

**ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
UNIVERSITÉ DU QUÉBEC**

**MÉMOIRE PRÉSENTÉ À  
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE**

**COMME EXIGENCE PARTIELLE  
À L'OBTENTION DE LA  
MAÎTRISE EN GÉNIE ÉLECTRIQUE  
M.Eng.**

**PAR  
ALI HADJ-ABDELHAFID**

**QoS POUR CLASSES VIRTUELLES  
DANS UN ENVIRONNEMENT MULTIMÉDIA**

**MONTRÉAL, LE 10 AVRIL 2002**

**© droits réservés de Ali Hadj-Abdelhafid**

**CE MÉMOIRE A ÉTÉ ÉVALUÉ  
PAR UN JURY COMPOSÉ DE:**

**M. Mohamed Cheriet, directeur de mémoire  
Département de génie de la production automatisée à l'École de Technologie Supérieure**

**M. Michel Kadoch, président du jury  
Département de génie électrique à l'École de Technologie Supérieure**

**M. Jocelyn Desbiens, professeur  
Département de génie électrique à l'Institut National de la Recherche Scientifique**

**IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC**

**LE 18 MARS 2002**

**À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE**

# **QoS POUR CLASSES VIRTUELLES DANS UN ENVIRONNEMENT MULTIMÉDIA**

**Ali Hadj-Abdelhafid**

## **Sommaire**

Des classes virtuelles permettant l'enseignement et l'apprentissage à distance dont l'interaction multimédia en mode de travail collaboratif produisent un volume considérable de transmission audio/vidéo et imposent un défi au réseau à cause des retards et des pertes que peut provoquer la congestion. Pour résoudre ce problème de gestion de qualité de service, nous avons entrepris une recherche basée sur la norme IEEE 802.1p décrivant les marquages du type de service de niveau 2 dans les réseaux locaux et ce, pour assurer une priorité au trafic multimédia. Ce mémoire présente le profil d'un outil de gestion de la qualité de service qui permet l'acquisition et le déploiement dynamique des paramètres de commandes des équipements du réseau.

Nous présentons en revue les possibilités existantes pour donner un aperçu sur ce qui se fait aujourd'hui dans ce domaine en expansion continue. Nous introduisons les différents types d'applications d'échanges multimédia et décrivons les méthodes et les procédures qui nous ont aidé à apporter une solution aux problèmes et exigences posés.

Dans le cadre de cette recherche, nous avons développé un nouveau profil qui fournit une priorité au trafic généré par une application multimédia. Ce profil est intégré au sein d'une application qui permet la création et la gestion des politiques dans le réseau. Dans ce travail, nous en expliquons le mode de fonctionnement et présentons les résultats obtenus lors des tests entrepris pour fin de validation.

La méthodologie que nous avons utilisée correspond à ce qui se fait à ce jour dans le domaine de la recherche sur la gestion de la qualité de service. Nous présentons aussi quelques-unes des recommandations à envisager pour l'amélioration de la méthode et de l'approche.

**Mots clés :** qualité de service, réseaux locaux, classes virtuelles, VLAN, modèle d'information DEN, CIM, politique de gestion, LDAP, IEEE 802.1Q/p, RSVP, SBM, CLI.

# **QoS FOR VIRTUAL CLASSES WITHIN MULTIMEDIA ENVIRONMENT**

Ali Hadj-Abdelhafid

## **Abstract**

Virtual classes which aim to allow remote teaching and learning with their multimedia interaction in collaborative mode produce an important volume of audio/vidéo transmission and impose a challenge on the network because of the delays and the losses caused by congestion. To solve this problem of quality of service management, we undertook a research based on the standard IEEE 802.1p describing markings of the type of service of level 2 in the local area networks, to ensure a priority for multimedia traffic. This research work presents a novel profile for a quality of service management which allows the acquisition and the dynamic deployment of the commands parameters of the network equipment.

Our contribution to this QoS for virtual classes aim to give an outline of what is made nowadays in this field continuously expanding and a review of the existing possibilities. We introduce the various types of multimedia exchanges applications and describe methods and procedures that helped us to bring a solution to the problem and which satisfied the requirements.

Within the framework of this search, we developed a new profile which provides a priority to the traffic generated by a multimedia application. This profile is integrated within an application allowing the creation and the management of the policies in the network. In this work, we explain the operating mode and show the results obtained during the tests.

The methodology we developed corresponds to what is done today in the research field on the management of the quality of service. We present also some of the recommendations to be considered for the improvement of the proposed approach.

**Keywords :** quality of service, local area networks, virtual classes, VLAN, information model, DEN, CIM, policy manager, LDAP, IEEE 802.1Q/p, RSVP, SBM, CLI.

## **AVANT-PROPOS ET REMERCIEMENTS**

**Ce mémoire a été réalisé dans le cadre du projet de télé-enseignement Synchronédia, présidé par M. Mohamed Cheriet, professeur à l'ÉTS, et auquel participe également M. Omar Cherkaoui, professeur à l'UQÀM.**

**J'aimerais exprimer ma gratitude à ces derniers, mon directeur de mémoire, M. Cheriet, et mon codirecteur, M. Cherkaoui, qui ont bien voulu accepter et diriger cette recherche, et dont les conseils ont été enrichissants.**

**Ma gratitude va aussi aux membres du jury pour avoir pris le temps de lire attentivement mon mémoire et pour m'avoir fait part de leurs commentaires très constructifs.**

**Je souhaite remercier l'équipe du laboratoire de téléinformatique de l'UQÀM pour m'avoir accueilli et pour m'avoir fourni les ressources nécessaires à la mise en œuvre de mon projet.**

**Enfin, ma reconnaissance va vers tous ceux qui m'ont aidé dans la réalisation de ce mémoire et en particulier à Kouider Bakhti, pour ses nombreux conseils en matière d'informatique, à Caroline Barré, pour la révision de mon manuscrit, et à ma famille, pour sa patience et ses encouragements.**

## TABLE DES MATIÈRES

	Page
SOMMAIRE .....	ii
ABSTRACT .....	iii
AVANT-PROPOS ET REMERCIEMENTS.....	iv
TABLE DES MATIÈRES .....	v
LISTE DES TABLEAUX.....	viii
LISTE DES FIGURES.....	ix
LISTE DES ABRÉVIATIONS ET SIGLES .....	x
INTRODUCTION .....	1
Situation du Problème.....	5
Objectifs du mémoire.....	7
Organisation des chapitres .....	8
CHAPITRE 1 REVUE DE LA LITTÉRATURE .....	10
1.1 Introduction.....	10
1.1.1 Exemples des applications convergées .....	11
1.1.2 Le besoin en QoS des applications convergées .....	12
1.1.3 Les besoins du trafic du meilleur-effort.....	12
1.1.4 Le trafic sur un réseau moderne et les deux dimensions de QoS.....	13
1.1.5 Comparaison des besoins du trafic à temps réel et du meilleur-effort.....	16
1.2 Cadre de réalisation.....	17
1.3 Méthodes existantes .....	19
1.4 Justification de notre approche.....	21
1.5 Contributions.....	23
1.6 Conclusion .....	23
CHAPITRE 2 NOTIONS DE POLITIQUE, MODÈLE ET NORME .....	25
2.1 Principe d'une politique de gestion.....	25
2.2 Une politique dans une structure réseau .....	26
2.3 Les services et modèles d'information (X500, CIM et DEN) .....	29
2.4 Répertoire LDAP .....	30
2.5 Classe virtuelle.....	31
2.6 Théorie des VLAN.....	32

2.7	Les normes IEEE.....	34
2.7.1	Norme 802.1Q.....	35
2.7.2	Norme 802.1p.....	35
2.8	Priorités dans le trafic.....	38
2.8.1	Définition de CoS et de QoS.....	38
2.8.2	Les options pour CoS/QoS dans des paquets LAN.....	39
2.8.3	Ethernet et IEEE-802.1Q/p .....	40
2.9	Gestionnaire de largeur de bande d'un sous-réseau .....	41
2.10	Conclusion .....	43
<b>CHAPITRE 3 IMPLÉMENTATION DU PROFIL QoS AU MODÈLE DEN .....</b>		<b>44</b>
3.1	Introduction .....	44
3.2	Création de la classe QoS.....	44
3.2.1	Modèle des classes du profil QoS .....	45
3.2.2	Identification des composants .....	47
3.2.3	Méthode d'activation de la politique QoS .....	49
3.2.4	Procédure d'activation du service.....	51
3.2.5	La définition des autres classes LDAP.....	52
3.3	Présentation exhaustive d'un exemple pratique .....	57
3.3.1	Applications de la politique .....	58
3.4	Classe du profil QoS dans LDAP .....	59
3.5	Implémentation orientée objet.....	61
3.5.1	Langage JAVA.....	61
3.5.2	JNDI ( <i>Java Naming and Directory Interface</i> ) .....	62
3.5.3	Intégration au PE des classes <i>JAVA</i> du service QoS.....	64
3.6	Conclusion .....	67
<b>CHAPITRE 4 ARCHITECTURE ET FONCTIONNALITÉS DU MODÈLE.....</b>		<b>68</b>
4.1	Introduction .....	68
4.2	Composition du gestionnaire .....	68
4.2.1	Éditeur de politiques(PE) .....	69
4.2.2	Exécuteur de politiques (PET) .....	69
4.3	Le service QoS dans un domaine administratif .....	70
4.4	Synchronisation.....	76
4.5	Déploiement des ressources .....	77
4.6	Interface graphique des profils d'équipement.....	78
4.6	Conclusion .....	80
<b>CHAPITRE 5 MISE EN ŒUVRE ET EXPÉRIMENTATION .....</b>		<b>81</b>
5.1	Introduction .....	81
5.2	Test d'évaluation au laboratoire.....	82
5.2.1	Composition du modèle .....	82
5.2.2	Serveur LDAP .....	84
5.2.3	Générateur de trafic.....	84

5.3	Paramètres de Configuration et commandes des équipements .....	85
5.4	Les résultats.....	90
5.5	Phases expérimentales sur réseau IP .....	92
<b>CHAPITRE 6 ANALYSE CRITIQUE ET RECOMMANDATIONS.....</b>		<b>96</b>
6.1	Introduction .....	96
6.2	Difficultés rencontrées .....	96
6.3	Analyse critique du travail effectué .....	97
6.4	Analyse critique de l'approche.....	98
6.5	Autres tests d'évaluation .....	99
6.6	Conclusion .....	100
<b>CONCLUSION.....</b>		<b>101</b>
<b>ANNEXE 1 : Définition des objets des classes impliquées.....</b>		<b>105</b>
<b>BIBLIOGRAPHIE .....</b>		<b>110</b>



## **LISTE DES TABLEAUX**

<b>Tableau I</b>	<b>Caractéristiques des applications conventionnelles du meilleur-effort .....</b>	<b>13</b>
<b>Tableau II</b>	<b>Caractéristiques des applications conversationnelles à temps réel.....</b>	<b>16</b>
<b>Tableau III</b>	<b>Acronymes des types de trafic .....</b>	<b>37</b>
<b>Tableau IV</b>	<b>Définition des types de trafic .....</b>	<b>38</b>
<b>Tableau V</b>	<b>Les paramètres des constituants de la figure 18 .....</b>	<b>57</b>
<b>Tableau VI</b>	<b>Les requêtes du protocole LDAP .....</b>	<b>71</b>
<b>Tableau VII</b>	<b>Représente les valeurs des résultats des tests .....</b>	<b>95</b>

## LISTE DES FIGURES

Figure 1	Les deux dimensions de QoS.....	14
Figure 3	Schéma synoptique d'une politique.....	21
Figure 4	En un simple sens, les politiques spécifient une ou plusieurs conditions qui résultent en une ou plusieurs actions.....	25
Figure 6	Modèle d'information DEN.....	29
Figure 7	Niveau2:IEEE 802.1p ajoute trois bits de priorité (tag) a l'entête du paquet	36
Figure 8	Meilleure comparaison et superposition entre les normes.....	41
Figure 9	Modèle de la classe QoS.....	46
Figure 10	Classe <i>QoServiceAction</i> .....	47
Figure 11	Sous-classe <i>QoServiceActionRouter</i> .....	48
Figure 12	Classe <i>QoServiceActionSwitch</i> .....	49
Figure 13	Classe <i>DeviceContainedAux</i> .....	52
Figure 14	Classe <i>GroupOfNames</i> .....	53
Figure 15	Diagramme de la classe <i>RouteurProfile</i> .....	54
Figure 16	Classe <i>ProfileContainedAuxClass</i> .....	54
Figure 17	Schéma global de DEN pour le gestionnaire de politiques.....	56
Figure 18	Activation du service QoS via une politique.....	57
Figure 19	Implémentation des API JNDI.....	63
Figure 20	Type de requête échangée entre le PE et le serveur LDAP.....	70
Figure 21	Interface d'authentification et de connexion.....	72
Figure 22	Message d'authentification.....	73
Figure 24	Création du nouveau service.....	74
Figure 25	Interface d'entrée des paramètres du service QoS.....	75
Figure 26	Fenêtre du <i>Schedule</i> .....	76
Figure 27	Fenêtre du <i>Scheduler</i> .....	77
Figure 28	Activation de politique.....	77
Figure 29	Mise en place des commandes de configuration.....	78
Figure 30	Interface du profil routeur.....	79
Figure 31	Interface du profil commutateur.....	79
Figure 32	Modèle de simulation.....	83
Figure 33	Trafic sans service QoS.....	91
Figure 34	Trafic avec service QoS-a.....	91
Figure 35	Trafic avec service QoS-b.....	91
Figure 36	Découverte des équipements sur le lien ÉTS-UQAM.....	94

## **LISTE DES ABRÉVIATIONS ET SIGLES**

<b>QoS</b>	<b>Quality of Service</b>
<b>CoS</b>	<b>Class of Service</b>
<b>IETF</b>	<b>The Internet Engineering Task Force</b>
<b>ATM</b>	<b>Asynchronous Transfer Mode</b>
<b>TCP</b>	<b>Transmission Control Protocol</b>
<b>CBQ</b>	<b>Class-Based Queueing</b>
<b>WFQ</b>	<b>Weighted Fair Queueing</b>
<b>RED</b>	<b>Random Early Detection</b>
<b>WRED</b>	<b>Weighted Random Early Direction</b>
<b>IP</b>	<b>Internet Protocol</b>
<b>RSVP</b>	<b>Resource Reservation Protocol</b>
<b>UDP</b>	<b>User Datagram Protocol</b>
<b>COPS</b>	<b>Common Operation Policy Services</b>
<b>ISO</b>	<b>International Standard Organisation</b>
<b>ODP</b>	<b>Open Distributed Processing</b>
<b>VLAN</b>	<b>Virtual Local Area Network</b>
<b>DEN</b>	<b>Directory-Enabled Network</b>
<b>DMTF</b>	<b>Desktop Management Task Force</b>
<b>CIM</b>	<b>Common Information Model</b>
<b>LDAP</b>	<b>Lightweight Directory Access Protocol</b>
<b>IEEE</b>	<b>Institute of Electrical and Electronics Engineers</b>
<b>ITU</b>	<b>International Telecommunication Union</b>
<b>CLI</b>	<b>Command Line Interface</b>
<b>RTT</b>	<b>Round-Trip Time</b>
<b>MPLS</b>	<b>Multi Protocol Labeling Switching</b>
<b>SBM</b>	<b>Subnet Bandwidth Management</b>
<b>PE</b>	<b>Policy Editor</b>
<b>PET</b>	<b>Policy Enforcement Tool</b>

## INTRODUCTION

Dans les standards de l'ISO (*International Standards Organisation*), la qualité de service est définie comme un concept spécifiant à quel point les services de réseaux offerts sont «bons» (Tannenbaum, 1992). Les services offerts sont entre autres la connexion à un réseau de télécommunication, le transfert ou la présentation d'un produit multimédia. Les informations de qualité de service sont représentées sous forme de paramètres dits de qualité de service, paramètres qui spécifient les caractéristiques d'un composant, des services et de l'objet des services (par exemple, un document multimédia est l'objet d'un service de transfert).

La qualité de service inclut tous les paramètres du système à savoir ses composants ou ses services qui influencent la présentation des données. C'est une donnée en tant que telle, il est difficile de la définir exactement. Il y a quelques approches qui donnent des définitions générales. Par exemple, dans le modèle de référence de l'ODP («*Open Distributed Processing*»), on définit la qualité de service comme «un ensemble d'exigences de qualité sur le comportement collectif d'un ou de plusieurs objets» (ITU/ISO, 1994). Cette définition semble malheureusement trop générale pour être exploitée.

La qualité de service se définit aussi en terme d'un ensemble de caractéristiques de performance d'un service perçu par l'utilisateur. Elle est alors exprimée dans un langage compréhensible par l'utilisateur et se traduit par un certain nombre de paramètres ayant des valeurs qui peuvent être subjectives ou objectives (pertinence des réponses, temps d'attente). Les paramètres de qualité de service comprennent ceux qui expriment les performances de comportement du système (probabilité de panne et débit, par exemple) et ceux qui expriment d'autres caractéristiques de service telles que la protection (sécurité) ou les priorités. Il y a donc deux types de paramètres :

- **Les paramètres objectifs qui peuvent être directement observés et mesurés là où l'utilisateur obtient le service (le délai ou le débit de transmission par exemple),**
- **Les paramètres subjectifs qui dépendent de l'opinion et des équipements (qualité des haut-parleurs par exemple)**

**Les appréciations sur la qualité de service sont du côté du fournisseur et de celui du consommateur de service. Dans la réalité, les deux opinions pourraient être différentes, chaque partie utilisant sa propre vision. Le fournisseur implante des fonctions spécifiques (classes de transport par exemple), et dans plusieurs cas, il utilise des services additionnels tels que les réseaux de télécommunications et leurs performances individuelles (taux de transmission par exemple).**

**D'autres approches utilisent différents niveaux d'abstractions telles qu'appliquées aux systèmes distribués avec les couches de l'ISO ou les points de vue de l'ODP (Raymond, 1993). Tout en considérant la qualité de service selon ces différents niveaux d'abstractions, nous l'avons identifié dans les différents composants d'un système de communication multimédia : protocoles de transport, systèmes d'exploitation, bases de données, serveurs de fichiers et interface usager.**

**Nous pouvons alors caractériser la qualité de service par les critères suivants :**

- **Critères orientés vers la performance (exemples : délai de transmission, débit du canal),**
- **Critères orientés vers le format; par exemple, la description du format de médium supporté, par les composants qui interagissent (format mpeg pour une vidéo ou ascii pour du texte, par exemple),**
- **Critères orientés vers la synchronisation (par exemple entre l'audio et la vidéo),**
- **Critères orientés vers le coût; par exemple, le coût des connexions ou des équipements.**

La nature continue de certaines données multimédia (vidéo et audio) exige la prise en compte de contraintes temporelles par des mécanismes d'expression qui les garantissent comme :

- Bande passante ou débit maximum : taux de transfert maximal pouvant être maintenu entre deux points terminaux,
- Gigue : variation du débit de bout en bout,
- Délai : temps écoulé entre l'envoi d'un paquet par un émetteur et sa réception par le destinataire,
- Disponibilité : taux moyen d'erreur d'une liaison.

Plusieurs facteurs peuvent avoir un impact sur ces paramètres.

Généralement, ces garanties constituent une qualité de service et sont appelées paramètres de qualité de service. Une façon de répondre à ces besoins est de pré-allouer les ressources nécessaires et de s'assurer qu'elles ne seront pas ré-allouées pendant la durée de la connexion. Il y a alors un besoin de négociations entre les garanties requises et celles qui peuvent être effectivement fournies.

Chaque composant du système multimédia (serveur, clients, système de transport, système d'exploitation, système de gestion de base de données) supporte un certain niveau de qualité de service. Il en est de même pour l'utilisateur voulant interroger le système. Les applications multimédia doivent alors faire abstraction des niveaux de qualité de service supportés individuellement par les composants et les configurer pour une session entière. Il est aussi important de coordonner les composants de l'infrastructure impliqués pour maintenir les niveaux de qualité.

La configuration de la qualité de service s'applique également à l'information de contrôle échangée entre composants. Les messages de contrôle, par exemple, permettent au système de réagir aux événements en temps réels, mais cela demande des garanties de service qui prennent en compte tous les composants.

La majorité des travaux effectués sur la qualité de service se sont concentrés sur la gestion de ressources dans les réseaux. Des ressources telles la largeur de bande et l'espace mémoire sont allouées et contrôlées dans ces réseaux à l'exception de TCP/IP qui permet le partage dynamique des ressources. Lorsque des garanties sont requises, la gestion de ressources utilise des services d'admission pour déterminer la disponibilité des ressources pour une requête donnée. C'est le cas des travaux de Crutcher et Waters (Crutcher, 1996).

Plus précisément la qualité de service est assimilée à la discrimination des services. autrement dit à la définition de classes différenciées. On retrouve principalement des chercheurs du groupe IETF (*Internet Engineering Task Force*) qui ont développés deux modèles (*DiffServ, IntServ*) de qualité de service pour un environnement IP qui utilisent des protocoles pour la réservation et pour la signalisation des ressources de réseaux.

Pour implémenter ces classes ou garantir des ressources, il faut définir une ou plusieurs politique sur les nœuds du réseau permettant d'implémenter la qualité de service demandée, en utilisant divers mécanismes (modelage de trafic, contrôle d'admission, gestion de la congestion, etc.)

En définitive, la gestion des ressources est une tâche importante dans la conception des réseaux. La qualité de service englobe tous les mécanismes permettant de différencier les types de trafic, ceux-ci pouvant être classés et administrés différemment à travers le réseau.

À la lumière des nouvelles avancées dans le domaine des télécommunications et des infrastructures d'équipements réseau et en nous appuyant sur les multiples ouvrages et divers travaux consacrés depuis ces dernières années à l'amélioration des niveaux de la qualité de service, nous présentons dans ce travail un outil permettant de gérer la priorité pour un flux de trafic multimédia. Le service développé permettra d'améliorer les

performances des échanges audio/vidéo entre usagers en offrant à l'administrateur un moyen de contrôle et de commande des constituants de son réseau.

### **Situation du Problème**

Comme cela a toujours été le cas, à l'intérieur de chaque organisation, la priorité se rapporte à l'établissement d'un ordre pour l'utilisation ou le fonctionnement des ressources spécifiques de cette organisation telles que les équipements, les installations.

Dans n'importe quelle organisation, la priorité exige un regard sur la nature de toutes ses activités. Par exemple, les hôpitaux doivent définir une échelle de priorité dans les admissions et les affectations des patients et du personnel. Le code de la route offre un privilège de droit de passage aux véhicules dits prioritaires (ambulance, service civil, police). Les manufacturiers ordonnent leur production selon des priorités établies sur un registre de commandes.

Dans ce monde de hiérarchisation et au titre de ce sujet de recherche, nous avons donc travaillé sur le thème de l'étude et l'implémentation d'un outil de contrôle de qualité de service pour une application multimédia. Nous avons apporté une contribution à l'amélioration de la négociation et du renforcement de la priorité pour le flux de trafic d'un système d'échange et d'interaction multimédia qui va relier différents campus. Notre contribution a consisté en l'introduction d'un nouveau profil de service en l'intégrant à un premier modèle constitué d'une plate-forme de gestion jugé insuffisant et incomplet. Les extensions apportées à ce premier modèle portent essentiellement sur le support et le déploiement dynamique des modes de configuration des équipements, dans le but de fournir à l'utilisateur un service dont la qualité se rapproche le plus possible de ce qu'il souhaite.



Le laboratoire de télécommunication de l'Université du Québec à Montréal, un des partenaires du projet "Synchromedia", s'est fait octroyer le sous-projet qui comporte le volet couvrant l'élaboration et la gestion des infrastructures réseau et leurs composants afin de supporter le déploiement de la plate-forme multimédia.

Au sein de ce laboratoire, une équipe en charge de ce sous-projet a pu développer un gestionnaire de politiques qui est une application permettant l'administration de la gestion de réseaux dynamiquement par le biais de politiques. L'administrateur qui utilise cette application définit tous les équipements qui constituent son ou ses réseaux ainsi que les personnes qui en sont les utilisateurs. Ensuite, l'administrateur n'a qu'à définir un service à implanter (par exemple, un VLAN) avec tous les paramètres nécessaires et par la suite définir la politique qui permettra de savoir quel service déployer, sur quel équipement et quand il sera déployé. Finalement, pour implanter la politique, l'administrateur nécessitera l'utilisation de l'exécuteur de politiques (PET) qui fera un échange via SNMP ou Telnet de la station de gestion à l'équipement réseau.

Dans ce contexte, après qu'un premier prototype de gestionnaire de politiques ait été développé, nous avons pour tâche de proposer des extensions au modèle de données du premier prototype pour introduire la priorité dans le trafic.

Nous avons d'abord procédé à une étude de l'existant qui nous a permis de répertorier toutes les informations nécessaires pour intégrer la qualité de service selon des catégories en analysant les besoins exprimés. L'hypothèse suivante a été émise pour guider notre recherche: «Améliorer la qualité de service signifie offrir un accès privilégié aux paquets du trafic caractérisant les données multimédia».

Nous avons réalisé toute la conception et par la suite l'implémentation du modèle a été complétée par le développement d'interfaces d'entrées des paramètres qui gèrent le

déploiement dynamique sur un réseau de ces informations de configuration des équipements.

