressource pour chacune d'elles ferait chuter les performances de l'équipement. En plus de transmettre ces conversations, l'équipement devrait alors maintenir les milliers d'états en plus de transmettre périodiquement les milliers de messages de rafraîchissement ce qui ferait grandement diminuer les performances de

l'équipement. Pour ces raisons, l'usage de services intégrés n'est pas recommandé dans un large réseau.

2.2.1.2 DiffServ

DiffServ a été principalement développé afin d'éliminer la contrainte présente dans IntServ [14]. En effet, le problème de déploiement d'IntServ est résolu avec DiffServ puisque ce dernier traite les trafics par agrégats au lieu de les traiter individuellement. Son principe de fonctionnement consiste à définir d'abord plusieurs classes (un maximum de 64 classes) dans lesquels seront agrégés plusieurs flots. Notez que les différentes classes servent à différentier les types de flots. Le fonctionnement consiste à classer et marquer les différents flots dès l'entrée dans le réseau. Ainsi la marque appliquée à un flot identifie le comportement que les routeurs suivants doivent appliquer à ce flot. Par conséquent, les équipements au cœur du réseau se baseront sur cette marque pour classer les différents flots de façon appropriée et appliqueront un certain comportement à chaque classe.

Cette façon de procéder est beaucoup plus efficace que l'approche IntServ et ne dégrade pas les performances comme le fait la précédente approche. Cependant, l'usage de DiffServ nécessite l'emploi de divers mécanismes de QdS tel la classification, le marquage, l'ordonnancement, l'évitement de congestion, le lissage ainsi que le *policing* du trafic qui seront tous présentés dans les sections suivantes.

2.2.2 La classification

Afin de pouvoir différentier un type de trafic par rapport aux autres, il est tout d'abord indispensable de le classer. Cette classification est basée sur des critères plutôt variés tel les numéros de port TCP ou UDP, la valeur du champ *DiffServ code Point* (DSCP) de l'entête IP, la valeur du champ *Class of Service* (CoS) de l'entête *Virtual LAN* (VLAN)

ou encore la valeur du champ EXP de l'entête protocole *Multi Protocol Label Switching* (MPLS). Dépendamment de l'endroit où la classification est effectuée dans le réseau, les critères utilisés varieront. Comme par exemple, en bordure du réseau, la classification sera effectuée sur l'information obtenue dans les entêtes des protocoles de couche transport (UDP/TCP) du modèle *Open System Interconnection* (OSI). Dans le cœur du réseau, lorsque les paquets seront routés, la classification sera plutôt basée sur l'entête de couche réseau du modèle OSI (IP/DSCP) tandis que lorsque les paquets seront commutés au niveau 2 (liaison) du modèle OSI, la classification devrait plutôt s'effectuer sur les entêtes de couche 2 tel que MPLS/EXP ou VLAN/CoS. Bien que ces critères soient ceux typiquement utilisés, il est tout de même possible d'utiliser d'autres critères de classification telle que les adresses IP, le type de protocole utilisé ou encore une adresse URL spécifique.

Une fois le trafic classé, il est possible d'y appliquer une action quelconque. Le genre d'action pouvant être effectué est également varié. Il peut entre autre s'agir d'effectuer un marquage bien spécifique, d'effectuer de l'ordonnancement, du lissage, du WRED, etc... La classification peut s'effectuer un peu partout dans le réseau (là où nécessaire) et ce, tant sur le trafic entrant que sur le trafic sortant d'une interface.

2.2.3 Le marquage

Le marquage est une des actions pouvant être appliquée aux différentes classes de trafic. Il consiste à marquer un paquet afin de lui définir un type de service offert. Le type de traitement qui sera attribué à ce paquet, ultérieurement dans le réseau, sera basé sur la marque qui lui a été attribuée lorsqu'il est entré dans celui-ci. Il est donc primordial de donner des marques aux paquets en fonction de ce qu'ils transportent de sorte que ces paquets soient traités selon leurs exigences. Les champs utilisés pour le marquage peuvent être le champ DSCP de l'en-tête IP, le champ CoS pour les VLAN, ou encore le champ EXP de l'en-tête MPLS.

Notez que les champs MPLS/EXP et VLAN/CoS ne sont constitués que de trois bits et peuvent par conséquent définir jusqu'à 8 niveaux de priorité (0 à 7). Le champ IP/DSCP quant à lui utilise 6 bits et peut définir jusqu'à 64 niveaux de priorité (0 à 63). Notez que la RFC 2597 [15] et 2598 [16] ont catégorisé l'usage des bits DSCP en définissant une classe prioritaire *Expedited Forwarding* (EF) ainsi que 4 classes *Assured Forwarding* (AF), puis en définissant 3 priorités de rejet pour chacune des classes AF. Le Tableau IV présente un résumé des noms des *Per Hop Behavior* (PHB) avec la valeur binaire du *Class Selector* et du D*rop Precedence* associés. Cependant, bien que les RFCs [15] et [16] définissent l'usage de ces bits, il revient à l'administrateur du réseau de définir le vrai comportement à adopter pour chacune de ces classes en configurant les mécanismes de QdS de façon appropriée.

Tableau IV

PHB DiffServ avec *class selector* et *drop precedence* associés

PHB	**Class Selector**	**Drop Precedence**	
EF	101	110	
AF41	100	010	Low
AF42	100	100	Medium
AF43	100	110	High
AF31	011	010	Low
AF32	011	100	Medium
AF33	011	110	High
AF21	010	010	Low
AF22	010	100	Medium
AF23	010	110	High
AF11	001	010	Low
AF12	001	100	Medium
AF13	001	110	High

Un dernier point extrêmement important lorsqu'il s'agit de marquage est la définition d'une frontière de confiance. C'est-à-dire définir un équipement à partir duquel les paquets sont marqués de manière fiable. Puisqu'un client pourrait volontairement marquer tous ses paquets à haute priorité afin d'avoir un meilleur service, il est donc nécessaire de remarquer les paquets de manière correcte dès leur entrée dans le réseau et ce, même si ceux-ci possèdent déjà une marque.

2.2.4 L'ordonnancement

Il a été vu précédemment que la classification consiste à trier le trafic et à le regrouper par classes selon le type de chacun. Le mécanisme d'ordonnancement est utilisé afin de gérer les différentes files d'attente (ou classe) du routeur en cas de congestion. Cette technique est principalement utilisée afin de traiter le niveau de priorité de chaque classe et de gérer la bande passante qui leur est allouée. Elle réorganise en fait l'ordre de sortie des paquets afin de leur donner la qualité de service à laquelle ils ont droit. Cette section présentera brièvement les principaux mécanismes d'ordonnancement ainsi que d'autres plus spécifiques à l'équipementier Cisco Systems.

