ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN GÉNIE
CONCENTRATION TECHNOLOGIES DE L'INFORMATION
M.Ing.

PAR
Rémi MENEGON

MÉCANISME D'INCITATION DISTRIBUÉ POUR LA GESTION DE RESSOURCES DE RÉSEAUX PRIVÉS VIRTUELS

MONTRÉAL, LE 21 AVRIL 2011

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

M. Jean-Marc Robert, directeur de mémoire
Département de génie logiciel et des TI à l'École de technologie supérieure

M. Michel Kadoch, président du jury
Département de génie électrique à l'École de technologie supérieure

M. Chamseddine Talhi, membre du jury
Département de génie logiciel et des TI à l'École de technologie supérieure

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

LE 14 AVRIL 2011

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

Cette année de recherche au sein de l'École de technologie supérieure a été passionnante. Accomplir une recherche sur un sujet complexe tout en partant de zéro n'est pas chose aisée. Heureusement, je n'ai pas été seul pour accomplir ce mémoire, et je tiens à remercier ces personnes.

Je souhaite remercier en premier lieu mon directeur de recherche, Jean-Marc Robert, de m'avoir accompagné tout au long de ce projet de recherche. Il a su me guider dans mes recherches, me rendre enthousiaste dès les premiers maigres résultats, me remotiver quand rien ne fonctionnait, me pousser quand le soleil et l'été frappaient à la porte du laboratoire et pour ces longues conversations au sujet du bon vin français.

Je souhaite également remercier ma famille, mes amis ainsi que ma tendre Cynthia Chalifour, de m'avoir soutenu au cours ce projet et me rappeler que je pouvais réussir.

Enfin, pour leur support indirect, mais au combien apprécié, je voudrais également remercier Linus Torvalds pour la création et la maintenance du noyau Linux, Mark Shuttleworth d'en avoir fait la fabuleuse distribution GNU/Linux Ubuntu, Jimmy Wales d'avoir créé la plus grande base de connaissance libre et gratuite qu'est Wikipédia et enfin Donald Knuth et Leslie Lamport du MIT d'avoir conçu réciproquement TEX et son évolution LATEX, le langage et le système de composition de documents utilisé pour la rédaction de ce mémoire.

Cette recherche a été partiellement financée par le Conseil de recherches en sciences mutuelles et en génie (CRSNG) du Canada et Bell Canada.

MÉCANISME D'INCITATION DISTRIBUÉ POUR LA GESTION DE RESSOURCES DE RÉSEAUX PRIVÉS VIRTUELS

Rémi MENEGON

RÉSUMÉ

Un fournisseur d'accès à Internet (ISP) peut offrir à ses clients des réseaux virtuels privés (VPN). Un contrat est défini entre l'ISP et l'administrateur du VPN. Ce contrat détermine, entre autres, le niveau de qualité de service (QoS), qui peut-être vu comme une garantie d'une certaine quantité de bande passante.

Nous proposons une architecture autonome et distribuée qui permet aux opérateurs de VPN de partager leur bande passante non utilisée avec d'autres administrateurs de VPN, et cela sans intervention de l'ISP. Pour amener ces opérateurs à collaborer, nous utilisons un mécanisme d'incitation basé sur le mécanisme de Vickrey-Clarke-Groves (VCG). Les opérateurs égoïstes mais rationnels n'auront pas d'autre choix que de réellement collaborer s'ils veulent obtenir le plus souvent possible une bonne qualité de service. Pour les clients de l'ISP, cette approche offre une meilleure qualité de service, de plus bas prix et leur permet de parer à une éventuelle surcharge sans aucune conséquence, même financière.

Mots-clés : Gestion autonome de ressources, Qualité de Service, Partage de bande passante, Mécanisme d'incitation réalisable

MÉCANISME D'INCITATION DISTRIBUÉ POUR LA GESTION DE RESSOURCES DE RÉSEAUX PRIVÉS VIRTUELS

Rémi MENEGON

ABSTRACT

An Internet Service Provider (ISP) can provide Virtual Private Networks (VPN) to its customers. There is an agreement between the ISP and a VPN operator. This Service Level Agreements (SLA) determines the Quality of Service (QoS) level, which can be seen as a guarantee of a certain amount of bandwidth.

We propose an autonomic distributed architecture to allow VPN operators to share extra bandwidth with each other without the intervention of the ISP. To force these operators to collaborate, we use a Vickrey-Clarke-Groves (VCG) mechanism design. Rational selfish operators would not have any other choice than to collaborate truthfully if they want to obtain their own QoS as often as possible. For users, this approach offers a better quality of service, lower price and allow them to undergo an overload without any drawback, even financially.

Keywords: Autonomic resource management, Service level agreement, Quality of service, Bandwidth sharing, Feasible Mechanism Design

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

AMD	Algorithmic Mechanism Design
ASA	Autonomic Service Architecture
DAMD	Distributed Algorithmic Mechanism Design
IP	Internet Protocol
ISP	Internet Service Provider
PL	Programmation Linéaire
PLNE	Programmation Linéaire en Nombres Entiers
QoS	Quality of Service
SIP	Session Initiation Protocol
SLA	Service Level Agreements
VCG	Vickrey-Clarke-Groves
VoIP	Voice over IP
VPN	Virtual Private Networks

LISTE DES SYMBOLES ET UNITÉS DE MESURE

$b_i(\bullet)$	Offre de prix pour l'achat de bande passante par le participant i
$g(\bullet)$	Fonction d'évaluation d'un algorithme d'allocation
$H(\bullet,\bullet)$	Fonction de hachage
i	Participant
$k(\bullet)$	Algorithme d'allocation
k_i	Proposition d'allocation du participant i
$m(\bullet)$	Mécanisme d'incitation
n	Nombre de participants
$\mathscr{N}$	Ensemble des participants
$nonce_i$	Entier généré aléatoirement
$\mathscr{O}$	Ensemble de solutions possibles
$p(\bullet)$	Fonction de paiement
p_i	Prix que doit acquitter le participant i
Q	Bande passante totale disponible chez l'ISP
q_i	Bande passante disponible chez le participant i
$\mathscr{R}$	Ensemble des participants demandant plus de ressources
R	Somme de la bande passante demandée.
r_i	Demande de bande passante du participant i
$\mathscr{S}$	Ensemble des participants proposant plus de ressources
S	Somme de la bande passante offerte.
s_i	Offre de bande passante du participant i
$\mathscr{T}$	Ensemble des participants neutres
T	Durée de la simulation

τ_O	Temps des offres
τ_C	Temps de calcul
$v_i(\bullet)$	Fonction d'utilité de i
$\mathscr{W}$	Ensemble des déclarations d'utilité des participants de $\mathscr{R}$
$w_i(\bullet)$	Déclaration de l'utilité du participant i

INTRODUCTION

Le fonctionnement des entreprises à travers le monde a nettement évolué. Aujourd'hui, les entités d'une même entreprise sont disséminées à travers la planète. Les employés sont eux-mêmes appelés à parcourir le monde. Seul l'avènement d'Internet a permis à tous ces éléments de communiquer entre eux. Malheureusement, la sécurité des données, dont la confidentialité, n'est pas assurée sur Internet. Pour répondre à ce problème, la solution la plus utilisée est l'utilisation d'un réseau privé virtuel (VPN).

Un VPN est un protocole réseau qui permet d'encapsuler toutes sortes de protocoles. En chiffrant à la source toutes données transmises et en les déchiffrant à la destination, le protocole VPN (principalement L2TP) permet de connecter différents réseaux de manière transparente. Dans certains cas, le protocole utilisé est basé sur des mécanismes cryptographiques permettant d'assurer la confidentialité et l'intégrité des données transmises. Ce service peut être directement offert par des fournisseurs d'accès à internet (ISP – *Internet Service Provider*) aux entreprises.

Le principal rôle des ISP dans la gestion des VPN est de gérer l'infrastructure du réseau ainsi que les ressources allouées à chaque VPN. Lors de la mise en place du VPN d'un client, celui-ci et l'ISP signent un contrat (SLA – *Service Level Agreements*). Ce contrat définit entre autres la qualité de service (QoS) requise par le client. On peut voir la QoS comme la garantie qu'une certaine quantité de bande passante soit toujours accessible pour le VPN.

Afin d'optimiser au maximum la bande passante totale que possède l'ISP, celui-ci en vend plus qu'il n'en possède, espérant que tous ses clients n'aient pas besoin en même temps du maximum de la bande passante achetée. Ceci implique principalement deux problèmes. D'une part, l'ISP doit en tout temps gérer les conflits potentiels, et d'autre part les clients peuvent être privés de ressources auxquelles ils ont droit selon leur contrat préalablement négocié. D'autre part, il n'y a aucune flexibilité pour le client afin d'augmenter à court terme sa bande passante allouée.

Afin de résoudre les problèmes liés à l'allocation de ressources en général, la littérature propose le modèle d'Architecture de Service Autonome (ASA)[4]. Cette architecture permet d'automatiser l'attribution et l'utilisation des ressources entre des VPN. Son principe repose sur le fait de permettre aux divers VPN surchargés d'emprunter des ressources inutilisées à d'autres VPN, afin de garantir leur SLA.

Malheureusement, le modèle ASA souffre de deux conséquences majeures, inhérentes à la prise de décision centralisée. Premièrement, l'ISP doit continuellement surveiller et identifier à l'aide d'un système d'évaluation des performances (logimètre) les ressources inutilisées afin de répondre aux nouvelles demandes des VPN surchargés. Ceci demande beaucoup de ressources systèmes. De plus, ce système d'échange peut occasionner une perte de QoS pour les VPN qui partagent leurs ressources. Deuxièmement, il n'y a absolument aucune incitation pour les VPN de partager leurs ressources inutilisées, d'autant plus que cela peut induire une perte momentanée de QoS pour eux-mêmes. Ceci est un problème sérieux, car même si l'ISP peut forcer un VPN à partager ses ressources, un VPN égoïste pourra toujours trouver une façon de simuler une utilisation maximale de ses ressources. Toutes solutions centralisées telles que l'ASA souffriront de ces défauts majeurs.

Pour éliminer ces deux problèmes, nous proposons ici une architecture distribuée et autonome. Cette architecture permet à chaque opérateur de VPN de communiquer avec les autres opérateurs afin de partager leurs ressources supplémentaires à leur souhait. Chaque opérateur devrait être en effet capable de « prédire » à court terme ses propres utilisations futures de bande passante et ceci bien mieux que n'importe quel autre intervenant centralisé. On peut citer l'exemple de l'établissement d'une vidéoconférence sur IP (VoIP), qui commence toujours par une période d'initialisation des divers paramètres multimédia via le protocole SIP. La surveillance de ce protocole par l'opérateur rend alors facile la prédiction de la bande passante nécessaire à cette vidéo-conférence, et cela avant même le début de celle-ci. Cette architecture est donc plus flexible et plus fiable. Nous démontrerons que notre architecture distribuée est meilleure qu'une architecture centralisée. Notre solution permet (1) une meilleure qualité de service pour (2) de meilleurs coûts d'utilisation pour les opérateurs VPN et (3) leur permet de

pallier une surcharge avec très peu d'inconvénients, même financier. De plus, elle permet une meilleure utilisation de la bande passante globale de l'ISP. Cette solution apporte des gains que ce soit à court ou à long terme.

Une architecture autonome et distribuée amène toutefois un nouveau lot de conséquences fâcheuses. Chaque opérateur VPN est maintenant indépendant, et de ce fait, peut agir comme bon lui semble. Il peut ainsi tirer uniquement avantage du système sans aucune considération pour les besoins des autres participants. Ainsi, il pourra, en bon égoïste, emprunter des ressources pour répondre à ses surcharges momentanées sans jamais prêter ses ressources inutilisées.

Afin de gérer ce problème, nous proposons ici d'utiliser un mécanisme d'incitation basé sur le système d'enchère de Vickrey-Clarke-Groves (VCG)[12], dans le but d'amener les opérateurs VPN égoïstes à se comporter comme des opérateurs coopératifs. Malheureusement, la résolution d'enchères combinatoires est NP-ardu[16]. Il est donc peu probable qu'une solution efficace puisse être trouvée. Il faut donc trouver un mécanisme introduisant une approximation tout en gardant sa propriété d'incitation à l'honnêteté, c'est-à-dire dont les participants ont tout intérêt à annoncer la vérité. Un tel mécanisme d'incitation est suggéré par Nisan et Ronen[22], qui propose une solution nommée *Second Chance Mechanism* (Mécanisme de Seconde chance) afin de résoudre le problème de calcul de VCG.

Le premier chapitre présentera le fonctionnement de plusieurs mécanismes d'enchère. Nous verrons les principaux modèles, ainsi que leurs caractéristiques. Une introduction à la théorie des jeux y sera également présentée et nous verrons comment sa principale branche, la théorie de la conception des mécanismes d'incitation (ou théorie de la conception des mécanismes de marché) est liée aux mécanismes d'enchère. Plus particulièrement, nous définirons le mécanisme d'incitation de Vickrey-Clarke-Groves (VCG). De tout cela nous dégagerons les enjeux et présenterons la problématique de notre mémoire.

Le deuxième chapitre traitera de différents modèles de gestion actuels, dont le modèle d'architecture ASA. Nous montrerons précisément les dernières avancées et les inconvénients de

ces modèles, liés principalement au fait qu'ils ne soient pas distribués. Nous verrons aussi comment Nisan et Ronen[22] ont réussi à rendre la résolution informatique de VCG possible, grâce à leur mécanisme de seconde chance.

Le troisième chapitre établira le modèle formel de notre système autonome et distribué. Ceci posera les bases mathématiques concrètes de notre problématique.

L'implémentation de la résolution numérique du modèle formel sera vue dans le quatrième chapitre.

Le dernier chapitre traitera des résultats ainsi que de leur analyse et leur interprétation.

Enfin, nous conclurons en rappelant nos diverses contributions à la gestion de ressources autonome et distribuée et nous en ferons une synthèse.

CHAPITRE 1

MÉCANISMES D'ENCHÈRE

1.1 Enchères

Les enchères à proprement parlé sont nées vers -500 avant J.C pour la vente d'esclaves. Deux systèmes étaient alors utilisés : la majorité des enchères étaient descendantes sauf pour la vente de belles femmes, qui étaient alors ascendantes[10](principale source de cette section). Aujourd'hui, l'utilisation d'enchères est omniprésente. Que ce soit au niveau d'un pays pour la vente du spectre hertzien[14] (CDMA, UMTS), la vente de l'électricité en Californie ou bien au niveau du simple internaute qui achète un vieil ordinateur portable sur eBay, tous utilisent un système d'enchères. En Europe, les appels d'offres des Marchés Publics représentent 8% du PIB.

Dans une enchère, chaque participant donne un prix qui serait prêt à payer pour l'acquisition d'un ou plusieurs objets. Un système évalue ensuite qui sont le ou les gagnants et indique le prix qui va être payé. Ce système varie suivant le type d'enchère[29].

1.1.1 Enchère anglaise

L'enchère anglaise, appelée aussi enchère ascendante, est la plus connue et utilisée. Les participants se réunissent dans un lieu où un commissaire-priseur organise la vente. Ce dernier annonce un premier prix et chaque membre de l'assemblée peut proposer un prix plus élevé. Si, après un certain moment le prix ne s'élève plus, alors l'enchère se termine et celui qui a donné le dernier prix, donc le plus élevé, gagne et paye ce prix. Dans le cas particulier où le commissaire-priseur achète au lieu de vendre, l'enchère anglaise est alors descendante, c'est-à-dire que le gagnant est celui qui a déclaré le prix le plus bas (et le commissaire-priseur paye ce prix).