Pour arriver à ces résultats, nous avons procédé à une analyse critique du modèle initial du premier prototype en cernant les lacunes et les exigences puis en envisageant toutes les possibilités offertes pour satisfaire les demandes des usagers. Cela nous a permis d'identifier les besoins restant à combler, notamment le choix et la compatibilité des configurations. Nous avons alors consulté la littérature pour ces aspects, ce qui nous a conduit à la proposition d'un modèle qui intègre tout à fait notre plate-forme de gestionnaire de politiques.

Sur la base des résultats obtenus de l'étude de l'existant, de l'étude des modèles et de celle des technologies existantes, nous avons réalisé et implémenté un nouveau service qui assure la priorité au trafic multimédia. Ce service a été implémenté en *Java* et accompagné par des développements d'interfaces appropriées pour la gestion de la configuration des équipements du réseau. Au stade actuel, nous pouvons globalement confirmer notre satisfaction si on considère les besoins de l'application multimédia.

Dans cette section, nous avons défini la nature du rapport entre la qualité du service et une application multimédia. Dans la section suivante, nous présenterons les objectifs de notre recherche.

### **Objectifs du mémoire**

Cette recherche vise essentiellement l'implémentation et la simulation de politique de contrôle de la qualité de service intégré dans une application de gestion du réseau et qui permet la priorisation du flux multimédia caractérisé par des paquets RTP en utilisant la méthode des classes de service (CoS) proscrit dans le protocole 802.1p de IEEE, et d'autre part, en faisant un traçage avec le protocole RSVP par le biais de la technique du

**SBM pour une réservation efficace de la largeur de bande, afin d'obtenir un meilleur service de bout-à-bout (émetteur, récepteur). Notre modèle de simulation sera défini comme un modèle Client/Serveur.**

**Notre modèle est implanté sur des stations de travail qui sont installées dans un laboratoire et a été testé à travers des équipements compris dans un réseau local.**

**Le but ultime des travaux de recherche décrits dans cet ouvrage est de fournir des outils importants à l'administrateur du réseau dans le but de l'assister à bien gérer les ressources de son réseau.**

**Malgré quelques contraintes matérielles et organisationnelles, nous estimons avoir contribué à l'avancement du projet "Synchromédia" en jouant un rôle dans la spécification du protocole et des paramètres de qualité de service et par son intégration au gestionnaire de politiques dans le réseau. Travail intéressant qui nous a permis d'acquérir de précieuses connaissances sur des sujets d'actualité.**

### **Organisation des chapitres**

**Ce mémoire est réparti en cinq chapitres. Le premier chapitre fait l'état des connaissances générales nécessaires à la compréhension du problème de la gestion de la qualité de service. Il précise la notion de qualité de service et le contexte dans lequel nous l'envisageons. Nous donnons un aperçu sur le projet "Synchromedia", décrivons le problème tel qu'il nous a été posé, le cadre de réalisation, la démarche méthodologique adoptée, ainsi que notre contribution. Le chapitre 2 porte sur l'environnement de travail décrivant les notions de politique, protocole et standards qui définissent notre projet de développement et les outils nécessaires pour sa réalisation. Au chapitre 3, nous présentons l'application cible en définissant le profil de notre service, son modèle**

**d'information organisationnel, le volet gestion de qualité de service et son profil. Nous décrivons également les méthodes de son activation par des exemples pratiques.**

**Le chapitre 4 fait état de l'architecture et des extensions que nous avons apportées au premier modèle, nous expliquerons le mode de fonctionnement. Le nouveau modèle est présenté avec les différents aspects implantés, les interfaces conçues et la contribution à l'avancement du vaste projet. Dans le chapitre 5, Il s'agit des tests de mise en œuvre selon les notions de technologie telles que vues dans la littérature mais aussi telles que souhaitées pour les besoins de l'application. Il s'agit également des informations de paramétrages telles que gérées dans les systèmes de gestion des réseaux. Enfin, puisque le projet se poursuit, nous avons écrit le chapitre 6 qui est une analyse critique de notre travail qui précise les difficultés rencontrées, les avantages et les insuffisances non seulement du travail mais aussi de l'approche et sur ce qui reste à faire avec la migration de l'application du gestionnaire de politiques qui continue pour une mise à jour de nouvelles versions de protocoles et de modèles, suivi par une conclusion globale sur notre travail.**

# **CHAPITRE 1**

## **REVUE DE LA LITTÉRATURE**

### **1.1 Introduction**

Le problème à l'étude est celui du développement du profil d'un modèle de service de gestion de la priorité dans un environnement multimédia. Avant de procéder au développement proprement dit, il est essentiel de définir non seulement la qualité de service mais aussi l'environnement dans lequel on en fait usage. C'est l'objectif de cette section.

Dans un premier temps, nous situons cette qualité de service dans le contexte des applications utilisant des informations multimédia et ce, dans un environnement réparti. Nous donnons enfin des aperçus généraux sur des aspects constituant notre contribution à l'amélioration du prototype.

La gestion de la qualité de service suppose au moins deux entités en interaction. Pour notre étude, les entités sont un système multimédia réparti et une application par laquelle l'utilisateur interagit avec le système. Ces entités échangent entre elles des informations multimédia.

De là, on retrouve les applications convergées qui fusionnent les trafics tels la voix, la vidéo, et les données afin d'augmenter la productivité, réduire les coûts, améliorer le service à la clientèle, et augmenter les réponses organisationnelles. Ces applications placent des demandes spéciales sur l'infrastructure réseau, des demandes qui ne peuvent pas être facilement rencontrées par simplement "balancer une largeur de bande" au problème. Heureusement, une nouvelle génération d'équipements de haut calibre, garantissant une gestion de qualité de service, permettra de déployer les applications qui laisseront les entreprises prospérer dans l'âge de l'information.

### 1.1.1 Exemples des applications convergées

Considérez les exemples spécifiques suivants : (VIPswitch, 2000)

- a) **messages unifiés** : Les systèmes des messages unifiés consolident les communications de la voix, de la vidéo et des données. Puisque les usagers peuvent compter sur un système simple pour contrôler toutes leurs communications, ils fonctionnent mieux. Généralement, des systèmes de transmission de messages unifiés sont intégrés avec des applications ordinateur-téléphonie pour améliorer le déroulement des opérations avec des fonctions communes de téléphone telles que les appels de conférence et réponse à un appel.
- b) **vidéoconférence bi-directionnelles** : Les applications de vidéoconférence sont une alternative viable aux voyages coûteux. Elles améliorent également le niveau de compréhension et d'attention des usagers en comparaison avec des appels téléphoniques simples. Les diminutions récentes du prix des appareils de caméra électroniques et du matériel de son rendent les standards des ordinateurs de bureau faciles à modifier pour supporter la vidéoconférence.
- c) **formation à base de vidéo à transmission continue**: Les serveurs vidéo peuvent stocker de l'information de formation par vidéo et les vidéos procéduraux qui aident les employés à acquérir de nouvelles qualifications. La formation à base de vidéo est moins chère et peu exigeante pour l'établissement d'un horaire de programme que les classes traditionnelles, et peut être faite sur demande.
- d) **la voix conversationnelle** : L'utilisation la plus attrayante actuellement des réseaux convergés est le trafic bi-directionnel de la voix. Les compagnies peuvent sensiblement réduire leurs factures d'appels longue distance par un routage du trafic de bureau-à-bureau à travers des réseaux de données. Une réduction des coûts vient aussi du déploiement et de la maintenance d'un câble commun pour le trafic de la voix et des données. En plus, un nombre réduit de circuits de boucles locales est nécessaire, donc le coût est plus bas.

- e) **l'intégration ordinateur-téléphonie** : En intégrant le trafic de la voix dans des applications à base d'ordinateur, des fonctions telles que le programme de maintenance et des systèmes consolidés de répertoire peuvent rendre des employés plus efficaces et réduit le temps nécessaire pour administrer et maintenir le système téléphonique. L'intégration ordinateur-téléphonie améliore les services et permet à une compagnie d'améliorer le pistage de ses systèmes de transmission.
- f) **application interactive partagée** : Les applications interactives comme *digital Whiteboarding* (tableau blanc numérique) changent la manière de travailler en groupe. En plus de raccourcir le temps pour le désigne, ces programmes réduisent la confusion et augmentent le taux du succès du produit (désigne-conception). Au-delà de ça, une compagnie peut permettre à des experts de collaborés sur des projets sans se soucier ni de la distance géographique qui les sépare ni de la tourmente des dépenses de voyage.

### 1.1.2 Le besoin en QoS des applications convergées

Toutes les applications convergées nécessitent un traitement spécial de la part du réseau, que ce soit pour une livraison plus rapide, pour diminuer la variance dans le retard, pour augmenter la fiabilité ou pour assurer des voies d'accès sans congestion à travers le réseau, ces nouvelles applications ont besoin d'une garantie de qualité de service (QoS.)

### 1.1.3 Les besoins du trafic du meilleur-effort

L'Internet a été conçu pour offrir du *best-effort* aux applications de données. Ces applications ont certaines caractéristiques spécifiques décrites dans le tableau I (VIPswitch, 2000). Elles sont relativement faciles à satisfaire, et fonctionnent bien dans les réseaux locaux, métropolitains, et étendus sans modification.

**Tableau I**  
**Caractéristiques des applications conventionnelles du meilleur-effort**

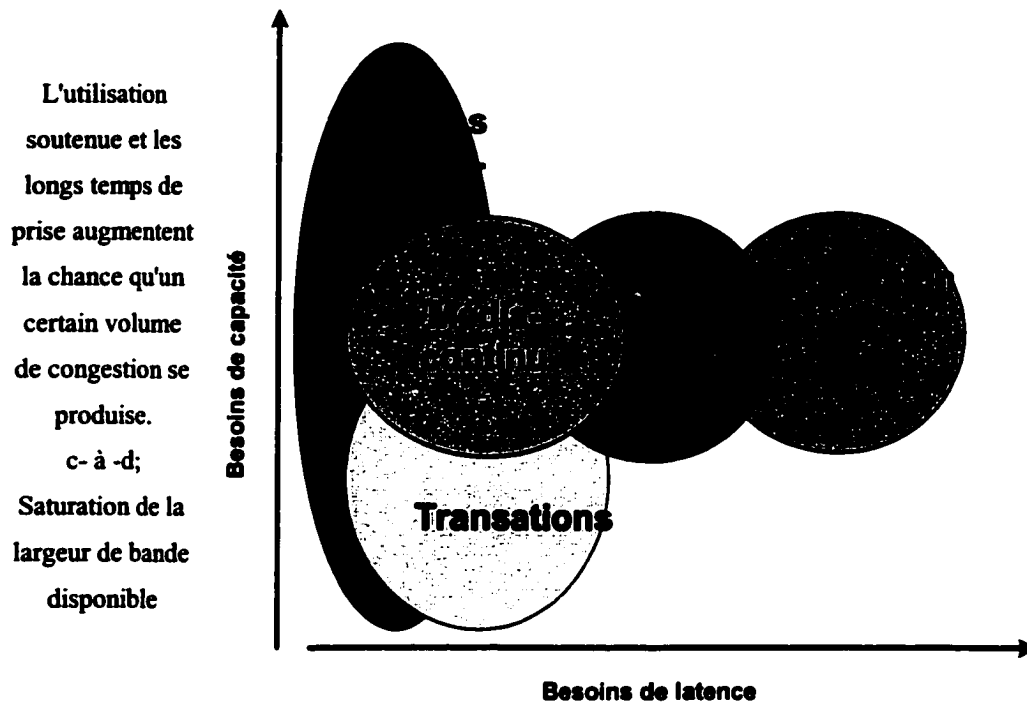
<b>Caractéristiques</b>	<b>Signification</b>
<b>Sensibilité au temps</b>	Si les données sont temporairement retardées, elles sont encore utiles. Un e-mail qui est retardé durant 3 secondes contient encore une information appropriée et utilisable
<b>Produit du trafic à éclat</b>	Demeure typiquement silencieux pour une période étendue (tandis que l'utilisateur lit une page Web ou compose un message) et puis envoie alors un soudain éclat de trafic qui consomme toute la capacité disponible.
<b>Supposer être un réseau à meilleur- effort</b>	L'application suppose que son trafic a une bonne chance d'être livré en même temps que les autres, et prévoit relancer en cas de défaillance réseau.
<b>Présente des retards variables</b>	L'application attend jusqu'à ce que toute l'information soit reçue, dans notre exemple, la lettre entière ou assez d'image Web, avant de l'afficher pour l'utilisateur.
<b>La reprise est relativement courts</b>	Tandis qu'un utilisateur peut surfer sur Internet pendant longtemps, l'interaction actuelle avec un site particulier s'achève en l'espace de quelques minutes ou secondes.

Dans un réseau moderne, les données du meilleur-effort sont seulement une partie du trafic. Une fois soumis aux réseaux du meilleur-effort, les vidéos et audios à transmission continue deviennent perturbés et inutilisables si le trafic est retardé d'une manière imprévisible. Les conversations bi-directionnelles dégradent dans la confusion si les retards bout-à-bout excèdent des seuils spécifiques. Les transactions mission-critiques échoueront si une livraison opportune (à temps) n'est pas garantie.

#### **1.1.4 Le trafic sur un réseau moderne et les deux dimensions de QoS**

Le trafic réseau a travers l'information qu'il véhicule peut être représenté sur deux dimensions : sa sensibilité au temps et la durée de temps où elle affectent le réseau. voir figure 1(VIPswitch,2000).





Points importants :

- Le temps de synchronisation entre l'expéditeur et le récepteur
- Le temps de l'aller-retour
- La déviance "Jitter" en raison du besoin à surmonter les tailles des mémoires tampons

Figure 1 Les deux dimensions de QoS

- a) **données du meilleur-effort** : Le trafic le plus facile à satisfaire est le trafic des données du meilleur-effort. Les données du meilleur-effort sont facilement prises en compte et survivront aux retards ou les rejets.
- b) **continu (streamed)** : Les applications à flux envoient du trafic sur demande. Elles sont à sens unique et le récepteur peut par exemple rechercher et enregistrer plusieurs minutes d'un film avant de commencer à le jouer. En dépit de ceci, les applications à flux ont des besoins spéciaux. Elles injectent une quantité constante de trafic dans le réseau sur une longue période. De quelques minutes (une chanson) à

quelques heures (un film). Dans beaucoup de cas, l'expéditeur ne veut pas que le récepteur enregistre le film ou la vidéo entier pour des raisons de droit d'auteur, ainsi le récepteur ne peut pas simplement télécharger le contenu et le jouer plus tard.

- c) **conversationnel** : Quand la plupart des gens pensent aux prochaines générations d'applications convergées, ils imaginent de la vidéoconférence face-à-face. Le trafic interactif de vidéo et de voix est conversationnelle, signifiant que les deux parties envoient et reçoivent de l'information. Le trafic conversationnel est le plus difficile à déployer dans un réseau à meilleur-effort. Les participants à une conversation sont souvent distants, géographiquement. Tous les deux se rendent intensément compte de tous les retards dans la transmission, puisqu'ils connaissent le sens réel d'une conversation normale. Le trafic dure pendant une période relativement longue dans le cadre d'un réseau, plusieurs minutes, au moins. Selon le type de trafic, les applications conversationnelles consommeront de 4Kbps à 1.5Mbps. Lucent Technologies<sup>1</sup> a déterminé que, pour le trafic conversationnel, une utilisation moyenne de juste 12,7 pour cent dégradera d'une façon inadmissible la vidéo interactive et le trafic de voix sur un réseau qui n'a pas une garantie de qualité de service. Le trafic conversationnel est le plus exigeant en termes de temps d'attente.
- d) **transactionnel** : Les données transactionnelles ne sont pas étrangères aux réseaux. Les blocs de touches pour la carte de crédit et les guichets automatiques exécutent des milliers de transactions par jour. Mais ils se font sur des réseaux qui ont été conçus pour garantir la livraison des messages courts et conversationnels. Avec l'augmentation excessive du e-commerce, on prévoit que l'Internet va manipuler les transactions aussi facilement que ces vieux réseaux. Les données transactionnelles ressemblent aux données conversationnelles, le RTT (*round-trip time*), dans ce réseau, est critique pour le succès de l'application.

---

<sup>1</sup>Compagnie américaine du New Jersey qui innove dans le domaine des réseaux de communication et les données optiques.

### 1.1.5 Comparaison des besoins du trafic à temps réel et du meilleur-effort

Le tableau II décrit les besoins du trafic conversationnel temps réel. Ces caractéristiques sont très différentes de ceux dans le tableau I du meilleur-effort (VIPswitch, 2000)

**Tableau II**  
**Caractéristiques des applications conversationnelles à temps réel**

Caractéristiques	Signification
Sensibilité au temps	Si les données sont retardées au-delà d'un certain temps (la profondeur du mémoire tampon) est inutile.
Générer du trafic à transmission continu avec un débit binaire constant	Il transmet une quantité constante du trafic pour la durée de la session (dans beaucoup de cas), même lorsque rien n'est envoyé.
Suppose un réseau sélectif	L'application peut incorporer quelques méthodes résistantes à la congestion, mais il suppose que le réseau a la capacité suffisante pour délivrer le trafic d'une façon opportune.
Suppose un retard relativement constant	L'application suppose que toute l'information sera traitée dans un même mode. Et sujette a un retard bout à bout semblable et établit la profondeur de mémoire tampon en conséquence.
Des temps de reprise assez longs	La session peut durer des minutes (voix) ou des heures (des applications partagées ou des médias continus.)

Les données du meilleur-effort, la transmission-continue à forte mise en tampon, et le trafic transactionnel peuvent être prises en compte par des applications intelligentes et par un excès de capacité du réseau local (LAN). Cependant, la diffusion à temps réel et le trafic interactif ne peuvent pas être protégés contre le retard en ajoutant simplement de la largeur de bande.

## 1.2 Cadre de réalisation

L'École des Technologies de l'Information (ÉTI, consortium de 4 universités québécoises : l'École de Technologie Supérieure (ÉTS), l'Institut National de Recherche Scientifique (INRS), l'Université du Québec à Montréal (UQAM) et la Télé-Université (TÉLUQ)), se veut à l'avant-garde du développement des applications multimédia réparties. Dans le cadre de ses activités de recherche, l'ÉTI, première institution universitaire "virtuelle" au Québec, a mis à profit l'expertise en formation à distance de ses institutions partenaires pour diffuser plusieurs de leurs cours par l'entremise du Web en mode *asynchrone*, c'est-à-dire en mode différé et en misant sur le courriel pour accroître l'interaction. Cette stratégie a mené au développement rapide de ces programmes et à un recrutement croissant d'étudiants. Toutefois, les étudiants dispersés en région ont vite ressenti le besoin de modes d'échanges plus directs et plus interactifs entre eux et leurs professeurs. Ce constat a été le fondement pour lancer l'idée du cadre du projet "Synchronédia". Son but est d'offrir des services d'enseignement, des simulations d'expériences en laboratoire, de transfert d'activités et la manipulation d'objets basés sur l'interaction en ligne via les technologies Internet afin de satisfaire une demande de plus en plus insistante de la part des milieux universitaires et professionnels. Ces services procurent plusieurs avantages (économie de temps, d'argent, accessibilité...). Le projet "Synchronédia" s'engage dans cette perspective en créant un environnement multimédia par la création d'une plate-forme qui implante une application de téléconférence vidéo (généralement une image de petite taille et un taux lent de rafraîchissement) permettant de concrétiser les classes en ligne. De cette manière, chaque étudiant devient un membre actif du cours.

Il ne faut pas non plus mettre de côté le fait qu'avec ce genre d'application, il sera facile d'intégrer des médias de toutes sortes comme les graphiques 3D, les flux de texte, de son et/ou de vidéo.

La création et le développement d'une telle application permettant la gestion sur Internet de centaines et même de milliers d'étudiants en même temps répartis sur différentes classes nécessiteront le déploiement de ressources réseaux diverses et onéreuses. Delà la nécessité d'un réseau fiable qui constituera le support physique sur lequel toutes les politiques de gestion seront déployées, ce qui nous conduit à l'étape suivante pour présenter le réseau de l'Université du Québec.

### Environnement réseau

L'Université du Québec est composée de dix institutions, lesquelles couvrent toutes les régions de la province du Québec. Un des buts de cette université est de donner un accès à tous les résidants (urbains, métropolitains et semi-urbains). Chaque institution est connectée par un réseau ATM, voir la figure 2 (Cherriet, Cherkaoui et al, 2000).

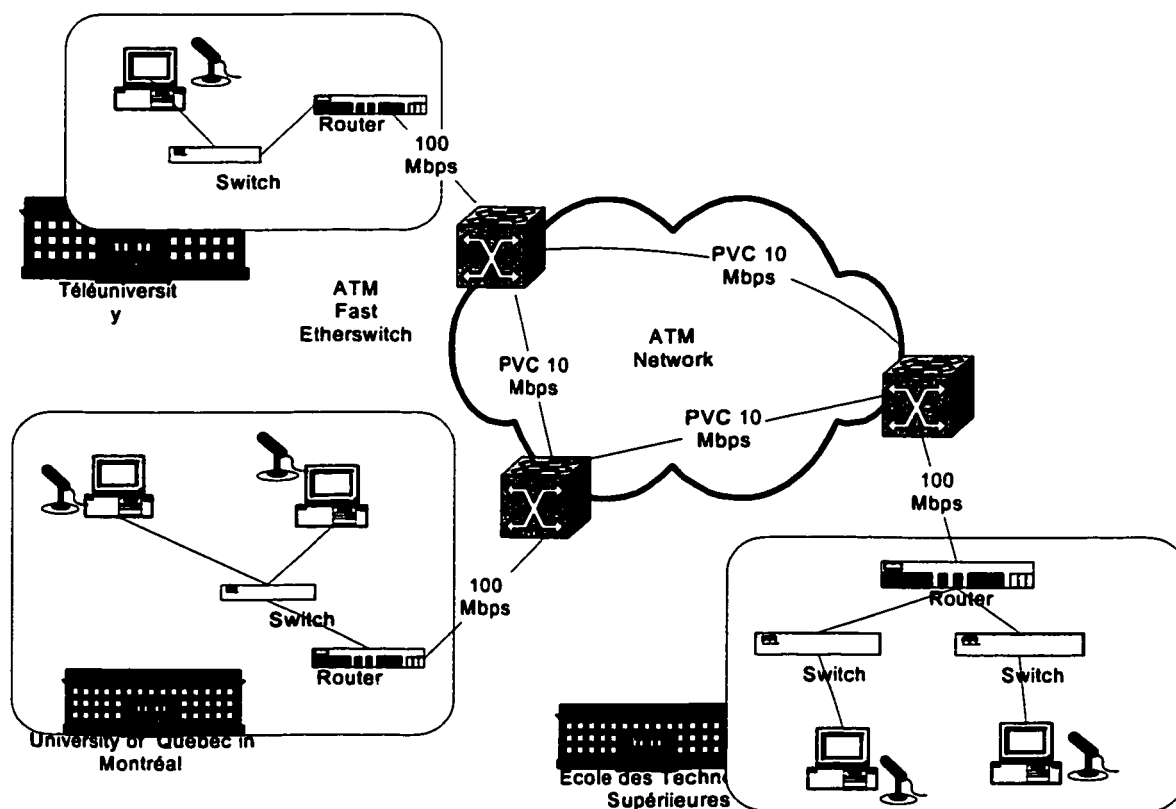


Figure 2 Schéma d'une architecture de connexion

Ce projet bénéficie d'une dynamique originale. Cette dynamique repose sur les progrès *technologiques* accomplis dans les domaines des télécommunications et de l'informatique ainsi que sur le développement et le déploiement accéléré des *standards* qui définissent l'Internet et ses applications, et favorisés par l'existence du réseau entendu de l'Université du Québec. Sa plate-forme doit être conçue pour rendre indépendante son application de vidéoconférence à l'infrastructure physique du réseau qui supporte la communication.

### 1.3 Méthodes existantes

Nous pouvons dans notre cas caractériser la qualité du service, d'une manière générale, par la capacité d'un élément de réseau (par exemple une application, un serveur ou un routeur) à garantir une livraison ponctuelle des données du réseau et qui respecte l'intégrité et la conformité de celles-ci. Quelques applications sont plus rigoureuses au sujet de leurs conditions de qualité de service que d'autres, et pour cette raison (entre autres), nous avons deux types de bases de qualité de service disponibles:

- **réserveation de ressource** (services intégrés) : des ressources de réseau sont réparties selon une requête qualité de service d'une application, et sujet à la politique de gestion de la largeur de bande
- **prioritisation** (services différenciés) : dans ce cas, le trafic réseau est classifié et les ressources sont réparties selon le critère de la politique de gestion de la largeur de bande. Pour permettre une qualité de service, les éléments de réseau donnent un traitement préférentiel aux classifications des trafics réseau identifiées en tant qu'ayant des exigences plus élevées.

Ces types de qualité de service peuvent être appliqués à des applications individuelles "flot" ou pour un flot d'agrégats. Par conséquent, il y a deux autres voies pour caractériser des types de qualité de service:

- **par flot** : Un flot est défini comme individuel, unidirectionnel de flux continu de données entre deux applications (émetteur et récepteur), identifiée seulement par un 5-tuplet (protocole de transport, adresse de source, numéro du port source, adresse de destination, et numéro du port de destination).
- **par agrégat** : Un agrégat est simplement la conjonction de deux flots ou plus. Typiquement, les flots auront quelque chose en commun (par exemple un ou plusieurs des paramètres des 5-tuplet, une étiquette ou un nombre prioritaire, ou peut-être de l'information d'authentification).