2.2.4.1 Priority Queueing (PQ)

Cet algorithme est l'un des plus simples algorithmes d'ordonnancement [17]. Le principe est illustré à la Figure 4 et consiste à classer d'abord le trafic et à traiter en premier lieu la file d'attente la plus prioritaire. Lorsque aucun paquet n'est présent dans cette file, l'ordonnanceur passe à la file suivante. Si un paquet est présent, il le traite et retourne traiter la file la plus prioritaire. Si aucun paquet n'est présent, il passe à la file suivante. Cet algorithme a l'avantage de pouvoir traiter en priorité les paquets les plus prioritaires et en second lieu les autres. Cependant il comporte comme principal inconvénient le simple fait que si la file d'attente la plus prioritaire n'est jamais vide, les

files moins prioritaires ne seront jamais traitées ce qui peut compromettre la qualité de service des applications moins prioritaires.

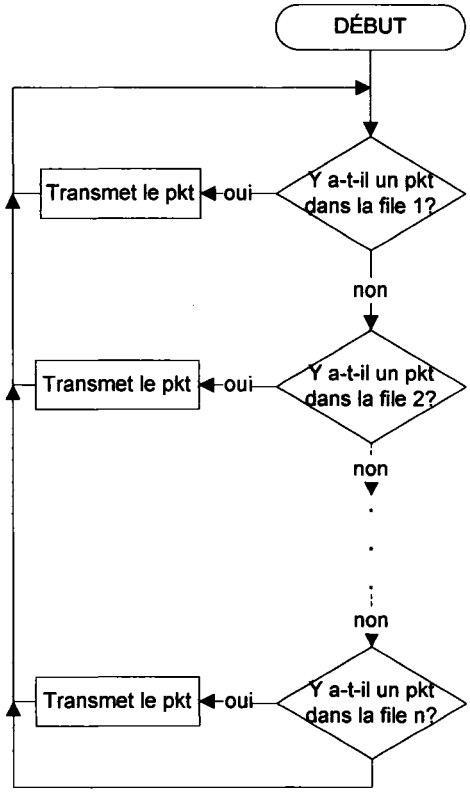


Figure 4 Fonctionnement de l'algorithme *Priority Queueing*

2.2.4.2 Round Robin (RR) et Fair Queueing (FQ)

Ces algorithmes ont été développés afin d'offrir un service équitable entre toutes les files d'attentes. Le principe de l'algorithme RR est simple, il consiste à parcourir chaque classe en boucle, les unes à la suite des autres, et à traiter un seul paquet par classe à chaque itération. Par conséquent, une classe moins occupée peut être traitée même si d'autres classes sont surchargées. Le problème avec ce mécanisme est qu'il ne fait pas de distinction entre les différentes tailles de paquets. Ainsi une classe où les paquets sont

plus grands recevra plus de capacité qu'une classe avec des paquets plus petits. (Voir aussi [18])

L'algorithme *Fair Queuing* [17] fonctionne de façon très similaire mais il a principalement été développé pour l'usage au sein d'un réseau informatique. Il classifie d'abord les paquets en fonction des flots auxquels ils appartiennent. Par la suite, il insère chaque flot dans une file d'attente propre à ce flot. Ensuite chaque file est parcourue octets par octets. Lorsqu'un paquet a fini d'être parcouru, celui-ci est inséré dans la file de sortie (Figure 5). Cette façon de faire est beaucoup plus équitable que le *Round Robin* puisque les classes ayant des plus petits paquets ne sont pas désavantagées. Cependant cet algorithme a le désavantage de ne pas pouvoir définir des poids (ou une bande passante) par classe.

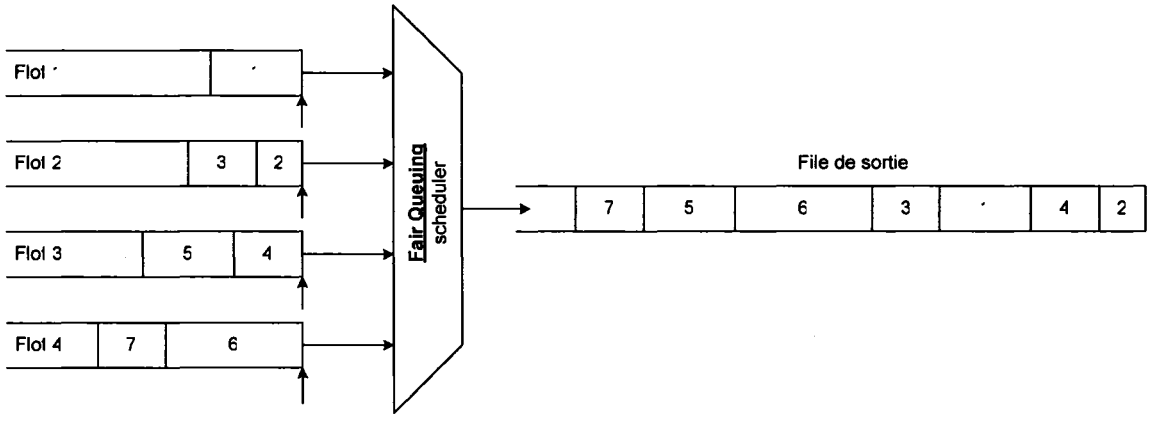


Figure 5 Fonctionnement de l'algorithme *Fair Queueing*

2.2.4.3 Class-Based Queueing (CBQ)

Cet algorithme d'ordonnancement est beaucoup plus adapté aux besoins des fournisseurs de service que les algorithmes précédemment décrits. D'abord il permet à l'administrateur de définir les types de trafic qui vont dans chacune des classes. Par la suite il lui permet de définir un poids pour chaque file d'attente (voir [19]). Par

conséquent, lorsque chaque file d'attente est pleine, la file d'attente j recevra une capacité égale à

$$C_j = \frac{W_j}{\sum_{\forall i} W_i} \times C \tag{2.1}$$

où W_j correspond au poids de la classe j, $\sum_{\forall i} W_i$ représente la somme des poids de toutes les files d'attente et C correspond à la capacité de l'interface. Le principal inconvénient avec cet algorithme est qu'il ne permet pas de prioriser un trafic dont les critères de QdS sont élevés. Cet inconvénient peut être réglé en ajoutant à cet algorithme une file d'attente de type *Priority Queue*. Cet ajout peut cependant comporter les mêmes problèmes soulevés dans la section 2.2.4.1. Cisco a défini un autre type de file d'attente qui sera vu à la section suivante.