Cette enchère est dite ouverte, car les offres des autres acheteurs sont connues de tous.

1.1.2 Enchère hollandaise

L'enchère hollandaise, ou enchère descendante, fut longuement utilisée pour la vente de fleurs aux Pays-Bas. Elle est basée sur le concept inverse de l'enchère anglaise. Ici, le prix est fixé très haut et diminue progressivement jusqu'à ce qu'un acheteur se signale.

Cette enchère est généralement bonne pour le vendeur, car elle oblige les participants à se déclarer tôt s'ils veulent acquérir l'objet.

Pour comparaison, si Alice et Bob désirent acquérir le même lot de fleurs, respectivement pour 5$ et 15$, alors le prix de la vente serait de 15$ en utilisant l'enchère hollandaise, alors qu'il ne serait que de 6$ avec l'enchère anglaise. En effet, dans l'enchère hollandaise, dès que le commissaire-priseur annonce 15$, Bob s'empresse d'annoncer qu'il prend l'objet à ce prix là, de peur qu'Alice s'annonce à 14$. Dans l'enchère anglaise, après l'annonce d'Alice de 5$, Bob surenchérie de 1$, et attend. Comme Alice ne surenchérie pas, Bob remporte l'enchère et paye uniquement 6$.

1.1.3 Enchère privée de premier prix

Comme le nom l'indique, l'enchère est de type fermé. Chaque participant remet dans une enveloppe fermée son offre. Le commissaire-priseur ouvre alors toutes les enveloppes et sélectionne la plus grande offre comme gagnante. Le gagnant paye le prix de son offre. Cette enchère impose une forte imputabilité des offres (tout en restant confidentiel) afin d'éviter une fraude.

1.1.4 Enchère de Vickrey

L'enchère de Vickrey est similaire à l'enchère privée de premier prix, si ce n'est que le prix payé par le gagnant est en fait la deuxième plus grande offre. Nous reviendrons sur tout ce que cela implique à la section 1.3.1.

1.1.5 Enchère double

L'enchère double se caractérise par le fait qu'il y a d'une part plusieurs acheteurs, mais aussi plusieurs vendeurs d'un même bien. Chaque acheteur propose une offre d'achat dans une enveloppe scellée et, de la même manière, chaque vendeur propose une offre de vente. Les offres des acheteurs sont ensuite triées de façon ascendante alors que celles des vendeurs de façon descendante. On parcourt ensuite chaque liste et on s'arrête lorsque les prix d'achat et de vente concordent, sélectionnant du coup les gagnants. C'est le système utilisé aujourd'hui par les marchés financiers pour l'échange de valeurs mobilières tels que les actions, obligations ou encore valeurs monétaires.

1.1.6 Enchère japonaise

L'enchère japonaise ou simultanée est l'enchère la plus utilisée au Japon. Elle sert surtout à la vente de poissons à la criée. Après un temps pour évaluer l'objet (ici le poisson), tous les acheteurs donnent leur offre en même temps. La meilleure offre gagne et le gagnant paye son offre.

Cette enchère a le mérite d'être la plus rapide au monde.

1.1.7 Enchère suisse

L'enchère suisse est du même type l'enchère privée de premier prix, si ce n'est que le gagnant peut se rétracter. On demande alors au gagnant suivant s'il veut garder son offre ou se rétracter lui aussi.

Cette enchère est communément utilisée en Suisse pour gérer le secteur de la construction. Les offres de chantiers étant nombreuses, cela limite les conflits de calendrier.

1.2 De la Théorie des jeux aux Mécanismes d'Incitation Algorithmique

1.2.1 Historique

Les mécanismes d'incitation sont une branche spécifique de la théorie des jeux. Cette dernière fut développée principalement par John von Neumann et Oskar Morgenstern[31]. C'est en recherchant une modélisation mathématique du jeu des échecs (problème soulevé alors par le mathématicien hongrois Ernst Zermelo), que John Von Neumann commença à jeter les premières bases de la théorie des jeux. Il trouva ainsi une stratégie unique déterminée selon toutes les possibilités choisies par l'adversaire. Il généralisera plus tard ce modèle avec l'aide de Oskar Morgensten à l'université de Princeton. Les résultats de ses recherches seront publiés dans *Theory of games and economic behavior*[31]. La théorie peut alors être appliquée sur des jeux à somme nulle, où ce que l'un gagne l'autre perd, et principalement pour deux joueurs. Loin encore de pouvoir être utilisée économiquement, la théorie fut alors très appliquée dans le domaine militaire.

Plus tard, John Forbes Nash[15] généralisa alors la théorie de John von Neumann. Grâce à sa théorie des minimax (démontré en 1928), John von Neumann démontra l'existence d'équilibres pour les jeux à somme nulle pour deux participants non coopératifs. Cet équilibre se distinguait par le moment où aucun des deux participants n'a intérêt de changer de stratégie si l'autre ne la change pas également. Intuitivement, cet équilibre est l'état dans lequel chaque participant y trouve satisfaction. John Forbes Nash démontra qu'il existe un même équilibre, appelé *équilibre de Nash*, pour tous jeux, qu'ils soient à somme nulle ou non nulle et pour n'importe quel nombre de joueurs. Bien que peu considérée au début, sa théorie est aujourd'hui appliquée à de très nombreux domaines, qu'ils soient économiques ou informatiques. Il reçut à ce titre le prix Nobel [1] d'économie en 1994.

Définition 1.1 *(Équilibre de Nash)*
Dans chaque jeu comprenant plusieurs joueurs connaissant les stratégies de chacun, il existe au moins un état stable appelé équilibre de Nash où tous les participants maximisent localement

1. Prix de la Banque de Suède en sciences économiques en mémoire d'Alfred Nobel.

leur fonction d'utilité, c'est-à-dire leur gain. En cet état, si un joueur modifie sa stratégie, il ne peut que réduire sa fonction d'utilité, donc son gain.

Dans les années 1960, Leonid Hurwicz[13], puis entre 1970 et 1980 Éric Maskin[19] et Roger Myerson[21] mirent au point une théorie à l'intersection de la théorie des jeux et de la microéconomie, la théorie des mécanismes d'incitation. Cette théorie permet alors d'analyser les divers comportements dans l'économie de marché entre divers acteurs. Alors que la théorie néoclassique des marchés, initiée par Adam Smith[28] à la fin du XVIII^{e} siècle avec le concept de la *Main Invisible*, possédait des contraintes fortes telles que l'utilisation d'un marché de concurrence pure et parfaite comprenant entre autres la transparence du marché, la théorie des mécanismes d'incitation s'affranchit de cette transparence. Elle indique qu'il est possible de mettre en place certains mécanismes d'incitation sur un marché avec des asymétries d'information, qui permettraient l'allocation optimale de bien au sens de Pareto, c'est-à-dire qu'il est impossible d'améliorer le confort de l'un sans réduire celui d'un autre. En 2007, le prix Nobel d'économie a été décerné conjointement à Leonid Hurwicz, Éric Maskin et Roger Myerson pour leurs travaux.

Noam Nisan et Amir Ronen[22] ont défini en 2000 les mécanismes d'incitation algorithmique (AMD). Ces mécanismes sont alors issus de l'intersection des domaines de l'économique, de la théorie des jeux et de l'informatique théorique. Grâce à un mécanisme d'incitation, des joueurs rationnels sont incités à tendre vers un équilibre de Nash. Le mécanisme d'incitation algorithmique le plus connu est le mécanisme de Vickrey-Clarke-Groves (VCG).

1.2.2 Définitions

Un mécanisme d'incitation algorithmique est un système qui propose des règles spécifiques afin d'optimiser l'allocation de biens entre divers joueurs rationnels.

Définition 1.2 *(Fonction d'utilité[23])*
La fonction d'utilité est une fonction représentant le bien-être ou la satisfaction obtenue par

l'obtention ou l'utilisation d'un bien ou d'un service. C'est une évaluation du besoin ou du désir par rapport au coût de son obtention.

Définition 1.3 *(Rationnalité[23])*
Une entité est dite rationnelle si son comportement correspond à son propre intérêt, c'est-à-dire qu'elle s'emploie à maximiser sa fonction d'utilité.

Définition 1.4 *(Problème d'incitation[22])*
Un problème d'incitation est décrit par :

- un ensemble $\mathscr{O}$ de solutions possibles. Chaque solution possible est un sous-ensemble de $\{1,\ldots,n\}$;
- chaque participant $i = 1,\ldots,n$ possède une fonction d'évaluation $v_i : \mathscr{O} \to \mathbb{R}^+$. Elle représente le gain personnel que représente l'acquisition d'un ou de plusieurs biens. Elle est absolument privée ;
- $p_i \in \mathbb{R}^+$ représente le prix que doit acquitter le participant gagnant i ;
- la fonction d'utilité (si quasi linéaire) de chaque participant i est égal à $v_i(o) - p_i$.

La résolution de ce problème d'incitation consiste à sélectionner la meilleure sortie $o \in \mathscr{O}$ qui maximise l'intérêt global $g(v,o) = \sum_i v_i(o)$. La demande $w_i = (r_i, b_i(r_i))$ signifie que le participant i demande r_i unités d'un bien et qu'il est prêt à offrir un prix de $b_i(r_i)$.

Définition 1.5 *(Mécanisme d'incitation[22])*
Un mécanisme d'incitation $m = (k(w), p(w))$ est caractérisé pour ensemble de participants déclarant chacun une valeur $w_i(o)$ par :

- un algorithme d'allocation $k(w) \in \mathscr{O}$, avec w l'ensemble des demandes $w_i = (r_i, b_i(r_i))$ des participants, définissant l'allocation composée des gagnants $i = 1,\ldots,n$;
- une fonction de paiement $p(w) \in \mathbb{R}^+$, définissant le paiement p_i que doivent s'acquitter chaque gagnant de l'allocation $k(w)$.

1.3 Mécanismes de Vickrey-Clarke-Groves

1.3.1 Mécanisme de Vickrey

En 1961, William Vickrey propose un nouveau mécanisme d'enchère[30], appelé depuis Mécanisme de Vickrey, en analysant précisément le comportement des participants. L'enchère utilisée ici est une enchère privée de second prix pour un unique objet mis en vente. Chaque participant émet une offre de façon confidentielle, mais imputable et le mécanisme sélectionne la plus grande offre. Le participant gagnant paye le prix de la deuxième meilleure offre.

On note le mécanisme de Vikrey $m_{Vikrey}(k(w), p(w))$ avec :

$$k(w) = \arg\max_{o \in \mathcal{O}} g(w, o) \tag{1.1}$$

$$p(w) = \arg \max_{o \in \mathcal{O},\ i \notin o} (g(w, o)) \in \mathbb{R}^+ \tag{1.2}$$

L'équation (1.1) permet de récupérer la meilleure allocation possible permettant de maximiser la valeur $g(v, k(w))$. Cette allocation détermine les gagnants de l'enchère. Tandis que l'équation (1.2) permet de calculer le paiement de chaque gagnant appartenant à $k(w)$ en fonction de toutes les autres offres d'achat.

La notation a été légèrement modifiée afin de refléter les offres. Ainsi, $g(w, o) = \sum_{i \in o} b_i(r_i)$.

Définition 1.6 *(Mécanisme efficace[22])*
Un mécanisme est efficace si le participant qui annonce la plus grande offre est sélectionné par le mécanisme.

On remarque que le mécanisme de Vickrey est efficace (voir équation (1.1)). De plus, le prix payé par i, s'il est déclaré gagnant n'est pas, directement ou indirectement, lié à la valeur de l'offre w_i déclarée. Un participant a donc toujours intérêt à déclarer la valeur issue de sa fonction d'évaluation, c'est-à-dire de déclarer $b_i(r_i) = v_i$ (donc *truthful*). Il maximise ainsi ses chances d'être sélectionné comme gagnant sans impacter le prix qu'il devra payer.

Chaque mécanisme d'incitation possède au moins une stratégie dominante.

Définition 1.7

Une *stratégie dominante* d'un mécanisme est une stratégie qui offre au participant des gains toujours supérieurs ou égaux aux autres stratégies, et cela, quel que soit les stratégies utilisées par les autres participants.

Le mécanisme de Vickrey-Clarke-Groves offre à ses participants une stratégie dominante : toujours déclarer la valeur issue de sa fonction d'évaluation, c'est-à-dire de déclarer $b_i(r_i) = v_i$.

Démonstration. En déclarant $b_i(r_i) < v_i$, non seulement le participant perd des chances de gagner, mais de plus, s'il gagne, il payera toujours le même prix $p_i = \arg \max_{o \in \mathcal{O},\ i \notin o} (g(w,o))$, donc sans aucun gain au niveau de sa fonction d'utilité. S'il déclare $b_i(r_i) > v_i$, il s'expose à payer plus cher que son évaluation. En effet, si $\exists w_j, j \neq i$ tel que $b_i(r_i) > b_j(r_j) > v_i$, alors il gagnera l'enchère et en tira un gain négatif. Ainsi, la stratégie dominante dans le mécanisme de Vikrey est de toujours annoncer $b_i(r_i) = v_i$, donc de toujours dire la vérité. □

Définition 1.8 *(Mécanisme d'incitation honnête[22])*

Un mécanisme d'incitation honnête $m = (k(v), p(v))$ est caractérisé pour ensemble de participants déclarant chacun leur vraie valeur d'évaluation $v_i(o)$ par :

- un algorithme d'allocation $k(v) \in \mathcal{O}$ définissant l'allocation composée des gagnants $i = 1, \ldots, n$;
- une fonction de paiement $p(v) \in \mathbb{R}^+$, définissant le paiement p_i que doivent s'acquitter chaque gagnant de l'allocation $k(v)$.

Un mécanisme où la stratégie dominante est de toujours annoncer $b_i(r_i) = v_i$, tel que le mécanisme de Vickrey, est appelé mécanisme d'incitation honnête.

1.3.2 Mécanismes de Clarke et de Grove

Le mécanisme de Clarke[5] est une généralisation du mécanisme de Vickrey publiée en 1971. Sa plus grande innovation est de permettre l'allocation de plusieurs biens. On note le mécanisme de Clarke $m_{Clarke}(k(w), p(w))$ avec :

$$k(w) = \arg\max_{o \in \mathscr{O}} g(w,o), \text{ où } k(w) \text{ est un vecteur de dimension } n \tag{1.3}$$

$$p(w) = \sum_{j \in k(w^{-i})} b_j(r_j) - \sum_{j \in k(w),\ j \neq i} b_j(r_j) \in \mathbb{R}^+ \tag{1.4}$$

L'équation 1.4 représente la somme des offres sélectionnées si le participant i ne participe pas moins la somme des offres des autres participants en la présence de i.

En 1973, Grove[12] généralisa celui de Clarke pour donner naissance au mécanisme de Grove, aussi nommé mécanisme Vickrey-Clarke-Groves en raison de sa parenté. On note ce mécanisme $m_{VCG}(k(w), p(w))$ avec :

$$k(w) = \arg\max_{o \in \mathscr{O}} g(w,o), \text{ où } k(w) \text{ est un vecteur de dimension } n \tag{1.5}$$

$$p(w) = h(w^{-i}) - \sum_{j \in k(w),\ j \neq i} b_j(r_j) \in \mathbb{R}^+ \tag{1.6}$$

$h : \mathscr{O} \to \mathbb{R}^+$ est la fonction d'évaluation d'un algorithme d'allocation arbitraire. Plus simplement, elle effectue la somme des offres des participants gagnants, donc inclus dans $k(w^{-i})$, obtenant ainsi le mécanisme de Clarke.