Les applications, la topologie de réseau et la politique dictent quel type de qualité de service est le plus approprié pour des flots individuels ou agrégats. Pour satisfaire le besoin de ces différents types de qualité de service, il y a un certain nombre de différents protocoles et algorithmes de qualité de service

- **Protocole de ReSerVation (RSVP)** : Fournit la signalisation pour permettre la réservation de ressources de réseaux (autrement connue sous le nom de services intégrés). Bien que typiquement utilisé à base d'un flux, RSVP est également utilisé pour réserver des ressources pour des agrégats.
- **Services différenciés (DiffServ)** : Fournit une voie solide et simple pour catégoriser et prioriser le trafic du réseau par flux et agrégats.
- **Multi Protocol Labeling Switching (MPLS)** : Fournit la gestion de largeur de bande pour des agrégats par l'intermédiaire de commande de routage de réseau selon des étiquettes (par encapsulation) dans l'entête de paquet.
- **Subnet Bandwidth Management (SBM)** : Permet la catégorisation et la priorisation à la couche 2 (la couche de lien des données dans le modèle OSI) sur les réseaux IEEE 802 partagés et commutés.

## 1.4 Justification de notre approche

Le but est de permettre à une application de gestion de réseaux d'activer des profils de service et d'action selon un horaire particulier pré-déterminé. On a déjà la définition des politiques qui sont rattachées à des *Schedulers*, qui s'appliquent à des équipements ou groupes d'équipements de différents constructeurs.

Le schéma suivant montre le synopsis de ce que doit contenir une politique et qui nous permettra de bien cerner le problème, suivi de quelques explications :

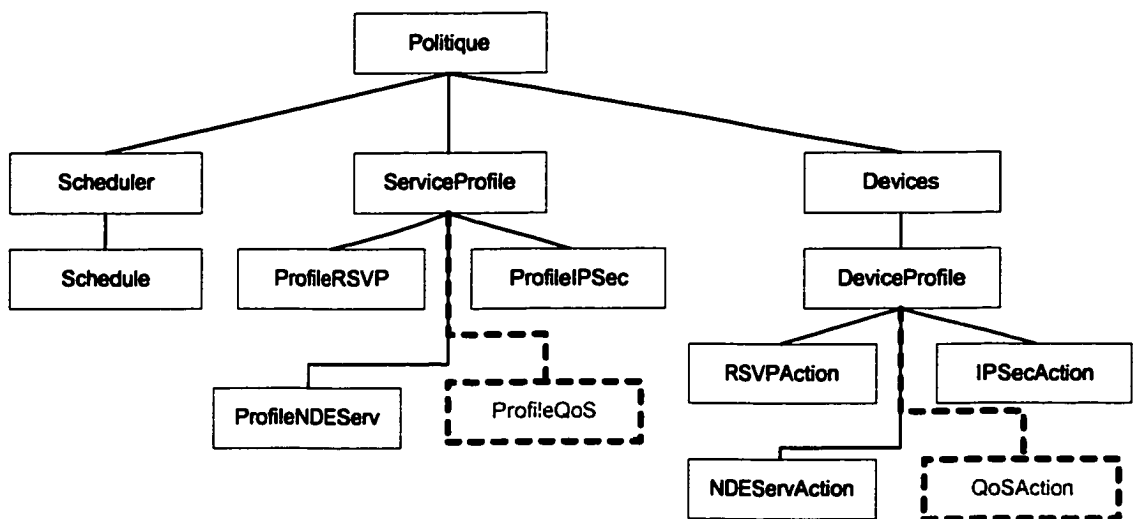


Figure 3 Schéma synoptique d'une politique

- Services et liens existants en traits pleins
- Service et liens rajoutés en traits discontinus

### Détails

Ce que l'on cherche à faire est de permettre à une application de gestion de réseau d'activer un profil de service selon un horaire particulier. On définit des politiques qui sont rattachées à des *Schedulers* (le *Scheduler* étant composé de un ou plusieurs



*Schedule* qui sont intégrées à l'éditeur). Ce profil de service doit s'appliquer à un équipement ou à un groupe d'équipements (*Devices*) qui lui sont associés.

Les *ServiceProfile* sont des objets qui permettent de détailler le profil du service qui est désiré de bout en bout. Ils décrivent les profils généraux des services en faisant abstraction des implémentations spécifiques sur le matériel. Par exemple, un *ServiceProfile* pourrait contenir des informations précisant que le service qualité de service doit être accordé sur les interfaces réseaux des routeurs à une valeur maximale de 5000 kbps. Un profil de service peut contenir des informations sur plusieurs services simultanés. Il est également associé à une politique de gestion à mettre en place.

Les *DeviceProfile* sont aussi des objets qui représentent les fonctionnalités des équipements et les méthodes pour les configurer en fonction de chacun des profils de service à implanter dans le réseau. Un profil de *Device* est associé à chaque dispositif défini sur le serveur. Le même profil peut être associé à plusieurs dispositifs, mais on ne retrouve qu'un seul profil par dispositif. C'est en fait ce qui permet de déterminer ces possibilités, et les méthodes utilisées pour le configurer en fonction des différents services. Par exemple, un routeur se verra associer le profil *Cisco2611\_113* qui indiquera que le routeur supporte la QoS sur l'IOS 11.3 et qu'il doit utiliser le script XYZ pour configurer ce service sur cet élément. Si un autre routeur possède les mêmes caractéristiques, il utilisera le même profil de configuration. Si une autre série de routeurs Cisco (ou d'une autre compagnie) a par exemple l'IOS 12.0, on pourra avoir un profil différent pour ces routeurs afin de refléter les différences entre les 2 IOS.

La nécessité d'une qualité de service se fait sentir dans la composition de ce gestionnaire à travers ces différentes politiques et leurs profils afin de combler le besoin pour une meilleure garantie dans le service et pour s'assurer de fournir un bon fonctionnement d'où un nouveau service QoS fiable s'impose. Notre tâche consiste à alors ajouter ce nouveau service portant sur la qualité de service via une priorisation du trafic engendré

par l'application multimédia et qui garantira les échanges entre usagers en améliorant les performances.

## **1.5 Contributions**

À la lumière de la problématique du sujet à traiter et vu les objectifs visés, notre contribution se résume en ce qui suit :

1. la modélisation orientée objet et l'implémentation du protocole 802.1Q/p pour les réseaux locaux et qui comporte :
  - a) la gestion de la largeur de bande par flux individuel et par agrégats ainsi que les ports UDP
  - b) la gestion des priorités dans le trafic via les classes de service
  - c) tracer les protocoles RSVP et 802.1p via le mécanisme de signalisation SBM
2. l'utilisation du schéma de DEN en tant que technique pour la description des modèles de classes de la qualité de service et leur intégration dans le gestionnaire de politiques.
3. la conception et l'implantation des interfaces de commande et de gestion dont l'architecture est du type client/serveur.
4. les tests et la mise en œuvre de ces implantations.

## **1.6 Conclusion**

Nous avons présenté dans ce chapitre la notion de qualité de service en insistant sur sa définition et son importance et en citant quelques types de base pour son implantation. Nous avons aussi mis en relief l'environnement dans lequel on envisage son application, à savoir les systèmes et les applications multimédia et les besoins à combler. Aujourd'hui, de multiples groupes de travail de l'IETF travaillent dans des domaines touchant la qualité de service. Certains tiennent compte de nouveaux développements dans les réseaux tels que les techniques de "*label switching*", tandis que d'autres

focalisent sur les politiques de contrôle dans les réseaux ou la définition de nouveaux protocoles de routage. Tous ces travaux sont néanmoins étroitement liés et permettent de construire progressivement les morceaux du puzzle qualité de service.

Plusieurs alternatives existent aujourd'hui. Cependant, parallèlement au modèle de réservation des ressources et de services intégrés posant des problèmes de déploiement à large échelle, il semble que les efforts s'orientent maintenant vers l'allocation prioritaire des ressources permettant de créer des distinctions des classes au sein du service meilleur-effort. Ce type de modèle étudié au sein du groupe *diffserv* repose essentiellement sur un champ dans l'en-tête du paquet IP, de telle sorte que les mécanismes de mise en file d'attente et de rejet des paquets puissent être mis en œuvre sur chaque nœud intermédiaire.

L'implémentation de tels algorithmes ajoutés au déploiement d'architectures adéquates (offrant un service prévisible, stable et cohérent) pourront offrir à terme, des niveaux de services différenciés tout en permettant une bonne stabilité.

Dans le chapitre suivant, nous présenterons l'environnement de travail et les outils nécessaires qui nous aideront dans notre démarche, Aussi nous aborderons l'étude des normes utilisées, leurs composants, et leur importance; entre autres nous allons montrer les avantages des produits qui utilisent 802.1Q/p dans leurs implantations.

## CHAPITRE 2

### NOTIONS DE POLITIQUE, MODÈLE ET NORME

Nous avons vu dans la section précédente que notre gestionnaire obéissait à des règles qui sont basées sur le principe d'une politique. En effet, on a d'un côté un horaire (*scheduler*) qui, lui, représente notre condition et un service qui, lui, évoque l'action à appliquer. Ces deux parties sont essentielles dans la constitution et le fonctionnement de notre gestionnaire; c'est à l'aide de la relation directe qui existe entre eux qu'il nous sera possible de mettre en place l'ensemble de l'architecture du système.

Dans ce chapitre, nous allons montrer et expliquer tous les concepts et les définitions relatifs au gestionnaire de réseau, en présentant tous les constituants qui forment cet environnement.

Nous tenterons de faire le point sur la notion de politique et de sa gestion, nous décrirons les mécanismes et approches qui nous permettront d'implanter les classes de service.

#### 2.1 Principe d'une politique de gestion

Dans son sens le plus simple, la politique est une ou plusieurs règles qui décrivent la ou les actions à se produire quand la ou les conditions spécifiques existent. Ce concept est illustré par la figure 4.



Figure 4 En un simple sens, les politiques spécifient une ou plusieurs conditions qui résultent en une ou plusieurs actions.

Une règle de politique comportera souvent d'autres règles. En effet "des politiques peuvent contenir des politiques". Cette notion de hiérarchie est cruciale, car elle permet à des politiques complexes d'être établies d'un ensemble de politiques plus simples, ceci simplifie leur gestion.

Les réseaux sont fondamentalement non déterministes et cela vaut particulièrement pour le standard meilleur-effort des réseaux IP, puisqu'ils ne consacrent pas des ressources à la manière des réseaux téléphoniques à base de circuit virtuel. La condition pour les réseaux où la qualité de service est à base de politique est là pour simplifier les situations. Les règles de politique (conditions et actions) doivent être vérifiables et non ambiguës. Il devrait y avoir seulement une règle correcte appropriée pour un groupe spécifique de conditions. En d'autres termes, les politiques doivent définir un état-machine fini.

## **2.2 Une politique dans une structure réseau**

Cette section présente une vue d'ensemble des services et des protocoles décrivant le cadre d'une politique et identifie les entités principales impliquées. Elle fournit un contexte permettant de comprendre la structure de la gestion de politique. Ce cadre vient combiner les résultats de différents groupes de travail de IETF et ITU pour définir les standards et les moyens nécessaires à une gestion dynamique des environnement réseaux par la mise en place de règles (politiques). Dans ce qui suit, on verra avec plus de détails ce concept global avec tous ces aspects.

La coordination des informations sur un réseau peut être gérée par un réseau à base de politique (RBP) qui est capable de tracer dynamiquement ces informations de concert avec les informations de configuration comprenant des mécanismes d'alignement, des procédés de traitement de paquet, la capacité de lien basée sur la classe de service, etc. L'élément principal dans RBP est un modèle d'information qui définit des abstractions de gestion des profils, des politiques, des dispositifs, des protocoles et des services. Les

mécanismes de RBP tels que les serveurs de politique peuvent localiser et identifier par eux-mêmes les divers dispositifs du réseau auxquels l'envoi de l'information de configuration appropriée est rendu possible par des protocoles tels que *Common Operation Policy Services* (COPS) ou *command line interface CLI*.

*Lightweight Directory Access Protocol* (LDAP) est un protocole de serveur-client qui permet la communication entre différents services d'annuaire électronique de sorte que les utilisateurs puissent créer et accéder à des répertoires. Fondamentalement, le but de LDAP est d'agir en tant que mécanisme de transmission compris entre le serveur de politique et les répertoires. Son premier avantage est de fournir un mécanisme à base de normes qui permet à n'importe quel client, serveur ou application l'accès à n'importe quel service de répertoire qui supporte l'interface de protocole de LDAP. La dernière version, LDAPv3, a été augmentée par l'ajout de plusieurs autres fonctions comprenant: sécurité, intégration simplifiée de schéma, extensibilité, références et caractéristiques internationales.

L'intérêt d'un service d'annuaire à répertoire (DEN/LDAP) est à la fois de mémoire et de gestion des politiques d'un réseau. En tant que tel, le répertoire pourrait contenir de l'information sur le mode du réseau, des tendances d'utilisation et l'utilisation des ressources. Les répertoires permettent également l'association d'un profil d'utilisateur au niveau du service, de l'accès, et de la priorité qu'ils recevront. Ensemble, ces informations peuvent être employées pour optimiser l'allocation de ressource de réseau.

Une fois que le serveur de politique a recueilli toute l'information nécessaire pour formuler sa décision, il renvoie une réponse (à la politique) au client (dans ce cas-ci un routeur/commutateur). Le protocole employé pour communiquer au client de politique sera typiquement identique à celui utilisé pour la demande de politique (COPS par exemple). Cependant, parce que COPS n'est pas encore officiellement une norme, une partie du matériel déjà déployé ne collaborera pas avec un serveur de politique utilisant

ces protocoles. Dans ces cas, le serveur de politique doit traduire la décision pour une politique vers un protocole que chaque dispositif peut comprendre. Ces protocoles seront typiquement SNMP et Telnet.

La responsabilité de fournir les fonctionnalités nécessaires aux ressources afin de rencontrer les politiques en place sera maintenant assuré par des équipements réseau comme les routeurs et les commutateurs. Essentiellement, le client d'une politique agit également en tant que renforceur de politique, utilisant des *hardware/software* spécifiques pour fournir le filtrage de paquet, la réservation de largeur de bande, la priorisation du trafic, etc.

Cette architecture est présentée par la figure 5.

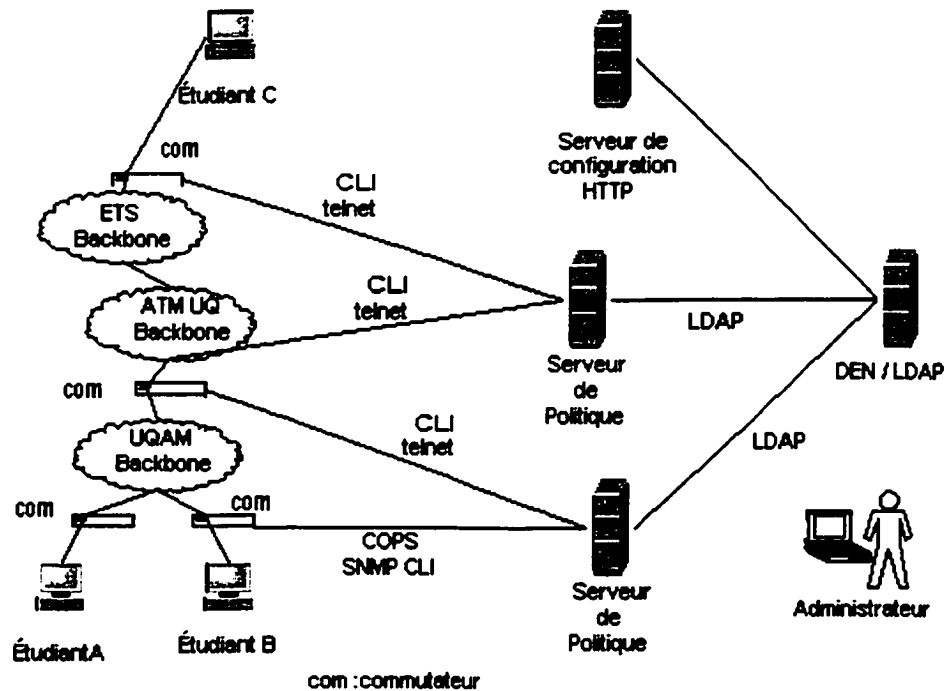


Figure 5 Encadrement d'une politique

### 2.3 Les services et modèles d'information (X500, CIM et DEN)

Nous venons de situer le serveur principal (DEN/LDAP) dans l'ensemble de son réseau. Maintenant, nous allons en donner une description plus détaillée. Étant donné le peu d'interopérabilité en ce qui concerne la représentation des données, des modèles d'information ont été développés. X500 et *Common Information Model* (CIM) ont été les précurseurs dans ce domaine. À ses débuts, X500 a été développé pour permettre la représentation des entreprises, des organisations et des personnes. Par la suite, le modèle CIM, conçu par DMTF (*Desktop Management Task Force*) a été étendu pour qu'il soit applicable à la description des équipements réseaux, des profils utilisateurs et des politiques réseaux. Le modèle DEN (*Directory Enabled Network*) est maintenant adapté aux applications utilisant le réseau Internet. C'est un progrès technologique considérable qui promet des évolutions majeures principalement dans le domaine de la gestion des réseaux (figure 6).

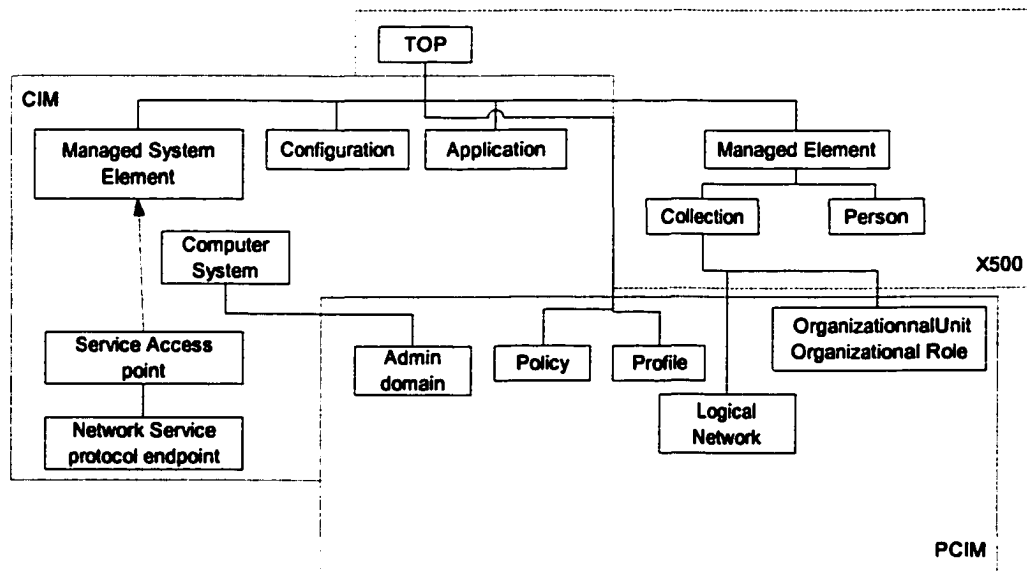


Figure 6 Modèle d'information DEN



Le modèle d'informations ne comporte pas seulement la description des éléments. Le schéma tient compte des relations logiques entre les objets. En effet, chaque objet se décrit grâce à deux types d'informations: les attributs du modèle objet ainsi que les relations entre ces objets. Le modèle objet disponible peut aussi être aisément étendu pour les besoins spécifiques de chaque application.

L'utilisation d'un modèle commun pour la représentation des données procure des avantages incontestables lors de l'implémentation d'applications. Le premier est l'accélération des développements puisque le modèle simplifie la période d'analyse et de design d'une application. À cela s'ajoutent les outils utilisables en rapport avec ce schéma. LDAP est l'un de ces outils. Le deuxième avantage réside dans l'interopérabilité entre des applications s'exécutant dans des environnements ayant des dépendances. Désormais, il est possible de choisir les modules séparément en évitant ainsi les modules défectueux d'un fournisseur.

## 2.4 Répertoire LDAP

Pour permettre l'échange d'information entre le serveur principal (DEN/LDAP) et les serveurs de politique, on a recours au *LightWeight Directory Access Protocol* (LDAP), qui est un protocole d'accès à une base de données. C'est un protocole nécessitant peu de bande passante utilisée généralement pour lire (plus rarement écrire) des informations sous forme d'objets et d'attributs. Le modèle DEN sert à la définition de ces objets. L'utilisation de LDAP est un atout considérable lorsqu'on sait qu'il est d'ores et déjà intégré aux systèmes d'exploitation arrivant sur le marché notamment *Windows 2000* et *Solaris 8*. On peut noter aussi le développement d'*OpenLDAP* dans le domaine des logiciels libres destinés aux UNIX et principalement *Linux*.

Des requêtes client (en format texte) sont envoyées à un serveur LDAP qui se charge ensuite d'y répondre. Deux aspects des opérations sont les points clé de la compréhension de LDAP.

Lors de la configuration d'un serveur LDAP, celui-ci doit avoir une connaissance de la structure des objets qui seront stockés dans sa base de données. Il doit connaître tous les objets et leurs attributs. Ils sont accessibles via le Serveur de Nom (*Name Server*), l'organisation des données suit le modèle des services de répertoire (*Directory Services*). C'est un modèle en arbre identique à la structure de systèmes de fichiers qui permet une recherche des informations beaucoup plus facile. LDAP autorise l'utilisation de filtres évolués afin d'augmenter considérablement les performances.

Pour démontrer ce point, nous pouvons considérer une politique d'affaire déclarant qu'un certain groupe d'utilisateurs devrait recevoir une priorité plus élevée dans le réseau à une heure particulière. Spécifiquement, les ingénieurs devraient obtenir la priorité de 9h-11h. C'est une règle statique. Elle est entrée dans le répertoire pour rester sans changement aussi longtemps qu'il est exigé. Cependant, durant la période requise de la mise en place, la règle de politique elle-même est " dynamiquement " installée sur le réseau à 9h. et " dynamiquement " alors retirée à 11h. En plus, une autre considération qui doit rentrer en ligne de compte, est qu'une recherche est exigée avant l'installation des filtres du paquet de la politique pour déterminer les membres du groupe d'ingénierie. Ce groupe peut changer fréquemment, spécialement s'il réside dans une grande organisation. Ainsi, il est clair que les règles au niveau de réseau sont assez dynamiques.

## **2.5 Classe virtuelle**

La classe virtuelle, généralement parlant, est un environnement d'enseignement et d'apprentissage réparti à travers un système de communication par ordinateur qui sont équipés par des outils (logiciel et matériel) multimédia. On peut dire qu'une classe

virtuelle est développée avec un ensemble d'applications dans un environnement de travail complètement intégré au niveau logiciel.

Tout cet environnement devient virtuel au lieu d'être un espace physique comme une salle de classe avec un bureau, des chaises, un tableau, etc. La classe virtuelle est un moyen efficace pour faciliter l'interaction entre les membres dans un environnement nouveau.

Plus spécifiquement, la classe virtuelle est un ensemble d'applications logicielles permettant un apprentissage en totale collaboration avec tous les membres de cette même classe. Les membres peuvent participer à une classe virtuelle sans tenir compte des facteurs temps et espace. Une classe virtuelle, pour être accessible ne nécessite, pour un utilisateur, qu'une connexion à un réseau IP fiable.

Cette méthode d'apprentissage permet au participant d'obtenir une période de réflexion et d'introspection avant qu'il décide de répondre à une question ou de participer à un débat.

## **2.6 Théorie des VLAN**

C'est le fondement de la classe virtuelle. Les réseaux virtuels permettent de diviser logiquement un réseau local en plusieurs domaines de diffusion (domaines de broadcast).

Etablir un réseau virtuel consiste à créer un groupe d'utilisateurs (regroupés au sein d'un domaine) indépendamment de leur position géographique. On peut ainsi définir des domaines de diffusion, sans utiliser les routeurs, indépendamment de l'endroit où se situent les stations. En effet, la division du réseau local est une division logique et non

physique et donc les stations d'un même domaine de diffusion ne sont pas obligées d'être sur le même segment LAN.

Les VLANs sont définis par le standard IEEE 802.1Q. Les VLANs doivent supporter les protocoles MAC de tous les standards LANs comme Ethernet, Token Ring, FDDI, Fast Ethernet, etc.. Ils simplifient l'administration de groupes de stations, limitent le trafic sur les réseaux locaux. Un commutateur est nécessaire pour développer un VLAN.

Les VLANs doivent permettre la gestion dynamique de la mobilité des utilisateurs tout en leur permettant de partager des données.

L'introduction des VLANs repose sur trois concepts :

- limiter les domaines de diffusion.
- garantir la sécurité.
- permettre la mobilité des utilisateurs.

Les VLANs apportent plusieurs avantages :

- L'optimisation de l'utilisation de la bande passante : dans un réseau local, les paquets transmis par un utilisateur sont envoyés à toutes les stations du réseau ; avec les réseaux virtuels, les paquets transmis par un membre d'un groupe ne sont reçus que par les autres membres de ce groupe et non par toutes les stations du réseau. Ainsi, il n'y a pas de trafic vers des destinations inutiles.
- Augmentation de la sécurité : dans un réseau local, des informations sensibles sont diffusées sur le réseau. Avec les VLANs, il est possible de créer un groupe composé uniquement d'utilisateurs qui peuvent avoir accès à ces informations. Etant donné que les émissions de paquets sont restreintes à des domaines, tout système à l'extérieur de ces domaines ne sera pas capable de recevoir ces informations.