2.2.4.4 Low Latency Queue (LLQ)

Une file d'attente de type LLQ (voir [17]) est utilisée pour diminuer autant que possible les délais des paquets mis dans cette file. En utilisant ce type de file, conjointement avec le CBQ, une file ayant une priorité supérieure à toutes les autres est créée. La LLQ utilise le principe du *Priority Queueing* afin que cette file soit vérifiée en premier à chaque fois qu'une place se libère dans la file de sortie. La différence avec le PQ consiste à imposer une limite de bande passante pour cette classe. Ainsi cette file d'attente ne pourra monopoliser toute la file de sortie au détriment des autres files d'attentes.

Tel que représenté dans la Figure 6, la file d'attente contenant les paquets prioritaires (LLQ) sera servie immédiatement puis, lorsqu'elle est vide ou qu'elle a atteint la bande passante maximale allouée, le routeur servira les files suivantes. Les informations contenues dans cette figure ont été prélevée implicitement de [17] mais leur véritable

algorithme n'est pas publié. Cependant, cette figure présente sans doute une bonne approximation du véritable algorithme.

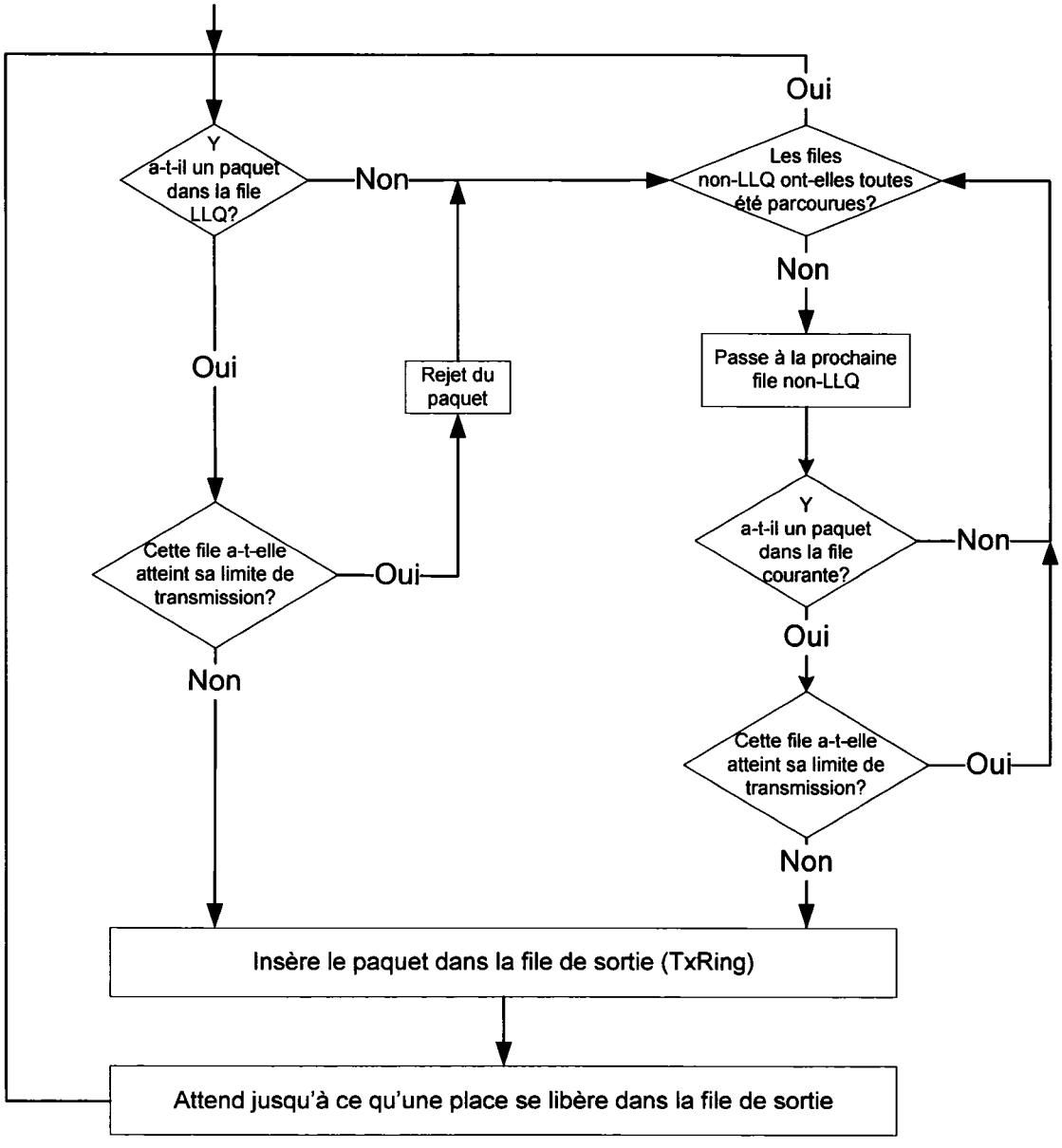


Figure 6 Inter fonctionnement entre une file LLQ et les files CBQ

2.2.5 Le lissage

Cette section présente le mécanisme de lissage utilisé pour réguler le trafic à un débit préalablement fixé. Avant d'expliquer le fonctionnement de ce mécanisme, il est important de connaître le rôle de certains paramètres de configuration.

Tableau V

Définition des paramètres de *shaping* et de *policing*

Terme	Définition
Tc	Intervalle de temps (en millisecondes) auquel la rafale Bc est transmise. De façon générale Tc = Bc/CIR
Bc	Nombre de bits contenu dans chaque rafale.
CIR	Débit défini dans le contrat de service (en bits/seconde).
Be	Nombre de bits, supérieur à Bc, pouvant être transmis ou reçus après une certaine période d'inactivité.

Concernant la rafale Bc, lorsque celle-ci n'est pas configurée, elle est calculée en multipliant Tc par CIR. Il est toutefois possible de modifier l'intervalle de temps Tc implicitement en configurant la rafale Bc. Ainsi, Tc devient égal à Bc/CIR. Notez que de façon générale, une rafale arrive ou sort de l'équipement à un taux égal au taux physique de l'interface. Par conséquent, la rafale ne dure pas nécessairement la totalité de l'intervalle Tc. Un exemple a été réalisé à la Figure 7 pour pouvoir mieux comprendre le fonctionnement du lissage et le calcul des paramètres. Dans cet exemple, le taux de lissage est de 96 kbps sur un lien ayant une capacité de 128 kbps. Notez que seul le paramètre CIR a été configuré. Par conséquent le paramètre Bc a été calculé à partir du CIR et de la valeur par défaut de Tc.