On remarquera que le mécanisme de Clarke ainsi que celui de Vikrey sont bien des mécanismes de VCG.

Grove prouva lors de ses recherches[12] que tous les mécanismes issus de la famille VCG ont pour propriétés :

a. d'être honnête *(Truthful)* ;

b. d'être efficace *(Ex-Post efficient)* ;

c. de maximiser la somme des utilités des participants ;

d. de ne pas maximiser les revenues du vendeur (malheureusement).

1.3.3 Complexité des Mécanismes d'incitation

La recherche de la meilleure solution $k(w)$, c'est-à-dire la meilleure sélection de participants, est nécessaire aux mécanismes d'incitation. Mais malheureusement celle-ci est, dans la majorité des cas, extrêmement complexe. Le lecteur attentif aura remarqué que la recherche de $\arg\max_{o \in \mathcal{O}} g(w,o)$, $k(w)$ est un problème NP-ardu.

Ce problème peut être réduit au problème du sac à dos, l'un des 21 problèmes NP-complets de Karp[16].

Définition 1.9 *(Problème NP[16])*
Un problème de décision est dans la classe NP s'il admet une résolution dans un temps polynomial grâce à une machine de Turing non déterministe, c'est-à-dire qu'il est possible de vérifier une solution en un temps polynomial.

Théorème 1.3.1 *(Problème NP-Complet[16])*
Tout problème de décision NP peut se réduire à un problème NP-Complet.

Sans approximation, la résolution d'un problème NP-Complet, tel que le problème d'optimisation combinatoire de VCG, se résout en un temps exponentiel. Quelques dizaines de participants sont suffisant pour poser problème si la résolution doit se faire en temps réel.

Malheureusement, introduire une approximation dans VCG rend certes l'application utile et efficace, mais élimine le caractère honnête du mécanisme. Alors, dire la vérité n'est plus la stratégie dominante.

Considérons le problème suivant. Si un algorithme approximatif $k(w)$ ne retourne pas la meilleure valeur possible (car son approximation est tombée, par exemple, sur un maximum local

et non global) alors que $h(w^{-i})$ retourne une valeur plus haute que la « meilleure » valeur possible, alors un paiement négatif sera demandé. Bien que la plupart du temps ce cas ne se produise pas, sa seule existence suffit à supprimer le caractère véridique du mécanisme.

Exemple 1.3.1

Prenons comme exemple le cas où trois participants veulent tous acheter un unique objet[22]. Les fonctions d'évaluation des participants A, B et C sont respectivement 3$, 2$ et 1$. Le mécanisme d'enchère étant de Vickrey, la stratégie dominante est de déclarer sa vraie évaluation. Normalement, l'agent A gagne l'enchère et paye 2$ (second meilleur prix). Si nous prenons le même exemple, mais cette fois-ci avec une approximation de l'algorithme $k(w)$ qui malheureusement se trompe et retourne comme meilleur prix 2$, c'est B qui gagne et qui payera soit 1$ soit, dans le pire des cas, $-1$$. Il est alors facile de voir qu'il n'y a ici aucune stratégie dominante.

1.4 Problématique

Lors de la gestion des ressources de ses clients opérant des VPN, l'ISP doit faire face à deux problèmes majeurs. D'une part, il doit en tout temps gérer les conflits potentiels qui peuvent exister quand les ressources se font rares, celles-ci étant surexploitées pour maximiser ses revenus. D'autre part, il doit gérer en plus une certaine flexibilité qu'exigeront ses clients pour, par exemple, augmenter à court terme leur bande passante allouée.

Conjecture 1

Seul l'opérateur d'un VPN peut réellement prédire ses futurs besoins, étant le seul à avoir accès aux informations nécessaires.

Notre problématique repose sur cette hypothèse forte. Nous croyons qu'il est raisonnable de la tenir pour acquise. L'exemple d'une connexion VoIP en démontre parfaitement sa pleine justification.

Exemple 1.4.1

Lors de l'établissement d'une téléconférence vidéo VoIP, l'initialisation est faite grâce au protocole SIP[27]. Il permet de définir tous les paramètres de la future communication, tels que les débits constants des flux audio et vidéo. L'opérateur du VPN peut, puisqu'il a accès aux échanges dans le VPN, capturer les messages SIP et en déduire, avant même que la communication s'établisse, la future hausse de bande passante.

De ce fait, il est le seul à pouvoir connaître sa véritable utilisation de la bande passante qui lui a été allouée et donc d'en disposer librement. Il est aussi par le même fait le seul pouvant demander une augmentation temporaire de sa bande passante allouée, et cela avant même que la ressource soit nécessaire.

Afin d'exploiter cette simple idée, nous devons mettre en place une architecture qui permettra à chaque opérateur de VPN client de l'ISP de communiquer avec les autres et d'ainsi pouvoir partager selon leur volonté leur ressource non utilisée.

Nous devons donc établir comment un opérateur de VPN va pouvoir partager ou emprunter des ressources avec les autres opérateurs de l'ISP. Cette architecture doit fournir un mécanisme d'incitation afin de rendre profitable et intéressant le partage de ressources inutilisées. Ce mécanisme doit être (1) calculable en temps réel, (2) distribué, (3) incitatif et (4) honnête et efficace.

À chaque temps $t \in T$, chaque participant appartient uniquement à l'un des trois sous-ensembles distincts partitionnant l'ensemble des participants $\mathscr{N}$:

– $i \in \mathscr{R}$ si i demande plus de ressources ;
– $i \in \mathscr{S}$ si i propose de partager des ressources ;
– $i \in \mathscr{T}$ si i ne propose ni ne demande rien (neutre).

En résumé, notre modèle que nous proposons dans ce mémoire doit être conforme à :

Partage

Les VPN $\in \mathscr{R}$ doivent pouvoir en tout temps emprunter des ressources supplémentaires aux VPN de $\mathscr{S}$ sans influencer les VPN neutres de $\mathscr{S}$.

Incitatif

Les VPN doivent être incités à appartenir à $\mathscr{S}$ et offrir leurs ressources inutilisées.

Honnête

Les VPN de $\mathscr{R}$ ne doivent demander que ce qu'ils ont vraiment besoin.

Calculable

Tous les calculs nécessaires doivent pouvoir être faits en un temps raisonnable, permettant le temps réel.

Distribué

Chaque VPN doit pouvoir faire ses propres choix.

Il est important que notre système puisse éviter que des participants égoïstes profitent des autres participants. Ce système devrait permettre à tout participant rationnel cherchant à maximiser son propre bénéfice de coopérer avec les autres participants.

CHAPITRE 2

TRAVAUX ANTÉRIEURS

2.1 Architecture autonome

Traditionnellement, l'ISP ne pouvait pas en tout temps corriger les problèmes de bande passante de ses VPN, puisque la gestion était essentiellement statique. Il n'y avait aucune anticipation et une négociation était à chaque fois nécessaire afin de rentre toute modification possible. Pour répondre à ce premier problème, et rendre la gestion de la bande passante des VPN dynamiques, il faut mettre en place une architecture autonome, qui se suffit à elle-même. Elle doit prendre les décisions, négocier les nouveaux paramètres et pouvoir les appliquer sans aucune intervention.

Afin de répondre à cette première problématique, l'*Autonomic Service Architecture* (ASA) est proposée par Yu Cheng et coll.[4] en 2006. C'est une architecture autonome qui gère l'ensemble des ressources. Elle est composée de nombreux composants qui interagissent ensemble automatiquement afin de réguler l'octroi de la bande passante aux clients du système, en suivant leurs besoins.

Chaque client signe un contrat (SLA) avec l'ISP indiquant la bande passante souhaitée ainsi que les divers termes de flexibilité que permet l'ASA. L'ensemble des SLA est stocké dans une base de données afin que l'architecture puisse le consulter en tout temps. Un système de surveillance des activités réseau (logimètre) est continuellement en service afin de pouvoir connaître en temps réel l'utilisation de la bande passante de chacun des VPN.

C'est à partir de cet ensemble d'informations que l'architecture ASA va prendre ses décisions. La surveillance d'activité et la base de données des SLA permettent de détecter les problèmes. La résolution se fait alors via un module de gestion des opérations. En accord avec les SLA, celui-ci va utiliser les ressources des VPN non surchargés et les réorienter vers les VPN surchargés.

Bien que cette architecture autonome permette une gestion automatique et plus souple de la bande passante entre les VPN, elle amène deux nouveaux problèmes majeurs, inhérents à sa structure centralisée. D'une part, l'ISP doit en tout temps maintenir une surveillance accrue de toutes les ressources de son réseau, et de pouvoir en identifier les flux de chaque VPN afin de pouvoir détecter les ressources inutilisées. Cette tâche est très exigeante et coûteuse. D'autre part, il n'y a aucune incitation pour les opérateurs de VPN pour bien vouloir prêter de la bande passante. Pire, étant donné que l'emprunt de ses ressources sans son consentement peut induire chez lui une perte de QoS, il sera alors incité à faire en sorte de ne pas être en mesure de prêter des ressources, en simulant par exemple leur utilisation maximale.

Plus récemment, Ahmad Quttoum et coll. publièrent en 2010 une amélioration de l'architecture ASA[26]. L'introduction d'un système d'enchère en remplacement du système de décision permet plusieurs améliorations. Ce système d'enchère n'est pas un système d'échange, mais plutôt un système permettant de gagner une priorité de ses données par rapport aux autres VPN, grâce à l'achat ou la vente d'indice QoS. Ainsi, un VPN momentanément peu utilisé pourra se passer de son haut niveau de QoS négocié dans son contrat (SLA) et passer à un niveau plus faible en vendant via le système d'enchère la différence de ressources inutilisées. Cette nouvelle architecture permet d'inciter les VPN à prêter des ressources, afin de réduire leur coût. Toutefois, on peut remarquer que l'application d'un tel mécanisme n'est pas réaliste[22, 18]. En effet, comme nous l'avons signalé plus tôt, la complexité des calculs à effectuer pour trouver le meilleur ensemble d'offres qui maximise les revenues du vendeur est grande, et au-delà d'un petit nombre de VPN, est infaisable en temps réel[9]. De plus, pour les mêmes raisons que l'architecture ASA initiale, ce processus prend d'énormes ressources processeur, et cela en tout temps.

2.2 Architecture distribuée

Internet est un très vaste champ d'applications pour les mécanismes d'incitations qui peuvent être appliqués à de nombreux modèles : routage entre domaines, partage de fichiers poste-à-poste (P2P), gestion de cache internet ou encore distribution de tâches. Tous ces modèles

ont un point commun, directement lié du fait de l'utilisation d'internet : ils utilisent tous des entités distribuées et indépendantes cherchant à maximiser leur profit sans toujours considérer l'intérêt des autres entités.

Afin de permettre l'utilisation des mécanismes d'incitation algorithmique (AMD) de façon distribuée, il est nécessaire d'en modifier quelque peu le fonctionnement pour permettre la participation de plusieurs entités indépendantes tout en gardant l'aspect incitatif.

En 2002, Joan Feigenbaum et coll. publièrent un ensemble de problèmes ouverts et de voies de recherche sur les mécanismes d'incitation algorithmique distribués (DAMD)[11]. Un des problèmes énoncés repose sur la difficulté d'utiliser des algorithmes distribués avec des entités indépendantes qui peuvent mentir ou se tromper. Ces difficultés sont liées au fait qu'il n'existe aucune entité ou aucun algorithme permettant de vérifier les réponses de chaque agent. L'intérêt des DAMD est alors d'inciter à donner les réponses attendues plutôt que de vérifier formellement celles-ci.

Klein et coll. proposent en 2008 un mécanisme d'incitation distribué en deux étapes[17] appliqué à la gestion de transfert d'informations tactiques militaires. Comme ces informations sont nombreuses et le temps de transfert important, il faut prioriser l'information qui va traverser le maillage d'entités indépendantes. Pour ce faire, les auteurs proposent un mécanisme avec deux étapes plutôt qu'une habituellement. Au début de chaque itération, chaque participant annonce son importance. Un centre décide alors d'un plan optimal pour le transfert, en fonction des priorités, et l'annonce à tout le monde. Enfin, les participants transfèrent leur message, en suivant ou non le plan. Plus ils suivent le plan, plus ils seront payés. Cette solution peut paraître naïve, mais repose sur le fait qu'aucun ne peut surévaluer l'importance de son message, car celui-ci pourra toujours être évalué à la réception. Cette restriction est malheureusement très peu présente dans le problème d'allocation de bande passante. De plus, le centre doit être constamment à portée de chaque participant. Enfin, comme le meilleur plan de transfert est calculé uniquement par le centre, des petites itérations de l'ordre de la seconde ne sont plus envisageables.

2.3 Approximation

Afin de rendre calculable en un temps raisonnable des problèmes combinatoires tels que l'utilisation du mécanisme d'incitation VCG, on peut avoir recourt à l'approximation des résultats. Toutefois, cette approximation peut-être parfois très problématique et supprimer certaines propriétés importantes d'un mécanisme. Nous avons vu à la section 1.3.3 qu'une telle approximation rendait le caractère honnête du mécanisme VCG, inopérant.

Afin de résoudre cette problématique, Ariel D. Procaccia et Moshe Tennenholtz ont proposé tout récemment deux mécanismes d'incitation approximés, et ne nécessitant aucun système monétaire[24]. Naturellement, puisqu'ils ne sont pas exacts, l'intérêt global n'est pas optimal. Le premier mécanisme, dit déterministe, arrive à un maximum de 2 fois le coût optimal, tout en ayant un coût minimal de facteur 1. Le deuxième mécanisme propose un coût maximum d'uniquement $\frac{3}{2}$, et utilise un système aléatoire. Le système monétaire est remplacé dans ce cas précis par une diminution des coûts pour les utilisateurs du système. Ceci n'est donc certainement pas applicable partout. Les utilisateurs donnent toujours leur véritable volonté, car c'est la stratégie dominante pour réduire les coûts (même si celle-ci n'est pas optimale).

Plus universel, Noam Nisan et Amir Ronen proposent en 2000 un mécanisme, appelé Mécanisme de deuxième chance, issu de VCG mais pouvant utiliser n'importe quel algorithme approximatif, tout en restant un mécanisme honnête[22]. Pour arriver à cela, ils proposent simplement que chaque participant puisse avoir la chance d'améliorer son propre sort. Après avoir défini un temps maximal de la recherche de la meilleure solution $k(w)$, le mécanisme demande à chacun de lui fournir un algorithme pouvant donner une solution dans cette plage de temps. Ainsi, chacun peut proposer un algorithme qui améliore sa condition et ne peut donc être mis de côté par un mauvais algorithme. Malheureusement, ce mécanisme demande un système central de calcul haute performance capable de résoudre plusieurs algorithmes en même temps. Toutefois, il est intéressant de constater que ce mécanisme peut-être utilisé sur tous les problèmes acceptant VCG comme mécanisme, ce qui en fait un atout considérant sa grande portée.

2.4 Systèmes monétaires et argent virtuel

Les systèmes d'enchères utilisés par les mécanismes d'incitation ont besoin d'un système de paiement utilisant, selon les cas, de la monnaie virtuelle ou non, afin de permettre un transfert d'argent entre les différents participants. De manière générale, ces systèmes de paiement doivent permettre des transferts ayant les propriétés d'être intègres, autorisés, confidentiels, disponibles et fiables. Ces cinq propriétés sont absolument nécessaires pour garantir qu'un transfert d'argent sera effectivement fait entre les deux participants désignés, de façon volontaire et du montant voulu. Tout manquement à une unique propriété met en faillite tout le système.

