

Groupe de travail Réseau
Request for Comments : 3644
 Catégorie : En cours de normalisation

Y. Snir & Y. Ramberg, Cisco Systems
 J. Strassner, Intelliden
 R. Cohen, Ntear LLC
 B. Moore, IBM
 novembre 2003

Traduction Claude Brière de L'Isle

Modèle d'information de politique de qualité de service (QS)

Statut du présent mémoire

Le présent document spécifie un protocole de l'Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. Prière de se référer à l'édition en cours des "Protocoles officiels de l'Internet" (STD 1) pour voir l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Notice de copyright

Copyright (C) The Internet Society (2003).

Résumé

Le présent document présente un modèle d'informations orienté objet pour représenter les politiques de gestion de réseau qui s'occupent de la qualité de service (QS). Le présent document se fonde sur le modèle d'informations de cœur de politique de l'IETF et ses extensions. Il définit un modèle d'informations pour l'application de la QS pour les services différenciés et intégrés qui utilisent la politique. Il est important de noter que le présent document définit un modèle d'informations qui par définition est indépendant de tout mécanisme particulier de mémorisation des données et de tout protocole d'accès.

Table des Matières

1. Introduction.....	2
1.1 Processus de définition de la politique de qualité de service.....	3
1.2 Objectifs de conceptions et ramifications.....	4
1.3 Modélisation des politiques de QS abstraites.....	8
1.4 Hiérarchie de règles.....	9
1.5 Publics visés.....	12
2. Hiérarchies de classes.....	12
2.1 Hiérarchie d'héritage.....	12
2.2 Hiérarchie de relations.....	14
3. Actions de qualité de service.....	14
3.1 Généralités.....	14
3.2 Actions de politique RSVP.....	15
3.3 Actions d'approvisionnement de politique.....	16
3.4 Actions de comportement par bond.....	19
4. Profils de trafic.....	21
4.1 Profils de provisionnement de trafic.....	21
4.2 Profils de trafic RSVP.....	22
5. Variables prédéfinies en rapport avec la QS.....	22
6. Valeurs en rapport avec la QS.....	23
7. Définitions de classe : hiérarchie d'association.....	24
7.1 Association "QoSPolicyTrfcProfInAdmissionAction".....	24
7.2 Association "PolicyConformAction".....	25
7.3 Association "QoSPolicyExceedAction".....	25
7.4 Association "PolicyViolateAction".....	26
7.5 Agrégation "QoSPolicyRSVPVariableInRSVPSimplePolicyAction".....	26
8. Définitions de classe : hiérarchie d'héritage.....	26
8.1 Classe QoSPolicyDiscardAction.....	27
8.2 Classe QoSPolicyAdmissionAction.....	27
8.3 Classe QoSPolicyPoliceAction.....	27
8.4 Classe QoSPolicyShapeAction.....	28
8.5 Classe QoSPolicyRSVPAdmissionAction.....	28
8.6 Classe QoSPolicyPHBAction.....	29
8.7 Classe QoSPolicyBandwidthAction.....	29
8.8 Classe QoSPolicyCongestionControlAction.....	30
8.9 Classe QoSPolicyTrfcProf.....	32

8.10	Classe QoSPolicyTokenBucketTrfcProf.....	32
8.11	Classe QoSPolicyIntServTrfcProf.....	33
8.12	Classe QoSPolicyAttributeValue.....	34
8.13	Classe "QoSPolicyRSVPVariable".....	35
8.14	Classe "QoSPolicyRSVPSourceIPv4Variable".....	35
8.15	Classe "QoSPolicyRSVPDestinationIPv4Variable".....	35
8.16	Classe "QoSPolicyRSVPSourceIPv6Variable".....	35
8.17	Classe "QoSPolicyRSVPDestinationIPv6Variable".....	35
8.18	Classe "QoSPolicyRSVPSourcePortVariable".....	36
8.19	Classe "QoSPolicyRSVPDestinationPortVariable".....	36
8.20	Classe "QoSPolicyRSVPIPProtocolVariable".....	36
8.21	Classe "QoSPolicyRSVPIPVersionVariable".....	36
8.22	Classe "QoSPolicyRSVPDCLASSVariable".....	37
8.23	Classe "QoSPolicyRSVPStyleVariable".....	37
8.24	Classe "QoSPolicyIntServVariable".....	37
8.25	Classe "QoSPolicyRSVPMessageTypeVariable".....	37
8.26	Classe "QoSPolicyRSVPPreemptionPriorityVariable".....	38
8.27	Classe "QoSPolicyRSVPPreemptionDefPriorityVariable".....	38
8.28	Classe "QoSPolicyRSVPUserVariable".....	38
8.29	Classe "QoSPolicyRSVPApplicationVariable".....	38
8.30	Classe "QoSPolicyRSVPAuthMethodVariable".....	38
8.31	Classe QoSPolicyDNValue.....	39
8.32	Classe QoSPolicyRSVPSimpleAction.....	39
9.	Déclaration de droits de propriété intellectuelle.....	40
10.	Remerciements.....	40
11.	Considérations sur la sécurité.....	40
12.	Références.....	40
12.1	Références normatives.....	40
12.2	Références Informatives.....	40
13.	Adresse des auteurs.....	41
14.	Déclaration complète de droits de reproduction.....	41

1. Introduction

Le modèle d'informations de politique de qualité de service (QPIM, *QoS Policy Information Model*) établit un cadre et des constructions standard pour spécifier et représenter les politiques qui administrent, gèrent, et contrôlent l'accès aux ressources de qualité de service du réseau. De telles politiques sont appelées des "politiques de qualité de service" dans le présent document. Le cadre consiste en un ensemble de classes et de relations qui sont organisées dans un modèle d'informations orienté objet. Il ignore toute mise en œuvre spécifique de point de décision de politique (PDP, *Policy Decision Point*) ou de point de mise en application de politique (PEP, *Policy Enforcement Point*) (voir les définitions dans la [RFC3198]) et est indépendant de tout mécanisme particulier de mise en œuvre de la qualité de service (QS).

QPIM est conçu pour représenter les informations de politique de qualité de service pour les domaines de politique de grande échelle (le terme "domaine de politique" est défini dans la [RFC3198]). Un des objectifs principaux de ce modèle d'informations est d'aider les administrateurs humains dans leur définition de politiques de contrôle des ressources de qualité de service (par opposition à la configuration d'élément de réseau individuel). Le processus de création des instances de données de QPIM est alimenté par des règles d'affaires, la topologie du réseau et la méthodologie de qualité de service (par exemple, les services différenciés).

Le présent document se fonde sur le modèle d'informations de cœur de politique de l'IETF et ses extensions telles que spécifiées par les [RFC3060] et [RFC3460]. QPIM s'appuie sur ces deux documents pour définir un modèle d'informations pour la mise en application de la qualité de service pour les services différenciés et intégrés ([RFC2475] et [RFC1633], respectivement) qui utilisent la politique. Il est important de noter que le présent document définit un modèle d'informations qui est par définition indépendant de tout mécanisme particulier de mémorisation de données et de protocole d'accès. Cela permet que divers modèles de données (par exemple, les schémas de répertoires, les schémas de base de données relationnelles, et les MIB SNMP) soient conçus et mis en œuvre selon un seul modèle uniforme.

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans la [RFC2119].

manière standard. L'utilisation d'un modèle d'informations de politique de qualité de service standard tel que QPIM est aussi nécessaire parce que différents appareils peuvent avoir des capacités sensiblement différentes. Un modèle d'équipement peut également avoir des fonctionnalités différentes si le système d'exploitation du réseau et les logiciels qui fonctionnent dans ces appareils sont différents. Donc, il est nécessaire d'avoir un moyen pour spécifier la fonctionnalité d'une façon standard qui soit indépendante des capacités des appareils de fabricants différents. C'est le rôle de QPIM.

Dans un scénario normal, l'administrateur va d'abord déterminer le ou les rôles que joue chaque interface de chaque élément de réseau dans la topologie de réseau globale. Ces rôles définissent les fonctions assurées par un certain élément de réseau indépendamment du fabricant et du type de l'appareil. La [RFC3060] et la [RFC3460] définissent le concept de rôle. Les rôles peuvent être utilisés pour identifier quelles parties du réseau ont besoin de quel type de conditionnement de trafic. Par exemple, les cartes d'interface réseau qui sont catégorisées comme interfaces "cœur" peuvent recevoir le nom de rôle "interface-cœur". Cela permet à l'administrateur de concevoir des politiques pour configurer toutes les interfaces qui ont le rôle "interface-cœur" indépendamment des appareils physiques réels eux-mêmes. QPIM utilise les rôles pour aider l'administrateur à transposer un certain ensemble d'appareils ou interfaces en un certain ensemble de constructions de politique.

Les constructions de politique définissent la fonctionnalité nécessaire pour effectuer le conditionnement de trafic désiré pour le ou les types de trafic particuliers. Les fonctions elles-mêmes dépendent du type particulier de technologies de réseautage choisies. Par exemple, la méthodologie DiffServ nous encourage à agréger les types de trafic similaires en allouant à chaque classe de trafic un comportement de transmission par bond particulier sur chaque nœud. RSVP permet de réserver la bande passante. Ces deux méthodologies peuvent être utilisées séparément ou en conjonction, comme défini par la politique d'affaires appropriée. QPIM permet à des classes spécifiques d'activer DiffServ et de modéliser le conditionnement RSVP.

Les définitions de classe QPIM sont utilisées pour créer des instances de diverses constructions de politique telles que des actions et des conditions de qualité de service qui peuvent être organisées hiérarchiquement dans des règles et des groupes (PolicyGroup et PolicyRule comme défini dans les [RFC3060] et [RFC3460]). Des exemples d'actions de politique sont la limitation de débit, le contrôle de la gigue et l'allocation de bande passante. Les conditions de politique sont des constructions qui peuvent choisir le trafic selon une expression booléenne complexe.

Une organisation hiérarchique a été choisie pour deux raisons. D'abord, elle reflète mieux la façon dont les hommes tendent à penser une politique complexe. Ensuite, elle permet que la politique soit facilement transposée dans les organisations administratives, car les organisations hiérarchiques de politique reflètent la plupart des organisations administratives. Il est important de noter que le processus de définition de politique décrit ici est fait indépendamment de toute capacité spécifique et option de configuration d'un appareil. La définition de politique est complètement indépendante des détails de la mise en œuvre et de l'interface de configuration des éléments de réseau individuels, ainsi que des mécanismes qu'un élément de réseau peut utiliser pour conditionner le trafic.

1.2 Objectifs de conceptions et ramifications

Ce paragraphe explique les objectifs de la conception de QPIM et comment ces objectifs sont traités dans le présent document. Il décrit aussi les ramifications des objectifs conceptuels et les décisions prises dans le développement de QPIM.

1.2.1 Orientation vers la définition de politique

Le principal objectif de la conception de QPIM est de modéliser les politiques de contrôle du comportement de qualité de service d'une façon qui reflète d'aussi près que possible la façon dont les hommes tendent à penser la politique. Donc, QPIM est conçu pour traiter les besoins de définition et de gestion de la politique, et non la configuration des appareils et du réseau.

Il y a plusieurs ramifications à cet objectif de conception. D'abord, QPIM utilise des règles pour définir les politiques, sur la base des [RFC3060] et [RFC3460]. Ensuite, QPIM utilise de façon extensive des organisations hiérarchiques de politiques et d'informations de politique. Enfin, QPIM ne force pas l'auteur de la politique à spécifier tous les détails de mise en œuvre; il suppose plutôt que les agents de configuration (les PDP) interprètent les politiques et s'efforcent eux-mêmes à s'adapter aux besoins des configurations spécifiques des appareils.

1.2.1.1 Modélisation fondée sur la règle

La politique est le mieux décrite en utilisant une modélisation fondée sur la règle comme expliqué et décrit dans les [RFC3060] et [RFC3460]. Une règle de politique de qualité de service est structurée comme une clause de condition et une clause d'action. La sémantique est simple : si la clause de condition s'évalue à VRAI, un ensemble d'actions de qualité de service (spécifiées dans la clause action) peut être exécuté. Par exemple, la règle

"Le trafic de la Toile devrait recevoir au moins 50% des ressources de bande passante disponibles, ou plus s'il en est de disponibles"

peut être formalisée par :

```
"<si protocole == HTTP> alors <minimum BW = 50%>"
```

où la première clause entre crochets est une condition de trafic et la seconde clause entre crochets est une action de QS.

Cette approche diffère de la modélisation du chemin des données qui décrit les mécanismes qui fonctionnent sur les flux de paquets pour réaliser l'effet désiré.

Noter que l'approche suivie dans QPIM ne sous classe pas spécifiquement la classe PolicyRule. Elle utilise plutôt les classes SimplePolicyCondition, CompoundPolicyCondition, SimplePolicyAction, et CompoundPolicyAction définies dans la [RFC3460], et définit des sous classes des classes suivantes : Policy, PolicyAction, SimplePolicyAction, PolicyImplicitVariable, et PolicyValue. Faire des sous classes de la classe PolicyRule aurait rendu plus difficile de combiner les actions et les conditions définies au sein des différents domaines fonctionnels [RFC3460] avec les mêmes règles.

1.2.1.2 Organiser hiérarchiquement les informations

L'organisation des informations représentées par QPIM est conçue comme hiérarchique. Pour ce faire, QPIM utilise l'agrégation PolicySetComponent [RFC3460] pour fournir une organisation incorporée arbitrairement des informations de politique. Un groupe de politique fonctionne comme un conteneur de règles de politique et/ou de groupes de politiques. Une règle de politique contient aussi des règles de politique et/ou des groupes, ce qui permet à une relation règle/sous-règle de s'établir.

La décision de la conception hiérarchique se fonde sur la réalisation qu'il est naturel pour l'homme d'organiser les règles de politique en groupes. Casser une politique complexe en un ensemble de règles simples est un processus qui suit la façon dont les gens pensent et analysent les systèmes. La complexité de la politique abstraite, tournée vers les affaires, est simplifiée et insérée dans une hiérarchie de règles simples et de groupements de règles simples.

L'organisation hiérarchique des informations aide à simplifier la définition et la lisibilité des instances de données fondées sur QPIM. Les hiérarchies peuvent aussi servir à porter une sémantique supplémentaire pour les actions de qualité de service dans un certain contexte. Un exemple, détaillé au paragraphe 2.3, montre comment les politiques d'allocation hiérarchique de bande passante peuvent être spécifiées sous une forme intuitive, sans qu'il soit besoin de spécifier des structures de programme complexes.

1.2.1.3 Définition de politique orientée vers les objectifs

QPIM facilite la définition des politiques de qualité de service tournées vers des objectifs. Cela signifie que le processus de définition de la politique de qualité de service est concentré sur l'effet désiré des politiques, et non sur les moyens de mettre en œuvre la politique sur les éléments de réseau.

QPIM est destiné à définir une spécification minimale du comportement désiré de la part du réseau. C'est le rôle des agents de configuration spécifiques de l'appareil d'interpréter la politique exprimée d'une façon standard et de combler les détails de configuration nécessaires qui sont demandés pour leur application dans un cas particulier. L'avantage de l'utilisation de QPIM est qu'il fournit un langage commun que chacun des appareils et/ou agent de configuration spécifique du fabricant peut utiliser. Cela aide à s'assurer d'une interprétation commune de la politique générale ainsi qu'à aider l'administrateur à spécifier une politique commune à mettre en œuvre à travers différents appareils. Ceci est analogue au paradigme fondamental orienté objet de séparation de la spécification et de la mise en œuvre. Avec QPIM, le conditionnement du trafic peut être spécifié d'une manière générale qui peut aider des mises en œuvre différentes à satisfaire un objectif commun.

Par exemple, une politique valide peut n'inclure qu'une seule règle qui spécifie que la bande passante devrait être réservée pour un certain ensemble de flux de trafic. La règle n'a besoin d'inclure aucun des divers autres détails qui peuvent être nécessaires pour mettre en œuvre un programmeur qui prenne en charge cette allocation de bande passante (par exemple, la longueur de la file d'attente requise). On suppose qu'un PDP ou les PEP vont combler ces détails en utilisant (par exemple) leurs réglages par défaut de longueur de file d'attente. Le rédacteur de la politique a seulement besoin de spécifier l'objectif principal de la politique, en s'assurant que l'application préférée reçoit assez de bande passante pour fonctionner de façon adéquate.

1.2.2 Modèle de domaine de politique

Un objectif de conception important de QPIM est de fournir un moyen de définir les politiques qui s'étendent sur de nombreux appareils. Cet objectif différencie QPIM des modèles d'informations de niveau appareil, qui sont conçus pour modéliser une politique qui contrôle un seul appareil, ses mécanismes et ses capacités.

Cet objectif de conception a plusieurs ramifications. D'abord, des rôles [RFC3060] sont utilisés pour définir des politiques à travers plusieurs appareils. Ensuite, l'utilisation de politiques abstraites libère le processus de définition de politique du besoin d'avoir à traiter les particularités des appareils individuels, et laisse la modélisation de l'interprétation et de la configuration aux PDP ou autres agents de configuration. Enfin, QPIM permet une large réutilisation de tous les blocs de construction de politique dans les multiples règles utilisées dans les différents appareils.

1.2.2.1 Modèle de politique de QS dans un appareil et manière indépendante du fabricant

QPIM modélise la politique de QS d'une façon qui est conçue comme indépendante de tout appareil ou fabricant particulier. Cela permet aux réseaux constitués d'appareils différents qui ont des capacités différentes d'être gérés et contrôlés en utilisant un seul ensemble standard de politiques. Utiliser un tel ensemble unique de politiques est important parce qu'autrement, la politique refléterait elle-même les différences entre les différentes mises en œuvre d'appareils.

1.2.2.2 Utilisation des rôles pour transposer la politique dans les appareils du réseau

L'utilisation des rôles permet qu'une définition de politique soit ciblée sur la fonction réseau d'un élément de réseau, plutôt que sur le type et les capacités de l'élément. L'utilisation des rôles pour transposer la politique dans les éléments de réseau donne une méthode efficace et simple pour compacter et abstraire la définition de politique. Une certaine politique abstraite peut être transposée dans un groupe d'éléments de réseau sans qu'on ait besoin de spécifier la configuration pour chacun de ces éléments sur la base des capacités d'aucun des éléments individuels.

La définition de politique est conçue pour permettre l'agrégation de plusieurs appareils au sein du même rôle, si on le désire. Par exemple, si deux interfaces de cœur de réseau opèrent à des débits différents, l'une d'elles n'a pas à définir deux règles de politique séparées pour exprimer exactement la même politique abstraite (par exemple, allouer 30 % de la bande passante de l'interface à un certain ensemble de flux préférés). L'utilisation d'un contexte hiérarchique et d'actions de QS relatives dans QPIM traite ce problème et d'autres en relation avec lui.

1.2.2.3 Réutilisation

Les objets réutilisables, comme définis par la [RFC3060] et la [RFC3460], sont les moyens pour partager les éléments de construction de politique, permettant ainsi une gestion centrale des concepts globaux. QPIM donne la capacité de réutiliser tous les blocs de construction de politique : variables et valeurs, conditions et actions, profils de trafic, et groupes de politiques et règles de politique. Cela donne la souplesse requise pour gérer de grands ensembles de règles de politique sur de grands domaines de politique.