Idéalement le lissage devrait être activé sur le trafic sortant d'une interface afin de limiter le trafic à un débit bien précis. Ce débit pourrait avoir été préalablement établi dans un contrat entre le client et le fournisseur de service. Par conséquent, le routeur émet par intervalle de temps et utilise des tampons afin de réguler le trafic au débit de sortie moyen souhaité. Des délais sont occasionnés aux paquets qui ont été insérés dans le tampon. De ce fait, ce mécanisme peut être problématique pour des applications temps réel comme la voix. Veuillez noter que ce mécanisme est actif uniquement lorsque le débit, devant être transmis, dépasse le débit défini dans le contrat. Dans le cas contraire, les paquets seront transmis directement et le mécanisme restera inactif.

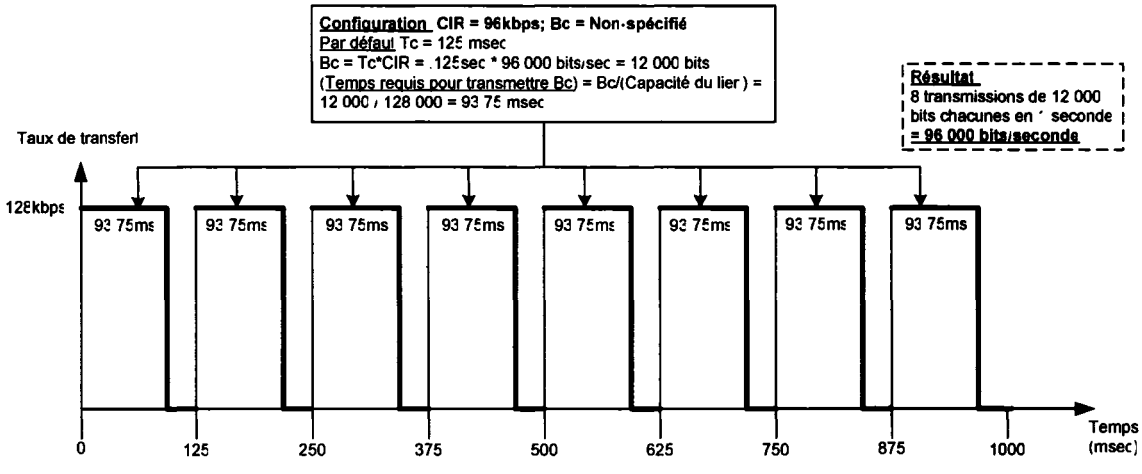


Figure 7 Exemple de calcul des paramètres de lissage

2.2.6 Le policing

Le *policing*, quant à lui, est appliqué sur le trafic entrant une interface afin de s'assurer que ce trafic est conforme au contrat établi. Le fonctionnement des divers paramètres décrits dans le Tableau V (CIR, Bc, Be) est identique dans le cas du *policing*. La façon de les configurer diffère légèrement puisque le *burst-normal* correspond à Bc tandis que le *burst-max* correspond à l'addition de Bc et Be. La principale différence avec le *shaping* est que pour le *policing*, il est possible de poser des actions sur le trafic.

La Figure 8 présente le fonctionnement du *policing* en utilisant l'algorithme *Token Bucket*. Notez que cette méthode de fonctionnement est tirée de [17]. Sachant qu'un jeton équivaut à un octet, à chaque fois qu'un paquet est reçu, un certain nombre de jetons sont ajoutés dans le *bucket* Bc. Si ce *bucket* est plein lorsque les jetons y sont ajoutés, les jetons excédentaires iront dans le *bucket* Be. De façon similaire, si le *bucket* Be est rempli de jetons et qu'il reste toujours des jetons excédentaires, ces jetons seront perdus. Le nombre de jetons à insérer dans le *bucket* est calculé de la façon suivante :

$$\frac{(Current\_pkt\_arrival\_time - Previous\_pkt\_arrival\_time) \times Police\_Rate}{8}$$

où les temps d'arrivés du paquet nouvellement reçu (*Current*) et du précédent (*Previous*) sont en seconde et le taux de *policing* est en bits par seconde. Par conséquent, le tout est divisé par huit afin d'obtenir un nombre de jetons.

À chaque fois qu'un paquet entre dans un *bucket*, un nombre de jetons égal au nombre d'octets dans le paquet est retiré du *bucket*. Ainsi les deux *buckets* peuvent accepter la venue de paquets pourvu que la taille des paquets n'excède pas le nombre de jetons dans le *bucket*. Si un paquet entre dans l'équipement et qu'il y a suffisamment de jetons dans le *bucket* Bc, alors le paquet sera conforme et sera traité comme tel (*Conform Action*). S'il n'y a pas suffisamment de jetons dans le *bucket* Bc mais qu'il y en a suffisamment dans le Be pour accepter le paquet, alors celui-ci est considéré excédentaire et subira l'action destinée à ces paquets (*Exceed Action*). S'il n'y a pas assez de jetons ni dans le *bucket* Bc ni dans le Be, alors ce paquet sera considéré comme un paquet dépassant le contrat et subira l'action adéquate (*Violate Action*). Le genre d'actions portées aux paquets peut être de le transmettre inchangé, le remarquer et le transmettre ou carrément le rejeter. Un scénario adéquat pourrait être de transmettre inchangés les paquets conformes, de remarquer avec une priorité de rejet plus élevée les paquets excédentaires et rejeter les paquets dépassant le contrat.