Un tel système est naturellement soumis à plusieurs attaques[1] menaçant ces propriétés essentielles :

Attaque par rejeu :
Dans ce type d'attaque, on intercepte dans un premier temps un ou plusieurs messages (sans forcément en comprendre le sens) qu'on enregistre afin de pouvoir, dans un deuxième temps les retransmettre. Ceci peut permettre, par exemple, de faire des *double spending*, c'est-à-dire des doubles paiements. Ainsi, un marchand pourrait écouter un paiement d'un de ses clients et retransmettre une ou plusieurs fois celui-ci à la banque, récupérant ainsi plusieurs fois le paiement initial.

Terminal falsifié :
Dans le cas où l'utilisation d'un terminal est nécessaire et qu'il ne peut être authentifié, on peut falsifier ce terminal et effectuer une attaque *man in the middle (MITM)* entre l'utilisateur du terminal et l'exploitant en laissant le terminal intercepter ou modifier la communication. Par exemple, un client d'une banque désirant retirer de l'argent a besoin de sa carte et de s'authentifier grâce à un code NIP, mais il ne peut pas authentifier le guichet automatique, qui peut être un faux.

Attaque par force brute :
Il s'agit ici d'essayer de manière automatisée et systématique toutes possibilités d'entrées les

d'un système de protection. Par exemple, on pourrait essayer toutes les possibilités de NIP d'une carte de crédit. Une variante plus efficace, mais moins probante utilise un dictionnaire plutôt que d'essayer toutes les possibilités.

Cryptanalyse :
La moins probable, mais la plus redoutable, cette méthode d'attaque repose sur la découverte d'une faille directement liée à la structure interne ou aux processus qui la composent. Elle peut être possible grâce à l'amélioration des connaissances ou des technologies rendant inutiles ou simples certains mécanismes d'encodage complexe. Par exemple, la découverte d'un algorithme de décomposition en temps polynomial de nombre en produit de facteurs premiers, rendrait tout à fait désuet les systèmes utilisant le fait qu'aucun algorithme n'est actuellement connu pour pouvoir factoriser un produit de nombres premiers en temps polynomial. Par exemple, le célèbre système cryptologique RSA serait alors cassé.

L'utilisation d'outils cryptographiques est alors généralement nécessaire pour se prémunir de ces divers attaques[6].

L'utilisation de fonction pseudo-aléatoire permet de garantir, par exemple, l'utilisation de nonce unique. Un nonce (*number used once*) est un nombre utilisé qu'une unique fois et qui sera lié, dans notre cas, à une communication entre deux clients désirant faire un transfert d'argent. Comme ce nombre ne peut être plusieurs fois le même, une attaque par rejeu est impossible.

Un système d'engagement numérique permet d'effectuer des paiements hors ligne, c'est-à-dire que le paiement ne s'effectue pas immédiatement, mais s'effectuera dans un avenir proche en utilisant un tiers actuellement non disponible (dans notre cas une banque). Ainsi, un participant peut s'engager à effectuer un virement monétaire depuis son compte en banque.

Tel qu'utilisé dans la vie courante, un système de signature numérique est indispensable pour certifier qu'un participant est bien l'émetteur de l'information. Lors d'une enchère, il serait par exemple désastreux si un participant pour faire une offre d'achat pour un autre participant à l'insu de son plein gré. L'utilisation d'une monnaie virtuelle composée de «jetons» signés pas

une banque centrale permet de rendre impossible la création de monnaie. Les signatures numériques utilisent généralement un système de chiffrement asymétrique[20], en utilisant une clé publique et une clé privée. La clé privée permet de signer un document et seul son propriétaire peut s'en servir, car elle est uniquement en sa possession et est protégée par un système authentification, tel un mot de passe. La clé publique, accessible à tous et sans authentification, permet de vérifier si un document a été signé avec sa clé privée associée.

2.5 Engagement numérique et mise en gage

Un système d'engagement numérique peut être mis en place pour imposer une forte imputabilité à un participant effectuant une action. Son utilisation, avec l'authentification du participant, permet de garantir qu'une action a bel et bien été entreprise par tel participant. Cet outil permet de nous assurer qu'aucun participant ne pourra nier qu'il a formulé une offre d'achat, proposer de vendre des ressources ou reçu un paiement.

L'engagement numérique, ou mise en gage (*bit commitment*) se déroule en deux phases :

a. La phase d'engagement permet à un participant de s'engager sans révéler cet engagement. Tout d'abord, il choisit l'information sur laquelle il va s'engager, puis il la passe dans un système d'engagement, appelé *commitment schemes* qui va encoder l'information afin de la rendre opaque. Il existe deux types d'encodage. Soit il est réversible, et dans ce cas l'information doit pouvoir être décodée ultérieurement, le couple d'encodage/décodage devant garantir que l'entrée et la sortie sont identiques. Soit il n'est pas réversible, et dans ce cas le système d'encodage doit pouvoir être rendu public. L'information encodée est alors envoyée au destinataire, qui possède alors votre information, sans toutefois pouvoir la lire.
b. La phase de révélation permet au deuxième participant de comprendre l'information qu'il a entre les mains. Si l'encodage utilisé est réversible, alors le premier participant lui donne simplement la clé de décodage. Sinon, il lui annonce exactement ce que contient l'information qu'il possède. En encodant lui-même cette information, il doit obtenir exactement l'information encodée qu'il possède, et en déduire que c'est bien l'information annoncée.

Une fois la phase d'engagement effectuée, le participant ne peut revenir en arrière ou mentir sur ce quoi il s'est engagé. Toute l'information sera vérifiée au moment de la phase de révélation. De plus, si l'information est signée, non seulement on ne pourra pas nier l'information, mais celle-ci sera attribuée de façon certaine à un participant.

Des systèmes de mise en gage plus évolués permettent de ne révéler qu'une partie de l'information envoyée, en fonction de la nécessité du moment. On parle alors de systèmes d'engagement à révélation minimum[2].

CHAPITRE 3

MODÉLISATION

Les chapitres 1 et 2 ont permis de dégager un ensemble de solutions qui tournent principalement autour de trois grands axes. Ces solutions permettront de définir des mécanismes améliorant significativement la gestion de ressources que vend un ISP à des opérateurs de VPN. Ces améliorations profiteront à la fois à l'ISP et aux opérateurs de VPN. Ces trois axes sont (1) l'autonomie de gestion, (2) la distribution de ressources et (3) l'approximation efficace de solutions optimales.

Une architecture autonome permet de répondre automatiquement et continuellement au problème de surcharge. Une architecture distribuée permet d'une part d'utiliser un système de partage de calculs, mais surtout de permettre à chaque entité d'agir comme bon lui semble. Ces entités ont une meilleure connaissance de leur besoin à court et moyen termes. Elles peuvent donc savoir combien de ressources elles peuvent offrir ou combien de ressources elles cherchent à obtenir. Enfin, l'approximation algorithmique permet de réduire la complexité des calculs, pouvant rendre un système complexe et non performant cherchant obligatoirement une solution optimale en un système rapide pouvant répondre en temps réel.

Nous proposons au cours de ce chapitre une nouvelle architecture de gestion de ressources pour ISP, qui allie tous les avantages des architectures autonomes distribuées avec un système d'approximation algorithmique. L'alliance de ces trois axes de recherche permet d'éliminer chacun des inconvénients introduits par chaque architecture.

Notre architecture permet aux opérateurs de VPN de décider par eux-mêmes des actions qu'ils vont entreprendre ainsi que de participer activement à l'assignation des ressources. En effet, chaque opérateur de VPN a accès à une bourse d'échange de ressources inutilisées. De plus, il est acteur lors de la vente de ces ressources.

La fluctuation des ressources nécessaires aux bonnes fonctions internes d'un VPN ne peuvent être reflétée à travers un simple SLA rigide qui ne prendra pas compte d'événements exceptionnels qui pourrait survenir.

Afin de garantir une bonne QoS, un opérateur de VPN qui détecte qu'une charge supplémentaire va survenir devrait acheter des ressources supplémentaires. Celles-ci lui permettront de garder sa QoS sans discontinuité pendant toute la durée de la charge supplémentaire exceptionnelle. L'achat se fait à l'aide d'une monnaie virtuelle. Comme il faut pouvoir gagner de l'argent avant de pouvoir en dépenser, chaque VPN devrait donc chercher à vendre ses ressources inutilisées. Ainsi, les opérateurs de VPN coopératifs pourront subvenir à leurs besoins exceptionnels de ressources puisqu'ils auront les ressources monétaires requises ayant permis à d'autres dans le passé de se prémunir contre une montée en charge inhabituelle. Quant aux opérateurs de VPN égoïstes, ils ne pourront se prévaloir de la moindre protection. En effet, ceux-ci n'ayant pas partagé de ressources au préalable, ils n'ont donc pas accumulés d'argent virtuel. Sans argent, on ne peut acheter de ressources supplémentaires en cas de surcharge.

Notre proposition est donc un mécanisme d'enchère efficace qui permet à un ensemble de participants coopératifs d'opérer sur une longue période de temps.

3.1 Généralités

Soit n le nombre d'opérateurs de VPN au sein d'un ISP. L'ensemble de ces opérateurs forme l'ensemble $\mathscr{N}$ dont la cardinalité est $\text{card}(\mathscr{N}) = n$. Ces opérateurs se partagent la bande passante Q disponible chez l'ISP. En raison de leur SLA, chaque participant $i \in \mathscr{N}$ possède à chaque round $t \in [\![1;T]\!]$ une quantité de bande passante $q_i(t)$. Comme on ne peut partager plus qu'on ne possède, la somme des bandes passantes reste fixe en tout temps, c'est-à-dire $\forall t \in [\![1;T]\!], \sum_{i=1}^{n} q_i(t) = Q$.

À chaque temps $t \in [\![1;T]\!]$, chaque participant appartient uniquement à l'un des trois ensembles ($\mathscr{R}_t$, $\mathscr{S}_t$ et $\mathscr{T}_t$) de la partition $\mathscr{N}$, c'est-à-dire trois ensembles mutuellement disjoints et dont l'union forme $\mathscr{N}$:

– $i \in \mathscr{R}_t$ si l'opérateur i demande des ressources supplémentaires au temps t ;
– $i \in \mathscr{S}_t$ si l'opérateur i propose de partager des ressources inutilisées au temps t ;
– $i \in \mathscr{T}_t$ si l'opérateur i ne propose ni ne demande aucune ressource au temps t.

Pour simplifier la notation, l'indice t sera omis par la suite. Implicitement, la partition de $\mathscr{N}$ ainsi que les requêtes et offres seront dépendantes de $t \in [\![1;T]\!]$.

Tout l'intérêt d'un système autonome distribué provient du fait que chaque opérateur peut déterminer localement à quel ensemble il appartient, ces derniers connaissant mieux que quiconque leur besoin à court terme.

Chaque participant $i \in \mathscr{R}$ va formuler une demande $w_i = (r_i, b_i(r_i)) \in \mathbb{R}^+ \times \mathbb{R}^+$, r_i représentant la quantité de bande passante demandée et $b_i(r_i)$ l'offre de prix associée à cette demande. On note $R = \sum_{i \in \mathscr{R}} r_i$ l'ensemble de la bande passante demandée. D'un autre côté, chaque participant $i \in \mathscr{S}$ va proposer une certaine quantité de bande passante s_i. On note $S = \sum_{i \in \mathscr{S}} s_i$ l'ensemble de la bande passante offerte. Naturellement, on considéra que $R > S$, c'est-à-dire que la demande est plus grande que l'offre, le problème étant sinon dénudé d'intérêt.

Le but du mécanisme d'incitation est de maximiser l'allocation de la bande passante, c'est-à-dire de vendre le plus de bande passante et au plus grand prix. Ceci est réalisé par l'algorithme $k(\mathscr{R}, \mathscr{W}, S)$, $\mathscr{W}$ étant l'ensemble des déclarations d'utilité des participants de $\mathscr{R}$, c'est-à-dire $\mathscr{W} = \{(r_i, b_i(r_i)) | i \in \mathscr{R}\}$. Ainsi, l'algorithme $k(\mathscr{R}, \mathscr{W}, S)$ retourne le meilleur sous-ensemble $\mathscr{R}^* \subseteq \mathscr{R}$ tel que :

$$\begin{aligned} \arg\max_{\mathscr{R}^* \subseteq \mathscr{R}} g(\mathscr{R}^*, \mathscr{W}) &= \sum_{i \in \mathscr{R}^*} b_i(r_i) \\ \text{t.q.} \sum_{i \in \mathscr{R}^*} r_i &\leq S \end{aligned} \tag{3.1}$$

Chaque participant peut communiquer à sa guise avec les autres participants. Il déclare les informations qu'il veut et peut mentir (mais ce ne sera pas dans son intérêt). Nous donne-

rons plus de détails sur les possibilités de communication, l'imputabilité des messages et des actions et les systèmes de paiements dans la section 3.5. Nous considérons donc que tous messages sont imputables à un participant, et qu'il existe un système de paiement fiable. La transmission de messages peut également être chiffrée, bien qu'inutile.

3.2 Menaces et hypothèses

Les participants envoient des messages, tels que la déclaration de leur intention, qui sont critiques. Il est essentiel pour le bon fonctionnement de notre mécanisme que ces messages ne soient pas modifiés et que leurs auteurs respectifs soient clairement désignés.

Hypothèse 1
Chaque participant possède une clé privée ainsi qu'une clé publique certifiée par le mécanisme. Ainsi, chaque participant peut signer ses messages.

3.3 Modèle simple

Nous présentons ici notre premier mécanisme $M_1(\mathscr{R}, \mathscr{S}, \tau_O, \tau_C)$, sous la forme de l'algorithme 1. Ce mécanisme est distribué et autonome, et il peut être évalué en temps réel. Ce mécanisme est appelé à tous les débuts d'intervalle t et fait agir tous les opérateurs de VPN. Le temps de calcul τ_C est fixé à l'avance de manière à rendre le mécanisme calculable en temps réel.

Le fait de signer sa demande à l'étape 3 permet d'éviter qu'un imposteur se fasse passer pour un autre et propose une offre à l'insu de celui-ci. Ceci peut avoir deux conséquences. D'une part l'opérateur de VPN usurpé peut voir son compte en banque se vider d'un coup dans le cas d'une fausse offre extravagante, d'autre part les vendeurs peuvent voir leur prix de vente diminuer dans le cas d'une fausse offre qui diminurait une vraie. Nous verrons plus en détail l'utilisation des signatures dans la section 3.5.1.