Par exemple, la règle suivante utilise les objets définis centralement qui sont réutilisés (référéncés) :

```
Si <DestinationAddress == FinanceSubNet> alors <DSCP = MissionCritical>
```

Dans cette règle, la condition se réfère à un objet nommé FinanceSubNet, qui est une valeur (ou éventuellement un ensemble de valeurs) définie et conservée dans un conteneur d'objets réutilisables. L'action de QS fait usage d'une valeur nommée MissionCritical, qui est aussi un objet réutilisable. L'avantage de spécifier une politique de cette façon est sa souplesse inhérente. Dans la politique ci-dessus, chaque fois que les affaires exigent un changement dans la définition du sous-réseau pour l'organisation, tout ce qui est nécessaire est de changer centralement la valeur réutilisable de FinanceSubNet. Toutes les règles de référence sont immédiatement affectées, sans qu'il soit besoin de les modifier individuellement. Sans cette capacité, le dépôt qui est utilisé pour mémoriser les règles devrait faire l'objet d'une recherche sur toutes les règles qui se réfèrent au sous-réseau Finance, et ensuite chaque condition de règle correspondante devrait être mise à jour individuellement. Ceci n'est pas seulement beaucoup moins efficace, mais aussi plus enclin à l'erreur. Pour une description complète des objets réutilisables, se reporter à la [RFC3060] et la [RFC3460].

1.2.3 Politique applicable

La politique définie par QPIM devrait pouvoir être mise en application. Cela signifie qu'un PDP peut utiliser la définition de politique de QPIM pour prendre les décisions nécessaires et mettre en application les règles de politique requises. Par exemple, les décisions d'admission RSVP devraient être prises sur la base des définitions de politique spécifiées par QPIM. Un PDP devrait être capable de transposer les définitions de politique de QPIM en configurations de PEP, en utilisant des protocoles standard ou propriétaires.

QPIM est conçu pour ignorer toute technologie particulière dépendante du fabricant. Cependant, les constructions de QPIM DEVRAIENT toujours être interprétées de telle sorte qu'un comportement conforme à la politique puisse être appliqué sur le

réseau géré. Donc, il y a trois exigences fondamentales que QPIM doit satisfaire :

1. La politique spécifiée par QPIM doit être capable de transposition dans les éléments de réseau réels.
2. La politique spécifiée par QPIM doit être capable de contrôler les fonctions de QS du réseau sans faire référence à un type spécifique d'appareil ou de fabricant.
3. La politique spécifiée par QPIM doit être capable de traduction dans la configuration des éléments de réseau.

QPIM satisfait aux exigences 1 et 2 ci-dessus en utilisant le concept de rôles (précisément, la propriété PolicyRoles, définie dans PCIM). En correspondant aux rôles alloués aux groupes de politique et aux éléments de réseau, un PDP (ou autre agent d'application) peut déterminer quelle politique devrait être appliquée à certains appareils.

L'utilisation des rôles dans la transposition de la politique aux éléments de réseau permet l'adaptabilité du modèle. La politique de QPIM peut être transposée à des domaines de politique de grande étendue en assignant un seul rôle à un groupe d'éléments de réseau. Ceci peut être fait même lorsque le domaine de la politique contient des appareils hétérogènes. Ainsi, un petit ensemble de politiques peut être déployé sur de grands réseaux sans avoir à respecifier la politique pour chaque appareil séparément. Cette propriété de QPIM est importante pour les applications de gestion de politique de qualité de service qui s'efforcent de faciliter la tâche de définition de la politique pour de grands domaines de politique.

L'exigence n° 2 est aussi satisfaite en donnant à QPIM une orientation domaine (voir dans la [RFC3198] une définition de "domaine"). En d'autres termes, la cible de la politique est un domaine, par opposition à un appareil ou interface spécifique.

L'exigence n° 3 est satisfaite par la modélisation des conditions et actions de qualité de service qui sont couramment configurées sur les divers appareils. Cependant, QPIM est extensible pour permettre de modéliser les actions qui ne sont pas incluses dans QPIM.

Il est important de noter que des PEP différents auront des capacités et fonctions différentes, ce qui nécessite des configurations individuelles différentes même si les différents PEP sont contrôlés par la même politique.

1.2.4 QPIM couvre la politique signalée aussi bien qu'approvisionnée

Les deux méthodologies prédominantes de qualité de service fondées sur les normes développées jusqu'à présent sont les services différenciés (DiffServ) et les services intégrés (IntServ). DiffServ donne un moyen pour mettre en application des politiques qui s'appliquent à un grand nombre d'appareils d'une façon adaptable. QPIM fournit des actions et conditions qui contrôlent la classification, la régulation et le formatage effectués au sein des frontières du domaine de service différencié, ainsi que les actions qui contrôlent le comportement bond par bond au sein du cœur du réseau DiffServ. QPIM ne rend pas obligatoire l'utilisation de DiffServ comme méthodologie de politique.

Les services intégrés, conjointement à leur protocole de signalisation (RSVP), donnent aux nœuds d'extrémité (et aux nœuds de bordure) le moyen de demander la qualité de service au réseau. QPIM fournit les actions qui contrôlent la réservation de telles demandes au sein du réseau.

Comme les deux méthodologies continuent d'évoluer, QPIM ne tente pas de fournir une couverture complète de tous les scénarios possibles. QPIM vise plutôt à fournir la modélisation du contrôle de politique pour tous les scénarios majeurs. QPIM est conçu comme extensible afin de permettre l'incorporation du contrôle sur les mécanismes de qualité de service nouvellement développés.

1.2.5 Interopérabilité pour les PDP et applications de gestion

Un autre objectif de la conception de QPIM est de faciliter l'interopérabilité entre les systèmes de politique tels que les PDP et les applications de gestion de politique. QPIM réalise cet objectif d'interopérabilité en normalisant la représentation de la politique. Les producteurs et les consommateurs de politique de qualité de service ont seulement besoin de s'appuyer sur les schémas fondés sur QPIM (et les modèles de données résultants) pour s'assurer d'une compréhension et d'un accord mutuels sur la sémantique de la politique de qualité de service.

Par exemple, supposons qu'une application de gestion de politique de qualité de service, construite par le fabricant A écrive ses politiques fondées sur le schéma LDAP qui transpose de QPIM en une mise en œuvre de répertoire utilisant LDAP. Supposons maintenant qu'un PDP construit séparément par le fabricant B s'appuie aussi sur le même schéma LDAP dérivé de QPIM. Même si ce sont deux fabricants avec deux PDP différents, chacun peut lire le schéma de l'autre et le "comprendre". C'est parce que l'application de gestion et le PDP ont tous deux une architecture qui leur permet de se conformer à la spécification QPIM. La même chose est vraie de deux applications de gestion de politique. Par exemple, l'application de politique du fabricant B peut avoir un outil de validation qui calcule si il y a un conflit entre les règles spécifiées par l'application de gestion de politique de l'autre fabricant.

L'interopérabilité des producteurs et consommateurs de QPIM est par définition à un haut niveau, et il n'est pas garanti que la même politique résulte en la même configuration de PEP. D'abord, des PEP différents vont avoir des capacités et fonctions différentes, qui nécessitent des configurations individuelles différentes même si les différents PEP sont contrôlés par la même politique. Ensuite, des PDP différents vont aussi avoir des capacités et fonctions différentes, et peuvent choisir de traduire différemment la politique QPIM de haut niveau selon les fonctionnalités du PDP, ainsi que des capacités des PEP qui sont contrôlés par le PDP. Cependant, les différentes configurations devraient quand même résulter en le même comportement réseau que spécifié par les règles de politique.

1.3 Modélisation des politiques de QS abstraites

Ce paragraphe discute la politique de qualité de service abstraite et la façon dont QPIM traite le problème.

Comme décrit ci-dessus, le principal objectif de QPIM est de créer un modèle d'informations qui puisse être utilisé à aider à combler partiellement le saut conceptuel entre le rédacteur humain de la politique et un élément de réseau qui est configuré pour appliquer la politique. Ce large écart implique clairement plusieurs niveaux de traduction, de l'abstrait au concret. Au niveau abstrait sont les règles d'affaires de politique de qualité de service. Une fois connues les règles d'affaires, un administrateur de réseau doit les interpréter comme politique de QS du réseau et la représenter en utilisant les constructions QPIM. QPIM facilite une représentation formelle des règles de QS, fournissant ainsi le premier niveau de concrétisation, représentant formellement la politique de qualité de service exprimée en termes de langage humain.

Lorsque un exécutant d'affaires humain définit la politique du réseau, il le fait généralement en utilisant des termes et un langage d'affaires humain. Par exemple, un homme peut formuler une déclaration de politique qui dit :

"les applications de ressources pour les hommes devraient avoir une meilleure QS que les simples applications de la Toile"

Ceci pourrait être traduit sous une forme légèrement plus sophistiquée, comme :

"Le trafic généré par des applications de ressources pour l'homme devrait avoir une plus forte probabilité de communiquer avec ses destinations que le trafic généré par les gens qui naviguent sur la Toile en utilisant des applications dont la mission n'est pas critique"

Bien que cette déclaration définisse clairement la politique de QS au niveau des affaires, elle n'est pas assez spécifique pour être applicable par les éléments de réseau. Une traduction est nécessaire en "termes et langage du réseau".

De l'autre côté de l'échelle, un élément de réseau qui fonctionne comme PEP, comme un routeur, peut être configuré avec des commandes spécifiques qui déterminent les paramètres de fonctionnement de ses mécanismes internes de qualité de service. Par exemple, la commande (imaginaire) "longueur-de-file-d'attente-de-sortie = 100" peut être une instruction à une carte d'interface réseau d'un routeur pour permettre de mémoriser jusqu'à 100 paquets avant que les paquets suivants soient éliminés (non transmis). Sur un appareil différent au sein du même réseau, la même instruction peut prendre une autre forme, parce qu'un fabricant différent a construit cet appareil ou parce qu'il a un ensemble de fonctions, et donc de mise en œuvre, différent, bien qu'il soit du même fabricant. De plus, un PEP particulier peut n'avoir pas la capacité de créer des files d'attentes qui font plus de, par exemple, 50 paquets, ce qui peut résulter en ce qu'une instruction différente mette en œuvre la même politique de QS.

Le premier exemple illustre la 'politique abstraite' tandis que le second illustre une 'configuration' concrète. De plus, le premier exemple illustre une politique de bout en bout, qui couvre le conditionnement du trafic d'application tout au long du réseau. Le second exemple illustre une configuration pour un PEP particulier, ou un ensemble de PEP. Alors qu'une déclaration de politique de bout en bout ne peut être mise en application que par configuration des PEP dans les diverses parties du réseau, le modèle d'informations de politique et celui des mécanismes qu'utilise un PEP pour mettre en œuvre cette politique, sont largement différents.

Le processus de traduction de la politique d'affaires abstraite en configuration de PEP concrète est en gros exprimé comme suit :

1. La politique de qualité de service d'affaires informelle est exprimée par un rédacteur humain de politique (par exemple, "Toutes les demandes sur la Toile émanant des dirigeants devraient avoir la priorité sur celles des autres employés")
2. Un administrateur de réseau analyse la topologie du domaine de politique et détermine les rôles des interfaces d'appareils particuliers. Un rôle peut être alloué à un grand groupe d'éléments, ce qui va résulter en la transposition en une politique particulière sur un large groupe d'interfaces d'appareils.

3. L'administrateur de réseau modélise la politique informelle en utilisant les constructions QPIM, créant ainsi une représentation formelle de la politique abstraite. Par exemple, "Si le protocole d'un paquet est HTTP et si sa destination est dans le groupe d'utilisateurs 'DIRIGEANTS', allouer alors IPP 7 à l'en-tête du paquet".
4. L'administrateur de réseau alloue des rôles aux groupes de politique créés dans l'étape précédente qui correspondent aux rôles des éléments de réseau alloués dans l'étape n° 2 ci-dessus.
5. Un PDP traduit les constructions de politique abstraite créées dans l'étape n° 3 en commandes de configuration spécifiques de l'appareil pour tous les appareils affectés par la nouvelle politique (c'est-à-dire, les appareils qui ont des interfaces auxquelles est alloué un rôle qui correspond aux rôles des constructions de la nouvelle politique). Dans ce procès, le PDP consulte les capacités des appareils particuliers pour déterminer les commandes de configuration appropriées qui mettent en œuvre la politique.
6. Pour chaque PEP du réseau, le PDP (ou un agent du PDP) produit les instructions appropriées spécifiques de l'appareil nécessaires pour mettre en application la politique.

QPIM, PCIM et PCIME sont utilisés dans l'étape 3 ci-dessus.

1.4 Hiérarchie de règles

La politique est décrite par un ensemble de règles de politique qui peuvent être groupées en sous ensembles [RFC3460]. Les règles de politique et les groupes de politique peuvent être incorporés au sein d'autres règles de politique, fournissant une définition de politique hiérarchique. Les règles incorporées sont aussi appelées des sous-règles, et on utilise les deux termes de façon interchangeable dans le présent document. L'agrégation PolicySetComponent (définie dans la [RFC3460]) est utilisée pour représenter l'incorporation d'une règle ou groupe de politique dans une autre règle de politique.

La définition de règle de politique hiérarchique améliore la lisibilité et la réutilisabilité de la politique. Au sein du modèle d'informations de politique de QS, la hiérarchie est utilisée pour modéliser le contexte ou la portée pour les actions de sous-règles. Dans QPIM, les actions de politique d'allocation de bande passante et les actions de seuil d'abandon utilisent ce contexte hiérarchique. On donne d'abord un exemple détaillé de l'utilisation de la hiérarchie dans les politiques d'allocation de bande passante. Les différences entre les représentations de politique plates et hiérarchiques sont discutées. L'utilisation de la hiérarchie dans les politiques de seuil d'abandon sont décrites dans un des paragraphes qui suivent. Enfin, mais non moins importantes, on décrit les restrictions sur l'utilisation des hiérarchies de règles au sein de QPIM.

1.4.1 Utilisation de hiérarchies au sein de politiques d'allocation de bande passante

Considérons l'exemple suivant dans lequel la politique informelle dit :

Sur toute interface sur laquelle ces règles s'appliquent, on garantit au moins 30 % de la bande passante de l'interface aux flux UDP, et au moins 40 % de la bande passante de l'interface aux flux TCP.

Le modèle d'informations de politique de QS suit le modèle d'informations de cœur de politique en utilisant les rôles comme moyen de spécifier l'ensemble des interfaces sur lesquelles cette politique s'applique. La politique ne suppose pas que toutes les interfaces fonctionnent à la même vitesse, ou aient d'autre propriété en commun à part d'être capables de transmettre les paquets. La bande passante est allouée entre les flux UDP et TCP en utilisant des pourcentages de la bande passante disponible de l'interface. En supposant que nous ayons une bande passante d'interface disponible de 1 Mbit/s, cette règle va garantir 300 kbit/s aux flux UDP. Cependant, si la bande passante de l'interface est seulement de 64 kbit/s, cette règle garantirait alors 19,2 kbit/s.

Cette politique est modélisée dans QPIM en utilisant deux règles de politique de la forme :

- Si (protocole IP est UDP) ALORS (garantir 30 % de la bande passante disponible) (1)
- Si (protocole IP est TCP) ALORS (garantir 40 % de la bande passante disponible) (2)

Supposons que ces deux règles soient groupées dans un PolicySet [RFC3460] portant la combinaison de rôles appropriée. Une mise en œuvre possible de ces règles au sein d'un PEP serait d'utiliser un programmeur Round-Robin pondéré avec trois files d'attente. La première file d'attente serait utilisée pour le trafic UDP, la seconde pour le trafic TCP et la troisième pour le reste du trafic. Les pondérations du programmeur Round-Robin seraient de 30 % pour la première file d'attente, 40 % pour la seconde et 30 % pour la dernière.

Les actions qui spécifient la garantie de bande passante supposent implicitement que la ressource de bande passante garantie est celle disponible au niveau de l'interface. Une PolicyRoleCollection est une classe définie dans la [RFC3460] dont l'objet

est d'identifier l'ensemble de ressources (dans cet exemple, les interfaces) qui sont allouées à un rôle particulier. Donc, le type d'éléments agrégés gérés au sein de la PolicyRoleCollection définit les ressources de bande passante qui sont contrôlées. Dans notre exemple, les interfaces sont agrégées au sein de la PolicyRoleCollection. Donc, les règles spécifient l'allocation de bande passante à toutes les interfaces qui correspondent à un certain rôle. Un autre comportement pourrait être défini de façon similaire en changeant ce qui a été agrégé au sein de la PolicyRoleCollection.

Normalement, une pleine spécification des règles exigerait d'indiquer la direction du trafic pour lequel est faite l'allocation de bande passante. En utilisant la variable direction définie dans la [RFC3460], les règles peuvent être spécifiées sous la forme suivante :

Si (la direction est sortante)

Si (protocole IP est UDP) ALORS (garantir 30 % de la bande passante disponible)

Si (protocole IP est TCP) ALORS (garantir 40 % de la bande passante disponible)

où le retrait est utilisé pour indiquer l'incorporation de règle. Pour économiser l'espace, on omet la condition de direction de la suite de la discussion.

L'incorporation de règle donne la capacité de préciser la portée de l'allocation de bande passante au sein d'une certaine classe de trafic transmise via ces interfaces. Les exemples ci-dessous ajoutent deux règles incorporées pour préciser l'allocation de bande passante pour les applications UDP et TCP.

Si (le protocole IP est UDP) ALORS (garantir 30 % de la bande passante disponible) (1)

Si (le protocole est TFTP) ALORS (garantir 10 % de la bande passante disponible) (1a)

Si (le protocole est NFS) ALORS (garantir 40 % de la bande passante disponible) (1b)

Si (le protocole IP est TCP) ALORS (garantir 40 % de la bande passante disponible) (2)

Si (le protocole est HTTP) ALORS (garantir 20 % de la bande passante disponible) (2a)

Si (le protocole est FTP) ALORS (garantir 30 % de la bande passante disponible) (2b)

Les sous-règles 1a et 1b spécifient l'allocation de bande passante pour les applications UDP. La ressource totale de bande passante qui est partagée entre les applications UDP est la bande passante disponible pour la classe de trafic UDP (c'est-à-dire, 30 %) et non la bande passante totale disponible au niveau de l'interface. De plus, TFTP et NFS ont la garantie d'obtenir au moins 10 % et 40 % de la bande passante totale disponible pour UDP, tandis que les autres applications UDP n'ont pas la garantie de recevoir quoi que ce soit. Donc, TFTP et NFS ont la garantie d'obtenir au moins 3 % et 12 % de la bande passante totale. Une logique similaire s'applique aux applications TCP.

L'objet de cette section sera de montrer qu'une représentation hiérarchique des politiques permet de spécifier un plus fin niveau de granularité de l'allocation de bande passante qu'il n'est autrement disponible en utilisant une représentation de politique non hiérarchique. Pour illustrer cela, on va comparer cet ensemble de règles à une représentation non hiérarchique (plate) de règles. Dans la représentation non hiérarchique, la bande passante garantie aux flux TFTP est calculée en prenant 10 % de la bande passante garantie aux flux UDP, résultant en une garantie de 3 % de la bande passante totale de l'interface.