- L'administration des réseaux est simplifiée : le déplacement physique d'une station d'un réseau local à un autre implique des coûts élevés, il faut faire des recâblages, reconfigurer les équipements etc.
- Le support d'organisations virtuelles : il est possible avec les réseaux virtuels de définir un groupe temporaire, par exemple, constitué d'employés d'une entreprise qui travaillent sur un même projet. Les VLANs leur permettent de partager des données et de travailler ensemble sur ce projet et ceci indépendamment de la localisation de ces employés.

Il existe différents types de VLAN :

1. **Les VLANs par port (Port-Based VLAN) ou de niveau 1** regroupent les systèmes en fonction du port sur lesquels ils sont connectés. Tel port du commutateur est associé à tel numéro de groupe.
2. **Les VLANs par adresse MAC (MAC Address-Based VLAN) ou de niveau 2** associent des stations par leur adresse MAC selon des tables d'adresses introduites par l'administrateur. L'administrateur indique au commutateur que telle adresse MAC appartiendra à tel numéro de groupe.
3. **Les VLANs d'adresse réseaux (Network Address-Based VLAN) ou de niveau 3** associent des sous-réseaux IP avec des groupes.

Il existe également un autre type de VLAN : **les VLANs selon un protocole (IP, IPX, ..)**. Deux stations peuvent communiquer uniquement si elles utilisent le même protocole.

## 2.7 Les normes IEEE

Notre choix s'est posé sur le protocole 802.1p qui est inclus dans le 802.1Q de la couche 2 dans le système ISO, car il a été utilisé dans l'application du gestionnaire de politiques comme un moyen de base d'identification et de communication entre VLAN. Ces

derniers sont gérés par le biais du protocole VTP *virtual trunking protocol* (à l'aide de jonctions).

L'alternative pour faire de l'inter-connexion au niveau 2 serait de prendre le protocole *Inter-Switch Link (ISL)*, propriété de la compagnie Cisco Systems.

### **2.7.1 Norme 802.1Q**

La norme IEEE 802.1Q définit une architecture pour des LAN Virtuels, ainsi que les services fournis dans celles-ci, en plus des protocoles et des algorithmes impliqués dans la fourniture de ces services.

Aucun mécanisme de qualité de service n'est défini dans ce standard, mais une importante condition pour fournir de la qualité de service est incluse dans ce standard, par exemple la faculté de régénérer la priorité de l'utilisateur au niveau des trames reçus en utilisant l'information sur la priorité contenue dans la trame et celle de la table de régénération de la priorité de l'utilisateur pour le port de réception.

### **2.7.2 Norme 802.1p**

La norme IEEE 802.1p, la classe du trafic expédiant et le filtrage dynamique de multicast offrent des méthodes importantes pour fournir de la qualité de service au niveau MAC.

IEEE 802.1Q ajoute 16 bits à l'en-tête de la trame à la couche 2, y compris trois bits qui peuvent être employés pour classifier la priorité (l'étiquette). Des trames qui implémentent 802.1p s'appellent " les trames étiquetées ". En outre, mettre en application le 802.1p dans un réseau avec des commutateurs qui ne supportent pas le 802.1p peut mener à l'instabilité, parce que des anciens commutateurs vont mal interpréter les 16 bits inattendus indiqués par la norme. De la même façon, mettre en

application 802.1p dans des réseaux plus anciens peut exiger une mise à niveau coûteuse de tous les commutateurs. Plus important encore, sachant que l'en-tête de niveau 2 est lu seulement au niveau commutateur, les routeur aux frontières du réseau, où le goulot d'étranglement se produit, ne vont pas tirer profit de la priorisation à base du 802.1p, à moins qu'il soit tracé avec la priorisation de couche 3.

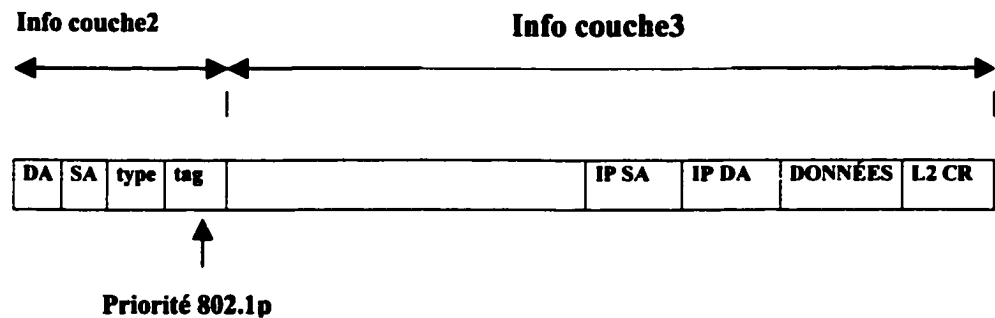


Figure 7 Niveau 2 : IEEE 802.1p ajoute trois bits de priorité (tag) a l'entête du paquet

### Les types de Trafic

Un exemple des différents types de trafic qui pourraient être présents sur un réseau local simple, regroupés en sept types et définis comme suit :

- a. contrôle de réseau : essentiel au support et au maintient des infrastructures réseau.
- b. voix interactive : exige un retard de moins de 10 millisecondes
- c. multimédia interactif : exige un retard de moins de 10 millisecondes
- d. application à charge contrôlée : quelques importantes applications
- e. service maximum : meilleur effort pour d'importants usagers
- f. meilleur effort : priorité ordinaire de LAN

g. arrière-plan : tout autre trafic, volume de transfert, jeux etc.

h. service économique

Les types et les acronymes qui définissent le trafic sont montrés dans les tableaux III et IV. Ils indiquent comment les différents types de trafic sont tracés selon le nombre de file d'attente de transmission existant :

**Tableau III**  
**Acronymes des types de trafic**

<b>Priorité (CoS)</b>	<b>Acronyme</b>	<b>Type de trafic</b>
1	BK	Arriere-plan
2	-	Service économique
0(par défaut)	BE	Meilleur Effort
3	EE	Service Maximum
4	CL	Charge Contrôlée
5	VI	Multimédia interactif (Vidéo)
6	VO	Voix interactive
7	NC	Contrôle de réseau



**Tableau IV**  
**Définition des types de trafic**

Nombre de file d'attente	Définition du type de trafic (dans un ordre croissant de priorité de gauche à droite)							
1	BE							
2	BE				VO			
3	BE				CL		VO	
4	BK	BE			CL		VO	
5	BK	BE			CL	VI	VO	
6	BK	BE	EE	CL	VI	VO		
7	BK	BE	EE	CL	VI	VO	NC	
8	BK	-	BE	EE	CL	VI	VO	NC

## 2.8 Priorités dans le trafic

Jusqu'à présent, on a longuement parlé des notions de qualité de service et de classe de service. Ici, on fera une étude comparative en définissant chacune d'elles et leur impact sur une architecture Ethernet afin d'établir le lien avec le réseau de l'Université du Québec, qui est principalement Ethernet.

### 2.8.1 Définition de CoS et de QoS

Le but de la qualité de service est de fournir un service prévisible (service défini en tant qu'un partage des capacités disponibles) durant les périodes de congestion. C'est la période de congestion qui est la cible de la qualité de service. Être capable de mesurer et rapporter sur le niveau de la qualité de service est aussi un important attribut des solutions de qualité du service. La classe de service est plus généralement convergée pour ce genre de tâche.

**a. classe de service**

- priorisation du trafic,
- petit nombre de classes individuelles.

**b. qualité de service**

- paramètres ou attributs qui décrivent le service,
  - réservation de largeur de bande,
  - retard bout-à-bout,
  - variation dans le retard (déviante "jitter"),
  - taux d'erreur,
  - perte dans le trafic.

**c. exemples**

- IEEE-802.1p (CoS),
- IETF RSVP (QoS),
- ATM Gestion du trafic.

CoS/QoS catégorise le trafic pour s'assurer que le trafic critique transitera toujours, en dépit des demandes de largeur de bande pour des applications moins importantes.

### **2.8.2 Les options pour CoS/QoS dans des paquets LAN**

Pour mettre en relief le rapport entre RSVP et le 802.1p, on regardera les caractéristiques de chacun :

### **Protocole RSVP**

- **RSVP fonctionne seulement avec des dispositifs de la couche 3,**
- **La commutation en couche 2 déplace le routage dans le LAN,**
- **RSVP est présent dans plusieurs système d'opération à ordinateurs,**
- **Obligation de réécrire les applications pour supporter RSVP.**

### **IEEE-802.1p**

- **Fourni du CoS a la couche 2,**
- **La commutation a la couche 2 représente la majorité des implémentations dans les LAN,**
- **Les commutateurs IEEE-802.1p sont offerts par de multiples manufacturiers.**

### **2.8.3 Ethernet et IEEE-802.1Q/p**

L'évolution du 802.1p dans un environnement Ethernet se présente ainsi:

#### **IEEE 802.1Q/p:**

##### **a. 802.1Q**

- **Normalisation de VLAN, y compris les formats d'étiquetage et de gestion.**

##### **b. 802.1p**

- **Définition du CoS à la couche 2,**

- Configuration dynamique / protocole de distribution (GARP<sup>2</sup>),
- 8 niveaux de priorité
- Des files d'attente multiple par port sont exigées.

### Avantages

- Priorité élevée d'alignement pour le trafic critique,
- GMRP<sup>2</sup> minimise l'inondation par du trafic multicast.

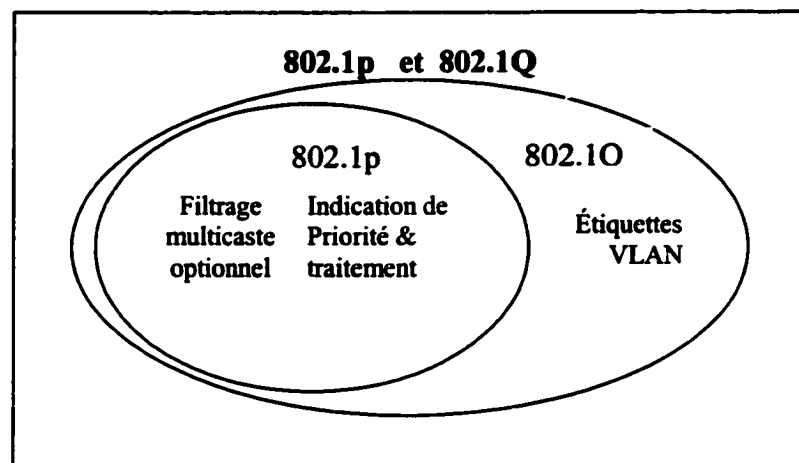


Figure 8 Meilleure comparaison et superposition entre les normes

## 2.9 Gestionnaire de largeur de bande d'un sous-réseau

**SBM (Subnetwork Bandwidth Manager)** est un protocole de signalisation des contrôles d'admission pour les LAN à base du protocole de réservation des ressources (RSVP). Il fournit une méthode pour tracer un protocole de la couche 3 tel que RSVP au LAN Ethernet. Il décrit le fonctionnement et le traçage d'une demande de largeur de bande de

<sup>2</sup> GARP(Generic Attribute Registration Protocol) Multicast Registration Protocol

la part d'un hôte supportant RSVP avec la couche du lien des dispositifs tels que des commutateurs et des passerelles pour supporter la réservation des ressources de réseau local pour des flux de données qui permettent le RSVP.

RSVP est un protocole de signalisation de la couche 3 qui permet la réservation de la largeur de bande pour un flot dans des réseaux à base de paquets. Cependant, quand les messages de RSVP doivent voyager sur des dispositifs de la couche 2 dans le LAN, aucune réservation de largeur de bande n'est faite sur ses dispositifs (commutateurs, passerelles) parce qu'elles ne comprennent pas l'information de la couche 3. Ce scénario peut conduire à la sur-utilisation des liens du LAN et peut par la suite mener à la congestion, particulièrement si le réseau de la couche 2 est un sous-réseau partagé ou un sous-réseau à vitesse mal adaptée (un sous-réseau dans lequel les connexions de serveur/client aussi bien que les connexions de circuit principal sont de la même vitesse). SBM est précisément le protocole qui traite des réservations dans un réseau partagé style IEEE 802 (Ethernet), commuté, ou à vitesse mal-adaptée de la couche 2 du réseau. Afin d'honorer les réservations, un dispositif indiqué agit au nom des autres dispositifs de la couche 2 pour manipuler les réservations du flot. Ce dispositif s'appelle *Designated Subnetwork Bandwidth Manager* (DSBM). Un DSBM est élu selon la priorité assignée.

### Avantages

- En étendant la réservation de largeur de bande d'une application qui supporte RSVP à la couche 2 du LAN ou à l'infrastructure de campus, SBM complète la qualité de service de bout à bout de la couche 3 pour des applications de micro-flot telles que des centres d'appels de multimédia, VoIP, et Web, et cela du WAN au campus au réseau local.
- SBM permet une meilleure réservation des ressources dans des environnements de médias partagés ou les environnements LAN, dont les Commutateurs/passerelles ne comprennent pas et ne supportent pas des protocoles de la couche 3.

- Les réservations de RSVP par-flot peuvent être faites de la même façon que dans la couche 3 du réseau, permettant l'extension de la qualité de service à l'extrémité du système.
- SBM permet le contrôle d'admission pour la qualité de service dans la couche 2 du LAN
- Un DSBM peut contrôler les réservations des ressources ou le contrôle d'admission sur un segment LAN.

## **2.10 Conclusion**

Les normes 802.1p/Q sont devenues un support de base important pour plusieurs compagnies et organisations telles que 3Com, Microsoft, Cisco, IBM, etc.

Le traçage avec le protocole RSVP de niveau 3 en utilisant la technique du protocole de signalisation SBM permet une meilleure utilisation des ressources du réseau afin d'offrir une bonne priorité au trafic et une utilisation efficace de la bande passante.

Dans le chapitre suivant, nous présenterons le profil de notre service QoS et son implémentation dans le modèle d'information, ainsi que les procédures et le scénario de fonctionnement (des exemples). Elle comporte aussi la méthode d'implémentation orientée objet et cela, en citant les étapes d'intégration des classes *JAVA* et les autres changements introduits sur les classes du *Policy Editor* pour une bonne complémentarité.

## **CHAPITRE 3**

### **IMPLÉMENTATION DU PROFIL QoS AU MODÈLE DEN**

#### **3.1 Introduction**

Le modèle d'information DEN (*Directory Enabled Network*) contient les spécifications et le schéma servant de fondation pour la gestion de réseau et de ses applications. Il décrit toutes les classes du gestionnaire de politiques ou on retrouve les services fournis, les profils des équipements et des personnes, etc.

Dans ce chapitre nous décrivons toutes les étapes de la création du nouveau profil QoS, nous établirons ses liens avec les autres classes et entités de notre application de gestionnaire de politiques. En plus, nous élaborons son principe d'interaction avec d'autres classes par des exemples de compréhension.

Enfin, nous présenterons les étapes de son intégration et de son implantation à l'aide de langage informatique *JAVA*.

#### **3.2 Création de la classe QoS**

Au chapitre 2 paragraphe 3, on a introduit le modèle DEN donné par la figure 6, qui nous permettait de représenter tous les éléments (ainsi que les rapports des liens entre eux) qui constituent notre gestionnaire de réseau. Maintenant, pour créer notre nouveau profil QoS, on aura besoin d'étendre ce modèle en rajoutant les classes appropriées pour ce service QoS.

Nous reprenons le diagramme des classes du modèle d'information à base de DEN et qui définit déjà notre gestionnaire de politiques, afin d'implanter les configurations nécessaires à la QoS dans le réseau, sachant que la fonctionnalité du QoS doit permettre

de prioriser les flux audio/vidéo donc RTP en adoptant le principe et la stratégie des classes de service (CoS) pour gérer le trafic.

D'une manière générale, l'objectif ici est de faire, en suivant les extensions du modèle d'information DEN, la création d'un nouveau service réseau permettant :

- De configurer les routeurs pour réserver une stricte priorité aux flots de paquets RTP qui appartiennent à une rangée de port UDP de destination
- De configurer les commutateurs afin qu'ils honorent les priorités du 802.1p et spécifient le seuil de rejet des paquets (selon leurs CoS) dans les files d'attente au niveau de tous les ports.
- Coller RSVP et 802.1p ensemble par le biais de la signalisation SBM.

### **3.2.1 Modèle des classes du profil QoS**

À partir du schéma global des classes donné par la figure 17, on propose le diagramme de classes du profil QoS représenté par la figure 9, qui est obtenu à partir d'une extension à DEN de la classe Top, il est divisé en sous-modèle pour une meilleure intégration :



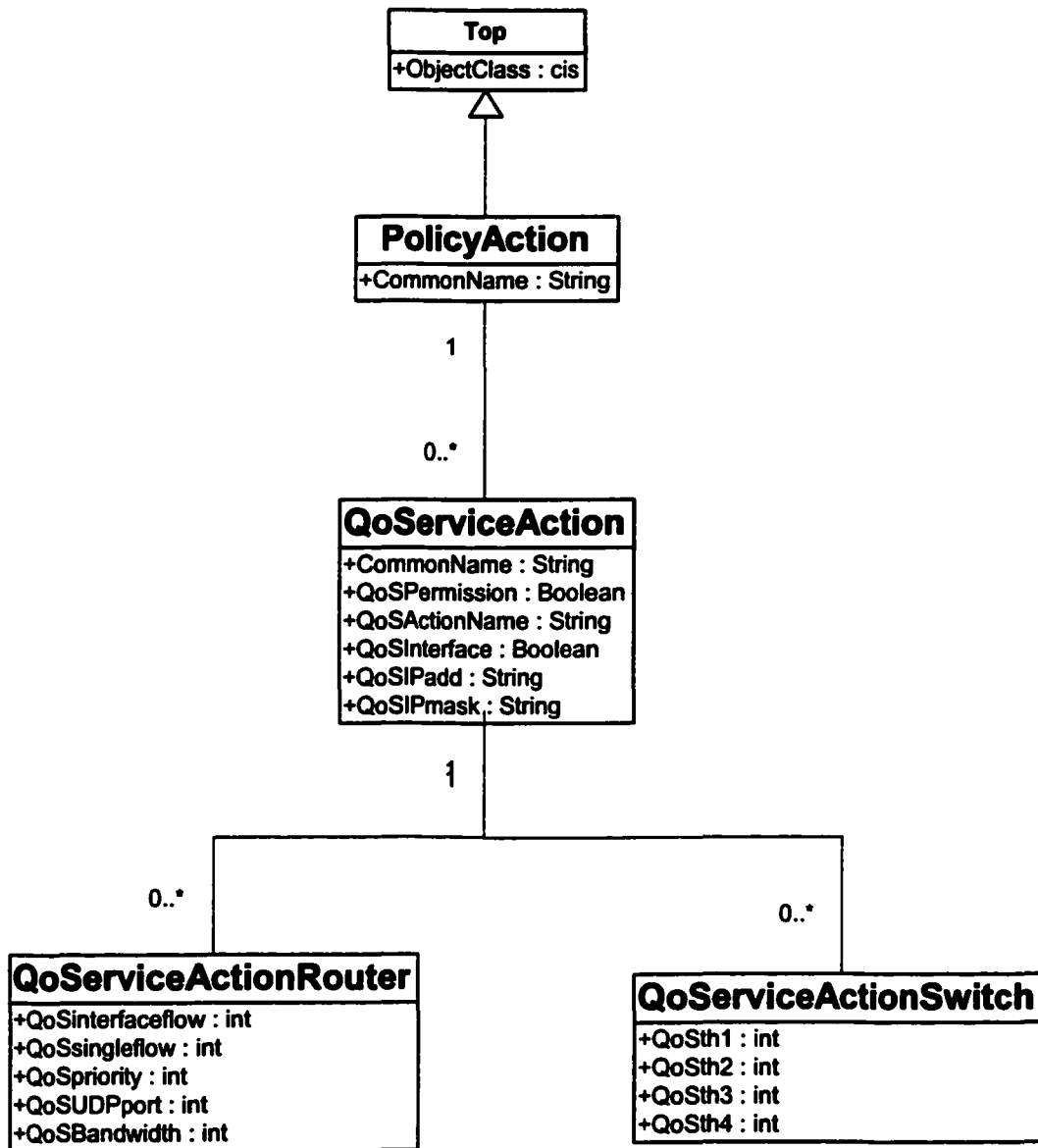


Figure 9 Modèle de la classe QoS

### 3.2.2 Identification des composants

Dans cette section, on identifie les composants de chaque classe contenu dans le diagramme de la figure 9.

Selon le schéma de l'IETF pour la gestion des services, un profil de service est défini par une sous-classe de *PolicyAction* qui hérite de la classe principale Top.

Pour notre application, nous avons cependant besoin de définir des informations sur le service QoS qui sont plus spécifiques à une interface réseau qu'au flux lui-même. Nous allons alors définir les classes suivantes:

#### a. la classe *QoServiceAction*

Cette classe permet de définir le profil de service pour chaque équipement. Elle contient les attributs suivants :

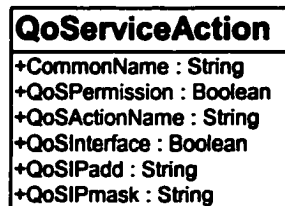


Figure 10 Classe *QoServiceAction*

#### 1. les attributs *CommonName* et *QoSActionName*

Nous permettent d'identifier le service en question par un identificateur qui le distingue.

#### 2. l'attribut *Permission*

Nous indique s'il s'agit d'une activation ou d'une désactivation du service (*Deny* ou *accept*).

### 3. les attributs Interface, IPadd et IPmask

Ce sont des paramètres nécessaires pour avoir accès à l'interface de l'équipement où le service est requis.

À partir de là, le choix d'une sélection est proposé selon qu'il s'agit d'un routeur ou d'un commutateur, on retrouve donc deux sous-classes.

#### b. la sous-classe *QoServiceRouter*

Cette sous-classe donne un accès spécifique au profil du routeur. Elle contient les attributs suivants :

<b>QoServiceActionRouter</b>
+QoSinterfaceflow : int
+QoSsingleflow : int
+QoSpriority : int
+QoSUDPport : int
+QoSbandwidth : int

Figure 11 Sous-classe *QoServiceActionRouter*

#### 1. l'attribut *interfaceflow*

Indique, en Kbps, ce que l'on désire permettre comme maximum de réservation de largeur de bande sur chacune des interfaces du routeur.

#### 2. l'attribut *singleflow*

Indique le maximum qu'une seule réservation peut effectuer sur une interface pour un flot de trafic donné. Ce dernier doit être plus petit que l'attribut *interfaceflow*.

#### 3. *UDPport*

Indique un intervalle des ports UDP de destination par lesquels les paquets seront reçus.

#### 4. *Bandwidth*

Spécifie un montant strict et garanti de largeur de bande à être réservée.

#### c. la sous-classe *QoServiceActionSwitch*

Cette sous-classe donne un accès spécifique au profil du commutateur. Elle contient les attributs suivants :

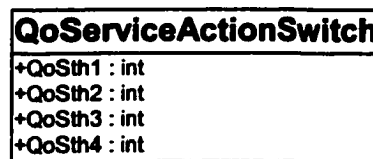


Figure 12 Classe *QoServiceActionSwitch*

#### 1. les attributs th1, th2, th3 et th4

Ils correspondent respectivement aux classes de services suivantes :CoS 0-1,CoS2-3, CoS 4-5 et CoS 6-7, qui donnent la priorité aux différents types de trafic du réseau qu'on a déjà vus au chapitre précédent.

### 3.2.3 Méthode d'activation de la politique QoS

On présentera d'une manière sommaire dans ce qui suit les étapes de la procédure de ce qu'un scénario donnera dans le concret. Les descriptions faites ici sont génériques; nous allons considérer une politique (réserver une largeur de bande de 5000 Kbps) qui doit activer un service QoS selon un horaire prédéfini. Notre politique ressemblerait donc à ceci:

***Policy-QoS***

Scheduler : 8h à 17h du lundi au vendredi valide à tous les mois.

ProfilDeService: *Service-QoS-exemple*

Device: *Cisco-routeur1* et *Cisco-routeur2*

Cette politique indique donc d'appliquer le profil *Service-QoS-exemple* selon l'horaire spécifié sur les routeurs *Cisco-routeur1* et *Cisco-routeur2*. Le profil de service aura donc une définition qui ressemble à ceci:

***Service-QoS-exemple***

ParamètresQoS: réserver une largeur de bande de 5000 kbps

La définition du service désiré ici fait abstraction des détails de configurations, mais précise des paramètres de haut niveau pour la description du service.

La politique doit s'appliquer à 2 routeurs ayant cette description:

***Cisco-routeur1***

ParamètreRéseau: 132.208.135.240, ...

...

ProfilDeRouteur: *CiscoModele2611\_IOS113*

***Cisco-routeur2***

ParamètreRéseau: 132.208.135.250, ...

...

ProfilDeRouteur: *CiscoModele2611\_IOS113*

Pour le besoin de cette présentation, les paramètres du routeur n'ont pas été précisés. Le profil de chacun de ces routeurs va permettre de connaître les paramètres de configuration de ces routeurs:

***CiscoModele2611\_IOS113***

SupporteQoS: oui

MéthodeDeConfiguration: telnet

ScriptQoS: commande1

commande2

...

**commandeX \$ParamètresQoS**

**(autres paramètres de profil de routeur)**

Par ce profil, on connaît la façon de configurer les routeurs qui possèdent ces mêmes caractéristiques. Les paramètres du profil de service sont passés au script (les commandes 1,2,...,X) pour activer le service avec ces paramètres.