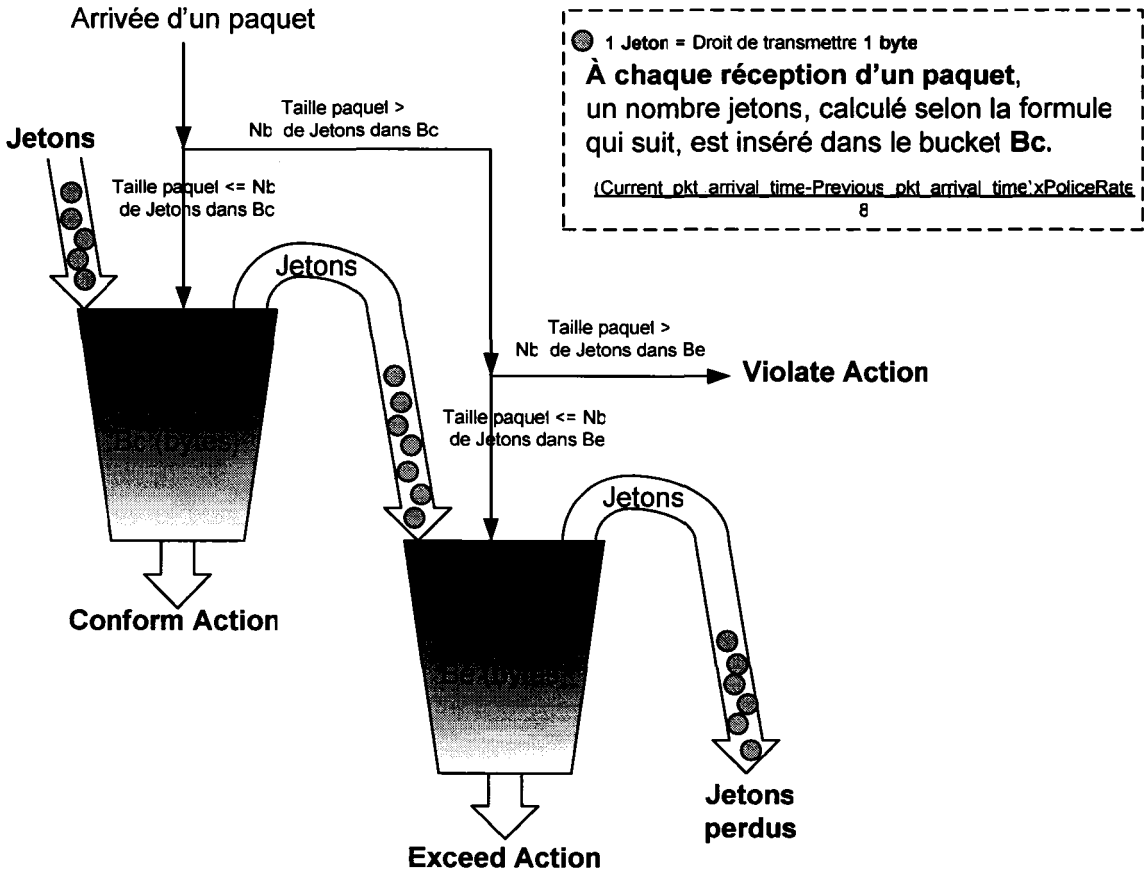


Figure 8 Gestion du *policing* par l'algorithme *Token Bucket*

2.2.7 L'évitement de congestion

Un bon moyen d'éviter la congestion dans un routeur est de l'anticiper. Le RED [17] profite du mécanisme de contrôle de congestion de TCP en rejetant des paquets selon un certain pourcentage. L'émetteur, n'ayant pas reçu d'acquittement durant un certain temps, réduira de moitié son taux de transmission puis recommencera à l'augmenter progressivement. De ce fait, l'application de RED sur du trafic UDP n'a aucun effet significatif sur le niveau de congestion. Il y a trois principaux paramètres à configurer pour le RED. Premièrement ce mécanisme permet de définir un premier seuil (Minimum Threshold), à partir duquel l'équipement commence à rejeter des paquets avec un certain

taux. Deuxièmement, le taux de rejet maximal peut être configuré en modifiant le paramètre *Mark Probability Denominator* (MPD). Ce taux de rejet maximal correspond en fait à 1/MPD. Finalement il est possible de définir le seuil maximal (Maximum Threshold) à partir duquel l'équipement rejettera 100% des paquets. Il est donc logique de configurer ce paramètre à la taille maximale de la file d'attente. Ainsi lorsque la file est pleine, 100% des paquets sont rejetés. Notez que lorsque le niveau de remplissage de la file se situe entre les deux seuils (Min et Max Threshold), le taux de rejet augmentera proportionnellement dépendamment d'où le niveau de remplissage se situe entre les deux seuils. Il devrait normalement atteindre un taux égal à 1/MPD une fois rendu à la limite du seuil maximal.

Par conséquent, le remplissage total des files d'attente est retardé et la congestion est prévenue. La différence entre le RED et le WRED [17] est que le WRED peut s'appliquer sur des classes de trafic afin de rejeter les paquets en fonction de leur niveau de priorité.

Concernant le WRED, il est clair que les paquets qui ont une priorité de rejet (*Drop Precedence*) plus élevée doivent être jetés les premiers. Le seuil minimum associé à une priorité de rejet élevée sera donc inférieur aux seuils minimums pour les paquets avec une priorité de rejet moins élevée et ce, afin que le mécanisme se déclenche plus tôt pour les paquets moins prioritaires. De plus un pourcentage de rejet plus élevé peut également être défini pour les paquets ayant une priorité de rejet élevée. Par conséquent, une classe pouvant contenir des paquets avec le même niveau de priorité mais avec des priorités de rejet différentes, pourrait définir des seuils minimums pour chaque priorité de rejet afin de rejeter les paquets dont la priorité de rejet est la plus élevée en premier.