Si (UDP ET TFTP) ALORS (garantir 3 % de la bande passante disponible) (1a)

Si (UDP ET NFS) ALORS (garantir 12 % de la bande passante disponible) (1b)

Si (autres APP UDP) ALORS (garantir 15 % de la bande passante disponible) (1c)

Si (TCP ET HTTP) ALORS (garantir 8 % de la bande passante disponible) (2a)

Si (TCP ET FTP) ALORS (garantir 12 % de la bande passante disponible) (2b)

Si (autres APP TCP) ALORS (garantir 20 % de la bande passante disponible) (2c)

Ces deux représentations sont-elles identiques ? Non, l'allocation de bande passante n'est pas la même. Par exemple, dans la représentation hiérarchique, les applications UDP ont la garantie de 30 % de la bande passante. Supposons que s'écoule un seul flux UDP d'une application différente de NFS ou TFTP. Cette application aurait la garantie de 30 % de la bande passante de l'interface dans la représentation hiérarchique mais de seulement 15 % de la bande passante de l'interface dans la représentation plate.

Un programmeur à deux étapes est mieux modélisé par une représentation hiérarchique alors qu'une représentation plate peut être réalisée par un programmeur non hiérarchique.

Une mise en œuvre schématique de programmeur hiérarchique de Round-Robin pondéré qui prend en charge la représentation hiérarchique des règles est décrite ci-dessous.

```
--File d'attente UDP ET TFTP --10%
--File d'attente UDP ET NFS --40%-Programmeur-30%--+
--File d'attente autre UDP --50% A1 |
```

```

--File d'attente TCP ET HTTP --20%
--File d'attente TCP ET FTP --30%-Programmeur-40%-Programmeur--Interface
--File d'attente autre TCP --50% A2 | B
-----Trafic non UDP/TCP -----30%---+

```

Le programmeur A1 extrait les paquets des trois files d'attente UDP conformément aux pondérations spécifiées par la sous règle de politique UDP. Le programmeur A2 extrait les paquets des trois files d'attente TCP spécifiées par la sous-règle de politique TCP. Le programmeur B de seconde étape programme entre UDP, TCP et tous les autres trafics conformément à la politique spécifiée dans le niveau de règle supérieur.

Une autre différence entre les représentations de règle plate et hiérarchique est la réalité de la division de la bande passante au delà de la garantie de bande passante minimale. Supposons deux flux de haut débit qui sont transmis via cette interface : un flux HTTP et un flux NFS. Supposons que le débit de chaque flux soit très au delà de la capacité de l'interface. Dans la mise en œuvre de programmeur plat, le ratio entre les pondérations est de 8:12 (c'est-à-dire, HTTP:NFS) et donc le flux HTTP consommerait 40 % de la bande passante alors que NFS consommerait 60 % de la bande passante. Dans la mise en œuvre de programmeur hiérarchique, le seul programmeur qui a deux files d'attente remplies est le programmeur B, donc le ratio entre le flux HTTP (TCP) et le flux NFS (UDP) serait de 30:40, et donc le flux HTTP consommerait approximativement 42 % de la bande passante de l'interface tandis que NFS consommerait 58 % de la bande passante de l'interface. Dans les deux cas, les deux flux HTTP et NFS auraient plus que la bande passante minimale garantie, mais les débits transmis réellement via l'interface diffèreraient.

La conclusion est que la représentation de politique hiérarchique fournit plus de structure et de contexte que la représentation plate de politique. De plus, les politiques qui spécifient une allocation de bande passante en utilisant des hiérarchies de règles devraient être mises en application en utilisant des programmeurs hiérarchiques lorsque le niveau hiérarchique des règles est transposé en niveau de programmation hiérarchique.

1.4.2 Utilisation de la hiérarchie des règles pour décrire les politiques de seuil d'abandon

Deux ressources majeures gouvernent le comportement par bond (PHB, *Per-Hop Behavior*) dans chaque nœud. La ressource d'allocation de bande passante gouverne le comportement de transmission de chaque classe de trafic. Les priorités et les pondérations d'un programmeur sont contrôlées par les politiques d'allocation de bande passante, ainsi que le nombre (minimal) de files d'attente nécessaires pour la séparation des trafics. Une seconde ressource, qui n'est pas contrôlée par les politiques d'allocation de bande passante, est la longueur de file d'attente et le comportement d'abandon. On utilise à cette fin des politiques de longueur de file d'attente et de seuil.

La hiérarchie des règles est utilisée pour décrire le contexte dans lequel agissent les seuils. La condition de la règle de politique décrit la classe de trafic et les actions de la règle décrivent l'allocation de bande passante, la priorité de transmission et la longueur de file d'attente. Si la classe de trafic contient différentes sous classes de présence d'abandon qui exigent des seuils différents au sein de la même file d'attente, les actions d'une sous règle décrivent ces seuils.

Voici un exemple de l'utilisation de l'incorporation de règle pour les besoins du contrôle de seuil. Les règles sont :

```

Si (le protocole est FTP) ALORS (garantir 10 % de la bande passante disponible)
    (la longueur de file d'attente égale 40 paquets)
    (la technique d'abandon est aléatoire)
    si (src-ip vient d'une adresse réseau 2.x.x.x) ALORS seuil min = 30 %, seuil max = 70 %
    si (src-ip vient d'une adresse réseau 3.x.x.x) ALORS seuil min = 40 %, seuil max = 90 %
    si (tous les autres) ALORS seuil min = 20 %, seuil max = 60 %

```

La règle décrit l'allocation de bande passante, la longueur de file d'attente et la technique d'abandon assignées au flux FTP. Les sous règles décrivent les priorités de seuil d'abandon au sein de ces flux FTP. Les paquets FTP reçus de tous les réseaux sauf les réseaux 2.x.x.x et 3.x.x.x sont abandonnés de façon aléatoire lorsque le seuil de file d'attente pour le flux FTP accumule 20 % de la longueur de la file d'attente. Une fois que la file d'attente atteint 60 %, tous ces paquets sont abandonnés avant la mise en file d'attente. Les deux autres sous règles fournissent d'autres seuils pour les paquets FTP qui viennent des deux sous réseaux spécifiés. Le comportement par bond de transmission assurée (AF, *Assured Forwarding*) est un autre bon exemple d'utilisation de la hiérarchie pour décrire les différentes préférences d'abandon au sein d'une classe de trafic. Cet exemple est fourni plus loin.

1.4.3 Restrictions à l'utilisation de la hiérarchie dans QPIM

L'incorporation de règle est utilisée dans QPIM pour deux objets importants :

- 1) Améliorer la clarté, la lisibilité et la réutilisabilité.
- 2) Fournir un contexte hiérarchique pour les actions.

Le second point vise la capacité de spécifier un contexte pour l'allocation de bande passante, ainsi que de fournir un contexte pour les politiques de seuil d'abandon.

Quand un niveau hiérarchique est-il supposé spécifier le contexte d'allocation de bande, quand est utilisée la hiérarchie pour spécifier le contexte de seuil d'abandon, et quand est-il utilisé simplement pour la clarté et la réutilisation ? La réponse dépend entièrement des actions. Les actions de contrôle de la bande passante au sein d'une sous règle spécifient comment la bande passante allouée à la classe de trafic déterminée par la clause de condition de la règle devrait être redistribuée entre les sous règles. Les actions de seuil d'abandon contrôlent le comportement d'abandon de la classe de trafic pour chacune des sous règles. Les actions de contrôle de la bande passante ont un pointeur implicite qui dit : l'allocation de bande passante se rapporte aux ressources de bande passante définies par la règle de plus haut niveau. Les actions de seuil d'abandon ont un pointeur implicite qui dit : les seuils sont tirés des ressources de file d'attente définies par la règle de plus haut niveau. Les autres actions n'ont pas un tel pointeur implicite, et pour ces actions, la hiérarchie n'est utilisée que pour les besoins de réutilisation et de lisibilité.

Chaque règle qui comporte une action d'allocation de bande passante implique qu'une file d'attente devrait être allouée à la classe de trafic définie par la clause de condition de la règle. Donc, une fois qu'une action d'allocation de bande passante au sein des actions d'une sous règle, une action de seuil au sein de cette sous règle ne peut pas se référer aux seuils de la file d'attente de la règle parente. Elle doit à la place se référer à la file d'attente de la sous règle elle-même. Donc, afin d'avoir une définition claire et sans ambiguïté, on devrait éviter les précisions des seuils et des allocations de bande passantes au sein des sous règles. Si ces deux précisions sont nécessaires pour la même règle, les règles de précision de seuil et de bande passante devraient être chacune agrégées à un groupe séparé, et ces groupes devraient être agrégés sous la règle de politique, en utilisant l'agrégation PolicySetComponent.

1.5 Publics visés

QPIM est destiné à divers publics. La liste suivante donne certaines des audiences prévues et leurs utilisations respectives :

1. Les développeurs d'applications de politiques de gestion de QS peuvent utiliser ce modèle comme un cadre extensible pour définir des politiques de contrôle des PEP et des PDP de façon interopérable.
2. Les développeurs de systèmes de points de décision de politique (PDP, *Policy Decision Point*) construits pour contrôler les allocations de ressources signalées par les demandes RSVP.
3. Les développeurs de systèmes de points de décision de politique construits pour créer des configurations de QS pour PEP.
4. Les constructeurs de bases de données et de connaissances de grandes organisations qui décident de combiner les informations de politique de QS avec d'autres informations de politique de réseautage, en supposant que toute la modélisation se fonde sur les [RFC3060] et [RFC3460].
5. Les auteurs de diverses normes peuvent utiliser les constructions introduites dans le présent document pour améliorer leur travail. Les auteurs de modèles de données qui souhaitent transposer une technologie spécifique de mémorisation en QPIM doivent utiliser aussi le présent document.

2. Hiérarchies de classes

2.1 Hiérarchie d'héritage

Les hiérarchies d'héritage de classe et d'association de QPIM prennent leurs racines dans les [RFC3060] et [RFC3460]. Les Figures 2 et 3 décrivent ces hiérarchies d'héritage de QPIM, tout en notant leurs relations avec les classes des [RFC3060] et [RFC3460]. Noter que de nombreuses autres classes utilisées pour former les politiques de QPIM, comme SimplePolicyCondition, sont définies dans les [RFC3060] et [RFC3460]. Donc, les figures suivantes NE représentent PAS TOUTES les classes et relations nécessaires pour définir les politiques de QPIM. Le concepteur qui utilise QPIM devrait plutôt utiliser les classes et relations appropriées des [RFC3060] et [RFC3460] en conjonction avec celles définies ci-dessous.

```
[ManagedElement] (abstraite, PCIM)
|
|--Policy (abstraite, PCIM)
| |
| +---PolicyAction (abstraite, PCIM)
```

```

| | |
| | | +---SimplePolicyAction (PCIMe)
| | | |
| | | | +---QoSPolicyRSVPSimpleAction (QPIM)
| | | +---QoSPolicyDiscardAction (QPIM)
| | | |
| | | +--QoSPolicyAdmissionAction (abstraite, QPIM)
| | | |
| | | | +---QoSPolicyPoliceAction (QPIM)
| | | |
| | | | +---QoSPolicyShapeAction (QPIM)
| | | |
| | | | +---QoSPolicyRSVPAdmissionAction (QPIM)
| | | |
| | | +--QoSPolicyPHBAction (abstraite, QPIM)
| | | |
| | | | +---QoSPolicyBandwidthAction (QPIM)
| | | |
| | | | +---QoSPolicyCongestionControlAction (QPIM)
| | |
| | | +--QoSPolicyTrfcProf (abstraite, QPIM)
| | | |
| | | | +---QoSPolicyTokenBucketTrfcProf (QPIM)
| | | |
| | | | +---QoSPolicyIntServTrfcProf (QPIM)
| | |
| | | +---PolicyVariable (abstraite, PCIMe)
| | | |
| | | | +---PolicyImplicitVariable (abstraite, PCIMe)
| | | | |
| | | | | +---QoSPolicyRSVPVariable (abstraite, QPIM)
| | | | |
| | | | | +---QoSPolicyRSVPSourceIPv4Variable (QPIM)
| | | | |
| | | | | +---QoSPolicyRSVPDestinationIPv4Variable (QPIM)
| | | | |
| | | | | +---QoSPolicyRSVPSourceIPv6Variable (QPIM)
| | | |
| | | +---PolicyVariable (abstraite, PCIMe)
| | | |
| | | | +---PolicyImplicitVariable (abstraite, PCIMe)
| | | | |
| | | | | +---QoSPolicyRSVPVariable (abstraite, QPIM)
| | | | |
| | | | | +---QoSPolicyRSVPDestinationIPv6Variable (QPIM)
| | | | |
| | | | | +---QoSPolicyRSVPSourcePortVariable (QPIM)
| | | | |
| | | | | +---QoSPolicyRSVPDestinationPortVariable (QPIM)
| | | | |
| | | | | +---QoSPolicyRSVPIPProtocolVariable (QPIM)
| | | | |
| | | | | +---QoSPolicyRSVPIPVersionVariable (QPIM)
| | | | |
| | | | | +---QoSPolicyRSVPDCLASSVariable (QPIM)
| | | | |
| | | | | +---QoSPolicyRSVPStyleVariable (QPIM)
| | | | |
| | | | | +---QoSPolicyRSVPDIntServVariable (QPIM)
| | | | |
| | | | | +---QoSPolicyRSVPMessageTypeVariable (QPIM)
| | | |

```

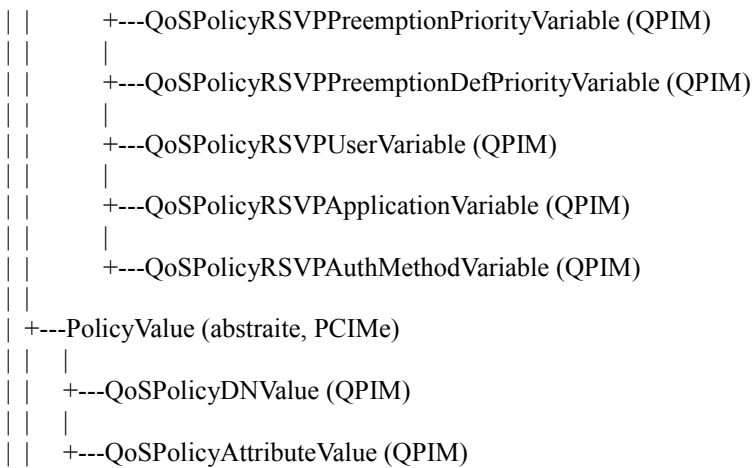


Figure 2 : Hiérarchie d'héritage de classe QPIM

2.2 Hiérarchie de relations

La Figure 3 montre la hiérarchie de relations QPIM.

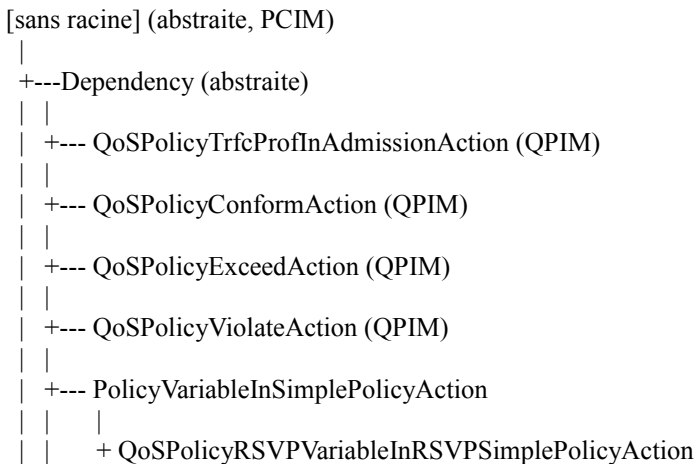


Figure 3. Hiérarchie d'héritage de classe d'association QPIM

3. Actions de qualité de service

Cette section décrit les actions de QS qui sont modélisées par QPIM. Les actions de QS sont des comportements du réseau mis en application par la politique qui sont spécifiés pour le trafic choisi par les conditions de QS. Les actions de QS sont modélisées en utilisant les classes de PolicyAction (définies dans la [RFC3060]), SimplePolicyAction (définie dans la [RFC3460]) et plusieurs actions de QS définies dans le présent document qui sont dérivées de ces deux classes, qui sont décrites ci-dessous.

Noter qu'il n'y a pas de discussion de PolicyRule, PolicyGroup, ou des différents types de classes de PolicyCondition dans le présent document. Ceci parce que ces classes sont pleinement spécifiées dans les [RFC3060] et [RFC3460].

3.1 Généralités

Les systèmes fondés sur la politique de QS permettent à l'administrateur de réseau de spécifier un ensemble de règles qui contrôlent le choix des flux qui ont besoin de recevoir un traitement préférentiel de transmission, ainsi que de spécifier l'ensemble particulier des comportements de transmission préférés. QPIM donne un modèle d'informations pour spécifier un tel ensemble de règles.

Les règles de politique de QS permettent de contrôler des environnements dans lesquels la signalisation RSVP est utilisée pour demander un traitement de transmission différent pour différents types de trafic de la part du réseau, ainsi que des environnements dans lesquels aucune signalisation n'est utilisée, mais un traitement préférentiel est désiré pour certains types de trafic (mais pas tous). Les règles de politique de QS permettent aussi de contrôler des environnements où de strictes garanties de QS sont fournies aux flux individuels, ainsi que des environnements où la QS est fournie à des agrégats de flux. Les actions de QS permettent à un PDP ou PEP de déterminer quelles demandes RSVP devraient être admises avant que les ressources du réseau soient allouées. Les actions de QS permettent le contrôle du contenu de la signalisation RSVP lui-même, ainsi que la différenciation entre les priorités des demandes RSVP. Les actions de QS permettent de contrôler la mise en application de la bordure du service différencié incluant la régulation, la mise en forme et le marquage, ainsi que le comportement par bond utilisé dans le cœur du réseau. Finalement, les actions de QS peuvent être utilisées pour contrôler la transposition des demandes RSVP à la bordure d'un nuage de service différencié en comportements par bond.

Quatre groupes d'actions sont dérivés des classes d'actions définies dans les [RFC3060] et [RFC3460]. Le premier groupe d'actions de QS contient une seule action, `QoSPolicyRSVPSimpleAction`. Cette action est utilisée pour le contrôle du signal RSVP et pour les actions d'installation. Le second groupe d'actions de QS détermine si un flux ou une classe de flux devrait être admis. Ceci est fait en spécifiant un profil de trafic approprié en utilisant la classe `QoSPolicyTrfcProf` et ses sous classes. Cet ensemble d'actions inclut aussi les actions de contrôle d'admission de QS, qui utilisent la classe `QoSPolicyAdmissionAction` et ses sous classes. Le troisième groupe d'actions contrôle les différenciations d'allocation de bande passante et de contrôle d'encombrement, qui ensemble spécifient le traitement de transmission du comportement par bond. Ce groupe d'actions inclut la classe `QoSPolicyPHBAction` et ses sous classes. La quatrième action de QS est une action inconditionnelle d'élimination de paquet, qui utilise la classe `QoSPolicyDiscardAction`. Cette action est utilisée soit par elle-même, soit comme bloc de construction de `QoSPolicyPoliceAction`.

Noter que certaines actions de QS ne sont pas directement modélisées. Elles sont plutôt modélisée en utilisant la classe `SimplePolicyAction` avec les associations appropriées. Par exemple, les trois actions de marquage (DSCP, IPP et CoS) sont modélisées en utilisant la classe `SimplePolicyAction`, et en associant cette classe à des variables et valeurs du type approprié défini dans la [RFC3460].

3.2 Actions de politique RSVP

Il y a trois types de décisions qu'un PDP (distant ou au sein d'un PEP) peut prendre pour évaluer une demande RSVP :

1. Admettre ou rejeter la demande
2. Ajouter ou modifier les paramètres d'admission de la demande
3. Modifier le contenu de la signalisation RSVP

La spécification de COPS pour RSVP [RFC2749] utilise différents types d'objet `Decision` pour modéliser chacune de ces décisions. QPIM suit la spécification de COPS pour RSVP et modélise chaque décision en utilisant une classe d'action différente.