### **3.2.4 Procédure d'activation du service**

En reprenant l'exemple du paragraphe précédent (3.2.3), voici les étapes qui décrivent comment le service sera activé en pratique:

- a. le *Scheduler* va déclencher à l'heure prévue la procédure d'activation du service,
- b. la procédure d'activation aura pour tâches:
  1. déterminer le service à activer : dans ce cas QoS,
  2. extraire les paramètres à activer : réserver 5000kbps sur les routeurs,
  3. extraire les paramètres utiles sur le prochain routeur (IP et profil),
  4. déterminer si le routeur (par le *DeviceProfile*) supporte ce protocole,
  5. si oui, dans le profil de ce routeur exécute le script qui correspond au service (ici QoS) avec les paramètres extraits en 2,
  6. optionnel : prendre un log des activités pour reporter à l'administrateur,
  7. recommencer le processus à l'étape 3 tant qu'il reste des routeurs à configurer sur ce profil.
- c. reprendre à l'étape 1 pour l'activation des autres profils si nécessaire.

### 3.2.5 La définition des autres classes LDAP

Dans cette section, on présentera les autres classes impliquées dans l'application du service QoS et qui nous aideront à mieux comprendre tous le processus de fonctionnement. Par la suite, on considérera le routeur comme équipement cible, sachant que la procédure s'applique de la même manière que pour un autre équipement, un commutateur par exemple.

#### a) La classe *DeviceContainedAux*

Pour attacher un routeur à une politique, une classe auxiliaire a été ajoutée:



Figure 13 Classe *DeviceContainedAux*

Cette classe auxiliaire est destinée à être ajoutée principalement à la classe *Policy*. Cette fonctionnalité existe déjà dans le serveur LDAP et l'éditeur de politique. L'attribut *DeviceContainedRef* pointe sur un élément de type *NetworkElement*. L'attribut *DeviceGroupContainedRef* pointe sur un objet de type *GroupOfNames*, qui est utilisé pour regrouper plusieurs routeurs dans une même entité.

### b) La classe *GroupOfNames*

Le *GroupOfNames* est une classe de X.500 qui est définie comme suit:

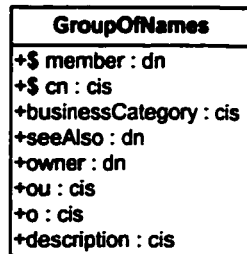


Figure 14 Classe *GroupOfNames*

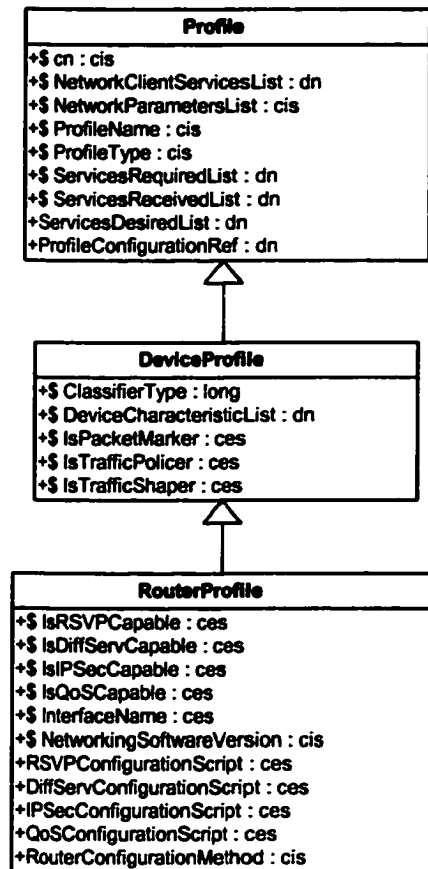
Nous utilisons les attributs *cn*, qui représentent le nom de l'objet, *member* qui regroupe les éléments de *NetworkElement* de ce groupe et leurs descriptions.

### c) La classe *RouteurProfile*

Le profil du routeur est donné par l'extension à DEN de la classe *DeviceProfile* de la figure 15.

Un *RouteurProfile* est associé à un *NetworkElement*. Il permet d'en connaître les fonctionnalités. Les attributs *IsRSVPCapable*, *IsDiffServCapable* et *IsQoS capable* permettent de savoir si le routeur supporte le protocole en question. L'attribut *InterfaceName* permet de connaître les interfaces à configurer sur le routeur. *RSVPConfigurationScript*, *DiffServConfigurationScript* et *QoSConfigurationScript* sont des attributs donne le script à utiliser sur ce type de routeur pour activer ce service. Si le service n'est pas activable sur ce routeur, l'attribut est simplement laissé vide. L'attribut *RouteurConfigurationMethod* détermine la façon de configurer le routeur, soit telnet, SNMP, etc.



Figure 15 Diagramme de la classe *RouteurProfile*

#### d) La classe *ProfileContainedAuxClass*

Permet d'attacher un attribut supplémentaire à une autre classe pour lui ajouter un profil:

Figure 16 Classe *ProfileContainedAuxClass*

Son seul attribut permet de faire une référence à un objet de type profil. Il est utilisé pour ajouter un profil de routeur aux éléments de *NetworkElement* qui représenteront ces derniers.

#### **e) Diagramme global des classes de DEN du gestionnaire de politiques**

Nous représentons dans la figure 17 l'aspect général du modèle d'information de notre gestionnaire de politiques.

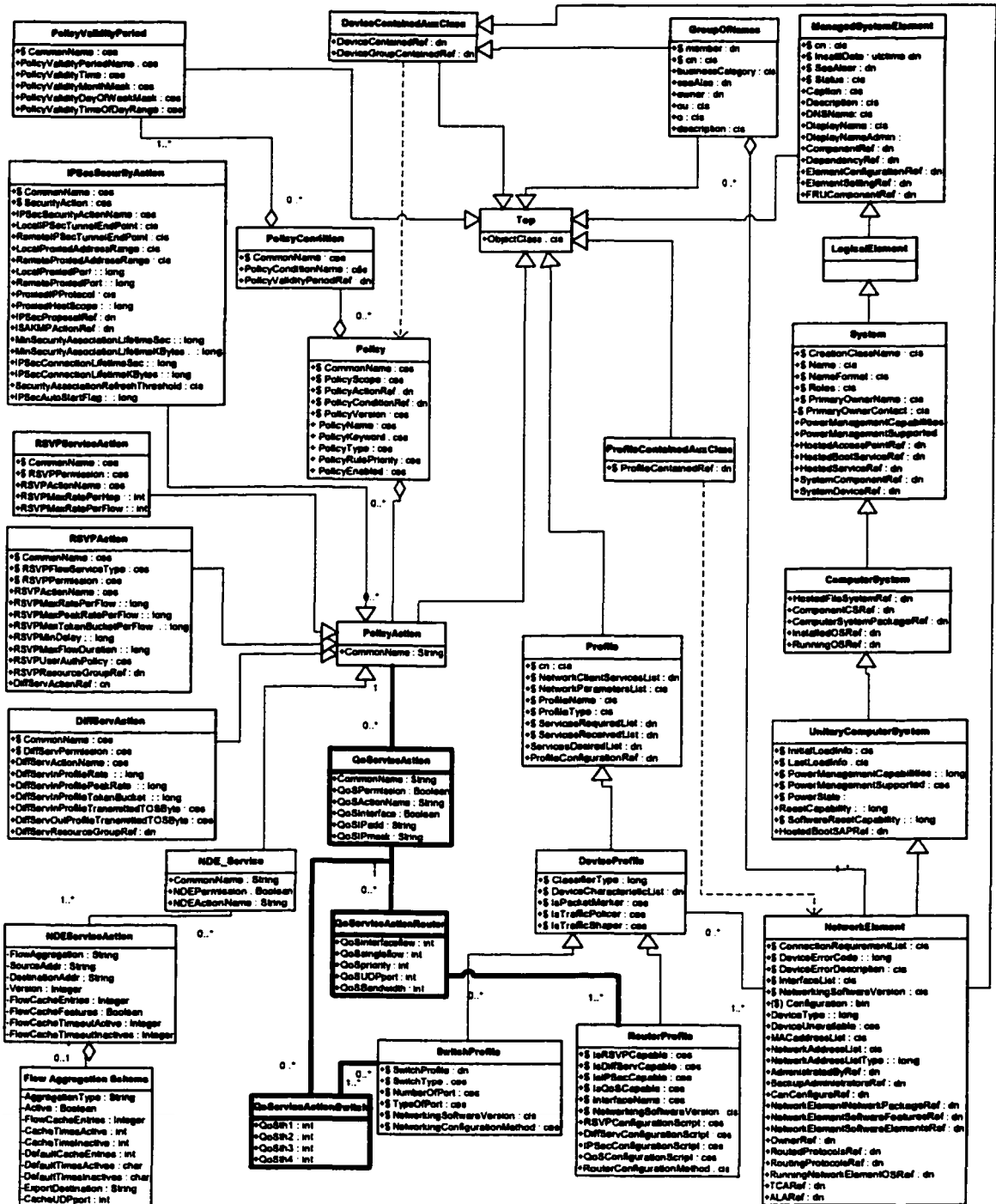


Figure 17 Schéma global de DEN pour le gestionnaire de politiques  
 - Le profil développé en trait gras

### 3.3 Présentation exhaustive d'un exemple pratique

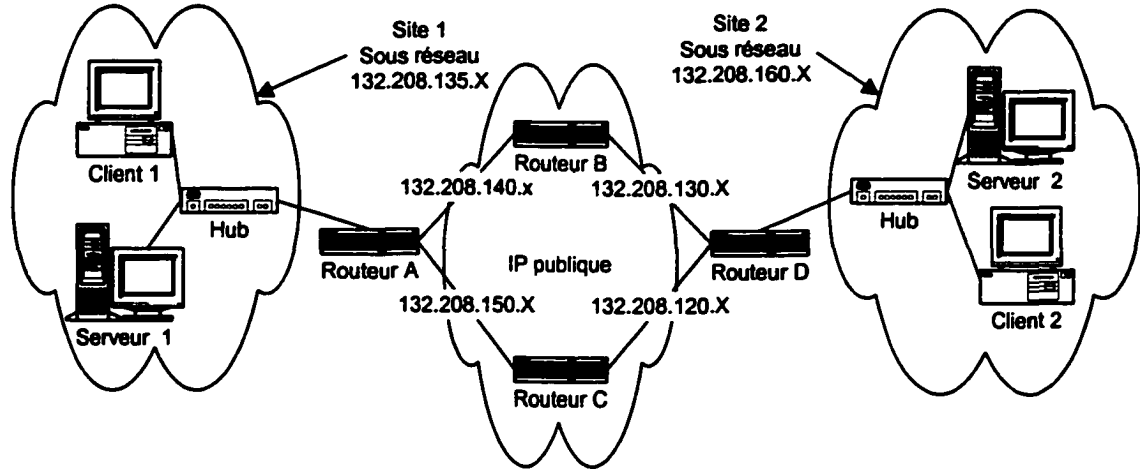


Figure 18 Activation du service QoS via une politique

Dans le tableau suivant, on inscrit les paramètres des composants de l'architecture de la figure 18.

Tableau V

Les paramètres des constituants de la figure 18

Équipement	Adresse IP	Description
Routeur A	132.208.135.240 132.208.140.1 132.208.150.1	Ce routeur est l'interface entre le sous réseau du site 1 et les deux sous-réseaux 132.208.140.X et 132.208.150.X. C'est un routeur d'accès.
Routeur B	132.208.140.240 132.208.130.240	Ce routeur est l'interface entre les sous réseaux 132.208.140.X et 132.208.130.X.
Routeur C	132.208.150.240 132.208.120.240	Ce routeur est l'interface entre les sous réseaux 132.208.150.X et 132.208.120.X.
Routeur D	132.208.160.240 132.208.130.1 132.208.120.1	Ce routeur est l'interface entre le sous réseau du site 2 et les deux sous-réseaux 132.208.130.X et 132.208.120.X. C'est un routeur d'accès.
Client 1	132.208.135.50	Client 1 situé au site 1.
Serveur 1	132.208.135.60	Serveur 1 situé au site 1.
Client 2	132.208.160.50	Client 2 situé au site 2.
Serveur 2	132.208.160.60	Serveur 2 situé au site 2.

### 3.3.1 Applications de la politique

Nous allons ici expliquer comment la politique QoS peut être appliquée en pratique. Nous avons une politique de gestion qui doit être exécuté à un horaire particulier. Lorsque le *Scheduler* détermine qu'il est temps d'activer une politique, il utilise l'attribut *PolicyScope* de la classe *Policy* pour déterminer à quel service s'applique la politique. Dans ce cas, la politique s'applique uniquement au service QoS. Le processus doit alors prendre en considération l'action (profil de service) à activer qui, dans le cas de QoS, est déterminé par la classe *QoServiceAction*.

Il doit alors appliquer ce profil de service aux équipements qui sont attachés à la politique et définis par les attributs *DeviceContainedRef* et *DeviceGroupContainedRef*. Le processus doit à ce moment exécuter le traitement sur chacun des routeurs de la liste, prenons exemple avec le routeurA. Il doit donc regarder l'attribut *ProfileContainedRef* pour déterminer le profil d'équipement qui s'applique à ce routeur. On sait que cette politique s'applique à QoS, alors le processus déterminera si ce service est supporté par cet équipement en regardant si le champ *IsQoSCapable* est *TRUE*. S'il est faux, le processus pour cet équipement s'arrête là et passe au suivant. Dans notre cas, le routeurA supporte effectivement QoS. Le champ *RouterConfigurationMethod* nous indique que ce routeur se configure par le protocole *telnet*. Le processus doit faire alors exécuter le script *QoSConfigurationScript* sur chacune des interfaces définies dans le *RouterProfile*. Les attributs précédés d'un \$ devant seront remplacés par leur valeur équivalente dans les classes qui les contiennent.

Quand le processus est complété, le service sera alors actif sur tous les équipements qui font partie de la liste du routeur à laquelle s'applique la politique. Si la politique s'appliquait aussi à *DiffServ* (donc mentionnée sur le *PolicyScope*), le processus aurait également effectué les même étapes (en plus de celle pour QoS) pour ce service. Voir l'annexe pour les détails sur cette exemple.

### 3.4 Classe du profil QoS dans LDAP

Une classe est un modèle sur lequel est basée une entrée. La classe détermine les attributs que doit posséder une entrée. Les attributs d'une classe peuvent être soit obligatoires ou encore elles peuvent être permises. Il est possible de restreindre l'accessibilité de certains attributs d'une classe d'après des conditions.

Les classes peuvent hériter d'autres classes. La façon de faire se présente seulement lors de la déclaration d'une entrée. C'est par l'attribut *objectclass* qu'on peut déterminer l'héritage. Les classes doivent être définies dans un fichier de configuration avant de pouvoir insérer une entrée dans le répertoire.

Le fichier de configuration qui permet de définir une classe doit contenir le nom de la classe, les attributs obligatoires et les attributs permis.

Voici la classe *QoServiceAction* et ses attributs écrits sur le serveur LDAP: elle contient les mêmes informations (attributs) que son modèle déjà décrit sur le schéma d'information de DEN et elle conserve aussi les mêmes liens qu'elle a avec les autres classes.

#### ## Déclaration des attributs de la classe QoS

attribute	QoSPermission	ces	single
attribute	QoSInterface	ces	single
attribute	QoSIPadd	ces	single
attribute	QoSIPmask	ces	single
attribute	QoSinterfaceflow	ces	single
attribute	QoSsingleflow	ces	single
attribute	QoSPriority	ces	single
attribute	QoSUDPport	ces	single

attribute	QoSBandwidth	ces	single
attribute	QoSth1	ces	single
attribute	QoSth2	ces	single
attribute	QoSth3	ces	single
attribute	QoSth4	ces	single

**## Présentation de la classe QoS**

**objectclass QoServiceAction**

**superior PolicyAction**

**requires**

**cn,**

**QoSpermission,**

**Allow**

**QoSInterface,**

**QoSIPadd ,**

**QoSIPmask,**

**QoSinterfaceflow,**

**QoSsingleflow,**

**QoSpriority,**

**QoSUDPport,**

**QoSbandwidth,**

**QoSth1,**

**QoSth2,**

**QoSth3,**

**QoSth4**

**## : précède l'écriture d'un commentaire sur LDAP**

**ces : *case exactmatch sensitive***

## **3.5 Implémentation orientée objet**

### **3.5.1 Langage JAVA**

Langage de programmation orienté objet mis au point par Sun Microsystems, et basé sur la réutilisabilité et la simplicité de mise en oeuvre. Java (qui veut dire café en argot américain) est un langage de programmation, proche de C++ dont on aurait retiré les incohérences et surtout les pointeurs. Ce langage permet à la fois de créer des programmes classiques et des applets intégrables à des pages Web. L'applet est téléchargé et exécuté par le navigateur web, offrant ainsi une dimension dynamique à une page HTML.

Les caractéristiques du langage Java sont :

- **Orienté objets**
- Intègre l'encapsulation, l'héritage (mais pas l'héritage multiple), la liaison dynamique, possède d'origine une librairie de classes standards (qui intègre notamment des classes gérant l'interface graphique utilisateur)
- **Portabilité** : Indépendant de la machine sur laquelle on exécute le programme
- Une source Java pré-compilée donne des "bytes codes", sorte de pseudo-assembleur qui s'adresse à la machine virtuelle Java (JVM). Cette JVM interprète ensuite ce byte code pour exécuter le programme. C'est le byte code qui est portable ; le tout est donc d'écrire un JVM propre à la machine. Les logiciels de navigation compatibles Java intègrent l'interpréteur de bytes codes qui simule cette machine virtuelle. Les applications ou applets Java sous forme de bytes codes, sont donc indépendantes de la machine physique.



- **Multi-tâche.** Un programme Java peut lancer plusieurs tâches qui s'exécutent simultanément. Java gère aussi le mécanisme des moniteurs, qui permet de synchroniser les "threads".
- **Sécurisé et fiable.** Java intègre dès la conception plusieurs mécanismes de sécurité, visant à rendre les programmes fiables et à éliminer les risques de virus.

Java est devenu très populaire et est en passe de remplacer C++. Il est toutefois menacé depuis peu par C# de Microsoft (qui doit encore faire preuve de la portabilité du langage).

### **3.5.2 JNDI (*Java Naming and Directory Interface*)**

JNDI est une extension standard de la plate-forme Java pour la recherche de ressources sur le réseau (service d'annuaire). JNDI se positionne comme une couche d'abstraction permettant d'accéder à diverses implémentations d'annuaires sous-jacents, tels que LDAP, COS Naming Service (CORBA), DNS, NIS, etc.

L'API JNDI se situe au-dessus de l'API SPI (*Service Provider Interface*) permettant à n'importe quel fournisseur d'interfacer son implémentation d'annuaire avec JNDI, voir figure 19.

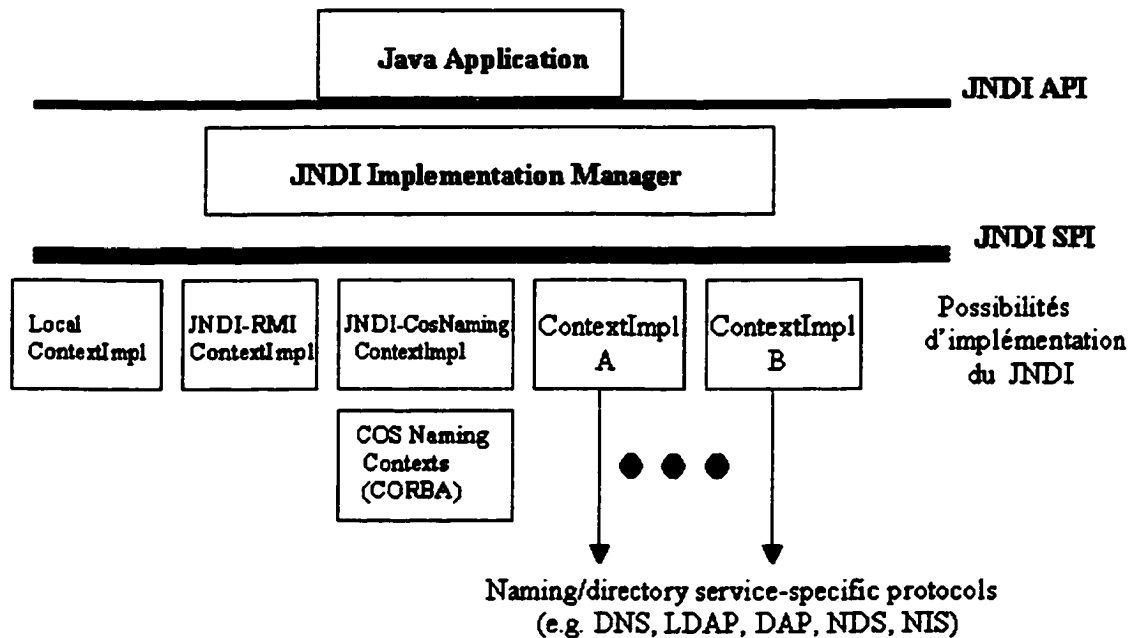


Figure 19 Implémentation des API JNDI

a- fonctionnalités

- Stockage et récupération de tout objet Java au travers du réseau,
- Opérations d'annuaire standard (association d'attributs à des objets, recherche d'objets répondant à tel ou tel critère),
- Support de divers protocoles implémentés par de divers fournisseurs :
  - JavaSoft RMI (*Remote Method Invocation*) Registry,
  - JavaSoft LDAP 3.0 (*Lightweight Directory Access Protocol*),
  - DNS (*Domain Name Server*),
  - JavaSoft NIS,
  - JavaSoft NIS+,

- **Novell NDS, Netware 3X's Bindery, Système de fichiers Novell, Extended NCP, etc,**
- **JavaSoft *COS Naming Service* (CORBA),**
- **JavaSoft *Système de fichiers* (accès aux fichiers du système via JDNI),**
- **SLP (*Service Location Protocol*),**

**b- avantages**

- **Indépendant d'un protocole d'annuaire spécifique,**
- **Abstraction objet,**
- **Bonne adaptation (utilisation de nouveaux protocoles implémentés sans modification de l'applicatif),**
- **Utilisation de différents protocoles en parallèle,**

**c- inconvénients**

- **Disponibilité progressive des protocoles utilisables avec JNDI.**

### **3.5.3 Intégration au PE des classes *JAVA* du service QoS**

L'application est implantée en langage *JAVA* pour la portabilité qu'il nous offre. Voici les étapes de la procédure d'implantation du QoS.

Au niveau du *Policy Editor*, nous avons dû rajouter des classes et faire les changements suivants :

1. ajout de la classe *QoServiceAction* :
  - Qui permet l'écriture sur LDAP des données de configurations entrées par l'utilisateur.
  
2. construction d'une *frame* pour la classe *QoServiceAction* :
  - Cette classe contient l'interface graphique d'utilisateur qui permet l'entrées les données des paramètres à mettre en place.
  
3. au niveau de la classe *PolicyAction* :
  - Ajout d'une constante qui représente l'objet *QoS\_service*,
  - Modification des méthodes *GetCN* et *GetDN* afin de récupérer respectivement le *CommonName* et *DomainName*.
  
4. ajout de QoS dans la classe *NewServiceDialog* :
  - Modification de la méthode *Private NewServiceDialog*,
  - Ajout d'un choix pour *NewServiceQoS*,
  - Modification de la méthode *OkButton\_actionPerformed* :
    - i. action à effectuer dans le cas où le bouton «ok» serait sollicité et que le choix effectué est *NewServiceQoS*,
    - ii. permet l'affichage de l'interface QoS .
  
5. dans la classe *RouteurProfile*:
  - Ajout d'une constante et de 2 méthodes *setIsQoSizable* et *getIsQoSizable* pour vérifier ou faire un changement si le routeur ne fait pas de la qualité de service.
  
6. dans la classe *DeviceFrame*:
  - Ajout de l'onglet *IsQoSizable* pour activer ou désactiver le service sur l'équipement.
  
7. dans la classe *Schedulerdaemon*:
  - Ajout de la méthode *readQoServiceAction* pour lire du serveur LDAP les données

des paramètres du service QoS,

- Modification de la méthode *readRouterProfile*,
- Modification de la méthode *getAttributeValue*,
- Modification de la méthode *enablePolicyDevice* permet d'activer ou désactiver les services de QoS pour le routeur et le commutateur;
- Ajout de la méthode *runTelnetRouter*. : Cette méthode permet de configurer en telnet le routeur afin qu'il puisse faire du QoS selon les paramètres entrés par l'utilisateur. Aussi, ajout de la méthode *runTelnetSwitch* : cette méthode permet de configurer en telnet le commutateur afin qu'il puisse faire du QoS selon les paramètres choisis par l'utilisateur.

#### 8. dans la classe *PolicyFrame*

- Modification de la méthode *CreatePolicyPane*.
- Ajout de l'élément QoS;

#### 9. dans la classe *PolicyEditor*

- Ajout de la méthode *displayQoSAction* : permet d'afficher les informations du service QoS dans la partie droite de la fenêtre principale,
- Modification de la méthode *updateDisplayPane* : permet entre autres d'afficher l'interface QoS,
- Modification de la méthode *editElementAction* : permet d'afficher entre autres un nouveau service QoS,
- Modification de la méthode *displayRouteurProfile* : permet d'afficher les informations provenant de LDAP dans le Jtree,
- Modification de la méthode *updateRouteurProfile*: permet le profil du Routeur dans la partie droite de la fenêtre principale,
- Modification de la méthode *updateServiceProfile* : permet de mettre et d'afficher les informations provenant de LDAP dans le Jtree de la fenêtre principale.