La forme du mécanisme n'est pas définie et peut prendre différentes formes. Il peut être complètement distribué parmi les participants. Dans ce cas, chaque participant offrant ou requérant devra calculer par lui-même les $p_i \forall i \in k^*$ ainsi que $\sum_{i \in k^*} p_i$ afin de connaître le gain final de

Algorithme 1 : Mécanisme simple $M_1(\mathscr{R}, \mathscr{S}, \tau_O, \tau_C)$

1. Chaque participant $i \in \mathscr{R}$ nécessitant plus de ressources formule en privé une demande $w_i = (r_i, b_i(r_i))$. Il produit une empreinte unique de cette demande $H(w_i, nonce_i)$ et l'envoie publiquement.
2. Chaque participant $i \in \mathscr{S}$ désirant vendre des ressources inutilisées formule en privé une offre s_i. Il produit une empreinte unique de cette demande $H(s_i, nonce_i)$ et l'envoie publiquement.
3. Après τ_O, chaque participant $i \in \mathscr{R}$ transmet publiquement sa demande signée $\{w_i, Sign(K_i, w_i\}$, et chaque participant $i \in \mathscr{S}$ transmet publiquement son offre signée $\{s_i, Sign(K_i, s_i)\}$.
4. Chaque participant $i \in \mathscr{N}$ peut, durant l'intervalle de temps donné τ_C et à l'aide d'une méthode approximative, calculer puis envoyer publiquement le meilleur sous-ensemble k_i tel que (1) $g(k_i, \mathscr{W})$ maximise le prix de la vente de l'ensemble de la bande passante offerte S parmi tous les sous-ensembles considérés durant τ_C, et (2) $\sum_{j \in k_i} r_j \leq S$.
5. Le mécanisme recherche dans $\mathscr{K} = \{k_i | \forall i \in \mathscr{N}\}$ la meilleure des solutions $k^* = \arg\max_{k_i \in \mathscr{K}} g(k_i, \mathscr{W})$.
6. Pour chaque gagnant $i \in k^*$, on considère la solution $k^* \setminus \{i\}$. Ainsi, l'ensemble $\mathscr{K}^+ = \mathscr{K} \bigcup \{k^* \setminus \{i\} | i \in k^*\}$.
7. Le mécanisme calcule $p_i = p_i(k^*, \mathscr{K}^+, \mathscr{W}) = \max_{k \in \mathscr{K}^+ \text{t.q. } i \notin k} g(k, \mathscr{W}) - [g(k^*, \mathscr{W}) - b_i(r_i)]$, avec $\mathscr{W} \in k^*$.
8. Chaque contributeur $i \in \mathscr{S}$ recevra un pourcentage du total des paiements au prorata de ce qu'il a offert, c'est-à-dire $\forall j \in \mathscr{S}, \frac{b_j}{S} \sum_{i \in k^*} p_i$.
9. Le mécanisme émet une liste signée de couples de paiements à effectuer $\{p(i,j,q), S(K_0, p(i,j,q))\} \in k^* \times \mathscr{S} \times R^+$.
10. Chaque participant gagnant $i \in k^*$ paie ce qu'il doit.

chaque offrant, c'est-à-dire au prorata de son offre, soit $\frac{b_j}{S} \sum_{i \in k^*} p_i, \forall j \in \mathscr{S}$. Les paiements se font de façon distribuée comme nous le verrons dans la section 3.5.2. La résolution de conflit pourra se faire si nécessaire par l'ISP faisant encore office ici d'arbitre neutre.

Il peut être aussi incarné directement et de façon centrale par l'ISP, telle une entité indépendante faisant office d'arbitre neutre.

Lemme 1 *(Information partielle)*

Le mécanisme $M_1(\mathscr{R}, \mathscr{S}, \tau_O, \tau_C)$ est un mécanisme à information partielle. En effet, l'utilisation dans une première phase d'envoi d'empreintes ne donne pas d'information sur les offres

d'achat ou de vente et est complètement anonyme. À la fin de cette période, l'envoi des véritables informations correspondantes aux empreintes est validé par tous[2].

Un mécanisme à information partielle garantie qu'aucun participant ne prendra de décision, ou même changera de décision, en fonction des autres puisque qu'il ne pourra pas accéder à cette information avant que tout le monde, y compris lui-même, n'aie déjà officialisé sa décision.

Remarque 3.3.1
$H(\bullet,\bullet)$ est une fonction de hachage à sens unique, c'est-à-dire résistante à la seconde préimage[20]. Le $nonce_i$, un entier généré aléatoirement suffisamment grand et à usage unique, est utilisé afin de prévenir les attaques par dictionnaire[20].

Lemme 2 *(Paiement positif)*
Le mécanisme $M_1(\mathscr{R},\mathscr{S},\tau_O,\tau_C$ ne propose que des paiements positifs. En effet, tous les paiements p_i sont positifs ou nuls.

Démonstration. Soit $i \in k^*$, $p_i(k^*,\mathscr{K}^+,\mathscr{W}) = \max_{k\in\mathscr{K}^+ \text{t.q. } i\notin k} g(k,\mathscr{W}) - [g(k^*,\mathscr{W}) - b_i(r_i)]$
Puisque $k^* \setminus \{i\} \in \mathscr{K}^+$, par construction on a : $\max_{k\in\mathscr{K}^+ \text{t.q. } i\notin k} g(k,\mathscr{W}) \geq g(k^* \setminus \{i\},\mathscr{W})$. Donc :

$$\begin{aligned} p_i(k^*,\mathscr{K}^+,\mathscr{W}) &\geq g(k^* \setminus \{i\},\mathscr{W}) - [g(k^*,\mathscr{W}) - b_i(r_i)] \\ &\geq [g(k^*,\mathscr{W}) - b_i(r_i)] - [g(k^*,\mathscr{W}) - b_i(r_i)] \\ &\geq 0 \end{aligned}$$

□

L'utilité d'avoir un mécanisme ne proposant que des paiements positifs est manifeste. En effet, si un paiement négatif pouvait avoir lieu, cela résulterait qu'une vente de ressources inutilisées pourrait coûter de l'argent au vendeur.

Lemme 3 *(Stratégie dominante k)*
Il existe une stratégie dominante pour chaque participant $i \in \mathscr{R}$ demandant des ressources supplémentaires. Chaque participant $i \in \mathscr{R}$ cherche à trouver le sous-ensemble $k_i \subseteq \mathscr{R}$ maximisant

le prix de la vente de l'ensemble de la bande passante offerte S tout en s'incluant lui-même, c'est-à-dire $i \in k_i$.

Démonstration. D'une part, si $i \notin k_i$, alors la fonction d'utilité de i est $u_i = 0$, sinon $u_i \geq 0$. Il n'y a donc aucun intérêt d'émettre l'ensemble k_i tel que $i \notin k_i$. D'autre part, il est essentiel que $g(k_i, \mathscr{W})$ soit maximal, car le mécanisme recherche la meilleure des solutions $k^* = \arg\max_{k \in \mathscr{K}^+} g(k, \mathscr{W})$. □

Ainsi, un participant demandant plus de ressources a tout intérêt à passer du temps à calculer un sous-ensemble gagnant, car c'est une chance pour lui de s'inclure lui-même. De plus, il a aussi intérêt de le faire bien, afin que son sous-ensemble, dont il fait partie, soit sélectionné.

Lemme 4 *(Mécanisme non honnête)*
Un participant peut manipuler le mécanisme en offrant un prix démesuré pour peu de ressources afin d'induire un paiement nul sous certaines conditions.

Démonstration. La stratégie dominante pour chaque participant $i \in \mathscr{R}$ est de trouver l'ensemble k_i maximisant le prix de la vente de l'ensemble de la bande passante offerte S tout en s'incluant lui-même. Si un participant j propose une offre démesurée pour une ressource r_j et que $\forall i \neq j \in \mathscr{R}, r_i + r_j \leq S$, alors il sera nécessairement choisi par tous. Par construction, $\mathscr{K}^+ = \{\{i, j\} | i \in \mathscr{R} \text{ et } i \neq j\} \bigcup \{\{i\} | i \in \mathscr{R} \text{ et } i \neq j\}$

$$\begin{aligned} p_j(\{i^*, j\}, \mathscr{K}^+, \mathscr{W}) &= \max_{k \in \mathscr{K}^+ \text{t.q. } i \notin k} g(\mathscr{K}, \mathscr{W}) - [g(\{i^*, j\}, \mathscr{W}) - b_j(r_j)] \\ &= g(\{i^*\}, \mathscr{W}) - [g(\{i^*, j\}, \mathscr{W}) - b_j(r_j)] \\ &= b_{i^*}(r_{i^*}) - [b_{i^*}(r_{i^*}) + b_j(r_j) - b_j(r_j)] \\ &= 0 \end{aligned}$$

□

Bien qu'une manipulation du mécanisme soit possible, celle-ci ne rend pas le mécanisme M_1 caduc malgré le risque potentiel. En effet, si deux participants utilisent cette stratégie détournée, il en résultera que l'un des deux subira une perte quasi totale de tout son argent

virtuel. Ce n'est donc pas une stratégie dominante, mais une faille qui peut être exploitée succinctement. Ce modèle est donc tout à fait exploitable si plus d'un participant est égoïste et cherche à exploiter les autres participants et reste simple d'utilisation.

3.4 Modèle avancé

Algorithme 2 : Mécanisme avancé $M_2(\mathscr{R}, \mathscr{S}, \tau_O, \tau_C)$

1. Chaque participant $i \in \mathscr{R}$ nécessitant plus de ressources formule en privé une demande $w_i = (r_i, b_i(r_i))$. Il produit une empreinte unique de cette demande $H(w_i, nonce_i)$ et l'envoie publiquement.
2. Chaque participant $i \in \mathscr{S}$ désirant vendre des ressources inutilisées formule en privé une offre s_i. Il produit une empreinte unique de cette demande $H(s_i, nonce_i)$ et l'envoie publiquement.
3. Après τ_O, chaque participant $i \in \mathscr{R}$ transmet publiquement sa demande signée $\{w_i, Sign(K_i, w_i\}$, et chaque participant $i \in \mathscr{S}$ transmet publiquement son offre signée $\{s_i, Sign(K_i, s_i)\}$.
4. Chaque participant $i \in \mathscr{N}$ peut, durant l'intervalle de temps donné τ_C et à l'aide d'une méthode approximative, calculer puis envoyer publiquement le meilleur sous-ensemble k_i tel que (1) $g(k_i, \mathscr{W})$ maximise le prix de la vente de l'ensemble de la bande passante offerte S parmi tous les sous-ensembles considérés durant τ_C, et (2) $\sum_{j \in k_i} r_j \leq S$.
5. Le mécanisme recherche dans $\mathscr{K} = \{k_i | \forall i \in \mathscr{N}\}$ la meilleure des solutions $k^* = \arg\max_{k_i \in \mathscr{K}} g(k_i, \mathscr{W})$.
6. Soit $i \in k^*$. Chaque participant j cherche, durant le temps donné τ_C, à trouver des sous-ensembles $k_{j,i}$t.q. $i \notin k_{j,i}$ et $g(k_{j,i}, \mathscr{W})$ est maximale, puis les publie.
7. L'ensemble $\mathscr{K}^+ = \mathscr{K} \bigcup k^* \setminus \{i\} \bigcup \{k_{j,i} | i \in K^*, j \in N\}$ est formé.
8. Le mécanisme calcule $p_i = p_i(k^*, \mathscr{K}^+, \mathscr{W}) = \max_{k \in \mathscr{K}^+ \text{t.q. } i \notin k} g(k, \mathscr{W}) - [g(k^*, \mathscr{W}) - b_i(r_i)]$, avec $\mathscr{W} \in k^*$.
9. Chaque contributeur $i \in \mathscr{S}$ recevra un pourcentage du total des paiements au prorata de ce qu'il a offert, c'est-à-dire $\forall j \in \mathscr{S}, \frac{b_j}{S} \sum_{i \in k^*} p_i$.
10. Le mécanisme émet une liste signée de couples de paiements à effectuer $\{p(i,j,q), S(K_0, p(i,j,q))\} \in k^* \times \mathscr{S} \times R^+$.
11. Chaque participant gagnant $i \in k^*$ paie ce qu'il doit.

Nous présentons ici le mécanisme avancé, exploitable à partir de l'algorithme 2. Il a pour but d'éliminer la possibilité d'exploiter la faille permettant d'obtenir ponctuellement des ressources pour un paiement nul.

Lemme 5 *(Information partielle)*
Le mécanisme $M_2(\mathscr{R}, \mathscr{S}, \tau_O, \tau_C)$ est un mécanisme à information partielle. En effet, l'utilisation dans une première phase d'envoi d'empreintes ne donne pas d'information sur les offres d'achat ou de vente et est complètement anonyme. À la fin de cette période, l'envoi des véritables informations correspondantes aux empreintes est validé par tous[2].

Le mécanisme $M_2(\mathscr{R}, \mathscr{S}, \tau_O, \tau_C)$ garantie donc qu'aucun participant ne prendra de décision, ou même changera de décision, en fonction des autres.

Lemme 6 *(Paiement positif)*
Le mécanisme $M_2(\mathscr{R}, \mathscr{S}, \tau_O, \tau_C$ ne propose que des paiements positifs. En effet, tous les paiements p_i sont positifs ou nuls.

Démonstration. La preuve est identique à celle du lemme 2, puisque par construction, $k^* \setminus \{i\} \in \mathscr{K}^+, \forall i \in k^*$ □

Comme la logique le veut, une vente ne pourra coûter de l'argent au vendeur.

Lemme 7 *(Stratégie dominante k avec $\mathscr{W}$)*
Il existe une stratégie dominante pour chaque participant $i \in \mathscr{R}$ demandant des ressources supplémentaires. Chaque participant $i \in \mathscr{R}$ cherche à trouver le sous-ensemble $k_i \subseteq \mathscr{R}$ maximisant le prix de la vente de l'ensemble de la bande passante offerte S tout en s'incluant lui-même, c'est-à-dire $i \in k_i$.

Démonstration. La preuve est identique à celle du lemme 3. □

Un participant demandant plus de ressources a tout intérêt à passer du temps à calculer un sous-ensemble gagnant, car c'est une chance pour lui de s'inclure lui-même. De plus, il a aussi intérêt de le faire bien, afin que son sous-ensemble, dont il fait partie, soit sélectionné.

Lemme 8 *(Stratégie dominante k avec $\mathscr{W}^{-i}$)*

Il existe une stratégie dominante pour chaque participant $i \in \mathscr{S}$ proposant des ressources inutilisées. Chaque participant $i \in \mathscr{S}$ cherche à trouver le sous-ensemble $k_i^* \subseteq \mathscr{R}$ maximisant le prix de la vente de l'ensemble de la bande passante offerte S.

Démonstration. Soit k^* la solution découverte dans la première phase du mécanisme et soit $i \in k^*$ un participant demandeur (donc $i \in \mathscr{R}$).

$$p_i = \max_{k \in \mathscr{K}^+ \text{t.q. } i \notin k} g(k, \mathscr{W}) - [g(k^*, \mathscr{W}) - b_i(r_i)], \text{ avec } \mathscr{W} \in k^*$$

Si un participant proposant des ressources trouve une solution k' t.q. $g(k') > g(k^*)$, alors :

– Si $i \notin k'$,
  $p'_i = g(k', \mathscr{W}) - [g(k^*, \mathscr{W}) - b_i(r_i)]$
  $p_i = \max_{k \in \mathscr{K}^+ \text{t.q. } i \notin k} g(k, \mathscr{W}) - [g(k^*, \mathscr{W}) - b_i(r_i)]$
  Donc $p'_i > p_i$.
– Si $i \in k'$,
  $p'_i = p_i = \max_{k \in \mathscr{K}^+ \text{t.q. } i \notin k} g(k, \mathscr{W}) - [g(k^*, \mathscr{W}) - b_i(r_i)]$
  Donc $p'_i = p_i$.

Ainsi, un participant proposant des ressources n'a aucun avantage à dissimuler un meilleur résultat, car en aucun cas il ne paiera plus. □

Un participant proposant à la vente des ressources inutilisées a tout intérêt à passer du temps à calculer, avec la plus grande rigueur, un sous-ensemble gagnant, car c'est pour lui une chance d'augmenter le gain total de la vente, donc son gain personnel.