`QoSPolicyRSVPAdmissionAction` contrôle la commande `Decision` et les objets `Fanions` de décision utilisés dans COPS pour RSVP. La classe `QoSPolicyRSVPAdmissionAction`, avec sa classe associée `QoSPolicyIntServTrfcProf`, est utilisée pour déterminer si on accepte ou rejette une certaine demande RSVP en comparant les paramètres `TSPEC` ou `RSPEC` de la demande RSVP au profil de trafic spécifié par le `QoSPolicyIntServTrfcProf`. Pour une description complète de la méthode de comparaison, voir la Section 4. Suivant la spécification de COPS pour RSVP, la décision d'admission a l'option d'accepter la demande et d'envoyer un avertissement au demandeur. Une `QoSPolicyRSVPAdmissionAction` peut être utilisée aussi pour limiter le nombre de réservations admises.

La classe `QoSPolicyRSVPSimpleAction`, qui est dérivée de la classe `PolicySimpleAction` [RFC3460], peut être utilisée pour contrôler les deux autres types de décision COPS RSVP. La propriété `qpRSVPActionType` désigne l'instance de la classe comme étant soit du type `'REPLACE'`, `'STATELESS'`, soit les deux (`'REPLACEANDSTATELESS'`). Pour les instances qui portent une valeur de propriété `qpRSVPActionType` de `'REPLACE'`, l'action est interprétée comme une décision COPS `Replace`, qui contrôle le contenu du message RSVP. Pour les instances qui portent une valeur de propriété `qpRSVPActionType` de `'STATELESS'`, l'action est interprétée comme une décision COPS `Stateless`, contrôlant les paramètres d'admission. Si ces deux actions sont requises, cela peut être fait en allouant la valeur `REPLACEANDSTATELESS` à la propriété `qpRSVPActionType`.

Cette classe est modélisée pour représenter les décisions COPS pour RSVP `Replace` et `Stateless`. Cela permet de la même façon l'utilisation à l'avenir de ces décisions COPS en étant directement contrôlées par une `QoSPolicySimpleAction`. La seule extension requise pourrait être la définition d'une nouvelle variable RSVP.

3.2.1 Exemple : Contrôle de la décision COPS Stateless

QoSPolicyRSVPSimpleAction permet la spécification des paramètres d'admission. Il permet la spécification de la priorité de préemption [RFC3181] d'une certaine demande de réservation RSVP. En utilisant la valeur de priorité de préemption, le PEP peut déterminer l'importance d'une réservation comparée aux réservations déjà admises, et si nécessaire peut préempter des réservations de priorité inférieure pour faire de la place pour celles de priorité supérieure. Cette classe peut aussi être utilisée pour contrôler la transposition des demandes RSVP dans le domaine des services différenciés en réglant la variable QoSPolicyRSVPDCLASSVariable à la valeur requise. Cela ordonne au PEP de marquer le trafic qui correspond aux spécifications de session et d'envoyeur portées dans une demande RSVP avec une certaine valeur de DSCP.

3.2.2 Exemple: contrôle de la décision COPS Replace

Un système de politique devrait être capable de contrôler les informations portées dans les messages RSVP. La QoSPolicyRSVPSimpleAction permet le contrôle du contenu des messages de signalisation RSVP. Un message RSVP peut porter un objet de politique de préemption [RFC3181] spécifiant la priorité de la demande de réservation par comparaison avec les autres demandes. Un message RSVP peut aussi porter un objet de politique pour des besoins d'authentification. Un message RSVP peut porter un objet DCLASS [RFC2996] qui spécifie au receveur ou à l'envoyeur la valeur particulière de DSCP qui devrait être mise sur le trafic de données. Une décision de données de remplacement de COPS pour RSVP contrôle le contenu du message RSVP en spécifiant un ensemble d'objets RSVP qui remplacent ou suppriment ceux existants.

3.3 Actions d'approvisionnement de politique

L'architecture de services différenciés [RFC2475] a été conçue pour fournir une différenciation de QS adaptable sans exiger qu'aucun protocole de signalisation fonctionne entre les hôtes et le réseau. Les actions de QS modélisées dans QPIM peuvent être utilisées pour contrôler tous les blocs de construction de l'architecture de services différenciés, incluant le comportement par bond, la classification de bordure, et la régulation et le formatage, sans qu'il soit besoin de spécifier les mécanismes de chemin des données utilisés par les mises en œuvre de PEP. Cela fournit un niveau d'abstraction qui cache les détails inutiles et permet à l'administrateur du réseau d'écrire les règles qui expriment les exigences du réseau sous une forme plus naturelle. Dans cette architecture, comme il ne se produit aucune signalisation entre l'hôte d'extrémité et le réseau avant que l'envoyeur ne commence à envoyer des informations, les mécanismes de QS devraient être établis à l'avance. Cela signifie habituellement que les PEP doivent être provisionnés à l'avance avec l'ensemble des règles de politique.

Les actions de régulation et de formatage sont modélisées comme des sous classes de l'action d'admission de QS. Le DSCP et le marquage de CoS sont modélisés en utilisant la classe SimplePolicyAction ([RFC3460]) associée aux variables et valeurs appropriées. Les actions d'allocation de bande passante et de contrôle d'encombrement sont modélisées comme des sous classes de QpQPolicyPHBAction, qui est elle-même une sous classe de la classe PolicyAction ([RFC3060])

3.3.1 Actions d'admission : contrôle des régulateurs et des formateurs

Les actions d'admission (QoSPolicyAdmissionAction et ses sous classes) sont utilisées pour réguler et /ou formater le trafic.

Chaque action d'admission est liée à un profil de trafic (QoSPolicyTrfcProf) via l'association QoSPolicyTrfcProfInAdmissionAction. Le profil de trafic est utilisé pour mesurer le trafic pour des besoins de régulation ou de formatage. Une action d'admission porte une propriété de portée (qpAdmissionScope) qui est utilisée pour déterminer si l'action contrôle des flux de trafic individuels ou des classes de trafic agrégées. Les concepts de "flux" et de "classe de trafic" sont expliqués dans la [RFC2475] en utilisant les termes de 'microflux' et de 'flux de trafic'. En gros, un flux est un ensemble de paquets portant un en-tête IP qui a les mêmes valeurs pour la source IP, la destination IP, le protocole et les accès de source et de destination de couche 4. Une classe de trafic est un ensemble de flux. Dans QPIM, des conditions simples et composées peuvent identifier les flux et/ou les classes de trafic en utilisant des termes booléens sur les valeurs des champs d'en-tête IP, incluant la valeur de l'octet Type de service.

Donc, l'interprétation de la propriété Portée est la suivante : si la valeur de la propriété Portée est 0 (par flux), chaque (micro) flux qui peut être confronté positivement à la condition de la règle est mesuré et régulé individuellement. Si la valeur de la propriété Portée est 1 (par classe) tous les flux confrontés à la condition de la règle sont mesurés comme un seul agrégat et régulés ensemble.

L'exemple suivant illustre l'utilisation de la propriété Portée. En utilisant deux actions de politique provisionnées, les politiques suivantes peuvent être mises en application :

- S'assurer que chaque flux HTTP ne va pas excéder 64 kbit/s
- S'assurer que le débit agrégé de tous les flux HTTP ne va pas excéder 512 kbit/s

Les deux politiques sont modélisées en utilisant la même classe `QoSPolicyPoliceAction` (dérivée de `QoSPolicyAdmissionAction`). La première politique a sa propriété `Portée` réglée à 'flux', tandis que la seconde a sa propriété `Portée` réglée à 'classe'. Les deux politiques sont modélisées en utilisant une règle avec deux actions de politique qui, dans une définition pseudo-formelle, ressemble à ce qui suit :

Si (HTTP) Action1=régulation, Profil de trafic =64 kbit/s, Portée1=flux
 Action2=régulation, Profil de trafic =512 kbit/s, Portée2=classe

L'action de régulation provisionnée `QoSPolicyPoliceAction` a trois associations, `QoSPolicyConformAction`, `QoSPolicyExceedAction` et `QoSPolicyViolateAction`.

Pour réaliser le résultat désiré déclaré ci-dessus, deux techniques de modélisation possibles peuvent être utilisées : les deux actions peuvent faire partie d'une seule règle de politique utilisant deux associations `PolicyActionInPolicyRule` [RFC3060]. Dans ce cas, la propriété `ExecutionStrategy` de la classe `PolicyRule` [RFC3460] DEVRAIT être réglée à "Faire Tout" afin que les deux flux individuels et les flux agrégés soient régulés.

Autrement, `Action1` et `Action2` pourraient être agrégés dans une instance de `CompoundPolicyAction` en utilisant les agrégations `PolicyActionInPolicyAction` [RFC3460]. Dans ce cas, afin que les flux individuels et les classes de trafic agrégées soient régulées, la propriété `ExecutionStrategy` de la classe `CompoundPolicyAction` [RFC3460] DEVRAIT être réglée à "Faire tout".

L'action de régulation est associée à un profil de trafic de baquet de jetons à trois niveaux qui porte les paramètres de débit, de salve et d'excès de salve. Le trafic mesuré par une unité peut être classé comme trafic conforme lorsque le débit mesuré est en dessous du débit défini par le profil de trafic, comme trafic excédentaire lorsque le trafic mesuré est au dessus de la salve normale et en dessous de la taille d'excès de salve, et comme trafic en violation lorsque le débit est au dessus de l'excès de salve maximum.

La [RFC3289] définit une mesure à deux niveaux, et fournit un moyen de combiner des mesures à deux niveaux dans une mesure plus complexe. Dans le présent document, un profil de trafic à trois niveaux est défini. Cela permet la construction à la fois de mesures à deux niveaux ainsi que de fournir une définition plus aisée des mesures à trois niveaux nécessaires pour créer les actions d'approvisionnement AF [RFC2597].

Une action de régulation qui modélise une régulation à trois niveaux DOIT associer trois actions séparées à un profil de trafic à trois niveaux. Ces actions sont une action de conformité, un action d'excédant et une action de violation. Une action de régulation qui modélise une régulation à deux niveaux utilise un profil de trafic à deux niveaux et associe seulement une action de conformité et une action d'excédant. Une action de régulation avec un profil de trafic à trois niveaux qui spécifie une action d'excédant mais ne spécifie pas d'action de violation implique que l'action entreprise lorsque le trafic est au dessus de la salve d'excès maximum est identique à l'action entreprise lorsque le trafic est au dessus de la salve normale. Un régulateur détermine si le profil est satisfait, mais les actions à effectuer sont déterminées par les associations `QoSPolicyXXXAction`.

Les formateurs sont utilisés pour retarder certains des paquets, ou tous, dans un flux de trafic, afin d'amener le flux à la conformité avec un profil de trafic. Un formateur a habituellement une mémoire tampon d'une taille finie, et les paquets peuvent être éliminés si il n'y a pas un espace suffisant dans la mémoire tampon pour contenir les paquets retardés. Le formatage est contrôlé par la classe `QoSPolicyShapeAction`. La seule association exigée est un profil de trafic qui spécifie les paramètres de débit et de salve auxquels les flux sortants devraient se conformer.

3.3.2 Marqueurs de contrôle

Trois types d'actions de contrôle de marquage sont modélisés dans QPIM : l'allocation des codets de services différenciés (DSCP, *Differentiated Services Code Point*) l'allocation de la préséance IP (IPP, *IP Precedence*) et l'allocation de la classe de service (CoS, *Class of Service*) de couche 2. Ces actions d'allocation elles-mêmes sont modélisées en utilisant la classe `SimplePolicyAction` associée aux variables et valeurs appropriées.

L'allocation des DSCP règle ("marques" ou "couleurs") le champ DS d'un en-tête de paquet à un codet DS particulier, en ajoutant le paquet marqué à un agrégat de comportement DS particulier.

Lorsque utilisée dans la forme de base, "Si <condition> alors 'DCSP = ds1'", l'action d'allocation alloue une valeur de DSCP (ds1) à tous les paquets qui ont pour résultat que la condition est évaluée à VRAI.

Lorsque utilisée en combinaison avec une action de régulation, une action d'allocation différente peut être produite via chacune des associations d'action 'conforme', 'excès' et 'violation'. De cette façon, on peut choisir un PHB dans un groupe de PHB en fonction de l'état d'une mesure.

La sémantique de l'allocation de DSCP est encapsulée dans l'appariement d'une variable DSCP et d'une valeur de DSCP au sein d'une seule instance de SimplePolicyAction via les associations appropriées.

L'allocation de l'IPP règle le champ IPP d'un en-tête de paquet à une valeur particulière d'IPP (de 0 à 7). La sémantique de l'allocation de l'IPP est encapsulée dans l'appariement d'une variable de type de service (ToS, *Type of Service*) (PolicyIPTosVariable) et d'une valeur de chaîne binaire () (définie dans la [RFC3460]) dans une seule instance de SimplePolicyAction via les associations appropriées. La valeur de la chaîne binaire est utilisée sous son format de gabarit de chaîne binaire. Le gabarit indique les trois bits pertinents du sous champ IPP dans l'octet de ToS, tandis que la chaîne binaire indique la valeur d'IPP à établir.

Les allocations de CoS contrôlent la transposition d'un comportement par bond en une classe de service de couche 2. Par exemple, la transposition d'un ensemble de valeurs de DSCP en valeur de priorité d'utilisateur 802.1p peut être spécifiée en utilisant une règle avec une condition décrivant le jeu de valeurs de DSCP, et une action d'allocation de CoS qui spécifie la transposition requise en la valeur de priorité d'utilisateur. La sémantique de l'allocation de CoS est encapsulée dans l'appariement d'une variable de CoS et d'une valeur de CoS (entier dans la gamme de 0 à 7) au sein d'une seule instance de SimplePolicyAction via les associations appropriées.

3.3.3 Contrôle des politiques de bordure - Exemples

En supposant que l'agrégation de comportement AF1 soit mis en application dans un domaine DS, les règles de politique sur les bordures du réseau devraient marquer les paquets à un des DSCP AF1x, selon la conformité du trafic à un profil de trafic prédéterminé à trois paramètres. QPIM modélise une telle action de régulation AF1 comme défini à la Figure 4.

```

+-----+ +-----+
| QoSPolicyPoliceAction |====| QoSPolicyTokenBucketTrfcProf |
| portée = classe      |    | débit = x, bc = y, be = z      |
+-----+ +-----+
*      @      #
*      @      #
*      @ +-----+ +-----+
*      @ | SimplePolicyAction |---| PolicyIntegerValue -AF13 |
*      @ +-----+ +-----+
*      @
* +-----+ +-----+
* | SimplePolicyAction |---| PolicyIntegerValue - AF12 |
* +-----+ +-----+
*
+-----+ +-----+
| SimplePolicyAction |---| PolicyIntegerValue - AF11 |
+-----+ +-----+

```

Légende des associations et agrégations :

**** : QoSPolicyConformAction

@@@@ : QoSPolicyExceedAction

: QoSPolicyViolateAction

==== : QoSTrfcProfInAdmissionAction

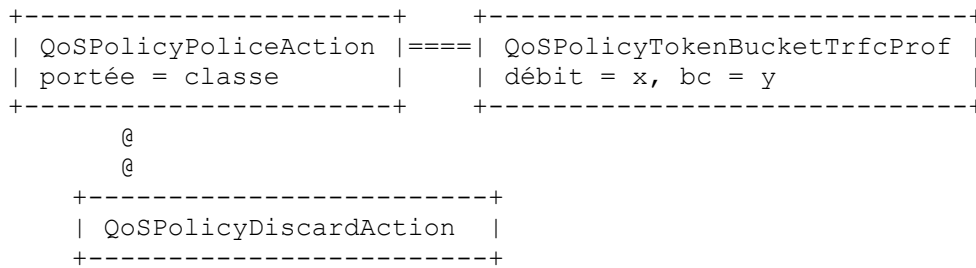
---- : PolicyValueInSimplePolicyAction ([RFC3460])

&&&& : PolicyVariableInSimplePolicyAction ([RFC3460], non montrée)

Figure 4. Régulation et marquage AF

L'action de régulation AF se compose d'une action de régulation, d'un profil de trafic à paquet de jetons et de trois instances de la classe SimplePolicyAction. Chacune des instances d'action simple de politique modélise une action de marquage différente. Chaque SimplePolicyAction utilise l'agrégation PolicyVariableInSimplePolicyAction pour spécifier que la PolicyDSCPVariable associée est réglée à la valeur d'entier appropriée. Ceci est fait en utilisant l'agrégation PolicyValueInSimplePolicyAction. Les trois agrégations PolicyVariableInSimplePolicyAction qui connectent les SimplePolicyActions appropriées aux variables DSCP appropriées, ne sont pas montrées dans cette figure pour la simplifier. AF11 est marqué sur le trafic conforme détecté; AF12 est marqué sur le trafic excédentaire détecté, et AF13 sur le trafic en violation détecté.

Le second exemple, à la Figure 5, est l'action de régulation la plus simple. Le trafic en dessous d'un profil de trafic à deux paramètres n'est pas modifié, tandis que le trafic qui excède le profil de trafic est éliminé.



Légende des associations et agrégations :

**** : QoSPolicyConformAction (non utilisée)

@@@@ : QoSPolicyExceedAction

: QoSPolicyViolateAction (non utilisée)

==== : QoSTrfcProfInAdmissionAction

Figure 5. Action simple de régulation

3.4 Actions de comportement par bond

Un comportement par bond (PHB, *Per-Hop Behavior*) est une description du comportement de transmission observable de l'extérieur d'un nœud de service différencié appliqué à un agrégat de comportement DS particulier [RFC2475]. L'approche retenue ici est qu'une action de PHB spécifie à la fois le comportement de transmission observable (par exemple, perte, retard, gigue) ainsi que la spécification des ressources de mémoire tampon et de bande passante qui doivent être allouées à chacun des agrégats de comportement pour réaliser ce comportement. C'est-à-dire, une règle avec un ensemble d'actions de PHB peut spécifier qu'un paquet EF ne doit pas être retardé plus de 20 ms à chaque bond. La même règle peut aussi spécifier qu'un paquet EF doit être traité avec une transmission prioritaire (par exemple, avec une mise en file d'attente prioritaire) et spécifier la bande passante maximum pour cette classe, ainsi que les ressources maximum de mémoire tampon. Les actions de PHB peuvent donc être utilisées à la fois pour représenter les exigences finales pour les PHB et fournir assez de détails pour être capables de transposer les actions de PHB en un ensemble de paramètres de configuration pour configurer les files d'attente, les programmeurs, les éliminateurs, et autres mécanismes.

La classe abstraite QoSPolicyPHBAction a deux sous classes. La classe QoSPolicyBandwidthAction est utilisée pour contrôler la bande passante, le retard et le comportement de transmission, tandis que la classe QoSPolicyCongestionControlAction est utilisée pour contrôler la taille de la file d'attente, les seuils et les algorithmes d'encombrement. La propriété qpMaxPacketSize de la classe QoSPolicyPHBAction spécifie en octets la taille de paquet, et est nécessaire lors de la traduction des actions de contrôle de bande passante et d'encombrement en configurations réelles de mise en œuvre. Par exemple, une mise en œuvre qui mesure la longueur de file d'attente en octets va avoir besoin d'utiliser cette propriété pour transposer la propriété qpQueueSize en la longueur de file d'attente désirée en octets.