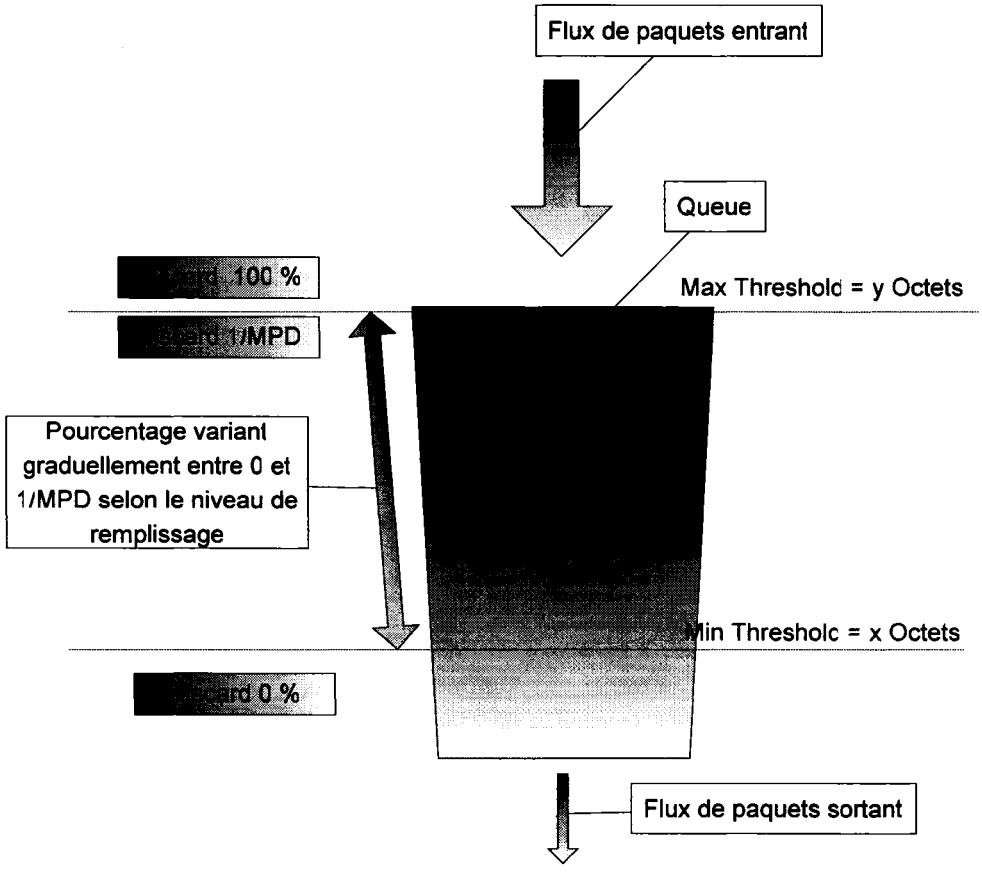


Figure 9 Fonctionnement du RED

2.3 La surveillance d'un réseau

Cette section présente un survol des différents protocoles et/ou des différentes architectures proposées pour effectuer la surveillance d'un réseau de télécommunications, plus particulièrement un réseau IP. On dénote principalement deux standards développés pour effectuer la surveillance d'équipements de télécommunications soit *Telecommunication Management Network* (TMN) et *Simple Network Management Protocol* (SNMP). Dans les deux cas, le principe est basé sur l'architecture Client/Serveur. Dans un couple *Gestionnaire/Agent*, le *gestionnaire* est le système qui interroge les agents situés sur les équipements réseau afin d'obtenir une information quelconque. Le *gestionnaire* est également nommé un *Network*

Management System (NMS). L'*agent* est, quant à lui, situé dans les équipements réseau et est en charge de répondre aux requêtes du NMS. De plus, peu importe le standard utilisé, les agents organisent leurs informations en suivant une structure arborescente nommée *Management Information Base* (MIB). Notez que [20] introduit un filtre permettant l'accélération de la recherche dans une MIB. Plus la MIB est grande, plus l'usage de leur filtre permet une diminution significative des délais de recherche.

Les sous-sections qui suivent présentent une introduction à ces deux standards largement utilisés dans les réseaux de télécommunications. Notez que [21] effectue une comparaison de ces deux standards tout en faisant ressortir les défis de la gestion de réseau dans les réseaux de prochaine génération.

2.3.1 TMN

TMN est un standard défini par l'ITU-T dans la série de recommandations M-3000. Il est basé sur le modèle OSI et a été développé pour gérer les réseaux de télécommunications en général. Bien qu'il ait été développé pour gérer tout type de réseau, son implémentation au sein d'un réseau IP peu s'avérer relativement complexe. Par exemple, la majorité des équipements IP supportent SNMP et non TMN, une passerelle de gestion (également nommée Q *Adapter*) est donc nécessaire afin de faire le lien entre le protocole *Common Management Information Protocol* (CMIP - [22]) et SNMP (voir [23]). Le protocole CMIP est l'équivalent du protocole SNMP dans l'architecture TMN. TMN comporte cinq fonctionnalités soit :

a. La gestion des configurations
b. La gestion des fautes
c. La gestion des performances
d. La gestion comptable
e. La gestion de la sécurité

En référant au Tableau I, on peut s'apercevoir que TMN supporte les mêmes fonctionnalités que le plan de gestion. Le Tableau VI présente, quant à lui, la liste des services *Common Management Information Service* (CMIS - [24]) offerts par TMN. On y retrouve également une courte description de ces services. On y retrouve, entre autres, des services permettant d'obtenir ou de modifier la valeur d'un objet, des services permettant d'obtenir des notifications lorsque certains événements surviennent mais également, des services permettant de créer ou de supprimer des objets.

Tableau VI

Définition des services offerts par TMN

Services	**Descriptions**
M-GET	Ce service est utilisé pour obtenir les valeurs des attributs d'un objet géré. Les paramètres de la portée et de filtrage peuvent être employés afin de sélectionner un groupe d'objets gérés pour lesquels on demande les valeurs des attributs.
M-CANCEL-GET	Ce service est utilisé pour arrêter un service M-GET en cours.
M-SET	Ce service est utilisé pour modifier les valeurs d'un attribut d'un objet géré.
M-EVENT-REPORT	Ce service est utilisé par un objet pour signaler un changement d'état ou une erreur. L'objet génère la notification qui est envoyée par l'agent au *manager*.
M-ACTION	Ce service est utilisé pour effectuer des actions sur un objet géré. Il a été introduit pour permettre à un mécanisme d'effectuer des opérations autres que M-GET, M-SET, M-CREATE. Quand un objet est créé, on définit aussi les opérations qu'on peut effectuer sur lui.
M-CREATE	Ce service est utilisé pour créer une nouvelle instance d'un objet géré. On spécifie aussi les valeurs d'information gérée.

Tableau VI (suite)

Services	Descriptions
M-DELETE	Ce service est utilisé pour effacer une instance d'un objet géré. Les paramètres de la portée et de filtrage peuvent être employés afin d'effacer un groupe d'objets gérés.

2.3.2 SNMP

SNMP est un standard défini par *Internet Engineering Task Force* (IETF), développé pour fonctionner au sein d'un réseau IP en utilisant la pile de protocole du modèle TCP/IP. Il utilise UDP comme protocole de couche transport. Il est le standard le plus largement déployé dans les réseaux IP dû à sa simplicité d'implémentation et d'utilisation. Trois versions ont vu le jour jusqu'à maintenant soit SNMPv1 [25], SNMPv2 [26] et SNMPv3 [27]. Le Tableau VII présente l'évolution des services offerts par SNMP selon la version de protocole utilisée.

Tableau VII

Définition des services offerts par les trois versions du protocole SNMP

Services	Version	Description
GET/GET_NEXT	1	Ces services sont utilisés pour demander à un agent de retourner la valeur d'un attribut d'un objet géré.
SET	1	Ce service est utilisé pour demander à un agent de modifier la valeur d'un attribut d'un objet géré.
TRAP	1	Ce service est utilisé par un objet géré pour signaler un changement d'état ou une erreur. L'agent envoie alors la notification au NMS.

Tableau VII (suite)

Services	Version	Description
GET_BULK	2	Ce service permet le transfert de plusieurs valeurs d'attributs d'objets gérés en un seul transfert.
Sécurité	3	Permet l'authentification et l'encryption de données.