### **3.6 Conclusion**

Dans ce chapitre, nous avons montré comment le service QoS a été rattaché au modèle d'information DEN qui, quant à lui, décrit toutes les classes que composent notre *Policy Editor* ainsi que le rapport des liens entre eux. On a expliqué le mode de fonctionnement par des exemples. Par la suite, on a cité les étapes de l'implantation dans le cadre d'une méthodologie orientée objet et un codage en langage informatique *JAVA*. L'architecture et les tests de ces composants seront présentés dans le chapitre qui suit.

Donc, dans le chapitre suivant, nous présenterons les fonctionnalités du gestionnaire de politiques ainsi que les tests de la mise en œuvre du service QoS à travers le schéma d'une architecture réseau de simulation qui permettra de mettre en place les différents composants présentés dans le chapitre précédent.

## CHAPITRE 4

### ARCHITECTURE ET FONCTIONNALITÉS DU MODÈLE

#### 4.1 Introduction

Dans le cadre de ce projet, l'équipe du laboratoire de recherche en téléinformatique de l'UQAM a développé un nouvel outil de gestion appelé gestionnaire de politiques, qui permet d'éditer des politiques réseaux rattachées à un facteur temporel (condition). Le but est de permettre au gestionnaire du réseau d'établir et d'offrir toutes les politiques réseaux nécessaires au bon fonctionnement des applications, et ce, sans intervention directe du gestionnaire sur l'équipement. Essentiellement, l'idée de son *Framework* (un ensemble de classes) est très simple, car il identifie deux composants primaires par leur fonctionnalité. Il est composé d'un *Policy Editor (PE)* et d'un *Policy Enforcement Tool (PET)* qui sont dans un sens le policier et le juge. Le PE est un renforceur, il définit la politique et le PET, prend des décisions basées sur les politiques qu'il retrace à partir du dépôt de politique ou peut-être dans d'autres emplacements tels des serveurs (d'authentification).

Ce chapitre présente les résultats de l'intégration du nouveau service QoS à l'architecture du prototype de notre gestionnaire de politiques et les diagrammes d'utilisation et de fonctionnement. Il porte sur le nouveau modèle de données et sur les interfaces développées.

#### 4.2 Composition du gestionnaire

Le gestionnaire de politiques qui est l'application permettant l'administration de la gestion de réseau dynamiquement par le biais de politique est constitué par deux entités qui sont :

#### **4.2.1 Éditeur de politiques(PE)**

L'éditeur de politiques est une application qui permet l'administration de la gestion de réseau dynamiquement. L'administrateur qui utilise cette application définit tous les équipements qui constituent son ou ses réseaux ainsi que les personnes qui en sont les utilisateurs. Ensuite, l'administrateur n'a qu'à définir un service à implanter (par exemple, un VLAN) avec tous les paramètres nécessaires et par la suite définir la politique qui permettra de savoir quel service déployer, sur quel équipement et quand il sera déployé. Finalement, pour implanter la politique, l'administrateur utilisera l'exécuteur de politiques (PET) qui fera un échange via SNMP ou Telnet de la station de gestion à l'équipement réseau

#### **4.2.2 Exécuteur de politiques (PET)**

L'exécuteur de politiques est l'application qui permettra la mise en œuvre des politiques définies antérieurement sur le serveur LDAP à l'aide de l'éditeur des politiques. De plus, il se devra de valider les politiques définies par l'administrateur sur la console de gestion. Donc, il activera les politiques en place sur les équipements réseau comme les routeurs.

Deux moyens sont utilisés pour parvenir à nos fins. Tout d'abord, il y a l'exécution des requêtes SNMP qui iront modifier directement les MIBs de l'élément réseau à configurer. Donc, LDAP sera utilisé pour aller chercher les configurations et ensuite les appliquer sur les équipements. Les requêtes "get" pourront être d'utilité en permettant d'interroger l'équipement afin d'en connaître la configuration actuelle.

Ensuite, il y a l'interface CLI utilisant Telnet. L'exécuteur des politiques ira chercher les paramètres pour activer un service sur un équipement et ensuite enverra la commande de



configuration directement sur l'élément réseau pour forcer l'application de la politique réseau.

Donc, de cette manière, les politiques implantées sur les équipements permettront à ces derniers de faire la gestion du trafic selon les critères de l'administrateur lors de la définition de ses politiques à l'aide de l'éditeur.

La séparation entre le PET et le PE (et également du dépôt de politique) est logique, basée sur la fonctionnalité, et n'est pas nécessairement une séparation physique. En d'autres termes, la séparation n'exclut pas la possibilité de pouvoir les installer sur le même dispositif.

### 4.3 Le service QoS dans un domaine administratif

Tout d'abord, la première étape consiste à lancer l'application *Policy Editor* installée sur le poste de l'administrateur du réseau. Ce programme détecte et établit un mode de connexion avec le serveur LDAP via le protocole LDAP, en utilisant des classes de l'API JNDI (vu que tout le code du programme est écrit en langage *JAVA*) qui lui permettent de faire des requêtes d'information et d'opération (figure 20).

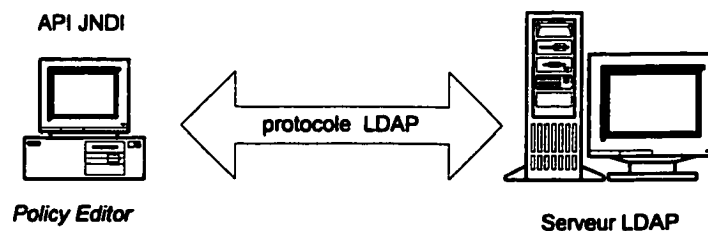


Figure 20 Type de requête échangée entre le PE et le serveur LDAP

Ce programme permet généralement d'une manière transparente de créer et de déployer des services sur des équipements à l'aide de politiques.

L'utilisateur pourra simuler et gérer les événements avec des interfaces graphiques lui permettant d'interagir avec le programme.

Le protocole LDAP doit être déployé au niveau du poste du gestionnaire de politiques et du serveur, ce dernier permet à chaque signalisation de gestionnaire de politiques de faire authentifier les entrées sur les usagers et les politiques, il s'agit là de définir le type d'opération à exécuter et la condition à satisfaire, comme présentées dans le tableau VI.

**Tableau VI**  
**Les requêtes du protocole LDAP**

Opération	Conditions
Ajouter	<ul style="list-style-type: none"> <li>. Le parent de la nouvelle entrée doit déjà exister dans le répertoire.</li> <li>. Il ne faut pas qu'il y ait deux entrées du même nom</li> <li>. La nouvelle entrée doit être conforme au schéma (DEN)</li> <li>. Le contrôle d'accès doit permettre l'opération</li> </ul>
Effacer	<ul style="list-style-type: none"> <li>. L'entrée à effacer doit exister</li> <li>. Elle ne doit avoir aucun enfant</li> <li>. Le contrôle d'accès doit permettre l'effacement de l'entrée</li> </ul>
Renommé ou modifier	<ul style="list-style-type: none"> <li>. L'entrée à renommer doit exister</li> <li>. Le nouveau nom pour l'entrée ne doit pas être en utilisation par une autre entrée</li> <li>. Le contrôle d'accès doit permettre l'opération</li> </ul>

#### Côté administrateur

Le lancement du *Policy Editor* du côté de l'administrateur est nécessaire avant de s'engager à faire de la simulation du service QoS. L'activation se fait en exécutant un

fichier dont l'extension est du type .Jar ou .bat qui lui-même fait appel à la classe principal du *Policy Editor*

Il est important de vérifier aussi, l'authentification de l'utilisateur avant de permettre l'introduction des paramètres obligatoires pour les configurations désirées. Ce qui, bien entendu nous mène à une étape souvent évidente, décrite prochainement.

Lors de l'exécution de la classe *Policy Editor*, une boîte de dialogue invite l'utilisateur à se connecter au serveur LDAP via un mot de passe et un choix de méthode d'authentification pour fin de sécurité (figure 21).

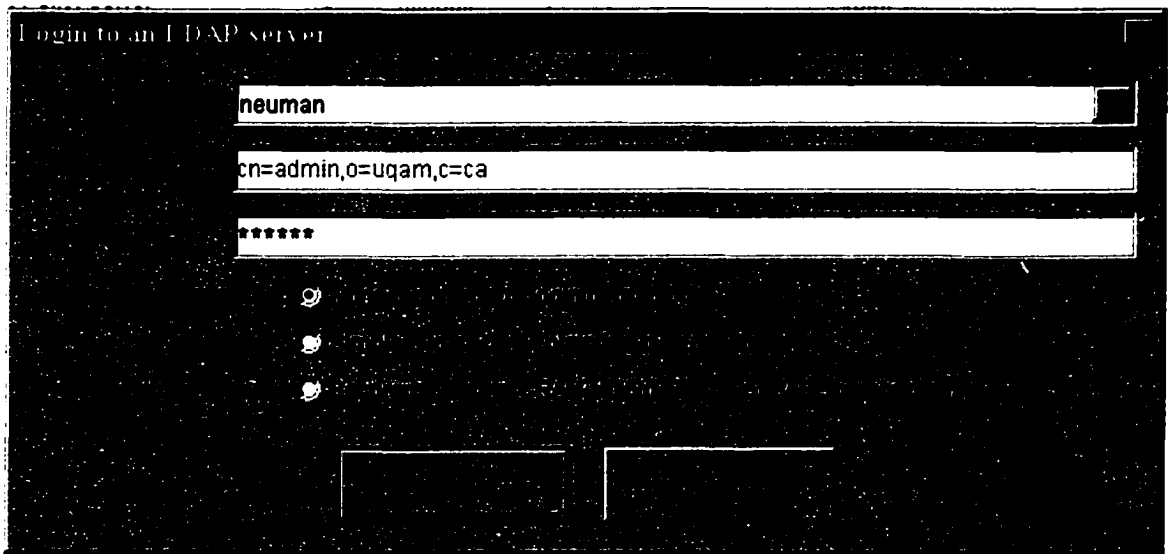


Figure 21 Interface d'authentification et de connexion

Toutes les informations seront envoyées au serveur, qui va détecter que la requête contient un message d'authentification, qu'il compare avec sa base de données pour vérifier sa validité (figure 22).

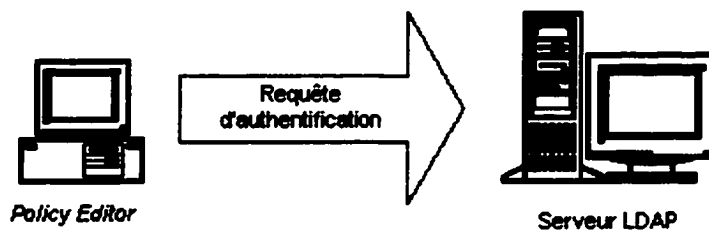


Figure 22 Message d'authentification

Si la réponse est positive, une boîte d'outils de l'interface du gestionnaire s'affiche, présentant toutes les informations et les services disponibles pour la gestion du réseau. Elle permet à l'utilisateur de faire une multitude d'opérations (Figure 23).

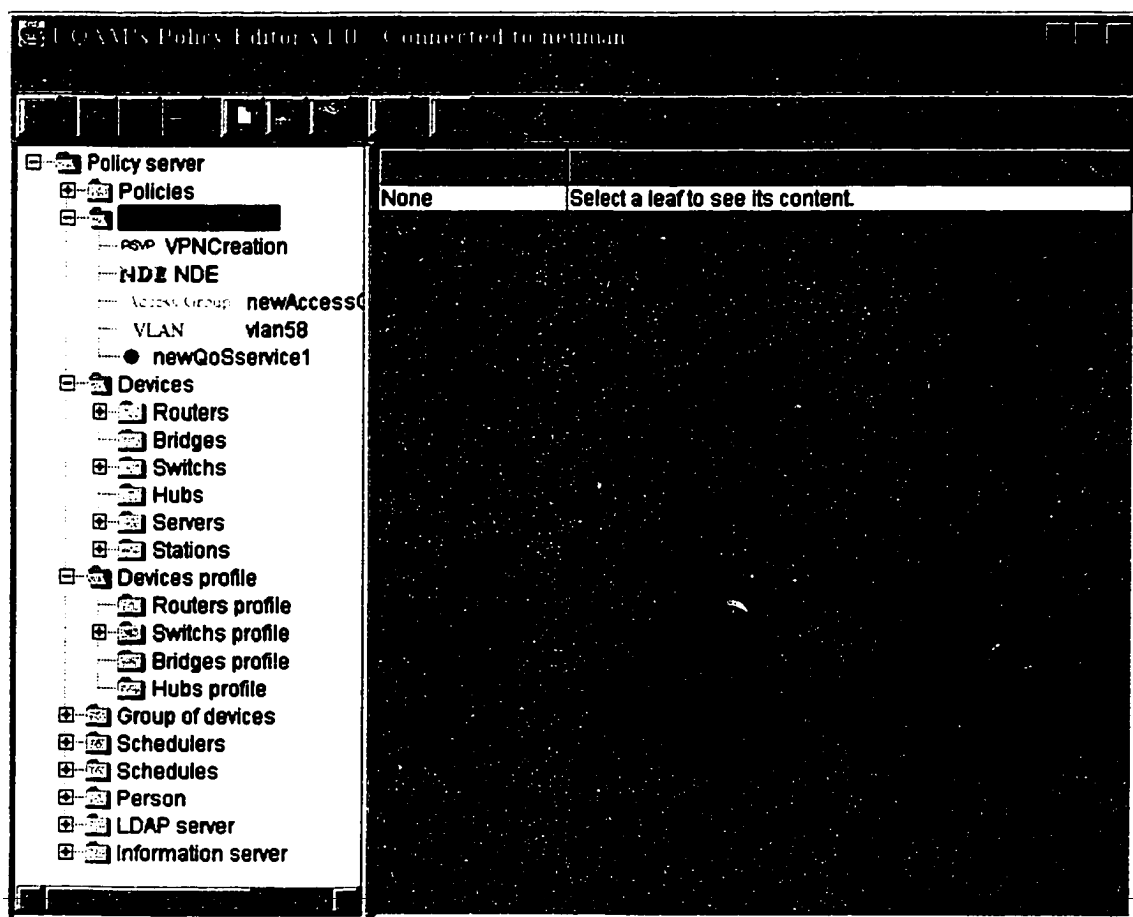


Figure 23 Fenêtre du gestionnaire

De là, une panoplie de choix de politiques, de services, de personnes etc. s'offre à l'utilisateur, mais dans notre cas et en liaison avec la figure 3, nous allons montrer comment établir le nouveau service QoS qu'on obtient en créant un nouveau profil du service QoS donné par la figure 24.

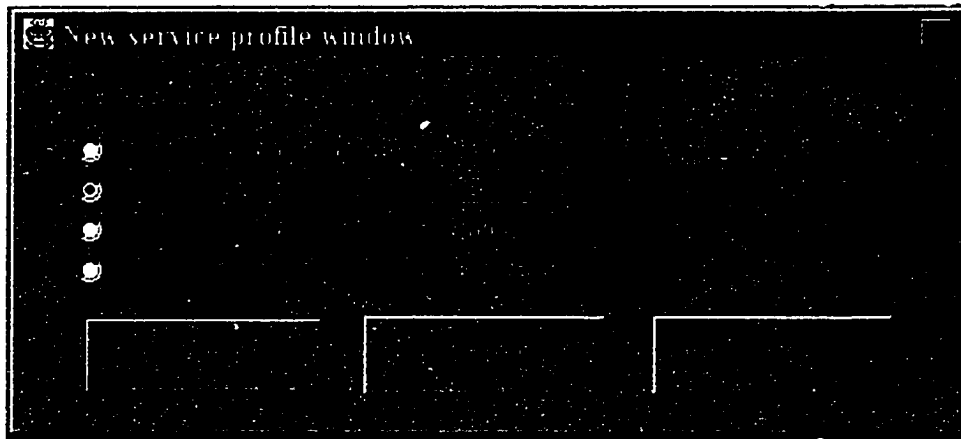


Figure 24 Création du nouveau service

À partir d'ici, il ne nous reste qu'à rajouter le profil du service désiré pour être pris en compte par le gestionnaire de politiques et ainsi l'appliquer aux équipements du réseau désignés par celui-ci.

Notre profil QoS nous donne une interface d'entrée des paramètres selon l'équipement sélectionné, qui est montré par la figure 25.

Figure 25 Interface d'entrée des paramètres du service QoS

Cette interface est structurée de façon à répondre aux besoins des équipements inclus dans l'architecture réseau et les paramètres nécessaires pour leurs configurations afin de répondre au besoin du service QoS. Les informations seront recueillies pour être enregistrées dans la mémoire du serveur LDAP pour un éventuel déploiement qui sera enclenché par l'horaire pré-défini sur le *Scheduler* ainsi que le *Schedule*.

#### 4.4 Synchronisation

La deuxième phase consiste au déploiement des ressources et des réservations sur le réseau à gérer. Cela est pris en charge par l'application *Policy Enforcement Tool*, qui démarre suivant l'horaire figurant au niveau du *Scheduler* (qui est un ensemble de *schedule*). Les requêtes seront envoyées par Telnet .

Voici les détails de ce que peut contenir un horaire (Figures 26 et 27)

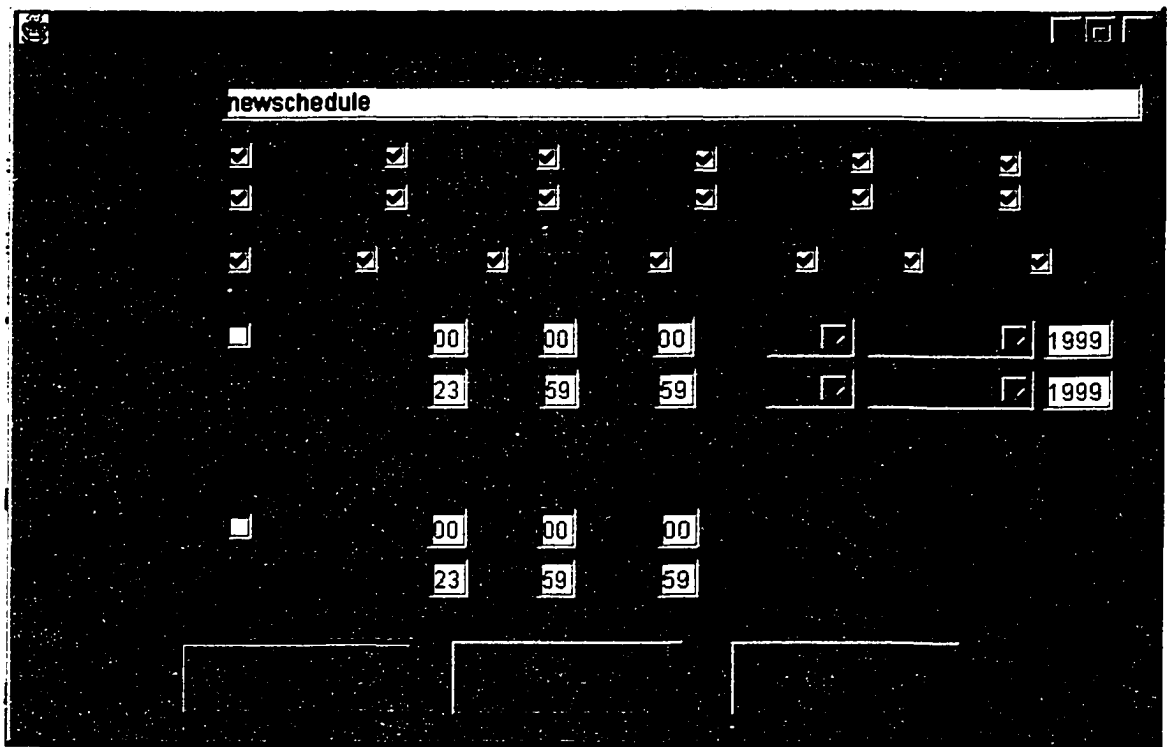


Figure 26 Fenêtre du *Schedule*

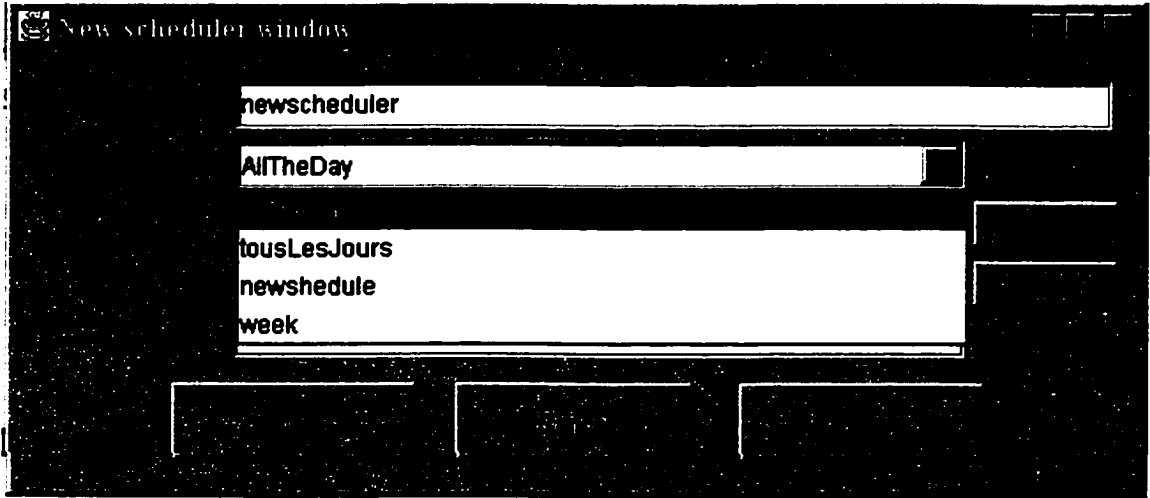


Figure 27 Fenêtre du *Scheduler*

#### 4.5 Déploiement des ressources

L'implémentation de la politique sur l'équipement se fait à l'aide d'un autre outil appelé *Policy Enforcement Tool*. Cet outil est chargé de vérifier si les politiques doivent être activées, et dans l'affirmative, il se charge d'appliquer la politique sur l'équipement. Figure 28.

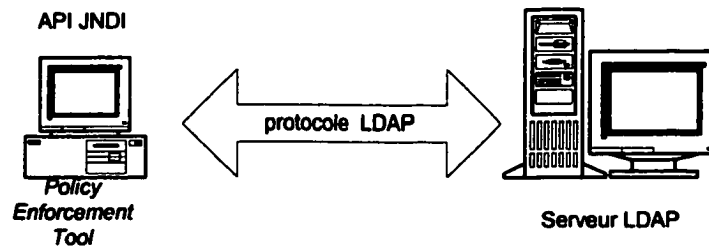


Figure 28 Activation de politique



À partir de ce stade, les commandes de configuration seront envoyées dynamiquement sur les équipements en utilisant l'application Telnet, comme montré sur la figure 29.

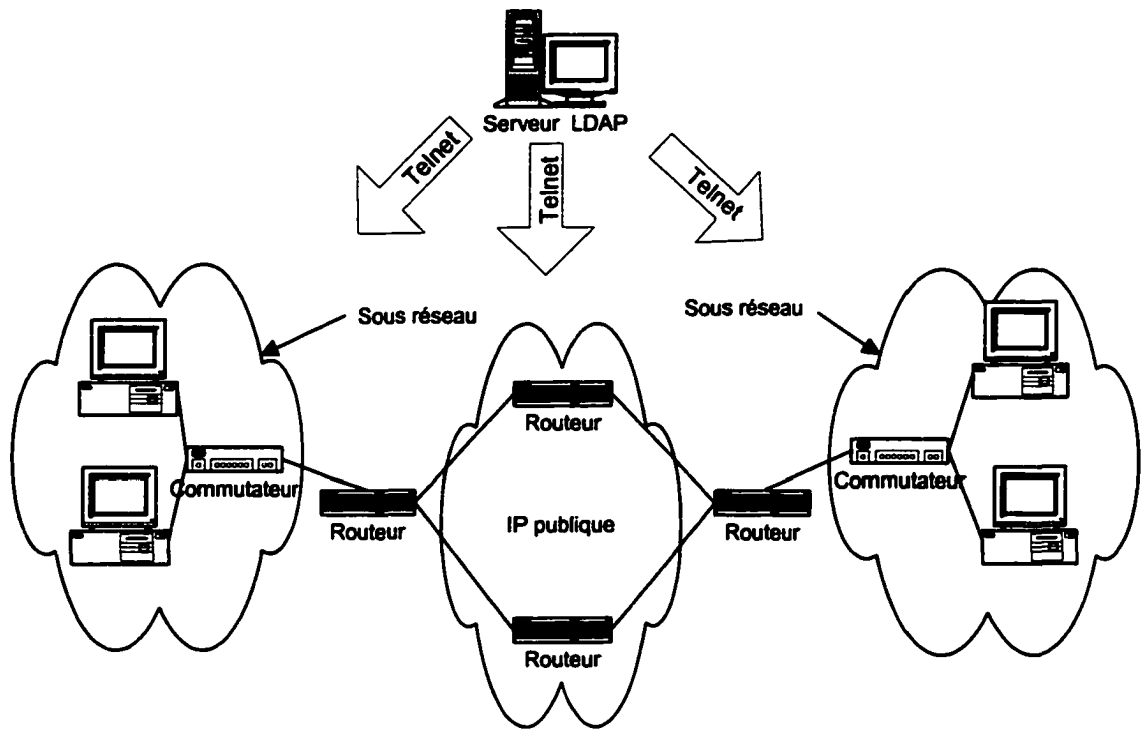


Figure 29 Mise en place des commandes de configuration

#### 4.6 Interface graphique des profils d'équipement

La création d'un nouveau profil routeur ou commutateur est donnée par l'interface graphique des figures ci-dessous. On retrouve tous les champs d'attributs déjà introduits et expliqués au chapitre 3.