3.4.1 Contrôle de la bande passante et du retard

QoSPolicyBandwidthAction permet de spécifier la bande passante minimale qui devrait être réservée pour une classe de trafic. La propriété qpMinBandwidth peut être spécifiée soit en kbit/s, soit comme un pourcentage de la bande passante totale disponible. La propriété qpBandwidthUnits est utilisée pour déterminer si les pourcentages ou des valeurs fixes sont utilisées.

La propriété qpForwardingPriority est utilisée chaque fois que la transmission préemptive est requise. Une règle de politique qui définit le PHB EF devrait indiquer une priorité de transmission non zéro. La propriété qpForwardingPriority contient une valeur d'entier pour permettre plusieurs niveaux de transmission préemptive lorsque des valeurs plus élevées sont utilisées pour spécifier une priorité plus élevée.

La propriété qpMaxBandwidth spécifie la bande passante maximum qui devrait être allouée à une classe de trafic. Cette propriété peut être spécifiée dans les actions de PHB avec une priorité de transmission non zéro afin de se protéger contre le risque de réduire à la famine les autres PHB.

Les propriétés qpMaxDelay et qpMaxJitter spécifient les limites au retard et à la gigue par bond en millisecondes pour tout paquet d'une certaine classe de trafic. La mise en application du retard et de la gigue maximums peut exiger l'utilisation d'une transmission préemptive ainsi que de contrôles de la bande passante minimum et maximum. L'application de valeurs faibles de retard et de gigue maximales peut aussi requérir des mécanismes de fragmentation et d'entrelaçage sur les liaisons à bas débit.

La propriété booléenne qpFairness indique si les flux devraient avoir une chance équitable d'être transmis sans abandon ou

retard. Une façon d'appliquer une action de bande passante avec `qpFairness` réglé à VRAI serait de construire une file d'attente par flux pour la classe de trafic spécifiée dans le filtre de la règle. De cette façon, des flux interactifs comme l'accès de terminal ne seraient pas mis en file d'attente derrière un flux saccadé (comme FTP) et aurait donc un temps de réponse raisonnable.

3.4.2 Actions de contrôle d'encombrement

La classe `QoSPolicyCongestionControlAction` contrôle la longueur de file d'attente, les seuils et les algorithmes de contrôle d'encombrement.

Un PEP devrait être capable de garder dans ses files d'attente `qpQueueSize` les paquets qui satisfont à la condition de la règle. Afin de fournir une taille de file d'attente indépendante de la vitesse de la liaison, la propriété `qpQueueSize` peut aussi être mesurée en millisecondes. L'intervalle de temps spécifie la durée nécessaire pour transmettre tous les paquets de la file d'attente si la vitesse de la liaison est dédiée entièrement à la transmission des paquets de cette file d'attente. La propriété `qpQueueSizeUnit` détermine si la taille de file d'attente est mesurée en nombre de paquets ou en millisecondes. La propriété `qpDropMethod` choisit des algorithmes de coupe en queue, en tête ou de façon aléatoire. L'ensemble des valeurs de seuil maximum et minimum peut être spécifié aussi en utilisant les propriétés `qpDropMinThresholdValue` et `qpDropMaxThresholdValue`, soit en paquets, soit en pourcentage de la taille totale de file d'attente disponible comme spécifié par la propriété `qpDropThresholdUnits`.

3.4.3 Utilisation de politiques hiérarchiques : exemples pour les actions de PHB

La définition hiérarchique de politique est un des principaux outils du modèle d'informations de politique de QS. L'incorporation de règle introduite dans la [RFC3460] permet la spécification de politiques hiérarchiques qui contrôlent les demandes RSVP, le formatage hiérarchisé, les actions de régulation et de marquage, ainsi que les programmeurs hiérarchiques et les définitions des différences entre les groupes de PHB.

Cet exemple donne un ensemble de règles qui spécifient les PHB mis en application dans un domaine de services différenciés. L'administrateur de réseau choisit de mettre en application les PHB EF, AF11 et AF13 et au mieux. Pour simplifier, AF12 n'est pas différencié. L'ensemble de règles prend la forme suivante :

```
Si (EF) alors faire les actions EF
Si (AF1) alors faire les actions AF1
    Si (AF11) alors faire les actions AF11
    Si (AF12) alors faire les actions AF12
    Si (AF13) alors faire les actions AF13
Si (défaut) alors faire les actions par défaut.
```

EF, AF1, AF11, AF12 et AF13 sont les conditions qui filtrent le trafic conformément aux valeurs de DSCP. La condition AF1 correspond au groupe de PHB AF1 entier incluant les valeurs de DSCP AF11, AF12 et AF13. La règle par défaut spécifie les règles au mieux (*Best Effort*). L'incorporation des règles AF1x au sein de la règle AF1 spécifie qu'il y a des précisions sur la façon dont le trafic AF1x devrait être traité par rapport au groupe de PHB AF1 entier. L'ensemble de règles réside dans un `PolicyGroup` avec une propriété `Stratégie de décision` réglée à 'FirstMatching' (*La première qui correspond*).

Les instances de classes ci-dessous spécifient l'ensemble d'actions utilisé pour décrire chaque PHB. Les tailles de file d'attente ne sont pas spécifiées, mais peuvent facilement être ajoutées à l'exemple.

Les actions utilisées pour décrire le PHB au mieux sont simples. Aucune bande passante n'est allouée au trafic au mieux. La première action spécifie que la classe de trafic au mieux devrait être traitée de façon équitable.

```
QoSPolicyBandwidthAction BE-B :
    qpFairness : VRAI
```

La seconde action spécifie que l'algorithme d'encombrement pour la classe de trafic au mieux devrait être aléatoire, et spécifie les seuils en pourcentage de la taille de file d'attente par défaut.

```
QoSPolicyCongestionControlAction BE-C :
    qpDropMethod : aléatoire
    qpDropThresholdUnits %
    qpDropMinThreshold : 10%
    qpDropMaxThreshold : 70%
```

EF exige une transmission préemptive. La bande passante maximum est aussi spécifiée pour s'assurer que la classe EF n'affame pas les autres classes. Le PHB EF utilise l'abandon de queue car les applications qui utilisent EF sont supposées être fondées sur UDP et ne pourraient donc pas tirer parti d'un abandon aléatoire.

QoSPolicyBandwidthAction EF-B :

qpForwardingPriority : 1
qpBandwidthUnits : %
qpMaxBandwidth : 50%
qpFairness : FAUX

QoSPolicyCongestionControlAction EF-C :

qpDropMethod : abandon en queue
qpDropThresholdUnits : paquet
qpDropMaxThreshold : 3 paquets

Les actions AF1 définissent l'allocation de bande passante pour le groupe de PHB entier :

QoSPolicyBandwidthAction AF1-B :

qpBandwidthUnits : %
qpMinBandwidth : 30%

Les actions AF1i spécifient les précisions de différenciation pour les PHB AF1x au sein du groupe de PHB AF1. Les différentes valeurs de seuil fournissent les différences de probabilité d'élimination des PHB AF1x au sein du groupe de PHB AF1.

QoSPolicyCongestionControlAction AF11-C :

qpDropMethod : aléatoire
qpDropThresholdUnits : paquet
qpDropMinThreshold : 6 paquets
qpDropMaxThreshold : 16 paquets

QoSPolicyCongestionControlAction AF12-C :

qpDropMethod : aléatoire
qpDropThresholdUnits : paquet
qpDropMinThreshold : 4 paquets
qpDropMaxThreshold : 13 paquets

QoSPolicyCongestionControlAction AF13-C :

qpDropMethod : aléatoire
qpDropThresholdUnits : paquet
qpDropMinThreshold : 2 paquets
qpDropMaxThreshold : 10 paquets

4. Profils de trafic

Les métriques mesurent l'état temporel d'un flux ou ensemble de flux par rapport à un profil de trafic. Dans le présent document, les profils de trafic sont modélisés par la classe QoSPolicyTrfcProf. L'association QoSPolicyTrfcProf InAdmissionAction lie le profil de trafic à l'action d'admission qui l'utilise. Deux profils de trafic sont dérivés de la classe abstraite QoSPolicyTrfcProf. Le premier est un baquet de jetons qui approvisionnent le profil de trafic en portant les paramètres de débit et de salve. Le second est un profil de trafic RSVP, qui permet de comparer les flux aux paramètres RSVP TSPEC et FLOWSPEC.

4.1 Profils de provisionnement de trafic

Les actions d'admission provisionnées, incluant de formatage et de régulation, sont spécifiées en utilisant un profil de trafic à baquet de jetons à deux ou trois paramètres. La classe QoSPolicyTokenBucketTrfcProf comporte les propriétés suivantes :

1. débit mesuré en kbit/s
2. salve normale mesurée en octets
3. salve excédentaire mesurée en octets

Débit détermine le débit moyen de transmission à long terme. Le trafic qui s'écoule en dessous de ce débit est conforme, tant que la salve normale n'est dépassée à aucun moment. Le trafic qui excède la salve normale mais est quand même en dessous de la salve excédentaire excède le profil de trafic. Le trafic au delà de la salve excédentaire est dit violer le profil de trafic.

La taille de la salve excédentaire est mesurée en octets en plus de la taille de salve. Une taille de salve excédentaire de zéro indique qu'aucune salve excédentaire n'est permise.

4.2 Profils de trafic RSVP

La politique d'admission RSVP peut conditionner la décision d'accepter ou refuser une demande RSVP sur la base de la spécification du trafic pour le flux (TSPEC) ou la quantité de ressources de QS demandées (FLOWSPEC). La décision d'admission peut se fonder sur la confrontation des demandes RSVP individuelles au profil de trafic ou en confrontant la somme agrégée de toutes les FLOWSPEC (TSPEC) actuellement admises, comme déterminé par la propriété `qpAdmissionScope` dans une `QoSPolicyRSVPAdmissionAction` associée.

La classe `QoSPolicyIntservTrfcProf` modélise ces deux profils de trafic. Cette classe a les propriétés suivantes:

1. Débit de jetons (r) mesuré en bit/s
2. Débit de crête (p) mesuré en bit/s
3. Taille de baquet (b) mesurée en octets
4. Unité régulée minimum (m) mesurée en octets
5. Taille maximum de paquet (M) mesurée en octets
6. Taux de réservation (R) mesuré en bit/s
7. Terme de surlongueur (s) mesuré en microsecondes

Les cinq premiers paramètres sont ceux de spécification du trafic utilisés dans l'architecture des services intégrés ([RFC1633]). Ces paramètres sont utilisés pour définir une TSPEC envoyeur aussi bien qu'une FLOWSPEC pour le service à charge contrôlée [RFC2211]. Pour une définition et une explication complète de leur signification, se reporter à la [RFC2210].

Les paramètres 6 et 7 sont utilisés pour la spécification de la FLOWSPEC du service garanti [RFC2212].

Un ordre partiel est défini entre les TSPEC (et les FLOWSPEC). La TSPEC A est plus grande que la TSPEC B si et seulement si $r_A > r_B$, $p_A > p_B$, $b_A > b_B$, $m_A < m_B$ et $M_A > M_B$. Une TSPEC (FLOWSPEC) mesurée par rapport à un profil de trafic utilise la même règle d'ordre. Un message RSVP n'est accepté que si sa TSPEC (FLOWSPEC) est inférieure ou égale au profil de trafic. Seuls les paramètres spécifiés dans les profils de trafic sont comparés.

La FLOWSPEC GS est comparée au débit R et au terme de surlongueur s . Le terme R ne devrait pas être supérieur au paramètre R du profil de trafic, tandis que le terme de surlongueur de la FLOWSPEC ne devrait pas être inférieur à celui spécifié dans le terme de surlongueur.

Des TSPEC ainsi que des FLOWSPEC peuvent être ajoutées. La somme de deux TSPEC se calcule en additionnant le débit r , le débit de crête p , la taille de baquet b , et en prenant la valeur minimum de l'unité régulée minimum m et la valeur maximum de la taille de paquet maximum M . Les FLOWSPEC GS sont additionnées en ajoutant le taux de réservation et en minimisant le terme de surlongueur s . Ces règles sont utilisées pour calculer l'état temporel des états RSVP admis qui correspondent à la classe de trafic définie par la condition de la règle. Cet état est comparé au profil de trafic pour arriver à une décision d'admission lorsque la portée de la `QoSPolicyRSVPAdmissionAction` est réglée à 'classe'.

5. Variables prédéfinies en rapport avec la QS

Des variables prédéfinies sont nécessaires pour assurer l'interopérabilité entre les serveurs de politique et les outils de gestion de politique provenant de fabricants différents. L'objet de la présente section est de définir les variables fréquemment utilisées dans les domaines de la politique de qualité de service.

On notera que cette section n'ajoute qu'aux classes de variable comme définies dans la [RFC3460] et réutilise les mécanismes définis ici.

Le modèle d'informations de politique de QS spécifie un ensemble de classes de variables prédéfinies pour prendre en charge un ensemble des termes de qualité de service fondamentaux qui sont couramment utilisés pour former les conditions et actions qui manquent dans la [RFC3460]. Les exemples incluent les variables qui se rapportent à RSVP. Toutes les classes de variables définies dans le présent document étendent la classe `QoSPolicyRSVPVariable` (définie dans le présent document) qui elle-même étend la classe `PolicyImplicitVariable`, définie dans la [RFC3460]. Des sous classes spécifient le type de données et la

sémantique des variables de politique.

Le présent document définit les classes de variable RSVP ; pour les détails, voir leur définition de classe :

Variables relatives à RSVP :

1. QoSPolicyRSVPSourceIPv4Variable – Adresse IPv4 de source du flux signalé comme RSVP, comme défini dans les objets RSVP PATH SENDER_TEMPLATE et RSVP RESV FILTER_SPEC [RFC2205].
2. QoSPolicyRSVPDestinationIPv4Variable – Accès de destination des flux signalés comme RSVP, comme défini dans les objets RSVP PATH et RESV SESSION [RFC2205] (pour le trafic IPv4).
3. QoSPolicyRSVPSourceIPv6Variable – Adresse IPv6 de source du flux signalé comme RSVP, comme défini dans les objets RSVP PATH SENDER_TEMPLATE et RSVP RESV FILTER_SPEC [RFC2205].
4. QoSPolicyRSVPDestinationIPv6Variable – Accès de destination du flux signalé comme RSVP, comme défini dans les objets RSVP PATH et RESV SESSION [RFC2205] (pour le trafic IPv6).
5. QoSPolicyRSVPSourcePortVariable – Accès de source du flux signalé comme RSVP, comme défini dans les objets RSVP PATH SENDER_TEMPLATE et RSVP RESV FILTER_SPEC [RFC2205].
6. QoSPolicyRSVPDestinationPortVariable – Accès de destination du flux signalé comme RSVP, comme défini dans les objets RSVP PATH et RESV SESSION [RFC2205].
7. QoSPolicyRSVPIPProtocolVariable – Protocole IP du flux signalé comme RSVP, comme défini dans les objets RSVP PATH et RESV SESSION [RFC2205].
8. QoSPolicyRSVPIPVersionVariable – Version des adresses IP portant le flux signalé comme RSVP, comme défini dans les objets RSVP PATH et RESV SESSION [RFC2205].
9. QoSPolicyRSVPDCLASSVariable – Valeurs de DSCP comme défini dans l'objet RSVP DCLASS [RFC2996].
10. QoSPolicyRSVPStyleVariable – Style de réservation (FF, SE, WF) comme défini dans le message RSVP RESV [RFC2205].
11. QoSPolicyRSVPIntServVariable – Type de service intégré (CL, GS, NULL) demandé dans le message RSVP Reservation, comme défini dans l'objet RSVP FLOWSPEC [RFC2205].
12. QoSPolicyRSVPMessageTypeVariable – Type de message RSVP, PATH, PATHTEAR, RESV, RESVTEAR, RESVERR, CONF ou PATHERR [RFC2205].
13. QoSPolicyRSVPPreemptionPriorityVariable – Priorité de réservation RSVP comme défini dans la [RFC3181].
14. QoSPolicyRSVPPreemptionDefPriorityVariable – Prémption de réservation RSVP défendant la priorité comme définie dans la [RFC3181].
15. QoSPolicyRSVPUserVariable – Identifiant de l'utilisateur qui a initié le flux comme défini dans la chaîne User Locator dans l'objet de politique Identity [RFC3182].
16. QoSPolicyRSVPApplicationVariable – Identifiant de l'application qui a généré le flux comme défini dans la chaîne de localisation d'application dans l'objet de politique Application [RFC2872].
17. QoSPolicyRSVPAuthMethodVariable - Type d'authentification RSVP utilisé dans l'objet de politique Identity [RFC3182].

Chaque classe restreint les types de valeurs possibles associés à une variable spécifique. Par exemple, la classe QoSPolicyRSVPSourcePortVariable est utilisée pour définir l'accès de source du flux signalé comme RSVP. La valeur associée à cette variable est du type PolicyIntegerValue.

6. Valeurs en rapport avec la QS

Les valeurs sont utilisées dans le modèle d'informations comme blocs de construction des conditions et actions de politique, comme décrit dans les [RFC3060] et [RFC3460]. La présente section définit un ensemble de valeurs auxiliaires qui sont utilisées pour les politiques de qualité de service ainsi que pour d'autres domaines de politique.

Toutes les classes de valeurs étendent la classe PolicyValue [RFC3460]. Les sous classes spécifient des types de données/valeurs spécifiques qui ne sont pas définis dans la [RFC3460].

Le présent document définit les deux sous classes suivantes de la classe PolicyValue :

QoSPolicyDNValue : Cette classe est utilisée pour représenter une seule valeur ou un ensemble de noms distinctifs [RFC2253], incluant des caractères génériques. Un nom distinctif est un nom qui peut être utilisé comme clé pour restituer un objet d'un service de répertoire. Cette valeur peut être utilisée par comparaison à des valeurs de référence portées dans des objets de politique RSVP, comme spécifié dans la [RFC3182]. Cette classe est définie au paragraphe 8.31.

QoSPolicyAttributeValue : Un terme de condition utilise la forme "la variable correspond à la valeur", et un terme d'action utilise la forme "régler la variable à la valeur" ([RFC3460]). Cette classe est utilisée pour représenter une seule ou un ensemble de valeurs de propriété pour le terme "valeur" dans une condition ou une

action. Cette valeur peut être utilisée en conjonction avec les valeurs de référence portées dans les objets RSVP, comme spécifié dans la [RFC3182]. Cette classe est définie au paragraphe 8.12.

Le nom de propriété est utilisé pour spécifier quelles propriétés dans l'instance de classe QoSPolicyAttributeValue est utilisée dans le terme de condition ou d'action. La valeur de cette ou ces propriétés sera alors restituée. Dans le cas d'une condition, une confrontation (qui dépend du nom de la propriété) sera utilisée pour voir si la condition est satisfaite ou non. Dans le cas d'une action, la sémantique sera alors "régler la variable à cette valeur".

Par exemple, supposons que les objets "usager" dans l'organisation incluent plusieurs propriétés, parmi lesquelles :

- Prénom
- Nom de famille
- Nom de connexion
- Département
- Titre

Une condition simple pourrait être construite pour identifier les flux par leur objet de politique RSVP porté par l'utilisateur. La condition simple : Nom de famille = "Smith" pour identifier un usager nommé Bill serait construite de la façon suivante :

Une SimplePolicyCondition [RFC3460] agrègerait un objet QoSPolicyRSVPUserVariable [QPIM], via l'agrégation PolicyVariableInSimplePolicyCondition [RFC3460].