Tel que stipulé dans [28], la première version offre des services de base utilisés pour aller chercher ou modifier les attributs d'un objet ou encore pour obtenir une notification lorsque certains événements surviennent. La seconde version, quant à elle, permet l'obtention d'une plus grande quantité d'information à l'aide d'une seule commande (GET_BULK). Une autre fonctionnalité importante de la version 2 est la gestion décentralisée qui fera ses preuves surtout dans la gestion des réseaux étendus. En fait, la communication NMS à NMS est dorénavant possible avec cette version, permettant ainsi l'établissement d'une structure de gestion hiérarchique. Par conséquent, des NMS pourraient être répartis de façon géographique où chacun gèrera sa portion du réseau et chacun d'eux pourraient communiquer avec un autre NMS père qui aura une vue plus globale du réseau. Après avoir constaté un manque flagrant de sécurité dans les deux premières versions, la troisième version a vu le jour offrant alors un service d'authentification et d'encryption des données.

2.3.3 TMN vs. SNMP

De façon générale, SNMP est beaucoup plus adapté à la gestion des réseaux IP que TMN [21]. Comme le font ressortir les auteurs, d'abord parce que TMN est beaucoup plus complexe que SNMP qui offre une architecture simple et une plus grande facilité d'utilisation. Du point de vue des fonctionnalités, les deux approches sont plutôt similaires à l'exception que TMN est considéré plus sécuritaire que SNMP. Cette différence réside principalement dans le fait que TMN utilise un réseau de gestion physiquement séparé du réseau géré ce qui le rend physiquement sécuritaire

comparativement à un réseau partagé comme IP. En considérant l'aspect multi-vendeur, les deux approches permettent de supporter cet aspect pourvu que les vendeurs implémentent les interfaces adéquates au standard choisi. Globalement, SNMP est préféré à TMN par les vendeurs, dû à la complexité de ce dernier et à la popularité du premier. Du point de vu des communications, TMN utilise la pile de protocoles du modèle OSI tandis que SNMP utilise la pile de protocoles du modèle TCP/IP. En considérant que les protocoles du modèle OSI sont rarement supportés dans les LAN et les réseaux étendus (WAN), SNMP est le choix privilégié. Du point de vue de l'implémentation, SNMP est beaucoup plus simple à implémenter dû à la simplicité du standard. Cependant, [29] fait ressortir qu'il y a eu plusieurs problèmes de vulnérabilité du protocole SNMP qui sont principalement dus à une mauvaise implémentation du standard pour la majorité des fabricants. De plus, ils notent que certaines de ces vulnérabilités persistent dans les dernières versions du standard et que celles-ci peuvent permettre à un attaquant de gagner des privilèges dans le système ou encore qu'ils peuvent causer des conditions de déni de service.

La section qui suit présente un survol des différents systèmes de surveillance présents dans le marché.

2.3.4 Survol du marché

Cette sous-section a été réalisée dans le but de faire un survol du marché des systèmes de gestion de la QdS. Ce marché est plutôt restreint et il n'existe pas encore, à ce jour, de système permettant la surveillance des mécanismes de QdS dans un environnement réseau hétérogène.

2.3.4.1 *QoS Policy Manager* (QPM)

Cet outil a été développé par l'équipementier Cisco System Inc. et permet principalement de définir des politiques de service et de configurer un ou plusieurs équipements selon la politique définie. Il permet également une surveillance des configurations afin d'informer l'administrateur du réseau d'un changement de configuration. De plus, ce système permet la visualisation de statistiques pour les différents mécanismes de QdS. Les statistiques sont récoltées par le biais de la MIB CBQoSMIB qui est propriétaire Cisco. Son architecture est présentée à la Figure 10 où on peut voir qu'il utilise le protocole SNMP pour interroger les équipements du réseau et Telnet et *Common Open Policy Service* (COPS) [30] pour configurer les équipements lorsque requis. QPM fait partie de Cisco Works et permet une gestion centralisée de la QdS.

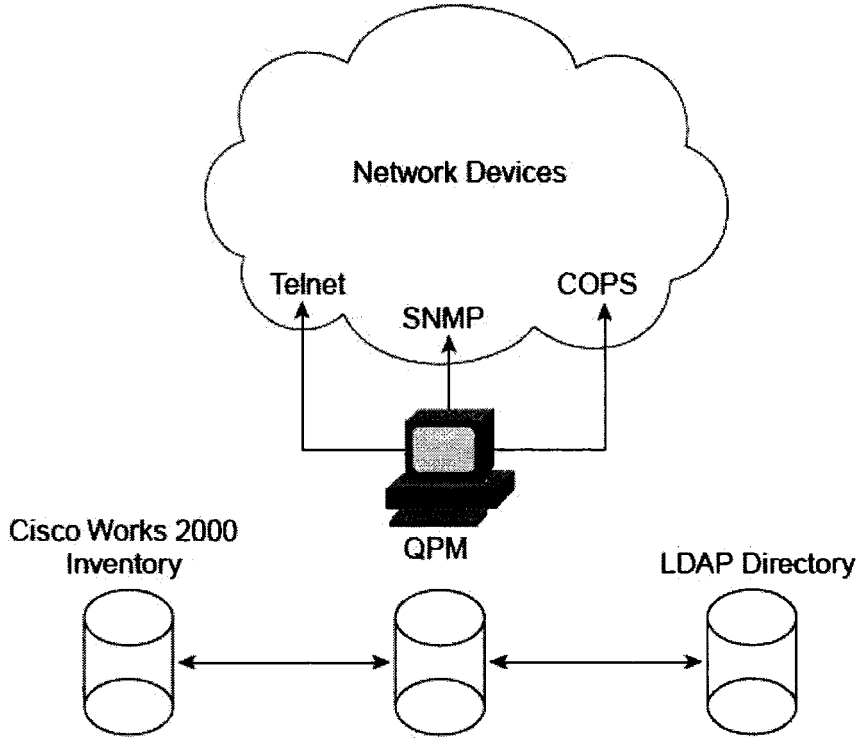


Figure 10 Architecture de QPM (tiré de [17])

Bien que ce système puisse répondre aux critères élaborés dans la problématique, le principal inconvénient avec les outils propriétaires réside dans le fait qu'ils ont été conçus pour un seul et unique équipementier. Par conséquent, ce type d'outil n'est pas tout à fait approprié dans un réseau hétérogène, ce qui est très souvent le cas chez les fournisseurs de service.