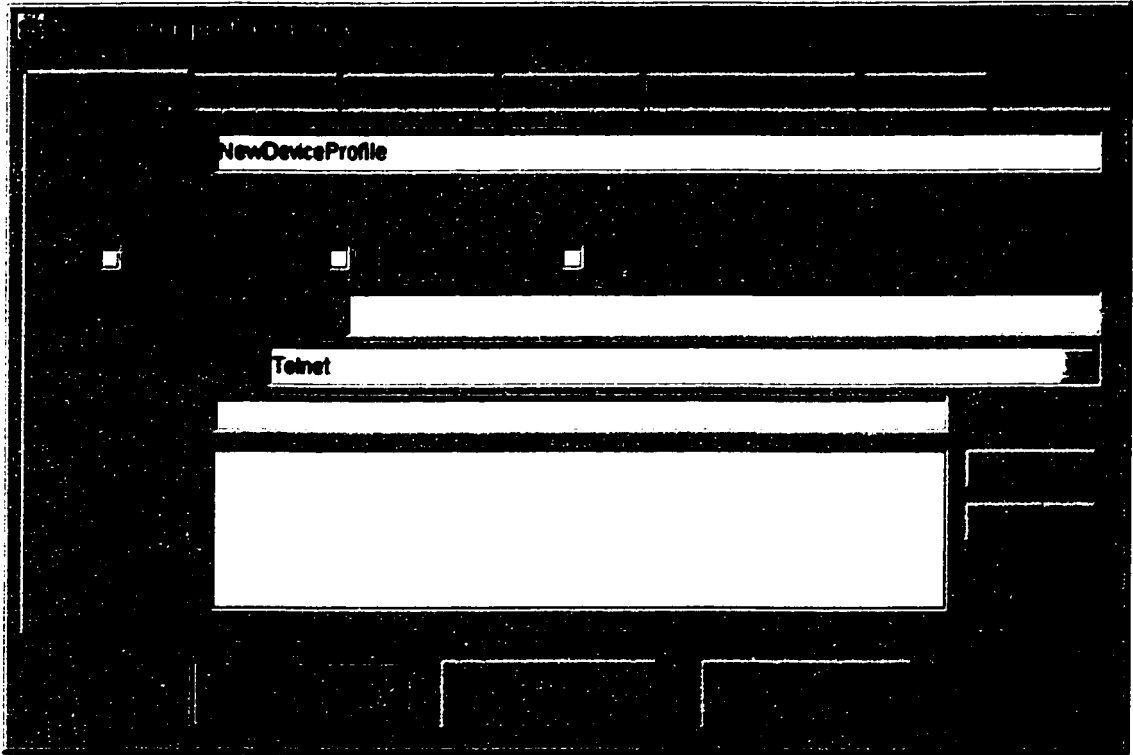


Figure 30 Interface du profil routeur

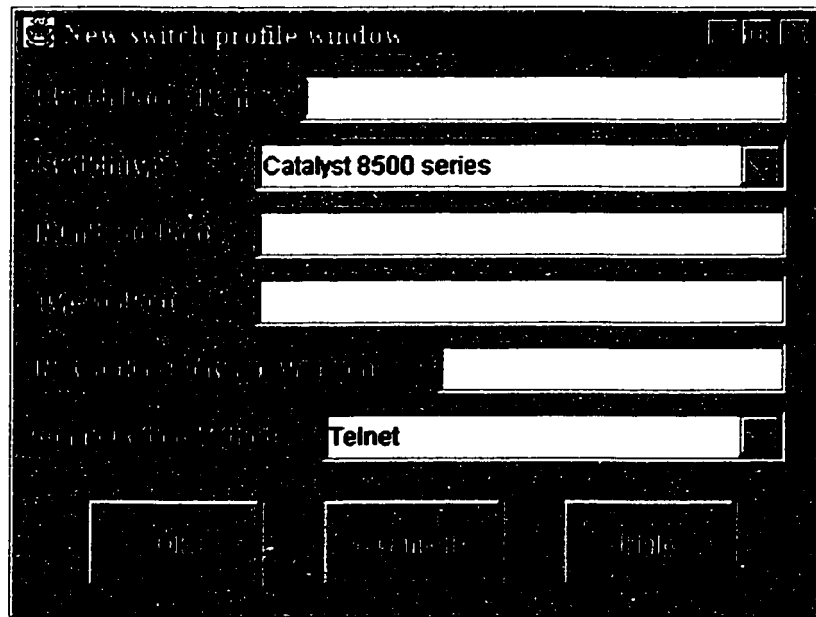


Figure 31 Interface du profil commutateur

#### **4.6 Conclusion**

**Nous avons montré et expliqué dans ce chapitre l'architecture, le principe et le mode de fonctionnement du service QoS au sein de l'application du gestionnaire de politiques. Nous avons présenté les interfaces graphiques mises à la disposition de l'administrateur par lesquelles il va interagir avec son environnement de réseau de télécommunication. Le test de ces composants sera arboré dans le chapitre qui suit.**

## **CHAPITRE 5**

### **MISE EN ŒUVRE ET EXPÉRIMENTATION**

#### **5.1 Introduction**

Afin de mettre en œuvre les différents constituants présentés dans les chapitres précédents, et qui représentent une partie fondamentale de notre démarche pour concrétiser l'implémentation de notre service de QoS, en premier lieu, nous abordons tour à tour la description du matériel utilisé, les multiples configurations proposées ainsi que l'utilisation proprement dite du programme via les interfaces graphiques.

Nous avons voulu procéder à plusieurs tests de fonctionnement et de performance à l'aide de certains scénarios pour sonder l'état des connexions et leurs composants qui vont constituer une partie du réseau où l'éventuelle plate-forme du projet "Sychromedia" va être déployée.

En deuxième lieu, nous avons conçu en laboratoire une série de modèles d'architecture réseau en s'appuyant sur les dispositifs disponibles et cela, en se basant sur les possibilités que peuvent offrir les commandes existantes à ce niveau-là, pour garantir la priorité du trafic audio/vidéo à travers le protocole 802.1p et les classes de service.

Dans cette section, nous présenterons les résultats qui ont été obtenus suite aux tests de validation effectués sur les équipements mis à notre disposition, ainsi que l'analyse de ces tests et des choix des paramètres de configuration.

## **5.2 Test d'évaluation au laboratoire**

L'idée principale derrière une politique de gestion est de fournir à un administrateur un accès spécifique aux ressources du réseau, et aux utilisateurs, une meilleure qualité de service.

En fait, l'application va être utilisée plus précisément par les interlocuteurs qui s'échangent des données multimédia. Donc, c'est principalement au réseau que revient la tâche de garantir les performances des communications à travers les possibilités et les performances des équipements qui le constituent. Alors, en partant de là, nous avons procédé à plusieurs essais en laboratoire pour qu'enfin nous aboutissions à un résultat qui sera détaillé dans le reste de ce chapitre. Au début, nous vous proposons l'architecture présentée dans ce qui suit.

### **5.2.1 Composition du modèle**

Avant de commencer à décrire notre modèle, il faut tout d'abord mentionner que le gestionnaire de politiques est installé sur un nœud du réseau pour la simulation. Il est à noter que l'installation du gestionnaire de politiques a été conçue sous forme d'un *daemon*.

Le modèle montré dans la figure 32 représente l'infrastructure qu'on a utilisé pour simuler notre modèle de contrôle de politique. Le système est implanté sur des postes de travail situés dans un même local, deux machines sont équipées de matériel audiovisuel, à savoir des caméras, des micros et des haut-parleurs, alors que deux autres machines servent à générer du trafic. L'ensemble est relié par une architecture de connexions commutateur/routeurs Ethernet. Les caractéristiques à prendre en compte dans cette analyse sont de deux types : d'une part, les caractéristiques architecturales, d'autre part les caractéristiques de performance.

### 1. les caractéristiques architecturales

- Hétérogénéité des équipements mis en jeu
- Disposition de ses équipements

### 2. les caractéristiques de performance

- Évolution du trafic (paquets UDP/RTP)

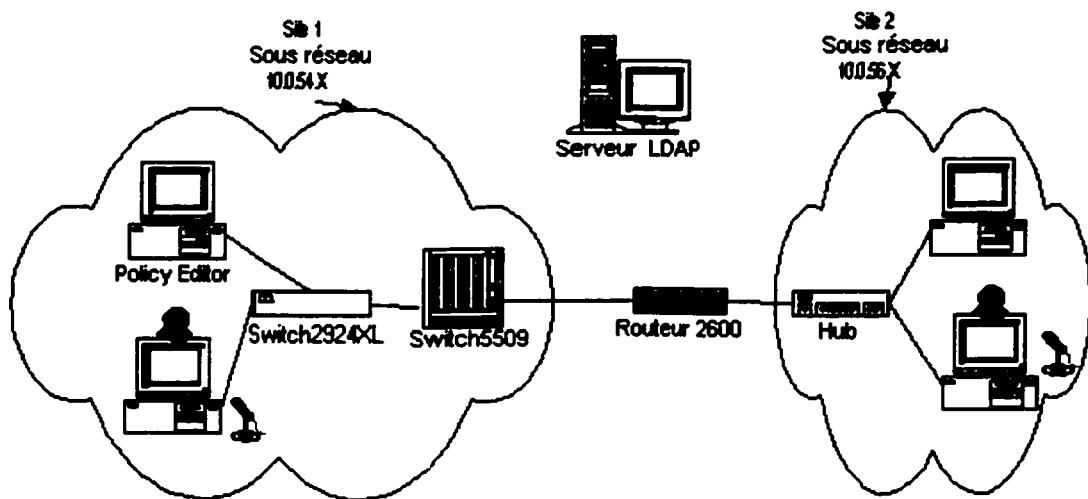


Figure 32 Modèle de simulation

### 3. équipements mis en jeu

	nombre
Cisco 2600 router Cisco IOS Software 12.0(7)T	1
Cisco2924XL switch Cisco IOS Software 12.0(5.2)XU	1
Cisco5509 switch Cisco IOS Software McpSW:5.2(2)	1
Stations PC	5

### Équipements multimédias

Seulement deux PC sont équipés de :

- Logiciel de support Mpat
- Carte réseau Intel PRO PCI
- Carte d'extension Mpat multimédia
- Carte sound blaster audio PCI 128
- Caméra, microphone et haut-parleurs

#### **5.2.2 Serveur LDAP**

L'aspect du serveur LDAP qu'on a décrit dans le chapitre précédemment est installé sur une station Unix pour simuler les événements des politiques proposés dans les profils de service.

#### **5.2.3 Générateur de trafic**

Nous avons utilisé un outil logiciel de tests appelé *Lantraffic V2* permettant de générer et de recevoir du trafic TCP sur un réseau en gérant jusqu'à 32 connexions simultanées.

C'est un générateur de trafic entièrement paramétrable, cet outil permet de réaliser des tests unitaires et des tests en charge d'un réseau avec des caractéristiques et des profils de charge très diversifiés.

Il est composé de 2 parties :

- L'émetteur ('Sender') assure la génération de trafic sur des connexions avec des profils de trafic paramétrables sur chaque connexion (jusqu'à 16 connexions simultanées).

- Côté récepteur ('Receiver'), chacune des 16 connexions est configurable selon différents modes : échoeur, absorbeur ou absorbeur fichier (les données reçues sont sauvegardées dans un fichier défini par l'utilisateur).

### 5.3 Paramètres de Configuration et commandes des équipements

Dans ce qui suit, nous exposerons les listes des commandes de configuration selon le type d'équipement mis en jeu dans notre modèle d'architecture de simulation :

#### 1. commutateur 5509

Pour indiquer les seuils de baisse de file d'attente de transmission sur tous les ports du commutateur, on utilise la commande suivante :

```
set qos wred-threshold port_type [rx | tx] queue {q# thr1 thr2 thr3 thr4}
```

#### Description de la syntaxe

<i>port_type</i>	Pour un type de port, une seule valeur est valide : 1q4t.
<b>rx</b>	(Facultatif) mot-clé pour indiquer l'alignement à l'entrée. Ce mot-clé n'est pas supporté.
<b>tx</b>	(Facultatif) mot-clé pour indiquer l'alignement de sortie.
<b>queue</b>	Mot-clé pour indiquer la valeur de la file d'attente.
<i>q#</i>	Nombre de file d'attente. La seule valeur valide est 1.
<i>thr#</i>	Valeur de seuil; les valeurs valides sont de 1 à 100 pour cent.

Ci-dessous, nous montrons comment configurer les seuils de baisse de file d'attente pour la transmission.

```
Console> (enable) set qos wred-threshold 1q4t tx queue 1 30 50 80 100
Transmit drop thresholds for queue 1 set at 30% 50% 80% 100%
Console> (enable)
```



Cet exemple montre comment afficher l'information de QoS d'une file d'attente de transmission:

```
Console> (enable) show qos info config 1q4t tx
```

```
QoS setting in NVRAM for 1q4t transmit:
```

```
QoS is enabled
```

```
Queue and Threshold Mapping:
```

```
Queue Threshold CoS
```

```
-----
```

```
1 1 0 1
```

```
1 2 2 3
```

```
1 3 4 5
```

```
1 4 6 7
```

```
Tx WRED thresholds:
```

```
Queue # Thresholds in percentage ( in abs values )
```

```
-----
```

```
1 14% 28% 80% 100%
```

```
Console> (enable)
```

## 2. routeur 2600

Pour réserver une file d'attente à stricte priorité pour un ensemble de flots de paquet RTP appartenant à un intervalle de port UDP de destination, on définit en premier lieu une configuration CBWFQ (classe à base de files d'attentes pondérées) selon les commandes suivantes dans un mode de configuration d'interface:

! Ces commandes définissent et créent une classe à tracer:

```
router(config)# class-map class1
```

```
router(config-cmap)# match access-group 101    index de la liste d'accés
```

! ces commandes crient et attachent une politique:

```
router(config)# policy-map policy1
```

```
router(config-pmap)# class class1
```

Activation du mécanisme WRED :

```
router(config-pmap-c)# random-detect
```

! cette commande applique la politique à l'interface :

```
router(config-if)# service-policy output policy1
```

! Cette commande réserve une file d'attente à stricte priorité:

```
router(config-if)# ip rtp priority 16384 16383 2000
```

La commande **random-detect** est une commande facultative pour les configurations de CBWFQ. Elle est utilisée pour configurer des limites de rejet des paquets pour une classe de file d'attente.

*Weighted RED (WRED)* avec la pondération gérée via le champ *IP Precedence*, permet au réseau de demander aux flux de trafic moins prioritaires de s'adapter au profit du trafic plus prioritaire. C'est un mécanisme qui empêche la congestion et évite le rejet des paquets par un ralentissement et une réduction du trafic.

*Weighted Fair Queuing (WFQ)* est un mécanisme de gestion des files d'attente pondérées par le champ *IP Precedence* qui permet au trafic prioritaire d'être mis en tête des files d'attente et donc d'avoir un meilleur temps de transmission.

### Commandes connexes

- **random-detect exponential-weighting-constant** : configure le facteur exponentiel du poids utilisé pour le calcul moyen de la taille de file d'attente
- **random-detect precedence** : configure des paramètres de WRED pour une priorité particulière du champ *IP Precedence*.

### 3. commutateur 2924XL

Pour établir une priorité aux trames entrant non-étiquetées ou la priorité des trames reçues par l'appareil relié au port indiqué, on utilise la commande de configuration d'interface :

```
Switch(config-if) # switchport priority {default default-priority-id | extend {cos value
| none | trust} | override}
```

#### Description de la syntaxe

<i>default-priority-id</i>	Valeur de la priorité pour le trafic non-étiqueté. La priorité est un nombre de 0 à 7. Sept est la priorité la plus élevée.
<b>extend</b>	Établir la priorité IEEE 802.1p de l'équipement. <ul style="list-style-type: none"> <li>• <b>cos value</b>— Réécrire la priorité 802.1p des appareils reliés à l'équipement. La valeur de cos est un nombre de 0 à 7. Sept est la priorité la plus élevée.</li> <li>• <b>none</b>— Il n'est pas indiqué à l'équipement quoi faire avec la priorité.</li> <li>• <b>trust</b>— Indique que l'équipement devrait faire confiance (honneur) à la priorité 802.1p reçue des appareils reliés à lui.</li> </ul>
<b>override</b>	Réécrire la priorité des trames étiquetées avec la valeur par défaut

#### Directives d'utilisation

La priorité par défaut du port s'applique si la trame entrante est non-étiquetée, et reçue d'un joncteur réseau (*trunk*) pour VLAN ou d'un port d'accès statique. Cette priorité du port ne s'applique pas aux trames étiquetées par des VLAN IEEE802.1Q ou du ISL. Si la trame entrante est une trame 802.1Q étiquetée, les bits prioritaires de l'utilisateur 802.1p seront utilisés.

La commande suivante montre comment configurer un port spécifique rattaché à un appareil pour honorer la priorité 802.1p reçue:

```
Switch(config-if)# switchport priority extend trust
```

#### 4. configuration d'un candidat SBM

Pour activer le protocole de réservation de ressource (RSVP) et le gestionnaire de largeur de bandes de sous-réseau (SBM) sur une interface, on utilise les commandes suivantes :

```
Router(config-if)# ip rsvp bandwidth [interface-kbps [single-flow-kbps]]
```

```
Router(config-if)# ip rsvp dsbm [priority]
```

#### Description de syntaxe

<i>interface-kbps</i>	(Facultative) Volume de largeur de bande (en kbps) à réserver sur une interface. L'intervalle est de 1 à 10000000.
<i>single-flow-kbps</i>	(Facultative) Volume de largeur de bande (en kbps) assignée à un flot simple. L'intervalle est de 1 à 10000000.
<i>priority</i>	Configurer une interface pour participer comme compétiteur au processus dynamique d'élection du DSMB, qui est basée sur la priorité la plus élevée

Dans l'exemple suivant, RSVP et SBM sont activés sur une interface Ethernet. Après que RSVP soit permis, l'interface est configurée en tant qu'un candidat a la fois DSMB et SBM avec une priorité de 100. La priorité configurée est élevée, faisant de cette interface un bon compétiteur pour le mode de DSMB. Cependant, la valeur configurable

maximale de priorité est 128, ainsi une autre interface configurée avec une priorité plus élevée pourrait gagner l'élection et devenir le DSBM.

```
interface Ethernet
ip address 145.2.2.150 255.255.255.0
no ip directed-broadcast
ip pim sparse-dense-mode
no ip mroute-cache
media-type 10BaseT
ip rsvp bandwidth 7500 7500
ip rsvp dsbm-candidate 100
end
```

#### Commande connexe

- **random-detect** : active le mécanisme WRED qu'on retrouve au niveau du commutateur 5509 et du routeur 2600

Les règles de l'application du gestionnaire de politiques nous autorisent le regroupement de plusieurs profils distincts sur une même session, et également de grouper les réservations.

#### **5.4 Les résultats**

Quand le service QoS n'est pas activé, nous constatons que la moyenne du trafic tourne au tour de la valeur de 34% pour les paquets UDP, montré à la figure 33.

The screenshot shows a 'Protocol Summary' window with a tree view on the left and a table on the right. The tree view is expanded to show Ethernet Type 2, IP, UDP, and TCP. The table shows the following percentages:

Protocol	Percentage
Ethernet Type 2	0,000%
IP	0,000%
UDP	34,194%
TCP	65,806%

Figure 33 Trafic sans service QoS

Dans le cas où le service QoS est mis en place, les résultats étaient assez variables, elles fluctuaient dans une plage à deux valeurs limites représentées par les figures 34 et 35.

The screenshot shows a 'Protocol Summary' window with a tree view on the left and a table on the right. The tree view is expanded to show Ethernet Type 2, IP, UDP, and TCP. The table shows the following percentages:

Protocol	Percentage
Ethernet Type 2	0,000%
IP	0,000%
UDP	45,763%
TCP	54,237%

Figure 34 Trafic avec service QoS-a

The screenshot shows a 'Protocol Summary' window with a tree view on the left and a table on the right. The tree view is expanded to show Ethernet Type 2, IP, UDP, and TCP. The table shows the following percentages:

Protocol	Percentage
Ethernet Type 2	0,000%
IP	0,000%
UDP	60,976%
TCP	39,024%

Figure 35 Trafic avec service QoS-b

Comme il est montré par ces résultats d'exécution obtenus, nous avons constaté que cet outil gère convenablement la priorité. Le flux de trafic UDP arrivent avec des taux plus élevés. Il donne un meilleur contrôle à travers notre système réseau en offrant des priorités relatives aux paquets UDP, notons que certaines contraintes nous ne permettent pas de tester l'application avec des usagers IP de l'extérieure ( plus de congestion) de notre architecture réseau fermé ou, nous pouvons relever des valeurs plus réelles et constater le vrai effet.

Le profil QoS influence la qualité de service dans le sens ou il contribue à l'améliorer avec ses paramètres nous avons vu que cela est fait dans le but de satisfaire les besoins requis.

### **5.5 Phases expérimentales sur réseau IP**

Dans le but de valider le service QoS, nous avons jugé pertinent aussi de recueillir les valeurs de certains paramètres du réseau. Pour cela, nous avons élaboré 3 scénarios de différentes architectures pour des tests de performance qui devaient nous renseigner de façon plus précise sur le comportement du réseau sur des liens entre l'UQAM , l'ÉTS et la Télé-Université. Ces trois scénarios sont :

Pour cela on a élaboré 3 scénarios de différentes architecture pour des tests de performances qui devait nous renseigner de façon plus précises du comportement du réseau sur des liens entre l'UQAM , l'ÉTS et la Télé-Université, c'est trois scénarios sont :

- a- Trafic routé « normalement » sur un réseau ouvert (classe non dédiée).
- b- Réservection d'un canal virtuel privé de 10Mbps entre les passerelles d'entrée des trois campus (routé à classe dédiée)
- c- Réservection d'un canal de bout-à-bout entre les postes engagés dans la communication

Même si tous les scénarios n'ont pas été réalisés, il est pertinent de présenter les résultats obtenus dans le scénario a (le seul a avoir été réalisé jusqu' à présent).

#### A. Scénario a

Il consiste en des communications audiovisuelles par le biais de l'application *Netmeeting* (dont les modules ont été intégrés à la plate-forme Synchromédia) entre l'UQAM et l'ETS avec un trafic routé « normalement » sur un réseau ouvert (classe non dédiée).

Les tests comportent à envoyer des *ping* de 64 octets à intervalle régulier d'une seconde sur une durée de plusieurs minutes à chaque nœud impliqué dans la communication puis de relever les valeurs des temps d'aller-retour (RTT). Les paramètres à évalués sont :

- La moyenne : C'est la somme des éléments divisée par leur nombre.
- Le maximum et le minimum des valeurs recueillies
- La gigue (déviante "Jitter") Une aberration qui se produit quand la vidéo ou la voix est transmise sur un réseau et les paquets n'arrivent pas à leurs destination dans un ordre consécutive ou selon une base temporelle, elle est égale au Max – Min
- L'écart-type : L'écart-type d'un échantillon est la moyenne quadratique des écarts des données par rapport à leur moyenne, il sert à quantifier la dispersion d'un échantillon autour de cette dernière et obtenu à partir de la racine carrée de la variance.

Deux formules équivalentes pour calculer la variance :

$$\begin{aligned} \text{Variance} &= \frac{\sum(X_i - M)^2}{\text{effectif}} \\ &= \frac{\sum X_i^2}{\text{effectif}} - M^2 \end{aligned} \quad (6.1)$$

On note aussi la variance  $\sigma^2$  ( sigma deux) et M étant la moyenne des valeurs de  $X_i$



**Variance (précisions) :** La variance est un indicateur de la dispersion des valeurs autour de leur moyenne. La première formule se lit : "moyenne des carrés des écarts à la moyenne". La seconde se lit : "moyenne des carrés moins le carré de la moyenne"

### B. Trace route

Au début on a lancé un " Trace route " pour découvrir les équipements engagés dans la connexion entre les 2 postes en communication, installés a l'UQAM et l'ÉTS :

The screenshot shows a 'Trace Route' window with a table of results. The table has four columns: IP address, status, hop count, and another hop count. The data is as follows:

IP Address	Status	Hop 1	Hop 2
132.208.135.240	Unresolved	1	4
132.208.2.180	Unresolved	2	1
192.139.150.17	Unresolved	3	4
192.168.104.2	Unresolved	4	4
142.137.40.105	Unresolved	5	5

Figure 36 Découverte des équipements sur le lien ÉTS-UQAM

### C. Mise au point

Les tests ont eu lieu sur différentes périodes de la journée, soient le matin, l'après-midi et le soir. Primo, l'idée est de faire des tests durant les heures habituelles de cours et en second, le but de procéder à trois phases de tests réparties le long d'une journée est d'avoir une meilleure perception sur le niveau et la qualité de la communication. Pour chacun des nœuds, deux tests ont été exécutés.