La valeur implicite associée à cette condition est créée de la façon suivante :

Un objet QoSPolicyAttributeValue serait agrégé à l'objet simple condition via une PolicyValueInSimplePolicyCondition [RFC3460]. L'attribut QoSPolicyAttributeValue de qpAttributeName serait réglé à "nom de famille" et la qpAttributeValueList serait réglée à "Smith".

Un autre exemple est une condition qui se rapporte au département de l'organisation de l'usager. Il peut être construit exactement de la même façon, en changeant l'attribut QoSPolicyAttributeValue de qpAttributeName en "Département" et la qpAttributeValueList serait réglée à la valeur particulière qui doit être satisfaite (par exemple, "ingénierie" ou "service client"). La condition logique serait alors évaluée à vrai si l'usager appartient au département d'ingénierie ou au service client.

Noter que de nombreux objets multi-attributs exigent l'utilisation de la classe QoSPolicyAttributeValue pour spécifier exactement lequel de ses attributs devrait être utilisé dans l'opération de confrontation à la condition.

7. Définitions de classe : hiérarchie d'association

Les paragraphes qui suivent définissent les associations spécifiées par QPIM.

7.1 Association "QoSPolicyTrfcProfInAdmissionAction"

Cette association lie un objet QoSPolicyTrfcProf (défini au paragraphe 8.9) qui modélise un profil de trafic spécifique, à un objet QoSPolicyAdmissionAction (défini au paragraphe 8.2). La définition de classe pour cette association est la suivante :

NOM : QoSPolicyTrfcProfInAdmissionAction

DESCRIPTION : Classe qui représente l'association entre une action d'admission de QS et son profil de trafic.

DÉDUIT DE : Dependency (voir la [RFC3060])

ABSTRAITE : FAUX

PROPRIÉTÉS : Antecedent[ref QoSPolicyAdmissionAction [0..n]] ; Dependent[ref QoSPolicyTrfcProf [1..1]]

7.1.1 Référence "Antecedent"

Cette propriété est héritée de l'association Dependency, définie dans la [RFC3060]. Son type est outrepassé pour devenir une référence d'objet à un objet QoSPolicyAdmissionAction. Cela représente la partie "indépendante" de l'association. La cardinalité [0..n] indique que n'importe quel nombre d'objets QoSPolicyAdmissionAction peut utiliser une certaine QoSPolicyTrfcProf.

7.1.2 Référence "Dependent"

Cette propriété est héritée de l'association Dependency, et est outrepassée pour devenir une référence d'objet à un objet

QoSPolicyTrfcProf. Cela représente un profil de trafic spécifique qui est utilisé par un nombre quelconque d'objets QoSPolicyAdmissionAction. La cardinalité [1..1] signifie que exactement un objet de QoSPolicyTrfcProf peut être utilisé par une certaine QoSPolicyAdmissionAction.

7.2 Association "PolicyConformAction"

Cette association relie une action de régulation à un objet définissant une action à appliquer au trafic conforme par rapport au profil de trafic associé. La définition de classe pour cette association est la suivante :

NOM : PolicyConformAction

DESCRIPTION : Classe représentant l'association entre une action de régulation et l'action qui devrait être appliquée au trafic conforme à un profil de trafic associé.

DÉDUIT DE : Dependency (voir la [RFC3060])

ABSTRAITE : FAUX

PROPRIÉTÉS : Antecedent[ref QoSPolicyPoliceAction[0..n]], Dependent[ref PolicyAction [1..1]]

7.2.1 Référence "Antecedent"

Cette propriété est héritée de l'association Dependency. Son type est outrepassé pour devenir une référence d'objet à un objet QoSPolicyPoliceAction. Cela représente la partie "indépendante" de l'association. La cardinalité [0..n] indique que n'importe quel nombre d'objets QoSPolicyPoliceAction peut recevoir la même action à exécuter comme action conforme.

7.2.2 Référence "Dependent"

Cette propriété est héritée de l'association Dependency, et est outrepassée pour devenir une référence d'objet à un objet PolicyAction. Cela représente une action de politique spécifique qui est utilisée par une certaine QoSPolicyPoliceAction. La cardinalité [1..1] signifie que exactement une action de politique peut être utilisée comme action "conforme" pour une QoSPolicyPoliceAction. Pour exécuter plus d'une action conforme, utiliser la classe PolicyCompoundAction pour modéliser l'action conforme.

7.3 Association "QoSPolicyExceedAction"

Cette association relie une action de régulation à un objet qui définit une action à appliquer au trafic excédant le profil de trafic associé. La définition de classe pour cette association est la suivante :

NOM : QoSPolicyExceedAction

DESCRIPTION : Classe représentant l'association entre une action de régulation et l'action qui devrait être appliquée au trafic qui excède un profil de trafic associé.

DÉDUIT DE : Dependency (voir la [RFC3060])

ABSTRAITE : FAUX

PROPRIÉTÉS : Antecedent[ref QoSPolicePoliceAction[0..n]], Dependent[ref PolicyAction [1..1]]

7.3.1 Référence "Antecedent"

Cette propriété est héritée de l'association Dependency. Son type est outrepassé pour devenir une référence d'objet à un objet QoSPolicyPoliceAction. Cela représente la partie "indépendante" de l'association. La cardinalité [0..n] indique que n'importe quel nombre d'objets QoSPolicyPoliceAction peut recevoir la même action à exécuter comme action excédentaire.

7.3.2 Référence "Dependent"

Cette propriété est héritée de l'association Dependency, et est outrepassée pour devenir une référence d'objet à un objet PolicyAction. Cela représente une action de régulation spécifique qui est utilisée par une certaine QoSPolicyPoliceAction. La cardinalité [1..1] signifie que exactement une action de politique peut être utilisée comme action "excédentaire" par une QoSPolicyPoliceAction. Pour exécuter plus d'une action conforme, utiliser la classe PolicyCompoundAction pour modéliser l'action excédentaire.

7.4 Association "PolicyViolateAction"

Cette association relie une action de régulation à un objet définissant une action à appliquer au trafic qui viole le profil de trafic associé. La définition de classe pour cette association est la suivante :

NOM : PolicyViolateAction

DESCRIPTION : Classe représentant l'association entre une action de régulation et l'action qui devrait être appliquée au trafic qui viole un profil de trafic associé.

DÉDUIT DE : Dependency (voir la [RFC3060])

ABSTRAITE : FAUX

PROPRIÉTÉS : Antecedent[ref QoSPolicyPoliceAction[0..n]], Dependent[ref PolicyAction [1..1]]

7.4.1 Référence "Antecedent"

Cette propriété est héritée de l'association Dependency. Son type est outrepassé pour devenir une référence d'objet à un objet QoSPolicyPoliceAction. Cela représente la partie "indépendante" de l'association. La cardinalité [0..n] indique que n'importe quel nombre d'objets QoSPolicyPoliceAction peut recevoir la même action à exécuter comme action qui commet la violation.

7.4.2 Référence "Dependent"

Cette propriété est héritée de l'association Dependency, et est outrepassée pour devenir une référence d'objet à un objet PolicyAction. Cela représente une action de politique spécifique qui est utilisée par une certaine QoSPolicyPoliceAction. La cardinalité [1..1] signifie qu'exactly une action de politique peut être utilisée comme action de "violation" par une QoSPolicyPoliceAction. Pour exécuter plus d'une action de violation, utiliser la classe PolicyCompoundAction pour modéliser l'action conforme.

7.5 Agrégation "QoSPolicyRSVPVariableInRSVPSimplePolicyAction"

Une simple action de politique RSVP est représentée comme une paire {variable, valeur}. Cette agrégation fournit le lien entre une instance de QoSPolicyRSVPSimpleAction et une seule QoSPolicyRSVPVariable. L'agrégation PolicyValueInSimplePolicyAction relie la QoSPolicyRSVPSimpleAction à une seule PolicyValue. La définition de classe pour cette agrégation est la suivante :

NOM : QoSPolicyRSVPVariableInRSVPSimplePolicyAction

DÉDUIT DE : PolicyVariableInSimplePolicyAction

ABSTRAITE : FAUX

PROPRIÉTÉS : GroupComponent[ref QoSPolicyRSVPSimpleAction [0..n]], PartComponent[ref QoSPolicyRSVPVariable [1..1]]

7.5.1 Référence "GroupComponent"

La propriété de référence "GroupComponent" est héritée de PolicyComponent, et est outrepassée pour devenir une référence d'objet à une QoSPolicyRSVPSimpleAction qui contient exactement une QoSPolicyRSVPVariable. Noter que pour toute instance seule de la classe d'agrégation QoSPolicyRSVPVariableInRSVPSimplePolicyAction, cette propriété est à une seule valeur. La cardinalité [0..n] indique qu'il peut y avoir 0, 1, ou plusieurs objets QoSPolicyRSVPSimpleAction qui contiennent tout objet de variable RSVP.

7.5.2 Référence "PartComponent"

La propriété de référence "PartComponent" est héritée de PolicyComponent, et est outrepassée pour devenir une référence d'objet à une QoSPolicyRSVPVariable qui est définie dans la portée d'une QoSPolicyRSVPSimpleAction. Noter que pour toute instance seule de la classe d'association QoSPolicyRSVPVariableInRSVPSimplePolicyAction, cette propriété (comme toutes les propriétés de référence) est à une seule valeur. La cardinalité [1..1] indique qu'une QoSPolicyRSVPVariableInRSVPSimplePolicyAction doit avoir exactement une variable RSVP définie dans sa portée pour être significative.

8. Définitions de classe : hiérarchie d'héritage

La présente section définit les classes d'objet spécifiées par QPIM.

8.1 Classe QoSPolicyDiscardAction

Cette classe est utilisée pour spécifier que les paquets devraient être éliminés. C'est la même chose que de déclarer que les paquets devraient être refusés à la transmission. La définition de la classe est la suivante :

NOM : QoSPolicyDiscardAction

DESCRIPTION : Cette action spécifie que les paquets devraient être éliminés.

DÉDUIT DE : PolicyAction (définie dans la [RFC3060])

ABSTRAITE : FAUX

PROPRIÉTÉS : Aucune

8.2 Classe QoSPolicyAdmissionAction

Cette classe est la classe de base pour prendre les décisions d'admission fondées sur une comparaison des mesures du comportement temporel d'un flux ou ensemble de flux avec un profil de trafic. La propriété qpAdmissionScope contrôle si la comparaison est faite par flux ou par classe (de flux). Seuls les paquets qui se conforment au profil de trafic sont admis à la suite du traitement ; les autres paquets sont éliminés. La définition de la classe est la suivante :

NOM : QoSPolicyAdmissionAction

DESCRIPTION : Cette action contrôle les décisions d'admission sur la base de la comparaison d'une mesure à un profil de trafic.

DÉDUIT DE : PolicyAction (définie dans la [RFC3060])

ABSTRAITE : FAUX

PROPRIÉTÉS : qpAdmissionScope

8.2.1 Propriété qpAdmissionScope

Cet attribut spécifie si la décision d'admission est prise par flux ou pour la classe entière de flux définie par la condition de la règle. Si la portée est le "flux", le débit réel ou demandé de chaque flux est comparé au profil de trafic. Si la portée est réglée à "classe", le débit agrégé réel ou demandé de tous les flux correspondants à la condition de la règle est mesuré par rapport au profil de trafic. La propriété est définie comme suit :

NOM : qpAdmissionScope

DESCRIPTION : Cette propriété spécifie si la décision d'admission est prise par flux ou pour la classe de flux entière.

SYNTAXE : Entier

VALEUR : Valeur d'entier. 0 spécifie que l'admission est faite par flux, et 1 spécifie que l'admission est faite par classe.

8.3 Classe QoSPolicyPoliceAction

Cette classe est utilisée pour définir les actions de régulation (c'est-à-dire, ces actions qui restreignent le trafic sur la base d'une comparaison avec un profil de trafic). En utilisant les trois associations QoSPolicyConformAction, QoSPolicyExceedAction et QoSPolicyViolateAction, il est possible de spécifier différentes actions à entreprendre selon que le trafic est conforme, en excès, ou en violation d'un profil de trafic. Le profil de trafic est spécifié dans une sous classe de la classe QoSPolicyTrfcProf. La définition de la classe est la suivante :

NOM : QoSPolicyPoliceAction

DESCRIPTION : Cette action contrôle le fonctionnement des régulateurs. Le débit des flux est mesuré par rapport au profil de trafic. Les actions qui doivent être effectuées sur le trafic conforme, en excès, et en violation sont indiquées en utilisant les associations d'action conforme, en excès et en violation.

DÉDUIT DE : QoSPolicyAdmissionAction (défini dans le présent document)

ABSTRAITE : FAUX

PROPRIÉTÉS : Aucune

8.4 Classe QoSPolicyShapeAction

Cette classe est utilisée pour définir des actions de formatage. Les formateurs sont utilisés pour retarder certains des paquets, ou tous, dans un flux de trafic afin d'amener un certain flux de trafic à conformité à un certain profil de trafic. Le profil de trafic est spécifié dans une sous classe de la classe QoSPolicyTrfcProf. La définition de la classe est la suivante :

NOM : QoSPolicyShapeAction

DESCRIPTION : Cette action indique que le trafic devrait être formaté pour être conforme à un profil de trafic.

DÉDUIT DE : QoSPolicyAdmissionAction (défini dans le présent document)

ABSTRAITE : FAUX

PROPRIÉTÉS : Aucune

8.5 Classe QoSPolicyRSVPAdmissionAction

Cette classe détermine si il faut accepter ou rejeter une certaine demande RSVP en comparant les paramètres TSPEC ou RSPEC de la demande au profil de trafic associé et/ou en appliquant la limite maximum de session pré établie. Le profil de trafic est spécifié dans la classe QoSPolicyIntServTrfcProf. Cette classe hérite de la propriété qpAdmissionScope de sa super classe. Cette propriété spécifie si l'admission devrait être faite par flux ou par classe. Si le profil de trafic n'est pas supérieur ou égal à la réservation demandée, ou à la somme de la réservation admise fusionnée avec la réservation demandée, le résultat est une décision de refus. Si aucun profil de trafic n'est spécifié, l'hypothèse est que tout le trafic peut être admis. La définition de la classe est la suivante :

NOM : QoSPolicyRSVPAdmissionAction

DESCRIPTION : Cette action contrôle l'admission des demandes RSVP. Selon la portée, une seule demande RSVP ou la totalité des demandes RSVP admises satisfaisant les conditions est comparée au profil de trafic.

DÉDUIT DE : QoSPolicyAdmissionAction (défini dans le présent document)

ABSTRAITE : FAUX

PROPRIÉTÉS : qpRSVPWarnOnly, qpRSVPMaxSessions

8.5.1 Propriété qpRSVPWarnOnly

Cette propriété est applicable lorsque satisfaisable ("admettre") une demande RSVP violerait les limites du régulateur (profil de trafic) ou lorsque le nombre maximum de sessions serait dépassé (ou les deux).

Lorsque cette propriété est réglée à VRAI, la demande RSVP est admise en dépit de la violation, mais un message d'erreur RSVP portant un avertissement est envoyé au générateur (envoyeur ou receveur). Lorsque réglée à FAUX, la demande devrait être refusée et un message d'erreur renvoyé à l'origine de la demande. Ainsi, la signification du fanion qpWarnOnly est : sur la base de la valeur de la propriété (VRAI ou FAUX) détermine soit d'admettre mais en avertissant le générateur que la demande est en violation de la règle, soit de refuser la demande tout en renvoyant une erreur.

Spécifiquement, une PATHERR (en réponse à un message Path) ou une RESVERR (en réponse à un message RESV) sera envoyé. Cela suit le fanion d'erreur de COPS pour RSVP envoyé dans l'objet Decision Flags. Cette propriété est définie comme suit :

NOM : qpRSVPWarnOnly

SYNTAXE : Booléen

Par défaut : FAUX

VALEUR : La valeur VRAI signifie que la demande devrait être admise ET qu'un message d'avertissement RSVP devrait être envoyé au générateur. La valeur FAUX signifie que la demande ne devrait pas être admise et qu'un message d'erreur approprié devrait être renvoyé à l'origine de la demande.

8.5.2 Propriété qpRSVPMaxSessions

Cet attribut est utilisé pour limiter le nombre total de demandes RSVP admises pour la classe de trafic spécifiée. Pour que cette propriété soit significative, la propriété qpAdmissionScope doit être réglée à classe. La définition de cette propriété est la suivantes :

NOM : qpRSVPMaxSessions

SYNTAXE : Entier

VALEUR : Doit être supérieure à 0.

8.6 Classe QoSPolicyPHBAction

Cette classe est une classe de base qui est utilisée pour définir le comportement par bond qui doit être alloué aux agrégats de comportements. Elle définit une propriété commune, `qpMaxPacketSize`, à utiliser par ses sous classes (`QoSPolicyBandwidthAction` et `QoSPolicyCongestionControlAction`). La définition de la classe est la suivante :

NOM : `QoSPolicyPHBAction`

DESCRIPTION : Cette action contrôle le comportement par bond fourni aux agrégats de comportements.

DÉDUIT DE : `PolicyAction` (définie dans la [RFC3060])

ABSTRAITE : VRAI

PROPRIÉTÉS : `qpMaxPacketSize`

8.6.1 Propriété `qpMaxPacketSize`

Cette propriété spécifie en octets la taille maximum de paquet des paquets dans le flux désigné. Cet attribut est utilisé dans la traduction des attributs QPIM en mécanismes de QS utilisés au sein d'un PEP. Par exemple, la longueur de file d'attente peut être mesurée en octets, tandis que le nombre minimum de paquets qui devraient être conservés dans un PEP est défini dans QPIM en nombre de paquets. Cette propriété est définie comme suit :

NOM : `qpMaxPacketSize`

SYNTAXE : Entier

VALEUR : Doit être supérieure à 0

8.7 Classe QoSPolicyBandwidthAction

Cette classe est utilisée pour contrôler la bande passante, le retard, et le comportement de transmission d'un PHB. La définition de la classe est la suivante :

NOM : `QoSPolicyBandwidthAction`

DESCRIPTION : Cette action contrôle la bande passante, le retard, et les caractéristiques de transmission du PHB.

DÉDUIT DE : `QoSPolicyPBHAction` (défini dans le présent document)

ABSTRAITE : FAUX

PROPRIÉTÉS : `qpForwardingPriority`, `qpBandwidthUnits`, `qpMinBandwidth`, `qpMaxBandwidth`, `qpMaxDelay`, `qpMaxJitter`, `qpFairness`

8.7.1 Propriété `qpForwardingPriority`

Cette propriété définit la priorité de transmission de cet ensemble de flux. Une valeur non zéro indique que la préemption de transmission est requise. Des valeurs supérieures représentent des priorités de transmission supérieures. Cette propriété est définie comme suit :

NOM : `qpForwardingPriority`

SYNTAXE : Entier

VALEUR : Doit être non négative. La valeur 0 signifie que la préemption de transmission n'est pas exigée. Une valeur positive indique la priorité qui doit être allouée à ce (cet ensemble de) flux. Les plus grandes valeurs représentent les plus fortes priorités.

8.7.2 Propriété `qpBandwidthUnits`

Cette propriété définit les unités qu'ont les propriétés `qpMinBandwidth` et `qpMaxBandwidth`. La bande passante peut être définie en bit/s ou en pourcentage de la bande passante disponible ou des ressources du programmeur. Cette propriété est définie comme suit :

NOM : `qpBandwidthUnits`

SYNTAXE : Entier

VALEUR : Deux valeurs sont possibles. La valeur 0 est utilisée pour spécifier des bit/s, tandis que la valeur 1 est utilisée pour spécifier un pourcentage de la bande passante disponible. Si cette propriété indique que la bande passante est un pourcentage, chaque propriété de bande passante exprime un pourcentage chiffré, et donc son maximum est 100.