Il n'y a plus ici de possibilité de manipuler le mécanisme afin d'obtenir ponctuellement des ressources pour un paiement nul.

3.5 Imputabilité et paiements

3.5.1 Imputabilité

Notre modèle utilise un système de signature numérique[20] lors de certains échanges de données entre toutes les entités de notre système, que ce soient les participants ou bien le mécanisme. Tel un document signé de façon manuscrite, il permet d'en définir l'authenticité à laquelle il rajoute d'autres propriétés intéressantes. Le fait de signer un échange permet de réunir les propriétés suivantes :

– Le document ne peut être modifié.
– La signature est unique pour chaque document.
– Le signataire peut être authentifié et est unique.
– Le signataire ne peut nier qu'il a lui-même signé le document (non-répudiation).

Nous utilisons principalement le fait que seul le véritable signataire peut signer à son nom un document, et de ce fait nous garantissons ainsi que personne ne pourra déclarer ou communiquer des données en se faisant passer pour une autre personne.

D'autre part, nous utilisons le système d'empreintes[20]. Une empreinte correspond à une version d'un document et le changement d'un unique élément atomique de celui-ci change complètement l'empreinte qui en résulte. Le fait de publier uniquement l'empreinte d'un document permet de créer un engagement numérique, c'est-à-dire de signaler que le document est prêt et qu'il ne changera pas avant sa publication[2].

Nous utilisons les empreintes dans notre modèle afin de permettre à tous de publier des informations dont personne ne connaîtra le contenu avant un certain moment. Ceci peut se faire grâce à ce protocole :

a. On envoie l'empreinte de nos données $C = H(\text{Data})$.
b. Quand tout le monde a envoyé ses données, on envoie nos données signées $\{\text{Data}, S(\text{SK}, \text{Data})\}$.

c. Chacun peut vérifier que (1) Data a bien été émis par la bonne personne en vérifiant sa signature grâce à sa clé publique PK et (2) chacun peut vérifier que les données non pas changées en vérifiant que $C = H(\text{Data})$.

Ceci nous permet de simuler un système où tout le monde peut émettre anonymement une participation qui ne sera révélée qu'à un moment donné, tel un système d'enchère de type fermé où chacun remet une enveloppe scellée.

Dans notre système, ceci garantit qu'aucun participant ne peut connaître les intentions des autres lorsqu'il devra émettre ses propres intentions, afin de rendre le mécanisme à information partielle.

3.5.2 Système de paiement

Afin de garantir l'intégrité de la richesse de chacun, et de ce fait garantir le bon fonctionnement du mécanisme, un système de paiement doit être mis en place. Il doit permettre un transfert d'argent virtuel entres les divers participants et de prévenir toute fraude, tel un paiement sans fond.

Nous proposons ici deux systèmes simples qui conviennent à notre problématique, mais qui pourront aisément être substitués par d'autres systèmes équivalents[25].

Banque centrale

Le mécanisme étant incarné par l'ISP. il est au coeur de notre architecture et représente une entité centrale. La greffe d'une banque centrale à celui-ci est donc tout à fait envisageable. Une banque liée au mécanisme est très intéressante, car étant donné que le mécanisme calcule le prix final de chaque transaction, il est alors simple de gérer les transferts d'argent directement. De plus, le mécanisme peut vérifier dès le début que chaque participant faisant une offre d'achat possède réellement les fonds nécessaires pour cette offre. De leur coté, les participants n'ont qu'à retenir leur montant en banque et de l'ajuster après chaque transfert. On pourra également rajouter au protocole du mécanisme un appel pour connaître l'état de son compte en banque.

Banque distribuée

Notre architecture reposant sur un réseau d'entités indépendantes, la possibilité d'utiliser une banque distribuée entre chaque entité est possible. Chaque participant arrivant avec une somme prédéterminée, il est facile pour chacun de tenir les comptes de tous, car le mécanisme donne des ordres de paiement publiquement en le diffusant à tous. Le mécanisme tient lui aussi les comptes de tout le monde et, dans un cas de tricherie, pourrait sévèrement punir le participant malveillant (en supprimant tout son argent virtuel par exemple). Comme pour la banque centrale, le mécanisme peut vérifier dès le début que chaque participant faisant une offre possède réellement les fonds nécessaires pour cette offre.

CHAPITRE 4

IMPLÉMENTATION

Nous devons simuler ici un réseau composé de plusieurs participants coopératifs utilisant un mécanisme d'enchères permettant d'échanger les ressources supplémentaires entre eux. Ne simulant que des interactions du point de vue de l'applicatif, nombreux paramètres et problèmes liés aux liaisons, transports et sessions ne sont pas pris en compte.

Cette implémentation permettra de valider notre modèle grâce à un ensemble de simulations basées sur plusieurs scénarios. Nous utiliserons pour cela le mécanisme 2 développé à la section 3.4.

4.1 Choix techniques

Afin de rendre notre implémentation valide, il faut qu'elle soit le reflet de notre modèle. Des choix techniques doivent être pris de façon à répondre au plus près des exigences qu'il impose. L'implémentation doit pouvoir prendre en compte plusieurs scénarios, dont nous verrons les aboutissements dans le chapitre suivant.

Notre modèle ne comporte que des ensembles mathématiques, mais il est bien plus commun et facile de manipuler des tableaux en programmation. De ce fait, tous les ensembles, tels que $\mathscr{N}$, $\mathscr{W}$ et $\mathscr{K}$ seront représentés par des tableaux.

Nos différents scénarios changeront les paramètres d'initialisation alors que le coeur même de notre modèle, la négociation et le partage, restera le même.

4.2 Initialisation

L'initialisation de l'ensemble des paramètres de la simulation est faite lors de cette étape. Les SLA de chacun des participants sont mises en place et la future utilisation de bande passante est générée.

4.2.1 Génération des SLA

Après avoir défini la capacité totale de l'ISP, nous allons définir pour chaque VPN un achat de bande passante. Cet achat est défini dans le SLA de chaque VPN. Afin de rendre la simulation la plus utile, nous faisons en sorte que la somme des SLA soit égale à la capacité totale de l'ISP. De plus, chacun des SLA sera spécifié suivant une méthode comportant un système pseudo-aléatoire, afin de rendre le tout plus réaliste.

Chaque SLA de VPN est initialisé de façon pseudo-aléatoire à une valeur entre 0 et 2 fois la moyenne. La moyenne n'est rien d'autre que la capacité totale de l'ISP divisé par le nombre de VPN. La somme totale des SLA est ensuite ramenée vers la capacité totale de l'ISP.

Exemple 4.2.1 *(Génération des SLA)*
Nous allons ici générer étape par étape $n = 5$ SLA qui partageront $Q = 500$ MO.

a. Calcul de la moyenne :
   $\bar{Q} = S/n = 500/5 = 100$
b. Calcul de 5 valeurs initiales pour les SLA, comprise dans $[0; 2\bar{Q}]$:
   $q = 2 * \bar{Q} *$ rand , avec rand un réel $\in [0; 1]$
   $q_1 = 2 * 100 * 0.21 = 42$
   $q_2 = 2 * 100 * 0.76 = 152$
   $q_3 = 2 * 100 * 0.33 = 66$
   $q_4 = 2 * 100 * 0.67 = 134$
   $q_5 = 2 * 100 * 0.62 = 124$
c. Calcul de la somme des bandes passantes et de l'écart par rapport à la bande passante mise à disposition par l'ISP Q :
   $\hat{Q} = \sum_{i=1}^{n} q_i = 518$
   $\delta_Q = Q - \hat{Q} = 500 - 518 = -18$
d. Calcul de l'écart moyen $\bar{\delta_Q}$ pour chaque SLA et sommation de cet écart :
   $\bar{\delta_Q} = \delta_Q/n = -18/5 = -3.6$
   $\bar{q_1} = 42 + (-3.6) = 38.4$

$\bar{q_2} = 152 + (-3.6) = 148.4$

$\bar{q_3} = 66 + (-3.6) = 62.4$

$\bar{q_4} = 134 + (-3.6) = 130.4$

$\bar{q_5} = 124 + (-3.6) = 120.4$

$\hat{Q} = \sum_{i=1}^{n} \bar{q_i} = 500$

e. On vérifie que $\forall i \in [1;n], \bar{q_i} \geq 0$. En cas de non-respect des conditions, on recommence toute la procédure de génération.

4.2.2 Génération de l'utilisation de la bande passante

Les besoins de bande passante sont générés dès le début de chaque itération. Ces besoins sont basés sur les SLA, sur la probabilité d'être surchargé et sur le taux de surcharge maximal.

Comme il s'agit d'un processus stochastique, le nombres de VPN surchargés ainsi que leur identité peuvent varier à chaque itération. S'ils ne sont surchargés, les VPN utiliseront un montant aléatoire de leur bande passante définie dans leur SLA.

4.2.3 Initialisation de l'argent de base

Tous les VPN participants reçoivent initialement la même somme d'argent. Cet argent est simulé dans les comptes d'une banque centrale logée dans le mécanisme. À la fin de chaque itération, les participants ayant vendu leurs ressources inutilisées seront dédommagés, tandis que les participants ayant acheté des ressources supplémentaires devront payer pour celles-ci.

Toutefois, il est aussi possible de distribuer la gestion des comptes à l'ensemble des participants. Chaque transaction authentifiée est alors diffusée à tous.

4.2.4 Mise en place des stratégies

Afin de démontrer le bon fonctionnement de notre système, nous incorporons à chaque VPN une stratégie. La première stratégie sera la stratégie dominante de notre mécanisme, c'est-à-dire de toujours dire la vérité sur ces besoins et de prêter au maximum aux autres. La deuxième

stratégie proposera l'utilisation des ressources d'autrui dès que nécessaire, sans pour autant proposer de partager ses propres ressources non utilisées.

4.3 Déroulement

Maintenant que l'environnement est initialisé, nous allons voir ici les différentes étapes lors de chaque itération. Également, nous expliquerons en détail les divers algorithmes qui seront utilisés lors de ces étapes.

À chaque itération, chaque participant i appartient à une unique catégorie :

– $i \in \mathscr{R}$ si i demande plus de ressources ;
– $i \in \mathscr{S}$ si i propose de partager des ressources ;
– $i \in \mathscr{T}$ si i ne propose ni ne demande rien (neutre).

Ces trois états sont liés directement à la stratégie mise en place, c'est-à-dire si le VPN est égoïste ou non.

4.3.1 Annonces

Au début de chaque itération, les VPN annoncent dans quelle catégorie ils sont. Chaque participant en surcharge va formuler une demande composée de la quantité de bande passante demandée et d'une offre de prix associée à cette demande. Le prix unitaire est défini de façon pseudo-aléatoire entre deux valeurs définies lors de la simulation. Bien qu'arbitraire, la fluctuation du prix unitaire correspond en fait à la nécessité absolue d'avoir à ce moment précis les ressources requises. Cette nécessité ne peut être perçue dans une simulation. Toutefois, chaque gestionnaire de VPN pourra, dans le cadre d'une application réelle, définir ses propres algorithmes de calculs de prix en fonction de sa charge, du prix du SLA, du prix de ses clients, de sa monnaie virtuelle restante, etc. Le mécanisme vérifie si chaque offre de prix n'excède pas le montant disponible dans le compte en banque du participant requérant.

D'un autre côté, chaque participant possédant des ressources supplémentaires et décidé à en vendre va proposer la quantité qu'il lui reste en bande passante.

Il n'y a pas de vente unitaire de ces lots de bande passante offerte, car ceci ajouterait inutilement de la complexité. En effet, on ne peut distinguer de valeur unitaire entre les offres de bande passante de chaque participant. À la place, l'ensemble de ces lots est globalisé dans un unique lot divisible, qui sera mis en vente.

4.3.2 Calculs

Notre mécanisme doit, durant l'intervalle de temps donné τ_C, calculer puis envoyer publiquement le meilleur sous-ensemble k tel que (1) $g(k, \mathscr{W})$ maximise le prix de la vente de l'ensemble de la bande passante offerte S parmi tous les sous-ensembles considérés durant τ_C, et (2) $\sum_{j \in k} r_j \leq S$. Comme le problème est NP-ardu, le calcul devra se faire à l'aide d'une méthode approximative afin de respecter l'intervalle de temps maximal τ_C.

L'algorithme $k(\mathscr{R}, \mathscr{W}, S)$ doit retourner le meilleur sous-ensemble $\mathscr{R}^* \subseteq \mathscr{R}$ tel que :

$$\begin{aligned} \arg\max_{\mathscr{R}^* \subseteq \mathscr{R}} g(\mathscr{R}^*, \mathscr{W}) &= \sum_{i \in \mathscr{R}^*} b_i(r_i) \\ \text{t.q.} \sum_{i \in \mathscr{R}^*} r_i &\leq S \end{aligned} \tag{4.1}$$

Chaque participant $i \in \mathscr{N}$ peut (et nous avons vu précédemment qu'il a tout intérêt à le faire) effectuer, durant le même intervalle de temps donné τ_C, calculer puis envoyer publiquement le meilleur sous-ensemble k_i. Il devra lui aussi trouver une méthode approximative (de son choix).

Nous devons résoudre un problème d'optimisation où les contraintes sont linéaires, la fonction d'optimisation est monotone et croissante, et dont les variables sont strictement entières. En effet, la condition de satisfaction d'un VPN impose que sa demande en ressources supplémentaires soit entièrement satisfaite.

La résolution de ce type de problème d'optimisation se fait généralement par Programmation Linéaire en Nombres Entiers (PLNE)[7]. Malheureusement, la résolution via la PLNE est NP-

ardu[7]. Bien que le nombre limité de VPN permette de résoudre ce problème par PLNE en un temps raisonnable, il est impossible de l'effectuer en temps réel.

Comme nous l'avons vu dans le mécanisme 2, chaque VPN qui demande plus de ressources va, le plus rapidement possible, tenter de trouver le meilleur vecteur possible, en s'incluant, permettant un prix de vente le plus élevé. Les vendeurs le feront également, sans s'inclure naturellement.

Chacun des VPN peut utiliser un algorithme différent, y compris un de type entièrement pseudo-aléatoire. Toutefois, notre implémentation ne propose qu'un unique algorithme pour tous. Bien que simple, nous pensons qu'il est suffisamment efficace dans notre cas. En condition réelle, les opérateurs VPN auront de toute façon intérêt à en trouver un meilleur et probablement différent les uns des autres.

Notre heuristique repose entièrement sur une relaxation continue de notre problème d'optimisation combinatoire. Pour cela, nous utilisons une résolution par Programmation Linéaire (PL), qui possède une complexité de seulement $O(n^2)$[8] en utilisant l'algorithme *Branch and cut*. À partir de la solution réelle (c'est-à-dire $k_i \in \mathbb{R}$) issue du PL, nous allons la relaxer jusqu'à obtenir une solution entière.

Algorithme 3 : Discrétisation des résultats issues de la relaxation continue

Entrées : Vecteur des scores des offres achats k_c, Somme de la bande passante offerte S
Sorties : Proposition d'allocation k

1. Soit m l'indice de la valeur maximale du vecteur k_c, on ajoute le VPN d'indice $i = m$ à l'ensemble k.
2. On calcule la somme de la bande passante demandée par tous les éléments de k, $R_k = \sum_{i \in k} r_i$.
3. Si $R_k < S$, on retire le score du VPN m du vecteur k_c et on retourne à l'étape 1, sinon on retire le VPN d'indice $i = m$ de l'ensemble k.