Note : Les nœuds 132.208.135.240 et 192.168.104.2 n'ont pas fait l'objet d'une analyse de notre part étant donné que le premier constitue le routeur par défaut pour notre station alors que le second ne retourne que des `time_out`, d'où un filtrage prévisible au niveau de l'ICMP (*Internet Control Message Protocol*), sachant que l'application Trace route est basée sur ce protocole.

#### D. Tableau récapitulatif

Tableau VII

Représente les valeurs des résultats des tests

	Noeud1 132.208.2.180						Noeud2 192.139.150.17						Noeud final 142.137.40.105					
	Jitter		Ecart-type		Moy		Jitter		Ecart-type		Moy		Jitter		Ecart-type		Moy	
Essai	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
Matin	71	82	8.09	8.19	4	4	67	48	9.22	8.41	5	6	187	300	15.06	42.07	48	48
Après midi	98	92	9.56	9.42	3	3	63	64	6.83	8.23	4	5	242	216	54.12	38.06	85	39
Soir	64	63	10.6	7.74	5	5	66	41	11.5	8.92	7	6	271	220	54.77	46.93	96	70

#### E. Bilan et observations

On a constaté durant le déroulement des tests que la qualité de la voix et des images vidéo étaient souvent altérées. De plus, lors des mouvements, l'image devenait floue, ça doit être effet du *Jitter*, étant donné que les valeurs du nœud final étaient plus élevées que les autres nœuds. En plus, les autres paramètres mesurés (ecart-type et moyenne) sont aussi assez élevée. Ces résultats étaient largement prévisibles étant donné que la communication se déroule sur un réseau IP où la gestion du trafic est dominé par le mode du meilleur-effort.

## **CHAPITRE 6**

### **ANALYSE CRITIQUE ET RECOMMANDATIONS**

#### **6.1 Introduction**

À la fin de ce projet de maîtrise, nous voulons faire de ce chapitre une autocritique de notre travail en vue d'en tirer des leçons pour l'avenir. Nous allons d'abord évoquer les difficultés rencontrées et les moyens utilisés pour les surmonter, puis, nous évaluerons et l'approche utilisée, en envisageant une autre façon de faire, et le travail que nous avons réalisé en le comparant à ce que nous aurions voulu faire. Enfin, une conclusion indiquera notre degré de satisfaction.

#### **6.2 Difficultés rencontrées**

Les difficultés que nous avons rencontrées sont essentiellement liées à une mauvaise évaluation de la faisabilité temporelle. Nous avons sous-estimé le temps d'apprentissage et les contraintes matérielles et logicielles (disponibilité, compatibilité, langage de programmation). Nous avons pris beaucoup de temps à comprendre la notion de qualité de service et le fonctionnement du système élaboré par l'équipe de l'UQAM. Notre tâche semblait bien plus proche de la gestion des données des paramètres que de la négociation.

Nous avions à poursuivre un travail commencé par d'autres chercheurs. Même si les quelques commentaires laissés étaient clairs, l'insuffisance de documentation ou de guide portant sur la description du prototype déjà réalisé et assorti d'un bilan sur les travaux faites jusqu'à présent par les anciens étudiants qui ont travaillé sur l'application du gestionnaire de politiques nous a rendu la période de transition assez longue pour maîtriser toutes les options de l'implémentation. Par exemple, l'existence de plusieurs autres services non achevés ou non fonctionnels et la création d'autres en cours nous

rendaient la situation encore plus compliquée à cause des manipulations très fréquentes et des changements assez réguliers portés sur le code de l'application du gestionnaire de politiques.

Au-delà de ça, les multiples expansions et modifications qu'a subi le gestionnaire de politiques depuis sa mise en service en 1999 en ont grandement affecté l'architecture interne qui n'était certes pas prévue pour supporter de tels changements. Il en a résulté une structure attaquée par les « patches », une surabondance de codes et une architecture vétuste. L'architecture qui semble avoir été utilisée à la base est du type 1-tier, c'est-à-dire que, par exemple, certains appels au serveur LDAP sont faits directement à partir de l'IPM (Interface personne-machine). Cette structure est, il va sans dire, extrêmement difficile à maintenir et la traçabilité aux exigences est impossible à faire.

Actuellement, de gros changements et une migration du prototype sont en cours afin de revoir les fondements même du programme et d'introduire de nouvelles techniques pour s'adapter aux tendances technologiques récentes qui nous ont rendu la tâche plus ardue et imposée des exigences difficiles à suivre, une grande patience était de mise.

L'équipe du laboratoire de téléinformatique de l'UQAM est tout à fait consciente de la situation et continue à développer de nouvelles idées pour apporter les améliorations nécessaires et introduire de nouvelles technologies.

### **6.3 Analyse critique du travail effectué**

Notre objectif principal au départ était de combler les manques d'un système de gestion de la qualité de service en proposant un nouveau profil au modèle existant et de s'assurer de son harmonisation et de sa compatibilité avec les autres services déjà installés, en plus d'être supporté par les versions des systèmes d'opérations des équipements mis en

opération. Il s'agissait en fait de mettre en place un nouvel outil de commande du réseau à la disposition du gestionnaire afin de gérer son trafic selon certaines priorités.

Nous avons pu concevoir et implanter notre profil QoS basé sur le critère des classes de service qui prend en compte le type de trafic en question et les modes de configuration ainsi que les disponibilités offertes par les équipements. Cela améliore la négociation de la qualité de service selon les responsables du projet. Il reste cependant à faire des mesures précises après une implantation avec la plate-forme du projet "Synchronédia". Nous estimons avoir atteint notre objectif qui était de répondre à la question suivante : "Les configurations selon le protocole 802.1p et les classes de service améliorent-elles la négociation de la qualité de service ?" La réponse est positive et nous avons un élément de preuve qui est le modèle du profil proposé. Nous déplorons toutefois une sous-estimation de la faisabilité temporelle et organisationnelle qui nous a empêché d'aller au bout de la démonstration.

Cependant, l'utilisation de notre service QoS reste restreint aux types d'équipements rencontrés et mis à notre disposition (contraintes matérielles) au niveau du laboratoire pour fin de tests de simulation, ce qui limite grandement sa généralisation à d'autres types de produits des autres manufacturiers d'équipements de télécommunication. Mais cela n'empêche pas qu'en redéfinissant quelques méthodes et attributs, une grande partie du code peut être réutilisée.

#### **6.4 Analyse critique de l'approche**

La méthode de travail n'a pas été facile à maîtriser, car chaque développeur de nouveau service inclu au sein du *Policy Editor* avait ses outils de développement qui ne sont pas toujours compatibles avec ceux des autres. De même, le gros volume de codage successif et permanent a produit des étapes difficilement intégrables et provoqué une longue série de débogages. Les profils et leurs classes étant parfois complémentaires,

un changement de version dans une entraînait des modifications dans les autres. Lorsque nous avons voulu fusionner notre service SBM de traçage entre les niveaux 2 et 3 avec le service RSVP déjà existant ce fut sans succès, vu qu'il n'était même pas encore achevé. Ce qui nous a obligé à opter pour une fonctionnalité indépendante par l'ajout de nos propres paramètres.

Nous avons dû jouer un rôle de médiateur entre les besoins requis de l'application et les possibilités offertes par les systèmes de gestion de base présentes dans les équipements. De même que nous avons mis un ensemble d'informations et de paramètres pertinents à la disposition de l'administrateur dans l'espoir de rendre la gestion plus flexible. Nous dirons aussi que le projet a mis à notre disposition des connaissances nouvelles et précieuses sur des sujets d'actualités (les politiques de gestion, qualité de service, multimédia, gestion des équipements). Nous nous en réjouissons.

Il faut néanmoins souligner que ce travail a été pour nous un apprentissage indispensable et que nous finissons heureux d'avoir acquis tant d'expérience et de connaissances sur la technologie et la programmation orientée-objet, le multimédia, la qualité de service et surtout la participation et l'exécution d'un projet de haut niveau scientifique regroupant les meilleures universités québécoises.

## **6.5 Autres tests d'évaluation**

Malheureusement, les deux derniers scénarios (b et c) de la phase expérimentale n'ont pas eu lieu à cause de la lenteur de la procédure réglementaire pour obtenir les autorisations légales dans le but de nous octroyer les 10Mbps de largeur de bande et la réservation du canal nécessaires. Nous avons donc seulement conservé le scénario " a " à cause de ces problèmes techniques hors de notre portée.

**Vu que le projet " Synchronmedia " se poursuit encore, nous recommandons fortement à l'équipe en charge de faire son possible pour poursuivre et compléter le reste des tests dont les résultats s'avéreront très importants pour une meilleure évaluation et perception de l'état et des performances du réseau actuel à travers tous ses composants, ce qui nous conduira forcément à mieux focaliser nos actions.**

## **6.6 Conclusion**

**Cette analyse critique nous permet de constater l'importance d'une bonne estimation de la faisabilité temporelle et organisationnelle d'un projet en tenant compte des risques et en prévoyant des alternatives. Nous avons manqué d'organisation dans le déroulement de notre processus, mais cela n'empêche pas que nous ayons apprécié cette première expérience en équipe de développement. Nous pensons avoir contribué à prouver qu'une bonne gestion dynamique des informations sur les paramètres de configuration des équipements améliore le niveau de négociation de la qualité de service.**

**Si les difficultés rencontrées limitent notre travail à une ébauche qui devrait être poursuivie et perfectionnée, il n'en demeure pas moins que ce travail nous a été bénéfique pour l'acquisition de connaissances variées qui nous aideront certainement dans notre carrière.**

## **CONCLUSION**

**Dans le but d'améliorer la flexibilité de la négociation de la qualité de service, nous avons entrepris de modéliser et d'implémenter un outil de gestion et de contrôle des paramètres de configuration d'équipements afin de garantir une meilleure priorité au flux de trafic multimédia. Ce travail été réalisé dans le cadre du projet "Synchrédia" pour le compte de L'École des Technologies de l'Information visant une meilleure compréhension des problèmes liés au développement des applications multimédia.**

**Arrivé dans une équipe qui avait déjà un premier prototype à son actif, nous avons procédé à une série d'études qui nous ont permis d'abord de passer en revue les méthodes existantes dans le domaine afin de mieux cerner et comprendre les besoins, ensuite d'intervenir et d'arriver aux résultats escomptés.**

**Dans la partie principale de ce travail, nous avons décrit en détails la technique élaborée et avons vu dans ces sections comment notre méthode tente de résoudre la problématique posée antérieurement en comblant le manque. Dans un premier temps, nous avons expliqué l'intérêt et les avantages dans l'utilisation des protocoles IEEE 802.1Q/p et l'importance de la combinaison des mécanismes et des méthodes des niveaux deux et trois du modèle ISO pour assurer une qualité de service de bout en bout par un traçage avec le protocole RSVP pour des meilleurs résultats. En outre, nous avons illustré les différentes étapes de l'intégration du nouveau profil QoS à l'application du gestionnaire de politiques, en se basant sur les classes du modèle d'information DEN appliqué au domaine des télécommunications. Finalement, nous avons cité et expliqué tous les points de la phase de codage et de programmation en langage *JAVA* des classes du profil QoS.**

**Dans un second temps, nous avons décrit comment nous procédons pour activer l'application du gestionnaire et, à travers elle, notre service QoS, en tenant compte de**



**l'interactivité, d'un côté, entre l'utilisateur et l'application et, de l'autre côté, entre les équipements eux-mêmes. En dernier, furent obtenus des résultats expérimentaux suite aux tests sur une architecture de simulation et de validation qui nous ont permis de constater l'efficacité de l'outil et de mettre à jour les faiblesses et les lacunes du système.**

**Des entreprises ont commencé dès maintenant à adopter les protocoles 802.1Q/p pour incorporer le principe des classes de service vu leur importance. On peut citer comme exemple la compagnie 3com qui a développé et doté ses cartes réseau (NIC) d'un logiciel appelé *Dynamic Access* qui transforme ces cartes en composants actifs assurant une interaction dynamique avec le réseau afin d'en optimiser globalement les performances et le contrôle et cela, en exploitant les avantages du protocole et des commutateurs 802.1p. La même chose s'applique au protocole RSVP qui est conçu pour être déployé dans tous les routeurs, même si la plupart des routeurs actuels ne sont pas prêts à supporter des paramètres nouveaux. CISCO est la première compagnie qui a commencé à fabriquer des routeurs compatibles avec le protocole RSVP.**

**Il faut noter que, dans le cadre de ce travail, certains points et issues n'ont pas été abordés parmi lesquels nous citons entre autres :**

**Hiérarchisation du trafic : permettre au client et au serveur de reconnaître le trafic spécifique des applications et d'ajouter des marques de priorité aux paquets afin de signaler leur priorité à l'infrastructure de réseau.**

**Efficacité du contrôle multicast : permettre aux cartes réseau de signaler aux commutateurs 802.1p l'appartenance aux groupes de multidiffusion afin de contrôler efficacement le flux multi-diffusion dans les réseaux locaux commutés.**

**Ces divers protocoles et mécanismes et services sont tous conçus pour fonctionner ensemble. En mélangeant et en associant leurs capacités dans une variété d'architectures possibles, le but des communications qui permettent du QoS de bout en bout et du haut vers le bas sont chaque jour plus près de la réalité. Les normes ne sont pas entièrement développées encore, et il reste certaines considérations importantes telles que le support de multicast qui exigent davantage d'attention, mais le déploiement est déjà en cours sur beaucoup de réseaux. Ce travail modeste marque néanmoins, les premiers pas dans cette direction.**

## **ANNEXE 1**

### **Définition des objets des classes impliquées**

## 1 : Définition des objets des classes impliquées

On considère une politique qui doit s'appliquer aux routeurs A et D (soit nos routeurs d'accès) pour laquelle on désire appliquer le service QoS avec un maximum de réservation de 5000 kbps par interface et un maximum de 1000Kbps par flot, avec une priorité de 100, un intervalle de port UDP de 16383 et une bande de 40 Kbps. Ceci, du lundi au vendredi de 8h30 à 17h. La politique sera la suivante:

```

objectclass: devicecontainedauxclass
objectclass: policy
objectclass: top
cn: QoS-5000K
policyname: QoS-5000K
dn: cn=QoS-5000K, ou=politique, ou=vpn, ou=teleinfo, o=UQAM, c=ca
policyscope: QoS
devicecontainedref:
    cn=routeurd,ou=routeur,ou=site2,ou=vpn,ou=teleinfo,o=uqam,c=ca
devicecontainedref:
    cn=routeura,ou=routeur,ou=site1,ou=vpn,ou=teleinfo,o=uqam,c=ca
policyconditionref: cn=horaire-jour,ou=politique ou=vpn,ou=teleinfo,o=uqam,c=ca
policyenabled: TRUE
policyversion: 1.0
policyactionref: cn=profile-QoS-5000K,
    ou=politique,ou=vpn,ou=teleinfo,o=uqam,c=ca

```

Ici, la politique a été associée à 2 routeurs, mais elle aurait aussi pu être associée à un groupe de routeurs défini comme suit:

```

objectclass: groupofnames
objectclass: top
dn: cn=VPN-group-1, ou=groupofdevices, ou=vpn, ou=teleinfo, o=UQAM, c=ca
cn: VPN-1
member: cn=routeurd,ou=routeur,ou=site2,ou=vpn,ou=teleinfo,o=uqam,c=ca

```

member: cn=routeura,ou=routeur,ou=site1,ou=vpn,ou=teleinfo,o=uqam,c=ca

et en remplaçant:

devicecontainedref:

cn=routeurd,ou=routeur,ou=site2,ou=vpn,ou=teleinfo,o=uqam,c=ca

devicecontainedref:

cn=routeura,ou=routeur,ou=site1,ou=vpn,ou=teleinfo,o=uqam,c=ca

par:

devicegroupcontainedref: cn= VPN-group-1, ou=groupofdevices, ou=vpn, ou=teleinfo,  
o=UQAM, c=ca

*Le Scheduler et le Schedule* sont définis comme suit:

objectclass: ippolicycondition

objectclass: policycondition

objectclass: top

cn: horaire-jour

policyconditionname: horaire-jour

dn: cn= horaire-jour, ou=politique, ou=vpn, ou=teleinfo, o=UQAM, c=ca

policyvalidityperiodref: cn=HeuresTravail,ou=politique,ou=vpn, ou=teleinfo, o=uqam,  
c=ca

objectclass: policyvalidityperiod

objectclass: top

cn: HeuresTravail

policyvalidityperiodname: HeuresTravail

dn: cn=HeuresTravail, ou=politique, ou=vpn, ou=teleinfo, o=UQAM, c=ca

policyvaliditydayofweekmask: 1111100

policyvaliditytimeofdayrange: 083000:170000

policyvaliditymonthmask: 111111111111

**Le profil de service que l'on veut appliquer sera alors:**

```

objectclass: qoserviceaction
objectclass: policyaction
objectclass: top
cn: profile-QoS-5000K
dn: cn=profile-QoS-5000K, ou=politique,ou=vpn,ou=teleinfo,o=uqam,c=ca
qosactionname: profile-QoS-5000K
qospermission: ACCEPT
QoSinterfaceflow: 5000
QoSsingleflow: 1000
QoSpriority:100
QoSUDPport:16383
QoSbandwidth:40

```

**Pour définir un profil de routeur:**

```

objectclass: top
objectclass: profile
objectclass: deviceprofile
objectclass: routerprofile
dn: cn=cisco-2600- IOS 12.0(7)T,ou=profile-routeur,ou=site1, ou=vpn, ou=teleinfo,
    o=UQAM, c=ca
cn: cisco-2600- IOS 12.0(7)T
profilename: cisco-2600- IOS 12.0(7)T
networkclientserviceslist: null
networkparameterslist: null
profiletype: router
servicesrequiredlist: null
servicesreceivedlist: null
classifiertype: 0
devicecharacteristicslist: null
ispacketmarker: FALSE
istrafficpolicer: FALSE

```

```

istrafficshapper: TRUE
isrsvpcapable: TRUE
isdiffservcapable: TRUE
isipseccapable: TRUE
isqoscapable: TRUE
interfacename: eth0
interfacename: eth1
networkingsoftwareversion: IOS 12.0(7)T
routerconfigurationmethod: telnet
qosconfigurationscript:
    interface $InterfaceName
    qos parametrs $QoSinterfaceflow $QoSsingleflow$QoSpriority$QoSUDPport
    $QoSBandwidth

```

Ici, on remarque que les attributs qui sont définis dans l'objet *QoServiceAction* sont ici passés en paramètres au script de configuration.

Pour définir un routeur qui utilise ce profil, on aura:

```

objectclass: top
objectclass: managedsystemelement
objectclass: logiquelement
objectclass: system
objectclass: computersystem
objectclass: unitarycomputersystem
objectclass: networkelement
objectclass: profilecontainedauxclass
dn: cn=routeura,ou=routeur,ou=site1, ou=vpn, ou=teleinfo, o=UQAM, c=ca
cn: routeura
name: routeura
powermanagementsupported: FALSE
connectionrequirementlist: 132.208.140.240
connectionrequirementlist: 132.208.150.240
deviceerrorcode: 0

```

**devicetype: 5**  
**networkingsoftwareversion: IOS 12.0(7)T**  
**initialloadinfo: null**  
**nameformat: text**  
**primaryownername: Ali**  
**deviceerrordescription: No error**  
**networkaddresslist: 132.208.135.240**  
**networkaddresslist: 132.208.140.1**  
**networkaddresslist: 132.208.150.1**  
**installdate: 2/10/01**  
**macaddresslist: 00:00:00:00:00:01**  
**macaddresslist: 00:00:00:00:00:02**  
**macaddresslist: 00:00:00:00:00:03**  
**roles: Network control device**  
**status: OK**  
**seealso: o=uqam,c=ca**  
**lastloadinfo: null**  
**description: Routeur d'accès du site 1**  
**creationclassname: router**  
**interfacelist: Ethernet**  
**powermanagementcapabilities: 0**  
**powerstate: 0**  
**softwareresetcapability: 0**  
**primaryownercontact: 988-3000 poste 6189**  
**resetcapability: 0**  
**ProfileContainedRef: cn=cisco-2600-IOS 12.0(7)T,ou=profile-routeur,ou=site1,  
ou=vpn, ou=teleinfo, o=UQAM, c=ca**

Sur la dernière ligne de la définition de notre équipement, on remarque le pointeur vers l'objet de profil défini plus haut. Notre autre routeur peut être défini de la même manière.



## BIBLIOGRAPHIE

Benford, S.D., Bowers, J., Fahlén, L.E., Greenhalgh, C.M., Mariani, J., & Rodden, T.R. (1995). *Networked Virtual Reality and Co-operative Work*. Presence: Teleoperators and Virtual Environments, 4 (4), 264-386, MIT Press.

Cambell, A., Coulson, G., & Hutchinson, D. (1994). *A Quality of Service Architecture*. ACM SIGCOMM Computer Communications Review, 24 (2), 6-27.

CISCO System. *Configuring QoS*, [En ligne].

[http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/12\\_1e/swconfig/qos.htm](http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/12_1e/swconfig/qos.htm)  
(Consulté en janvier 2001)

Crutcher, L.A., & Waters, A.G. (1996) . *Connection Management for an ATM Network*. IEEE Network, 6 (6), 42-55.

Diedrich, J., Lohmar, T., Zitterbart, M., & Keller, R. (2000). *Traffic Phase Effects with RED and Constant Bit Rate UDP-Based Traffic*. In J. Crowcroft, J. Roberts & M. I. Smirnov (Eds.), *Quality of Future Internet Services*, First COST 263 International Workshop, QoSIS 2000, Berlin, September 25-26.

3COM corporation. *DynamicAccess Software*, [En ligne].

<http://www.3com.fr/produits/dynamicaccess.html> (Consulté en février 2001)

Ferguson, P., & Huston, G. (1998) *Quality of service: delivering QoS on the Internet and in Corporate Networks*. John Wiley & Sons.

Ek, N. *IEEE 802.1 P,Q - QoS on the MAC level*, [En ligne].

<http://www.tml.hut.fi/Opinnot/Tik110.551/1999/papers/08IEEE802.1QosInMAC/qos.html>  
(Consulté en janvier 2001)

Follows, J. & Straeten, D. (1998). *Application-Driven Networking: Class of Service in IP, Ethernet and ATM Networks*. IBM Redbook.

Gen Suzuki. (1995). *Interspace: Toward Networked Reality of Cyberspace*, Proc. Imagina, 1-3 February. Monte Carlo, ISBN 2-86938-113-1.

Greenhalgh, C. & Benford, S. (1998). *Supporting Rich and Dynamic Communication in Large-Scale Collaborative Virtual Environment*. Presence: Teleoperators and Virtual Environments. MIT Press, (in press).

Greenhalgh, C., & Benford, S. (1995). *MASSIVE: A Virtual Reality System for Teleconferencing*. ACM Transactions on Computer Human Interaction (TOCHI), 2 (3), 239-261, ACM Press, September.

Han, J. & Smith, B. (1996). *CU-SeeMe VR Immersive Desktop Teleconferencing*. Proc. Multimedia, Nov 18-22, Boston, USA, 199-208, ACM Press.

IEEE groups. *IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture*, [En Ligne]. <http://standards.ieee.org/catalog/olis/802-1990.pdf> (Consulté en juillet 2001)

McCanne, S., Jacobson, V., & Vetterli, M. (1996). *Receiver-driven Layered Multicast*. Computer Communications Review, ACM SIGCOMM, 26(4), October, ISSN 0146-4833.

Nakanishi, H., Yoshida, C., Nishimura, T. & Ishida, T. (1996). *Freewalk: Supporting Causal Meetings in a Network*, Proc. CSCW, 308-314, ACM Press.

Oh, S., Hiroyuki, S., Kazutoshi, F., Toshio, M., Shinji, S., Masatoshi, A. & Hideo, M. (1997). *A Dynamic QoS Adaptation Mechanism for Networked Virtual Reality*. Proceedings of Fifth IFIP International Workshop on Quality of Service, 397-400, Columbia University, New York, USA, May.

PUYBARET, E. *Manuel consultable gratuitement pour débuter et se perfectionner en Java*. [En ligne]. <http://www.eteks.com> (Consulté en avril 2001)

VIPswitch. *Building a guaranteed, quality-of-service (QoS) infrastructure for emerging business applications*. [En ligne]. [http://www.vipswitch.com/PDF/VIPswitchQoSWhitePaper\\_Dec2000\\_PDF.pdf](http://www.vipswitch.com/PDF/VIPswitchQoSWhitePaper_Dec2000_PDF.pdf) (consulté septembre 2001)

QoSforum. *QoS Protocols & Architectures white papers*. (1999). [En ligne] <http://www.qosforum.com> (Consulté en avril 2001)

Raymond, K. (1993). *The reference Mode of Open Distributed Processing: A Tutorial*. In J. De Meer, B. Mahr & O. Spanniol (Eds.), In Proceedings of the first international Conference on ODP, 3-14. Berlin.

Reynard, G., Benford, S., & Greenhalgh, C. (1998). *Awareness Driven Video Quality of Service in Collaborative Virtual Environments*. Proc. ACM Conference on Human Factors in Computing Systems, CHI, 464-471, Los Angeles, April 18-23, ACM Press.

Tannenbaum, A.S. (1992). *Computer Networks*. Textbook, Prentice Hall.

Tassel, J., Bob B. & Smith, A. (1997). *An End to End Price-Based QoS Control Component Using Reflective Java*. Fourth COST 237 Workshop, Lisboa, Portugal, December 15-19.

Vogel, A., Kerherve, B., von Bochmann, G., & Gecsei, J. (1995). *Distributed Multimedia and QoS: a Survey*. IEEE Multimedia, Summer, 10-19.