8.7.3 Propriété `qpMinBandwidth`

Cette propriété définit la bande passante minimum qui devrait être réservée pour cette classe de trafic. Les valeurs relatives (c'est-à-dire, en pourcentage de la bande passante) et absolues (c'est-à-dire, en bit/s) peuvent être spécifiées selon la valeur de la propriété `qpBandwidthUnits`. Cette propriété est définie comme suit :

NOM : `qpMinBandwidth`

SYNTAXE : Entier

VALEUR : La valeur doit être supérieure à 0. Si la propriété `qpMaxBandwidth` est définie, la valeur de `qpMinBandwidth` doit alors être inférieure ou égale à la valeur de `qpMaxBandwidth`.

8.7.4 Propriété `qpMaxBandwidth`

Cette propriété définit la bande passante maximum qui devrait être allouée à cette classe de trafic. Les valeurs relatives (c'est-à-dire, en pourcentage de la bande passante) et absolues (c'est-à-dire, en bit/s) peuvent être spécifiées selon la valeur de la propriété `qpBandwidthUnits`. Cette propriété est définie comme suit :

NOM : `qpMaxBandwidth`

SYNTAXE : Entier

VALEUR : La valeur doit être supérieure à 0. Si la propriété `qpMaxBandwidth` est définie, alors la valeur de `qpMinBandwidth` doit être inférieure ou égale à la valeur de `qpMaxBandwidth`.

8.7.5 Propriété `qpMaxDelay`

Cette propriété définit le retard maximal par bond que devrait subir le trafic de cette classe lorsque il est transmis sur ce bond. Le retard maximum est mesuré en microsecondes. Cette propriété est définie comme suit :

NOM : `qpMaxDelay`

SYNTAXE : Entier (microsecondes)

VALEUR : La valeur doit être supérieure à 0.

8.7.6 Propriété `qpMaxJitter`

Cette propriété définit la variance maximale du retard par bond que le trafic de cette classe devrait subir lors de sa transmission sur ce bond. La gigue maximum est mesurée en microsecondes. Cette propriété est définie comme suit :

NOM : `qpMaxJitter`

SYNTAXE : Entier (microsecondes)

VALEUR : La valeur doit être supérieure à 0.

8.7.7 Propriété `qpFairness`

Cette propriété définit si une mise en file d'attente équitable est requise pour cette classe de trafic. Cette propriété est définie comme suit :

NOM : `qpFairness`

SYNTAXE : Booléen

VALEUR : La valeur de FAUX signifie que la mise en file d'attente équitable n'est pas requise pour cette classe de trafic, tandis que la valeur de VRAI signifie que la mise en file d'attente équitable est exigée pour cette classe de trafic.

8.8 Classe `QoSPolicyCongestionControlAction`

Cette classe est utilisée pour contrôler les caractéristiques de l'algorithme de contrôle d'encombrement utilisé. La définition de la classe est la suivante :

NOM : `QoSPolicyCongestionControlAction`

DESCRIPTION : Cette action contrôle les caractéristiques du contrôle d'encombrement du PHB.

DÉDUIT DE : `QoSPolicyPBHAction` (défini dans le présent document)

ABSTRAITE : FAUX

PROPRIÉTÉS : qpQueueSizeUnits, qpQueueSize, qpDropMethod, qpDropThresholdUnits, qpDropMinThresholdValue, qpDropMaxThresholdValue

8.8.1 Propriété qpQueueSizeUnits

Cette propriété spécifie les unités dans lesquelles l'attribut qpQueueSize est mesuré. La taille de file d'attente est mesurée en nombre de paquets ou en unités de temps. L'intervalle de temps spécifie le temps nécessaire pour transmettre tous les paquets dans la file d'attente si la vitesse de la liaison est dédiée entièrement à la transmission des paquets de cette file d'attente. La définition de la propriété est :

NOM : qpQueueSizeUnits

SYNTAXE : Entier

VALEUR : Cette propriété peut avoir deux valeurs. Si la valeur est réglée à 0, l'unité de mesure est alors le nombre de paquets. Si la valeur est réglée à 1, l'unité de mesure est alors la milliseconde.

8.8.2 Propriété qpQueueSize

Cette propriété spécifie la taille maximum de la file d'attente en paquets ou en millisecondes, selon la valeur de qpQueueSizeUnits (0 spécifie les paquets, et 1 spécifie les millisecondes). Cette propriété est définie comme suit :

NOM : qpQueueSize

SYNTAXE : Entier

VALEUR : Cette valeur doit être supérieure à 0.

8.8.3 Propriété qpDropMethod

Cette propriété spécifie l'algorithme d'abandon du contrôle d'encombrement qui devrait être utilisé pour ce type de trafic. Cette propriété est définie comme suit :

NOM : qpDropMethod

SYNTAXE : Entier

VALEURS : Trois valeurs sont actuellement définies. La valeur 0 spécifie un algorithme d'abandon aléatoire, la valeur 1 spécifie un algorithme d'abandon de queue, et la valeur 2 spécifie un algorithme d'abandon en tête.

8.8.4 Propriété qpDropThresholdUnits

Cette propriété spécifie les unités dans lesquelles les deux propriétés qpDropMinThresholdValue et qpDropMaxThresholdValue sont mesurées. Les seuils peuvent être mesurés en paquets ou en pourcentage des tailles de file d'attente disponible. Cette propriété est définie comme suit :

NOM : qpDropThresholdUnits

SYNTAXE : Entier

VALEURS : Trois valeurs sont définies. La valeur 0 définit l'unité comme nombre de paquets, 1 définit l'unité comme pourcentage de la taille de la file d'attente et 2 définit l'unité en millisecondes. Si cette propriété indique que l'unité de seuil est un pourcentage, chacune des propriétés du seuil exprime alors un pourcentage entièrement numérique, et donc son maximum est 100.

8.8.5 Propriété qpDropMinThresholdValue

Cette propriété spécifie le nombre minimum de ressources de mise en file d'attente et de mise en mémoire tampon qui devraient être réservées pour cette classe de flux. Le seuil peut être spécifié comme valeur relative (c'est-à-dire, un pourcentage) ou absolue (c'est-à-dire, le nombre de paquets ou de millisecondes) selon la valeur de la propriété qpDropThresholdUnits. Si cette propriété spécifie une valeur de 5 paquets, des ressources de mémoire tampon et de file d'attente suffisantes devraient alors être réservées pour contenir 5 paquets avant de faire fonctionner l'algorithme spécifié d'abandon du contrôle d'encombrement. Cette propriété est définie comme suit :

NOM : qpDropMinThresholdValue

SYNTAXE : Entier

VALEUR : Cette valeur doit être supérieure ou égale à 0. Si la propriété qpDropMaxThresholdValue est définie, la valeur de la propriété qpDropMinThresholdValue doit alors être inférieure ou égale à celle de la propriété qpDropMaxThresholdValue.

8.8.6 Propriété `qpDropMaxThresholdValue`

Cette propriété spécifie le nombre maximum de ressources de mise en file d'attente et de mise en mémoire tampon qui devraient être réservées pour cette classe de flux. Le seuil peut être spécifié comme valeur relative (c'est-à-dire, un pourcentage) ou absolue (c'est-à-dire, nombre de paquets ou de millisecondes) selon la valeur de la propriété `qpDropThresholdUnits`. Les mécanismes d'abandon du contrôle d'encombrement ne devraient pas conserver plus de paquets que la valeur spécifiée dans cette propriété. Noter cependant que certains mécanismes d'abandon peuvent calculer des moyennes d'occupation de file d'attente, et donc les ressources de file d'attente maximum réelles devraient être plus grandes. Cette propriété est définie comme suit :

NOM : `qpDropMaxThresholdValue`

SYNTAXE : Entier

VALEUR : Cette valeur doit être supérieure ou égale à 0. Si la propriété `qpDropMinThresholdValue` est définie, la valeur de la propriété `qpDropMinThresholdValue` doit alors être inférieure ou égale à la valeur de la propriété `qpDropMaxThresholdValue`.

8.9 Classe `QoSPolicyTrfcProf`

C'est une classe de base abstraite qui modélise un profil de trafic. Les profils de trafic spécifient les paramètres de débit maximum utilisés dans les décisions d'admission. L'association `QoSPolicyTrfcProfInAdmissionAction` lie la décision d'admission au profil de trafic. La définition de la classe est la suivante :

NOM : `QoSPolicyTrfcProf`

DÉDUIT DE : `Policy` (définie dans la [RFC3060])

ABSTRAITE : VRAI

PROPRIÉTÉS : Aucune

8.10 Classe `QoSPolicyTokenBucketTrfcProf`

Cette classe modélise un profil de trafic à deux ou trois niveaux de baquet de jetons. Des profils supplémentaires peuvent être modélisés en mettant en cascade plusieurs instances de cette classe (par exemple, en connectant le résultat d'une instance à l'entrée d'une autre instance). Ce profil de trafic porte les valeurs du régulateur ou formateur à mettre en application sur un flux ou ensemble de flux. La définition de la classe est la suivante :

NOM : `QoSPolicyTokenBucketTrfcProf`

DÉDUIT DE : `QoSPolicyTrfcProf` (défini dans le présent document)

ABSTRAITE : FAUX

PROPRIÉTÉS : `qpTBRate`, `qpTBNormalBurst`, `qpTBExcessBurst`

8.10.1 Propriété `qpTBRate`

C'est un entier non négatif qui définit le débit de jetons en kilobits par seconde. Un débit de zéro signifie que tous les paquets seront hors du profil. Cette propriété est définie comme suit :

NOM : `qpTBRate`

SYNTAXE : Entier

VALEUR : Cette valeur doit être supérieure à zéro

8.10.2 Propriété `qpTBNormalBurst`

Cette propriété est un entier qui définit la taille normale d'une salve mesurée en octets. Cette propriété est définie comme suit :

NOM : `qpTBNormalBurst`

SYNTAXE : Entier

VALEUR : Cette valeur doit être supérieure à zéro

8.10.3 Propriété qpTBExcessBurst

Cette propriété est un entier qui définit l'excès de la taille de salve mesurée en octets. Cette propriété est définie comme suit :

NOM : qpTBExcessBurst

SYNTAXE : Entier

VALEUR : Cette valeur doit être supérieure ou égale à qpTBNormalBurst

8.11 Classe QoSPolicyIntServTrfcProf

Cette classe représente un profil de trafic IntServ. Les valeurs des profils de trafic IntServ sont comparées à la spécification de trafic (TSPEC) et aux demandes de réservation de QS (FLOWSPEC) portées dans les demandes RSVP. La définition de la classe est la suivante :

NOM : QoSPolicyIntServTrfcProf

DÉDUIT DE : QoSPolicyTrfcProf (défini dans le présent document)

ABSTRAITE : FAUX

PROPRIÉTÉS : qpISTokenRate, qpISPeakRate, qpISBucketSize, qpISResvRate, qpISResvSlack, qpISMinPolicedUnit, qpISMaxPktSize

8.11.1 Propriété qpISTokenRate

Cette propriété est un entier non négatif qui définit le paramètre de débit de jetons, mesuré en kilobits par seconde. Cette propriété est définie comme suit :

NOM : qpISTokenRate

SYNTAXE : Entier

VALEUR : Cette valeur doit être supérieure ou égale à 0

8.11.2. Propriété qpISPeakRate

Cette propriété est un entier non négatif qui définit le paramètre débit de crête, mesuré en kilobits par seconde. Cette propriété est définie comme suit :

NOM : qpISPeakRate

SYNTAXE : Entier

VALEUR : Cette valeur doit être supérieure ou égale à 0

8.11.3 Propriété qpISBucketSize

Cette propriété est un entier non négatif qui définit le paramètre taille de baquet de jetons, mesuré en octets. Cette propriété est définie comme suit :

NOM : qpISBucketSize

SYNTAXE : Entier

VALEUR : Cette valeur doit être supérieure ou égale à 0

8.11.4 Propriété qpISResvRate

Cette propriété est un entier non négatif qui définit le taux de réservation (R-Spec) dans la réservation de service garanti RSVP. Elle est mesurée en kilobits par seconde. Cette propriété est définie comme suit :

NOM : qpISResvRate

SYNTAXE : Entier

VALEUR : Cette valeur doit être supérieure ou égale à 0

8.11.5 Propriété qpISResvSlack

Cette propriété est un entier non négatif qui définit le terme de surlongueur RSVP dans la réservation de service garanti RSVP. Elle est mesurée en microsecondes. Cette propriété est définie comme suit :

NOM : qpISResvSlack
 SYNTAXE : Entier
 VALEUR : Cette valeur doit être supérieure ou égale à 0

8.11.6 Propriété qpISMinPolicedUnit

Cette propriété est un entier non négatif qui définit l'unité régulée RSVP minimum, mesurée en octets. Cette propriété est définie comme suit :

NOM : qpISMinPolicedUnit
 SYNTAXE : Entier
 VALEUR : Cette valeur doit être supérieure ou égale à 0

8.11.7 Propriété qpISMaxPktSize

Cette propriété est un entier positif qui définit la taille maximum admise de paquet pour les messages RSVP, mesurée en octets. Cette propriété est définie comme suit :

NOM : qpISMaxPktSize
 SYNTAXE : Entier
 VALEUR : Cette valeur doit être un entier positif, notant le nombre d'octets dans la plus grande charge utile de paquet d'un flux ou classe signalé RSVP.

8.12 Classe QoSPolicyAttributeValue

Cette classe peut être utilisée pour représenter indirectement des références de variable et de valeur soit dans une condition simple (" $\langle x \rangle$ satisfait $\langle y \rangle$ ") soit une simple action (" $\langle x \rangle = \langle y \rangle$ "). Dans les deux cas, $\langle x \rangle$ et $\langle y \rangle$ sont connues comme la variable et la valeur de la condition ou action. La valeur des propriétés qpAttributeName et qpAttributeValueList est utilisée pour substituer respectivement $\langle x \rangle$ et $\langle y \rangle$ dans la condition ou action.

La substitution est faite comme suit : la valeur de la propriété qpAttributeName est utilisée pour substituer $\langle x \rangle$ et la valeur de la propriété qpAttributeValueList est utilisée pour substituer $\langle y \rangle$.

Une fois la substitution faite, la condition peut être évaluée et l'action peut être effectuée.

Par exemple, supposons qu'on veuille définir une condition sur un nom d'utilisateur de la forme "usager == 'Smith'", en utilisant la classe QoSPolicyRSVPUserVariable. Les informations d'utilisateur dans le message RSVP fournissent un DN. Le DN pointe sur des objets d'utilisateur détenant de nombreux attributs. Si l'attribut pertinent est "nom de famille", on va utiliser la classe QoSPolicyAttributeValue avec qpAttributeName = "Nom de famille", qpAttributeValueList = {"Smith"}.

La définition de la classe est la suivante :

NOM : QoSPolicyAttributeValue
 DÉDUIT DE : PolicyValue (définie dans la [RFC3460])
 ABSTRAITE : FAUX
 PROPRIÉTÉS : qpAttributeName, qpAttributeValueList

8.12.1 Propriété qpAttributeName

Cette propriété porte le nom de l'attribut qui est à utiliser pour substituer $\langle x \rangle$ dans une simple condition ou une simple condition de la forme " $\langle x \rangle$ satisfait $\langle y \rangle$ " ou " $\langle x \rangle = \langle y \rangle$ " respectivement. Cette propriété est définie comme suit :

NOM : qpAttributeName
 SYNTAXE : Chaîne

8.12.2 Propriété qpAttributeValueList

Cette propriété porte une liste de valeurs qui est à utiliser pour substituer $\langle y \rangle$ dans une simple condition ou simple action de la forme " $\langle x \rangle$ satisfait $\langle y \rangle$ " ou " $\langle x \rangle = \langle y \rangle$ " respectivement.

Cette propriété est définie comme suit :

NOM : qpAttributeValueList
 SYNTAXE : Chaîne

8.13 Classe "QoSPolicyRSVPVariable"

C'est une classe abstraite qui sert de classe de base pour toutes les variables implicites qui ont à voir avec le conditionnement RSVP. La définition de la classe est la suivante :

NOM : QoSPolicyRSVPVariable
 DESCRIPTION : Classe de base abstraite utilisée pour construire d'autres classes qui spécifient différents attributs d'une demande RSVP.
 DÉDUIT DE : PolicyImplicitVariable (définie dans la [RFC3460])
 ABSTRAITE : VRAI
 PROPRIÉTÉS : Aucune

8.14 Classe "QoSPolicyRSVPSourceIPv4Variable"

C'est une classe concrète qui contient l'adresse de source IPv4 du flux RSVP signalé, comme défini dans les objets RSVP PATH SENDER_TEMPLATE et RESV FILTER_SPEC [RFC2205]. La définition de la classe est la suivante :

NOM : QoSPolicyRSVPSourceIPv4Variable
 DESCRIPTION : Adresse de source IPv4 du flux signalé RSVP, comme défini dans les objets RSVP PATH SENDER_TEMPLATE et RSVP RESV FILTER_SPEC [RFC2205].
 TYPES DE VALEUR ADMIS : PolicyIPv4AddrValue
 DÉDUIT DE : QoSPolicyRSVPVariable (défini dans le présent document)
 ABSTRAITE : FAUX
 PROPRIÉTÉS : Aucune

8.15 Classe "QoSPolicyRSVPDestinationIPv4Variable"

C'est une classe concrète qui contient l'adresse de destination IPv4 du flux signalé RSVP, comme défini dans les objets RSVP PATH SENDER_TEMPLATE et RSVP RESV FILTER_SPEC [RFC2205]. La définition de la classe est la suivante :

NOM : QoSPolicyRSVPDestinationIPv4Variable
 DESCRIPTION : Adresse de destination IPv4 du flux signalé RSVP, comme défini dans les objets RSVP PATH et RESV SESSION [RFC2205].
 TYPES DE VALEUR ADMIS : PolicyIPv4AddrValue
 DÉDUIT DE : QoSPolicyRSVPVariable (défini dans le présent document)
 ABSTRAITE : FAUX
 PROPRIÉTÉS : Aucune

8.16 Classe "QoSPolicyRSVPSourceIPv6Variable"

C'est une classe concrète qui contient l'adresse de source IPv6 du flux signalé RSVP, comme défini dans les objets RSVP PATH SENDER_TEMPLATE et RSVP RESV FILTER_SPEC [RFC2205]. La définition de la classe est la suivante :

NOM : QoSPolicyRSVPSourceIPv6Variable
 DESCRIPTION : Adresse de source IPv6 du flux signalé RSVP, comme défini dans les objets RSVP PATH SENDER_TEMPLATE et RSVP RESV FILTER_SPEC [RFC2205].
 TYPES DE VALEUR ADMIS : PolicyIPv6AddrValue
 DÉDUIT DE : QoSPolicyRSVPVariable (défini dans le présent document)
 ABSTRAITE : FAUX
 PROPRIÉTÉS : Aucune

8.17 Classe "QoSPolicyRSVPDestinationIPv6Variable"

C'est une classe concrète qui contient l'adresse de destination IPv6 du flux signalé RSVP, comme défini dans les objets RSVP PATH SENDER_TEMPLATE et RSVP RESV FILTER_SPEC [RFC2205]. La définition de la classe est la suivante :

NOM : QoSPolicyRSVPDestinationIPv6Variable

DESCRIPTION : Adresse de destination IPv6 du flux signalé RSVP, comme défini dans les objets RSVP PATH et RESV SESSION [RFC2205].