2.3.4.2 Netscout nGenious

Cette section présente un système de gestion développé par l'entreprise *NetScout* (http://www.netscout.com/) qui se spécialise dans le développement de système de surveillance réseau. Par conséquent, leur système n'est pas limité à un seul équipementier et peut donc offrir la possibilité de surveiller un réseau hétérogène. *Netscout nGenious* est un système permettant la surveillance des performances d'un réseau de la couche 2 à la couche 7 du modèle OSI. Son principe de fonctionnement est représenté à la Figure 11. Pour l'expliquer brièvement, des sondes sont insérées dans le réseau afin de récolter le trafic qui y circule et en dériver des statistiques sur la performance des liens. Le serveur local récolte les informations afin que le serveur global puisse les afficher, au client, dans un format approprié. De plus, les serveurs locaux peuvent interroger les routeurs et les commutateurs afin de récolter des informations générales par le biais du protocole SNMP.

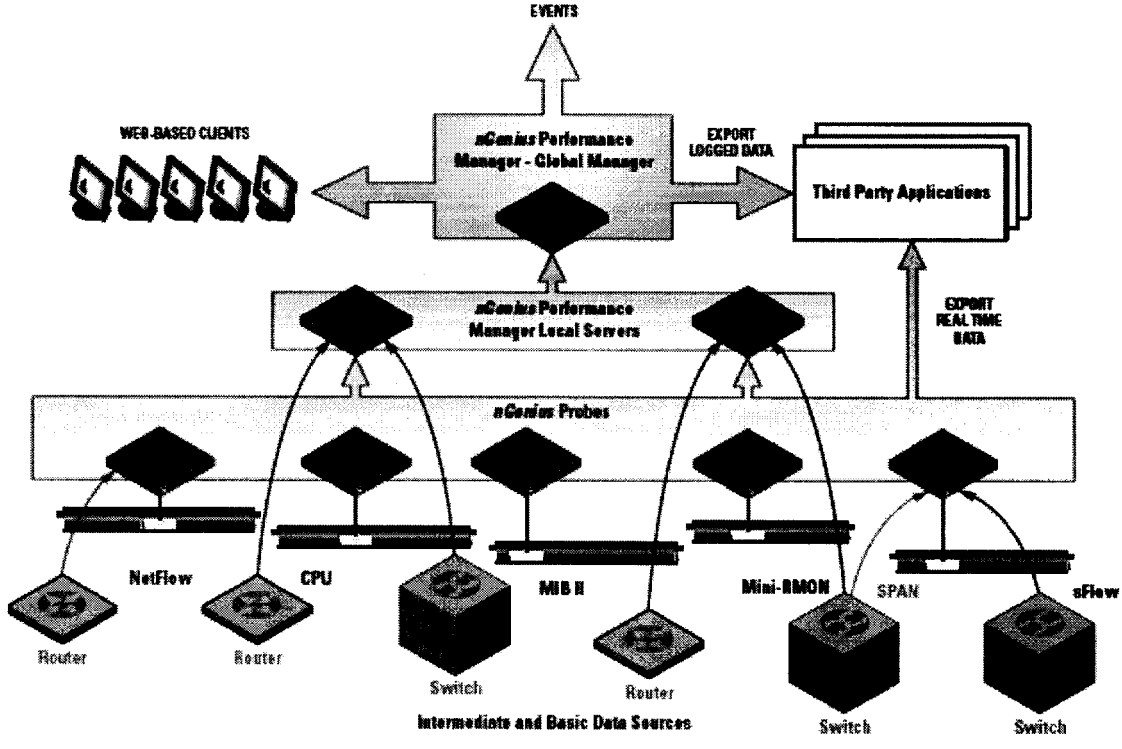


Figure 11 Structure générale de *Netscout nGenious* (http://www.netscout.com/)

Ce système permet de visualiser l'utilisation de la capacité des liens par type d'application et même par utilisateur. De plus, il est en mesure de générer des rapports sur mesure journaliers, hebdomadaire ou mensuels. Cependant, le seul moyen possible pour surveiller la QdS est de créer des filtres IP/DSCP appropriés. Le système présentera alors les statistiques d'utilisation en fonction des filtres et, pour un filtre spécifique, la répartition de la charge entre les diverses applications. Par conséquent, ce système est plutôt mal adapté à la surveillance des mécanismes de QdS puisqu'il ne permet que de connaître le débit associé à chaque classe de service et n'est pas en mesure de donner des statistiques sur les autres mécanismes de QdS tel le lissage, le *policing*, ou l'évitement de congestion. Cet inconvénient est également valable pour d'autres modèles de gestionnaire de performance similaire à celui présenté dans la présente section. C'est d'ailleurs le cas du modèle SCE1000 de Cisco System.

2.3.4.3 WhatsUp et HP OpenView

Les systèmes de gestion WhatsUp de IPswitch (http://www.ipswitch.com/) et OpenView de HP (http://www.openview.hp.com/products/nnm/) sont deux exemples de systèmes complets de gestion. Tous les deux permettent une découverte de la topologie du réseau et permettent une surveillance des performances générales des équipements. De plus, des alarmes sont générées lorsque certains événements se produisent dans le réseau ou encore, lorsque l'utilisation des ressources dépasse un certain seuil.

Concernant la surveillance des mécanismes de QdS, WhatsUp n'intègre absolument pas cet aspect alors que HP OpenView permet la surveillance de la téléphonie IP. Ce dernier surveille les performances des équipements de VoIP en interrogeant principalement le *Call Manager* de Cisco. Ainsi, il peut retirer des statistiques sur les appels ainsi que donner des recommandations pour de meilleures performances et une plus grande disponibilité.

Ces systèmes de gestion permettent tous deux le développement de scripts[1]. Ainsi un utilisateur pourrait construire son propre script afin d'obtenir les informations qu'il désire. Cependant, les opérations pouvant être effectuées avec ces scripts étant plutôt limitées, le développement de scripts pour obtenir des informations concernant les mécanismes de qualité de service n'est pas à considérer. Après une plus grande investigation auprès de ces produits, on peut conclure que l'aspect de qualité de service n'a pas été abordé par ceux-ci.

---

[1] Série d'instructions servant à exécuter une tâche particulière (tiré de http://w3.granddictionnaire.com/).

2.4 Contexte du projet par rapport à l'état de l'art

Comme les NGN consistent en une architecture permettant, entre autre, une gestion de réseau dans un environnement multi-services et qu'un des rôles clé de QMA s'insère dans cette architecture, une part importante de ce chapitre a été consacrée à l'expliquer.

Par ailleurs, comme la qualité de service est un élément essentiel des réseaux multi-services et que ce mémoire se consacre spécifiquement à la surveillance de ces mécanismes, une introduction technique aux mécanismes de QdS ainsi qu'un survol des protocoles et systèmes de surveillance existants ont été réalisés.

Le chapitre qui suit présente la proposition d'un système permettant une surveillance de ces mécanismes. La solution proposée s'insère dans le contexte des NGN. Par conséquent, la place qu'elle y occupe sera décrite.