Pour cela, nous utilisons une heuristique vorace, présentée dans l'algorithme 3. Alors que toutes les valeurs $k_i \in \mathbb{R}^+$, nous prenons la plus grande valeur pour la fixer à 1 et vérifions la bande passante restante. Nous répétons cette démarche jusqu'à qu'il ne reste plus de bande passante disponible. L'optimisation est une succession de maxima locaux et n'est pas une

optimisation globale, mais nous pensons que ce cas d'étude reste une optimisation discrète globale approchante.

Exemple 4.3.1 *(Discrétisation des résultats issues de la relaxation continue)*
Soit $S = 14$MO de bande passante à partager entre 4 VPN dont les offres $w_i = (r_i, b_i(r_i))$ sont :

$VPN_1 : (3,3)$ $VPN_2 : (7,7)$ $VPN_3 : (5,5)$ $VPN_4 : (3,3)$

La sortie de notre algorithme *Branch and cut* nous donne les scores suivants pour chaque VPN :

$VPN_1 : 0.7246$ $VPN_2 : 0.7037$ $VPN_3 : 0.9453$ $VPN_4 : 0.7246$

a. On initialise l'ensemble $k = \emptyset$.
b. On sélectionne le VPN ayant le plus grand score, ici VPN_3, puis on l'ajoute à k. $k = \{VPN_3\}$.
c. On vérifie que la somme des demandes en bande passante des éléments de k ne dépasse pas S, $R_k = \sum_{i \in k} r_i = 5 < 14$.
d. Comme $R_k < S$, on recherche le prochain score maximum, ici VPN_4, puis on l'ajoute k. $k = \{VPN_3, VPN_4\}$.
e. On vérifie que la somme des demandes en bande passante des éléments de k ne dépasse pas S, $R_k = \sum_{i \in k} r_i = 5 + 3 = 8 < 14$.
f. Comme $R_k < S$, on recherche le prochain score maximum, ici VPN_1, puis on l'ajoute k. $k = \{VPN_3, VPN_4, VPN_1\}$.
g. On vérifie que la somme des demandes en bande passante des éléments de k ne dépasse pas S, $R_k = \sum_{i \in k} r_i = 5 + 3 + 3 = 11 < 14$.
h. Comme $R_k < S$, on recherche le prochain score maximum, ici VPN_2, puis on l'ajoute k. $k = \{VPN_3, VPN_4, VPN_1, VPN_2\}$.
i. On vérifie que la somme des demandes en bande passante des éléments de k ne dépasse pas S, $R_k = \sum_{i \in k} r_i = 5 + 3 + 3 = 18 \not< 14$.
j. Comme $R_k > S$, on retire de k le dernier élément rajouté, VPN_2. $k = \{VPN_3, VPN_4, VPN_1\}$.

Comme la stratégie dominante est de s'inclure à tout prix dans le vecteur k_i (lemme 7), chaque VPN nécessitant des ressources réduit la bande passante disponible avec son propre besoin

puis calcule le meilleur vecteur k_i avec l'heuristique précédente. Il s'ajoute ensuite au vecteur avant de le publier. Le mécanisme et les autres VPN ne font naturellement pas cette étape.

Le calcul des $k_j(w^{-i})$ se fait de la même façon, en excluant i des possibilités lors de l'application de notre heuristique. Ces calculs ne sont font uniquement après la sélection telle que vu dans la section suivante.

4.3.3 Sélection

Une fois que toutes les solutions candidates ont été calculées par les divers participants, il faut sélectionner la solution maximisant la somme des offres. Ceci se fait rapidement par le calcul du gain de chaque vecteur. Le gain de chaque vecteur k_i est représenté par la somme des offres des gagnants (c'est à dire $\forall j, j \in k_i$). Ainsi le gain $G(k) = \sum k_i.b_i(r_i)$. Nous élisons alors le vecteur $k^* = \arg_{k \in \mathscr{K}} \max G(k)$.

4.3.4 Coût total et paiements

Le coût total est la somme des coûts de chaque VPN. Le prix que chaque VPN doit payer est $p_i(w) = g(w^{-i}) - g(w) + w_i \geq 0$. La banque centrale, c'est-à-dire dans notre cas le mécanisme, récupère automatiquement les paiements des comptes des gagnants. Il stocke l'ensemble dans un compte temporaire.

La distribution des paiements se fait ensuite proportionnellement à l'offre des $i \in \mathscr{S}$, c'est-à-dire $p_i = -(s_i/S) \leq 0$. Le prix est négatif, car on reçoit ce paiement.

La bande passante allouée à chaque VPN est modifiée en fonction de la sélection. Ces modifications ne seront actives que pour cette itération et chaque SLA redeviendra à sa valeur d'origine à la prochaine itération.

4.4 Résultats

Après la fin de la simulation, nous possédons toute l'information pour effectuer une analyse de la qualité de service. Nous analysons toutes les bandes passantes consommées ainsi que les bandes passantes nécessaires, et nous pouvons en déduire la satisfaction de chaque VPN.

Nous déclarons que deux types possibles de QoS : `MauvaisQoS` et `BonQoS`. Si la bande passante nécessaire d'un VPN est en dessous de la bande passante allouée, alors le VPN a un `BonQoS`, sinon un `MauvaisQoS`. Nous verrons en détail ceci dans le chapitre 5.

Nous cherchons à voir l'évolution du nombre de participants ayant obtenu les ressources nécessaires afin d'atteindre leur QoS au fil du temps.

CHAPITRE 5

RÉSULTATS ET INTERPRÉTATION

Nous allons analyser ici notre modèle issu de notre mécanisme 2 présenté au chapitre 3. Pour cela, nous avons mis en place une implémentation respectant entièrement le chapitre 4. Notre principal objectif est de montrer comment les VPN coopératifs, qui prêtent volontiers leurs ressources inutilisées, s'en sortent par rapport aux VPN égoïstes qui ne prêtent jamais les leurs.

Pour cela, nous utilisons une mesure simple de qualité de service (QoS), dont la valeur n'a que deux états. Le premier, soit 1, indique que le VPN possède assez de ressources pour combler tous ses besoins au court de la prochaine itération de la simulation. Le deuxième état, soit 0, indique qu'il n'a pas assez de ressources pour répondre à la demande de trafic de son réseau. Ainsi, la mesure instantanée de QoS est prise à chaque unité de temps. De ce fait, la mesure globale de QoS sur un intervalle t unités de temps est simplement la somme des t mesures instantanées. Plus un VPN possède une mesure globale de QoS élevée, moins il a eu de problèmes de qualité de service.

Nos simulations reposent sur un fournisseur de service (ISP) et un réseau de 40 VPN, dont une moitié est composée de VPN coopératifs et l'autre de VPN égoïstes. L'ISP possède une bande passante de 600MO qu'il a entièrement vendue à ses 40 clients. Pour chaque client, il existe une entente de service (Service Level Agreements ou SLA) ou contrat qui détermine les clauses de vente, tels que prix, bande passante, etc. via des SLA.

La valeur de la bande passante allouée pour chaque VPN est déterminée de façon pseudo-aléatoire. De plus, la somme des bandes passantes allouées est égale à la capacité totale de l'ISP, soit 600MO, afin d'utiliser toute la bande passante possible offerte par ce dernier.

Les prix des offres sont définis pour chaque unité de bande passante entre 0.8$ et 1.2$ dans toutes les simulations. Chaque VPN possède au début de la simulation 40$ virtuels. La somme

des montants disponibles lors de la simulation est donc de 1600$ et n'évoluera jamais, car il n'y a pas de création de richesse.

La durée totale de chaque simulation dure 5000 itérations et à chaque itération, 20 VPN seront en surcharge.

Les différents paramètres globaux des simulations peuvent être résumés dans le tableau 5.1.

Tableau 5.1 Ensemble des paramètres communs aux simulations

Durée (Nombre d'itérations)	5000
Bande passante ISP (MO)	600
Nombre de VPN	40
… dont coopératifs	20
… dont égoïstes	20
Nombre de VPN en surcharge	20
Monnaie initiale par VPN	40
Prix par unité minimum	0.8
Prix par unité maximum	1.2

5.1 Simulation avec demandes équilibrées

5.1.1 Context

L'ensemble des SLA initialisés au départ de cette simulation est disponible dans le tableau 5.2, et correspond à des valeurs entre 0 et 200% de la moyenne de la capacité totale de l'ISP. La somme correspond bien à la bande passante mise à disposition par l'ISP.

Pour chaque itération, les VPN qui ne seront pas en surcharge utiliseront leur bande passante entre une valeur comprise entre 50 et 100% de leur SLA. Les 20 VPN sélectionnés au hasard comme ayant besoin de plus de ressources tenteront d'utiliser quant à eux entre 100 et 150% de leur SLA. Bien que tout à fait arbitraires, nous considérons que ces valeurs-là sont somme toute assez réalistes et permettront de bien tester notre mécanisme.

Tableau 5.2 Ensemble des SLA initialisés

VPN	1	2	3	4	5	6	7	8	9	10
SLA	13,0	11,3	22,8	23,7	5,4	14,5	13,2	19,2	21,1	22,5
VPN	11	12	13	14	15	16	17	18	19	20
SLA	8,1	20,2	19,5	4,7	3,4	14,8	28,6	10,0	17,4	6,5
VPN	21	22	23	24	25	26	27	28	29	30
SLA	22,4	7,5	15,0	20,8	26,5	28,6	16,2	4,0	4,3	7,5
VPN	31	32	33	34	35	36	37	38	39	40
SLA	25,0	7,5	24,3	7,1	27,7	10,3	5,7	7,4	18,3	14,0

Les différents paramètres spécifiques de cette simulation peuvent être résumés dans le tableau 5.3.

Tableau 5.3 Ensemble des paramètres de la simulation avec demandes équilibrées

Durée (Nombre d'itérations)	5000
Bande passante ISP (MO)	600
Nombre de VPN	40
… dont coopératifs	20
… dont égoïstes	20
Nombre de VPN en surcharge	20
Monnaie initiale par VPN	40
Prix par unité minimum	0.8
Prix par unité maximum	1.2
Utilisation sans surcharge du SLA	$[50\%; 100\%]$
Utilisation avec surcharge du SLA	$[100\%; 150\%]$

5.1.2 QoS

La figure 5.1 présente les courbes cumulatives des QoS de chaque catégorie de VPN en fonction du temps. Sa génération se fait de la façon suivante : à chaque itération, si après le processus d'échange un VPN a suffisamment de ressources pour répondre à ses besoins, il obtient une mesure de QoS instantanée de 1, sinon cette mesure est simplement de 0. Sur le graphique, chaque mesure instantanée de QoS gagné équivaut à une unité. Nous procédons ensuite à la

somme de ces unités pour chaque catégorie puis, pour chaque intervalle, nous procédons à une somme cumulative (c'est-à-dire que nous reprenons toujours la valeur précédente à laquelle nous ajoutons la valeur courante de l'intervalle). Les courbes sont donc monotones et croissantes.

Nous pouvons observer que globalement les VPN qui coopèrent ont plus de QoS que les VPN égoïstes. L'écart est ici d'environ 25%. Bien que déjà élevé, cet écart ne permet pas de bien comprendre la portée du mécanisme. En effet, il ne faut pas oublier que malgré un nombre élevé de la mesure globale de QoS, nombreux proviennent simplement du fait que certains VPN n'étaient pas en surcharge avant même le processus de partage de ressources.

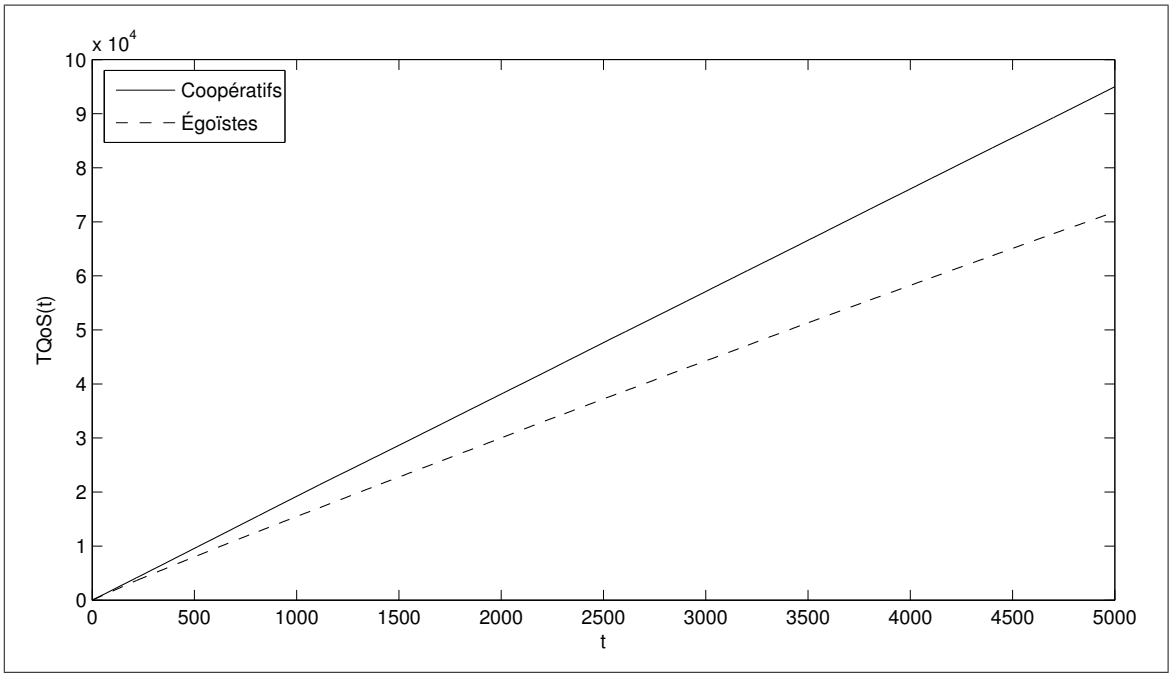

Figure 5.1 Somme totale des QoS

En partant de cette idée, nous avons produit un nouveau graphique. La figure 5.2 présente le même cumul de QoS que la figure précédente, mais nous avons supprimé toutes les mesures instantanées de QoS des VPN qui n'étaient pas surchargés avant le processus d'échange. N'affichant plus que les mesures des VPN en surcharge avec le partage de ressources, nous pouvons observer que les VPN égoïstes obtiennent 3 fois moins de bons QoS par rapport aux

VPN coopératifs. On peut remarquer que la croissance de la mesure globale de QoS des VPN égoïstes. De plus, l'écart entre les deux catégories augmente à chaque intervalle. En fait, à terme la courbe des VPN égoïstes atteint une asymptote lorsque leur 40$ sont épuisés.