TYPES DE VALEUR ADMIS : PolicyIPv6AddrValue

DÉDUIT DE : QoSPolicyRSVPVariable (défini dans le présent document)

ABSTRAITE : FAUX

PROPRIÉTÉS : Aucune

8.18 Classe "QoSPolicyRSVPSourcePortVariable"

Cette classe contient l'accès de source du flux signalé RSVP, comme défini dans les objets RSVP PATH SENDER_TEMPLATE et RSVP RESV FILTER_SPEC [RFC2205]. La définition de la classe est la suivante :

NOM : QoSPolicyRSVPSourcePortVariable

DESCRIPTION : Accès de source du flux signalé RSVP, comme défini dans les objets RSVP PATH SENDER_TEMPLATE et RSVP RESV FILTER_SPEC [RFC2205].

TYPES DE VALEUR ADMIS : PolicyIntegerValue (0..65535)

DÉDUIT DE : QoSPolicyRSVPVariable (défini dans le présent document)

ABSTRAITE : FAUX

PROPRIÉTÉS : Aucune

8.19 Classe "QoSPolicyRSVPDestinationPortVariable"

C'est une classe concrète qui contient l'accès de destination du flux signalé RSVP, comme défini dans les objets RSVP PATH SENDER_TEMPLATE et RSVP RESV FILTER_SPEC [RFC2205]. La définition de la classe est la suivante :

NOM : QoSPolicyRSVPDestinationPortVariable

DESCRIPTION : Accès de destination du flux signalé RSVP, comme défini dans les objets RSVP PATH et RESV SESSION [RFC2205].

TYPES DE VALEUR ADMIS : PolicyIntegerValue (0..65535)

DÉDUIT DE : QoSPolicyRSVPVariable (défini dans le présent document)

ABSTRAITE : FAUX

PROPRIÉTÉS : Aucune

8.20 Classe "QoSPolicyRSVPIPProtocolVariable"

C'est une classe concrète qui contient le numéro de protocole IP du flux signalé RSVP, comme défini dans les objets RSVP PATH et RESV SESSION [RFC2205]. La définition de la classe est la suivante :

NOM : QoSPolicyRSVPIPProtocolVariable

DESCRIPTION : Numéro de protocole IP du flux signalé RSVP, comme défini dans les objets RSVP PATH et RESV SESSION [RFC2205].

TYPES DE VALEUR ADMIS : PolicyIntegerValue

DÉDUIT DE : QoSPolicyRSVPVariable (défini dans le présent document)

ABSTRAITE : FAUX

PROPRIÉTÉS : Aucune

8.21 Classe "QoSPolicyRSVPIPVersionVariable"

C'est une classe concrète qui contient le numéro de version du protocole IP du flux signalé RSVP, comme défini dans les objets RSVP PATH et RESV SESSION [RFC2205]. Les numéros de version bien connus sont 4 et 6. Cette variable permet une définition de politique du type : "Si version IP = IPv4 alors ...". La définition de la classe est la suivante :

NOM : QoSPolicyRSVPIPVersionVariable

DESCRIPTION : Numéro de version IP des adresses IP portées dans le flux signalé RSVP, comme défini dans les objets RSVP PATH et RESV SESSION [RFC2205].

TYPES DE VALEUR ADMIS : PolciIntegerValue

DÉDUIT DE : QoSPolicyRSVPVariable (défini dans le présent document)

ABSTRAITE : FAUX

PROPRIÉTÉS : Aucune

8.22 Classe "QoSPolicyRSVPDCLASSVariable"

C'est une classe concrète qui contient la valeur du DSCP comme défini dans l'objet RSVP DCLASS [RFC2996]. La définition de la classe est la suivante :

NOM : QoSPolicyRSVPDCLASSVariable

DESCRIPTION : Valeur du DSCP comme définie dans l'objet RSVP DCLASS [RFC2996].

TYPES DE VALEUR ADMIS : PolicyIntegerValue, PolicyBitStringValue

DÉDUIT DE : QoSPolicyRSVPVariable (défini dans le présent document)

ABSTRAITE : FAUX

PROPRIÉTÉS : Aucune

8.23 Classe "QoSPolicyRSVPStyleVariable"

C'est une classe concrète qui contient le style de réservation comme défini dans l'objet RSVP STYLE dans le message RESV [RFC2205]. La définition de la classe est la suivante :

NOM : QoSPolicyRSVPStyleVariable

DESCRIPTION : Style de réservation comme défini dans l'objet RSVP STYLE dans le message RESV [RFC2205].

TYPES DE VALEUR ADMIS : PolicyBitStringValue, PolicyIntegerValue (entier parmi { Fixed-Filter=1, Shared-Explicit=2, Wildcard-Filter=3})

DÉDUIT DE : QoSPolicyRSVPVariable (défini dans le présent document)

ABSTRAITE : FAUX

PROPRIÉTÉS : Aucune

8.24 Classe "QoSPolicyIntServVariable"

C'est une classe concrète qui contient le service intégré demandé dans le message RSVP Reservation, comme défini dans l'objet FLOWSPEC RSVP [RFC2205]. La définition de la classe est la suivante :

NOM : QoSPolicyRSVPIntServVariable

DESCRIPTION : Service intégré demandé dans le message RSVP Reservation, comme défini dans l'objet FLOWSPEC RSVP [RFC2205].

TYPES DE VALEUR ADMIS : PolicyIntegerValue (une des valeurs { CL=1, GS=2, NULL=3})

DÉDUIT DE : QoSPolicyRSVPVariable (défini dans le présent document)

ABSTRAITE : FAUX

PROPRIÉTÉS : Aucune

8.25 Classe "QoSPolicyRSVPMessageTypeVariable"

C'est une classe concrète qui contient le type de message RSVP, comme défini dans l'objet d'en-tête commun de message RSVP [RFC2205]. La définition de la classe est la suivante :

NOM : QoSPolicyRSVPMessageTypeVariable

DESCRIPTION : Type de message RSVP, comme défini dans l'objet d'en-tête commun de message RSVP [RFC2205].

TYPES DE VALEUR ADMIS : Entier (une des valeurs de { PATH=1, PATHTEAR=2, RESV=3, RESVTEAR=4, RESVERR=5, CONF=6, PATHERR=7})

DÉDUIT DE : QoSPolicyRSVPVariable (défini dans le présent document)

ABSTRAITE : FAUX
PROPRIÉTÉS : Aucune

8.26 Classe "QoSPolicyRSVPPreemptionPriorityVariable"

C'est une classe concrète qui contient la priorité de réservation RSVP, comme défini dans l'objet [RFC3181]. La définition de la classe est la suivante :

NOM : QoSPolicyRSVPPreemptionPriorityVariable
DESCRIPTION : Priorité de réservation RSVP comme défini dans la [RFC3181].
TYPES DE VALEUR ADMIS : PolicyIntegerValue
DÉDUIT DE : QoSPolicyRSVPVariable (défini dans le présent document)
ABSTRAITE : FAUX
PROPRIÉTÉS : Aucune

8.27 Classe "QoSPolicyRSVPPreemptionDefPriorityVariable"

C'est une classe concrète qui contient l'objet priorité de défense de réservation RSVP, comme défini dans la [RFC3181]. La définition de la classe est la suivante :

NOM : QoSPolicyRSVPPreemptionDefPriorityVariable
DESCRIPTION : Priorité de défense de réservation de préemption RSVP comme défini dans la [RFC3181].
TYPES DE VALEUR ADMIS : PolicyIntegerValue
DÉDUIT DE : QoSPolicyRSVPVariable (défini dans le présent document)
ABSTRAITE : FAUX
PROPRIÉTÉS : Aucune

8.28 Classe "QoSPolicyRSVPUserVariable"

C'est une classe concrète qui contient l'identifiant de l'utilisateur qui a initié le flux comme défini dans la chaîne Localisateur d'utilisateur dans l'objet de politique Identité [RFC3182]. La définition de la classe est la suivante :

NOM : QoSPolicyRSVPUserVariable
DESCRIPTION : Identifiant de l'utilisateur qui a initié le flux comme défini dans la chaîne Localisateur usager dans l'objet de politique Identité [RFC3182].
TYPES DE VALEUR ADMIS : QoSPolicyDNValue, PolicyStringValue, QoSPolicyAttributeValue
DÉDUIT DE : QoSPolicyRSVPVariable (défini dans le présent document)
ABSTRAITE : FAUX
PROPRIÉTÉS : Aucune

8.29 Classe "QoSPolicyRSVPApplicationVariable"

C'est une classe concrète qui contient l'identifiant de l'application qui a généré le flux comme défini dans la chaîne Localisateur d'application dans l'objet de politique Application [RFC2872]. La définition de la classe est la suivante :

NOM : QoSPolicyRSVPApplicationVariable
DESCRIPTION : Identifiant de l'application qui a généré le flux comme défini dans la chaîne Localisateur d'application dans l'objet de politique Application [RFC2872].
TYPES DE VALEUR ADMIS : QoSPolicyDNValue, PolicyStringValue, QoSPolicyAttributeValue
DÉDUIT DE : QoSPolicyRSVPVariable (défini dans le présent document)
ABSTRAITE : FAUX
PROPRIÉTÉS : Aucune

8.30 Classe "QoSPolicyRSVPAuthMethodVariable"

C'est une classe concrète qui contient le type d'authentification utilisé dans l'objet de politique Identité [RFC3182]. La

définition de la classe est la suivante :

NOM : QoSPolicyRSVPAuthMethodVariable

DESCRIPTION : Type d'authentification RSVP utilisé dans l'objet de politique Identité [RFC3182].

TYPES DE VALEUR ADMIS : PolicyIntegerValue (énumération de { NONE=0, PLAIN-TEXT=1, DIGITAL-SIG = 2, KERBEROS_TKT=3, X509_V3_CERT=4, PGP_CERT=5})

DÉDUIT DE : QoSPolicyRSVPVariable (défini dans le présent document)

ABSTRAITE : FAUX

PROPRIÉTÉS : Aucune

8.31 Classe QoSPolicyDNValue

Cette classe est utilisée pour représenter une seule valeur ou un ensemble de valeurs de noms distinctifs [RFC2253], incluant des caractères génériques. Un nom distinctif est un nom qui peut être utilisé comme clé pour restituer un objet à partir d'un service de répertoire. Cette valeur peut être utilisée pour comparaison à des valeurs de référence portées dans des objets de politique RSVP, comme spécifié dans la [RFC3182]. La définition de la classe est la suivante :

NOM : QoSPolicyDNValue

DÉDUIT DE : PolicyValue

ABSTRAITE : FAUX

PROPRIÉTÉS : qpDNList

8.31.1 Propriété qpDNList

Cet attribut donne une liste non ordonnée de chaînes dont chacune représente un nom distinctif avec des caractères génériques. Le format d'un nom distinctif est défini dans la [RFC2253]. Le caractère astérisque ("*") est utilisé comme caractère générique pour une seule valeur d'attribut ou pour un RDN. L'ordre des RDN est significatif. Par exemple, un attribut qpDNList qui porte la valeur suivante :

"CN=*, OU=Ventes, O=Widget Inc., *, C=US" correspond à :

"CN=J. Smith, OU=Ventes, O=Widget Inc, C=US"

et correspond aussi à :

"CN=J. Smith, OU=Ventes, O=Widget Inc, L=CA, C=US".

L'attribut est défini comme suit :

NOM : qpDNList

SYNTAXE : Liste des noms distinctifs mis en œuvre comme chaînes, dont chacune sert de référence à un autre objet.

8.32 Classe QoSPolicyRSVPSimpleAction

Cette action contrôle le contenu des messages RSVP et la façon dont les demandes RSVP sont admises. Selon la valeur de sa propriété qpRSVPActionType, cette action se traduit directement en une décision COPS Replace ou en une décision COPS Stateless, ou les deux comme défini dans COPS pour RSVP. Seules les variables qui sont des sous classes de QoSPolicyRSVPVariable peuvent être associées à cette action. La définition de la propriété est la suivante :

NOM : QoSPolicyRSVPSimpleAction

DESCRIPTION : Cette action contrôle le contenu des messages RSVP et la façon dont les demandes RSVP sont admises.

DÉDUIT DE : SimplePolicyAction (définie dans la [RFC3460])

ABSTRAITE : FAUX

PROPRIÉTÉS : qpRSVPActionType

8.32.1 Propriété qpRSVPActionType

Cette propriété est une énumération d'entiers notant le ou les types d'action RSVP. La valeur 'REPLACE' note une action de décision COPS Replace. La valeur 'STATELESS' note une action de décision COPS Stateless. La valeur REPLACEANDSTATELESS note les deux actions de décision. Voir les détails dans la [RFC2749].

NOM : qpRSVPActionType

DESCRIPTION : Cette propriété spécifie si le type d'action est pour le type de décision COPS Replace, Stateless, ou les deux.

SYNTAXE : Entier

VALEUR : C'est une énumération d'entiers. Une valeur de 0 spécifie une décision COPS Replace. Une valeur de 1 spécifie une décision COPS Stateless. Une valeur de 2 spécifie les deux décisions COPS Replace et Stateless.

9. Déclaration de droits de propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourrait être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur répertoire en ligne des IPR de l'IETF à <http://www.ietf.org/ipr>.

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à ietf-ipr@ietf.org.

10. Remerciements

Les auteurs souhaitent remercier de leurs apports les participants au groupe de travail Cadre de politique, et en particulier le groupe combiné des co-auteurs de PCIMe, Lee Rafalow, Andrea Westerinen, Ritu Chadha et Marcus Brunner. De plus, on remerciera de leurs précieuses contribution Ed Ellesson, Joel Halpern et Mircea Pana. Merci pour tous les commentaires, critiques, idées et contributions générales.

11. Considérations sur la sécurité

Le modèle d'informations de cœur de politique [RFC3060] décrit les considérations générales de sécurité qui se rapportent au modèle général de cœur de politique. Les extensions définies dans le présent document n'introduisent aucune considération supplémentaire en rapport avec la sécurité.

12. Références

12.1 Références normatives

[RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.

[RFC3060] B. Moore et autres, "Spécification du [modèle d'information de cœur de politique](#) -- version 1", février 2001. (*MàJ par RFC3460*) (P.S.)

[RFC3460] B. Moore, éd., "[Extensions au modèle d'information](#) de cœur de politique (PCIM)", janvier 2003. (P.S.)

12.2 Références Informatives

[RFC1633] R. Braden, D. Clark et S. Shenker, "[Intégration de services](#) dans l'architecture de l'Internet : généralités", juin 1994. (*Info.*)

[RFC2205] R. Braden, éd., L. Zhang, S. Berson, S. Herzog, S. Jamin, "[Protocole de réservation de ressource](#) (RSVP) – version 1, spécification fonctionnelle", septembre 1997. (*MàJ par RFC2750, RFC3936, RFC4495, RFC6780*)

[RFC2210] J. Wroclawski, "Utilisation de [RSVP avec les services intégrés](#) de l'IETF", septembre 1997. (P.S.)

- [RFC2211] J. Wroclawski, "Spécification du service d'[élément de réseau à charge contrôlée](#)", septembre 1997. (P.S.)
- [RFC2212] S. Shenker, C. Partridge, R. Guerin, "Spécification de la [qualité de service garantie](#)", septembre 1997. (P.S.)
- [RFC2253] M. Wahl, S. Kille et T. Howes, "[Protocole léger d'accès à un répertoire](#) (LDAPv3) : Représentation de chaîne UTF-8 des noms distinctifs", décembre 1997.
- [RFC2475] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang et W. Weiss, "[Architecture pour services différenciés](#)", décembre 1998. (MàJ par [RFC3260](#))
- [RFC2597] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, "[Groupe PHB Transmission assurée](#)", juin 1999. (MàJ par [RFC3260](#)) (P.S.)
- [RFC2749] S. Herzog, et autres, "[Utilisation de COPS avec RSVP](#)", janvier 2000. (P.S.)
- [RFC2872] Y. Bernet, R. Pabbati, "[Élément de politique d'identité d'application](#) et de sous-application à utiliser avec RSVP", juin 2000. (P.S.)
- [RFC2996] Y. Bernet, "[Format de l'objet DCLASS RSVP](#)", novembre 2000. (P.S.)
- [RFC3181] S. Herzog, "[Élément de politique de priorité](#) par préemption signalée", octobre 2001. (Remplace [RFC2751](#)) (P.S.)
- [RFC3182] S. Yadav et autres, "[Représentation d'identité](#) pour RSVP", octobre 2001. (P.S.)
- [RFC3198] A. Westerinen et autres, "[Terminologie pour la gestion fondée sur la politique](#)", novembre 2001. (Information)
- [RFC3289] F. Baker, K. Chan, A. Smith, "Base de données d'informations de gestion pour l'architecture de services différenciés", mai 2002. (P.S.)

13. Adresse des auteurs

Yoram Ramberg
Cisco Systems
4 Maskit Street
Herzliya Pituach, Israel 46766
téléphone : +972-9-970-0081
mél : yramberg@cisco.com

Yoram Snir
Cisco Systems
300 East Tasman Drive
San Jose, CA 95134
téléphone : +1 408-853-4053
mél : ysnir@cisco.com

John Strassner
Intelliden Corporation
90 South Cascade Avenue
Colorado Springs, Colorado 80903
mél : john.strassner@intelliden.com

Ron Cohen
Ntear LLC
téléphone : +972-8-9402586
Fax : +972-9-9717798
mél : ronc@lyciumnetworks.com

Bob Moore
IBM Corporation
P. O. Box 12195, BRQA/501/G206
3039 Cornwallis Rd.
Research Triangle Park, NC 27709-2195
téléphone : +1 919-254-4436
mél : remoore@us.ibm.com

14. Déclaration complète de droits de reproduction

Copyright (C) The Internet Society (2003). Tous droits réservés.

Le présent document et ses traductions peuvent être copiés et fournis aux tiers, et les travaux dérivés qui les commentent ou les expliquent ou aident à leur mise en œuvre peuvent être préparés, copiés, publiés et distribués, en tout ou partie, sans restriction d'aucune sorte, pourvu que la déclaration de droits de reproduction ci-dessus et le présent paragraphe soient inclus dans toutes telles copies et travaux dérivés. Cependant, le présent document lui-même ne peut être modifié d'aucune façon, en particulier en retirant la notice de droits de reproduction ou les références à la Internet Society ou aux autres organisations Internet, excepté autant qu'il est nécessaire pour le besoin du développement des normes Internet, auquel cas les procédures de droits de reproduction définies dans les procédures des normes Internet doivent être suivies, ou pour les besoins de la traduction dans d'autres langues que l'anglais.

Les permissions limitées accordées ci-dessus sont perpétuelles et ne seront pas révoquées par la Internet Society ou ses

successeurs ou ayant droits.

Le présent document et les informations contenues sont fournis sur une base "EN L'ÉTAT" et le contributeur, l'organisation qu'il ou elle représente ou qui le/la finance (s'il en est), la INTERNET SOCIETY et la INTERNET ENGINEERING TASK FORCE déclinent toutes garanties, exprimées ou implicites, y compris mais non limitées à toute garantie que l'utilisation des informations encloses ne violent aucun droit ou aucune garantie implicite de commercialisation ou d'aptitude à un objet particulier.

Remerciement

Le financement de la fonction d'édition des RFC est actuellement fourni par l'Internet Society.