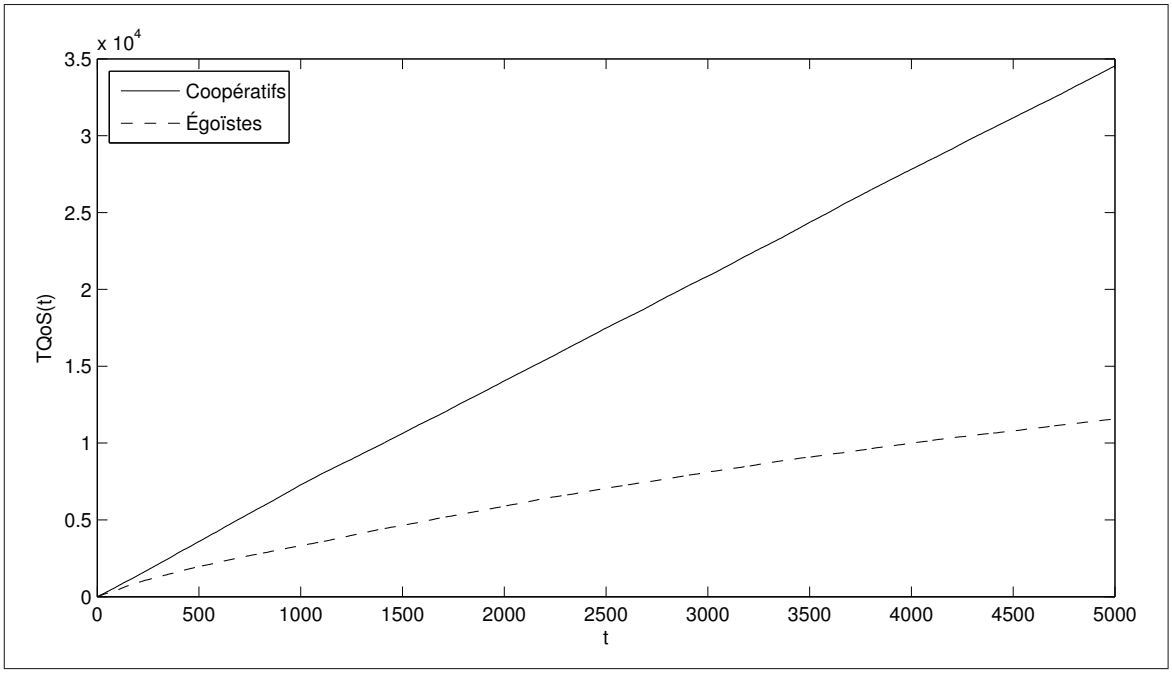

Figure 5.2 Somme totale des QoS si surchargé

5.1.3 Comptes

La figure 5.3 montre la répartition de l'argent virtuel entre les deux catégories. Chaque VPN commençant avec le même montant, et étant donné qu'il y a autant de VPN dans chaque catégorie, on peut remarquer que la somme d'argent totale de chacune est égale.

Alors que la courbe des VPN coopératifs ne cesse d'augmenter en tendant vers la somme totale d'argent virtuel, celle des VPN égoïstes tend vers zéro. On observe en fait un échange d'argent continu et unidirectionnel entre les deux catégories de VPN.

Les VPN coopératifs gagnent de l'argent en vendant leurs ressources inutilisées et ils profitent de cet argent amassé pour en acheter le cas échéant aux autres VPN coopératifs. Les VPN égoïstes ne partagent pas de ressources, et de ce fait ne gagnent point d'argent. Ils profitent de

l'argent initial pour en acheter aux autres VPN coopératifs ; il est néanmoins clair que ceci ne dure pas longtemps.

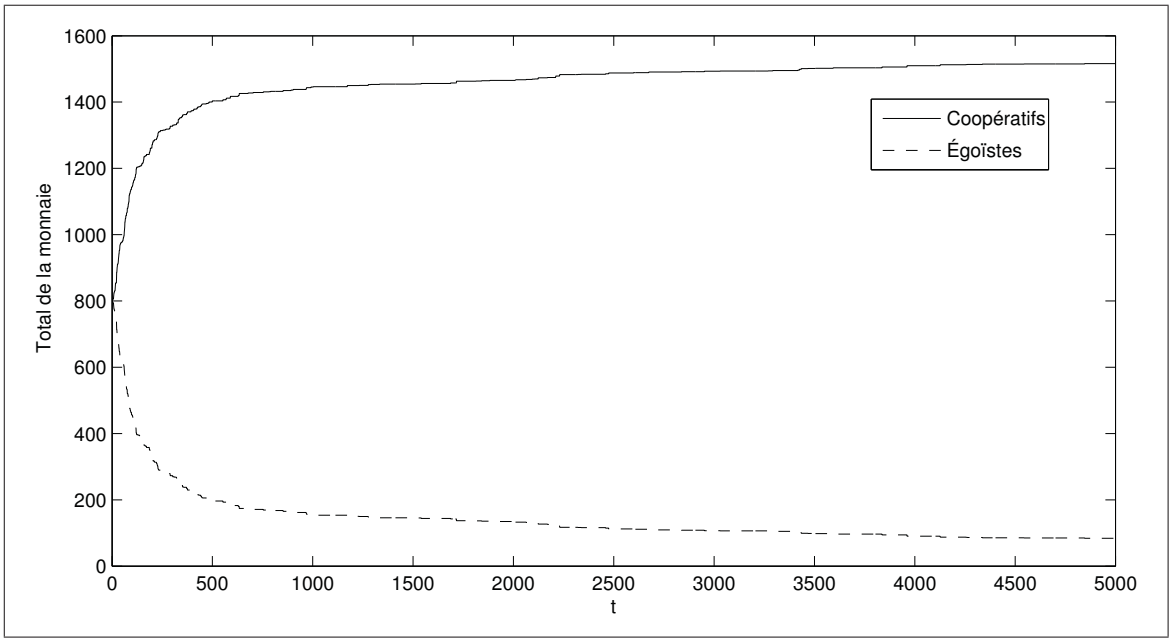

Figure 5.3 Répartition de l'argent virtuel

Alors que les VPN égoïstes n'ont quasiment plus d'argent virtuel à leur disposition, il est curieux de voir tout de même leur mesure globale de QoS de continuer d'augmenter doucement. Pour répondre à cette question, nous avons produit le graphique de la figure 5.4. Cette figure est semblable à la figure 5.2, mais comporte une contrainte supplémentaire. En plus de ne considérer que les VPN en manque de ressources avant le lancement du processus d'échange, il ne prend en compte que les demandes en ressources supérieures à 25%. Cela nous permet d'éliminer toutes les petites demandes. En effet, plus les demandes en ressources sont petites, moins elles coûtent chères, le prix étant directement proportionnel au montant de bande passante achetée. Ainsi, nous éliminons du graphique les demandes les moins onéreuses. Le taux de 25% a été choisi de façon tout à fait arbitraire et n'est ici que pour différencier les « petites » demandes aux « grandes » demandes en ressources.

Cette fois-ci, on observe sur la figure 5.4 qu'une asymptote horizontale commence à se dessiner beaucoup plus rapidement sur la courbe de la mesure globale de QoS des VPN égoïstes.

Ceci démontre que les QoS qui apparaissent encore sur la courbe des VPN égoïstes de la figure 5.2 sont donc bel et bien pour des petites demandes, ce qui correspond à leur faible somme d'argent disponible. Une fois cette somme à zéro, aucune mesure instantanée de QoS ne pourra être accordée en cas de surcharge. En effet, en simulant suffisamment longtemps, on observe bien une asymptote horizontale.

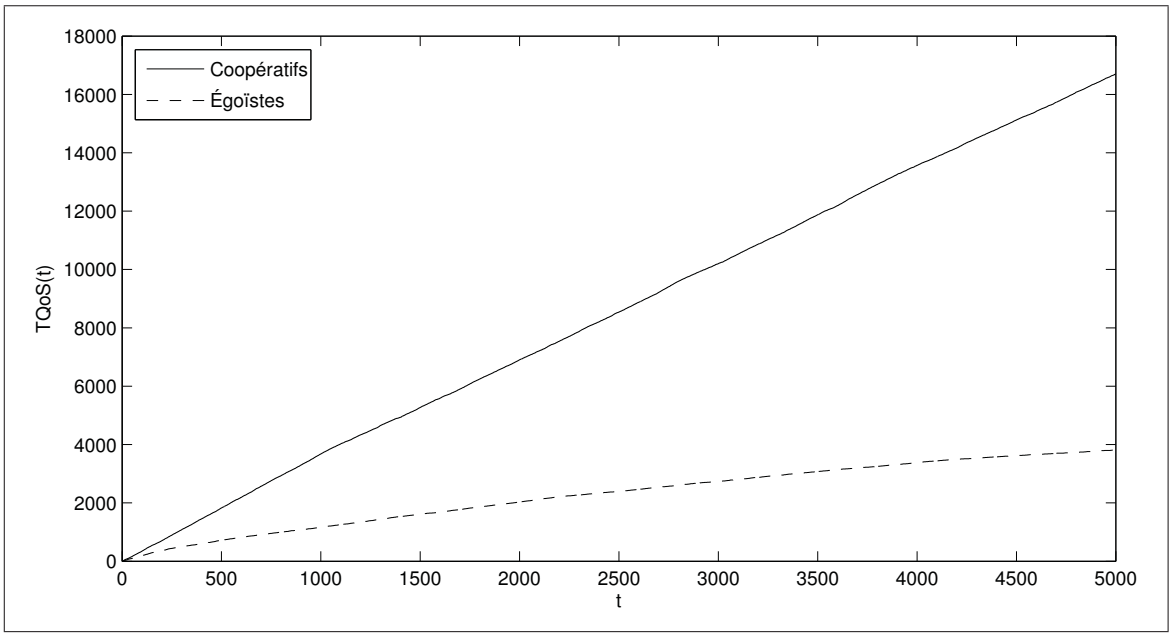

Figure 5.4 Somme totale des QoS si surchargé de plus de 25%

5.2 Simulations avec demandes déséquilibrées

5.2.1 Context

La simulation précédente n'est pas de facto réaliste au sujet des demandes en ressources. En effet, tous les VPN, s'ils sont dans un état de surcharge au niveau de la bande passante, ont une demande de ressources comprise entre 100 et 150% de leur SLA respective. Bien que chaque SLA est unique, car générée de façon aléatoire, les valeurs de bande passante sont toutefois du même ordre de grandeur.

Nous proposons donc deux nouvelles simulations, possédant des demandes en bande passante déséquilibrées. Pour cela, nous utiliserons deux algorithmes de calcul de la demande en bande

passante lors d'une surcharge. Le premier algorithme est identique à celui de la simulation précédente, c'est-à-dire qu'il génère une valeur de demande en ressources comprise entre 100 et 150% de la SLA du VPN en état de surcharge. Le deuxième algorithme se différencie par le fait qu'il rendra la surcharge beaucoup plus importante. En effet, il génère une valeur de demande en ressources comprise entre 100 et 300% de la SLA du VPN en état de surcharge. Nous avons donc maintenant deux classes de surcharge. La première, issue du premier algorithme, de petite ampleur et que nous nommerons OL_{min}. La deuxième, issue du second algorithme, de grande ampleur et que nous nommerons OL_{max}.

Les deux nouvelles simulations sont identiques à la première si ce n'est qu'elles utilisent les deux nouveaux algorithmes de génération de surcharge OL_{min} et OL_{max}. De plus, ces deux simulations se distinguent entre elles par la proportion d'utilisation des deux algorithmes.

5.2.2 Simulation avec demandes déséquilibrées 5M-15m

Comme pour la première simulation, sur 40 VPN, 20 seront en état de surcharge ; 5 utiliseront l'algorithme OL_{max} alors que 15 utiliseront l'algorithme OL_{min}. L'ensemble des paramètres de la simulation est résumé dans le tableau 5.4.

Comme pour la première simulation, nous produisons 4 graphiques faisant état du cumul de QoS ainsi que de la répartition de l'argent virtuel.

La figure 5.5 est le graphique montrant la somme totale des QoS pour chaque catégorie. On peut tout de suite constater que, pour les deux catégories, le cumul total a diminué. Ceci ne démontre aucunement que notre algorithme souffre de quelques maux lorsque la demande est déséquilibrée, mais bel et bien qu'il possède une efficacité équivalente. En effet, la demande en ressources à considérablement augmentée alors que l'offre n'a pas été modifiée. Il est donc évident que moins de VPN surchargés auront accès à des ressources supplémentaires. À ceci s'ajoute que la différence entre le cumul des QoS entre les deux catégories a augmenté, montrant alors qu'en cas de demande importante, les VPN égoïstes ont encore plus de difficulté à s'approvisionner.

Tableau 5.4 Ensemble des paramètres de la simulation 5M-15m

Durée (Nombre d'itérations)	5000
Bande passante ISP (MO)	600
Nombre de VPN	40
… dont coopératifs	20
… dont égoïstes	20
Nombre de VPN en surcharge	20
Monnaie initiale par VPN	40
Prix par unité minimum	0.8
Prix par unité maximum	1.2
Utilisation sans surcharge du SLA	$[50\%;100\%]$
Utilisation avec surcharge du SLA	
… avec 75% de chance	$[100\%;150\%]$
… avec 25% de chance	$[100\%;300\%]$

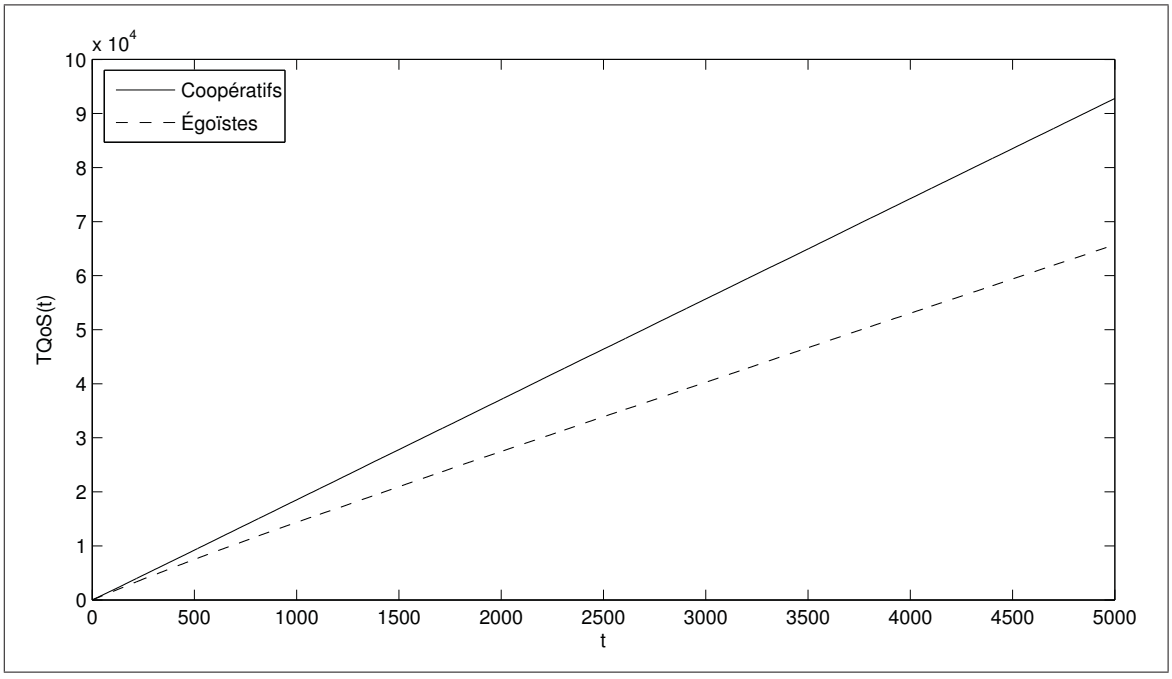

Figure 5.5 Somme totale des QoS - 5M-15m

Cette observation est d'autant plus évidente sur la figure 5.6, montrant le graphique de la somme totale des QoS pour les VPN surchargés pour chacune des catégories, c'est-à-dire

comportant uniquement les mesures de QoS des VPN en état de surcharge. On peut alors remarquer que les VPN coopératifs obtiennent 7 fois plus de mesure de QoS que les VPN égoïstes.

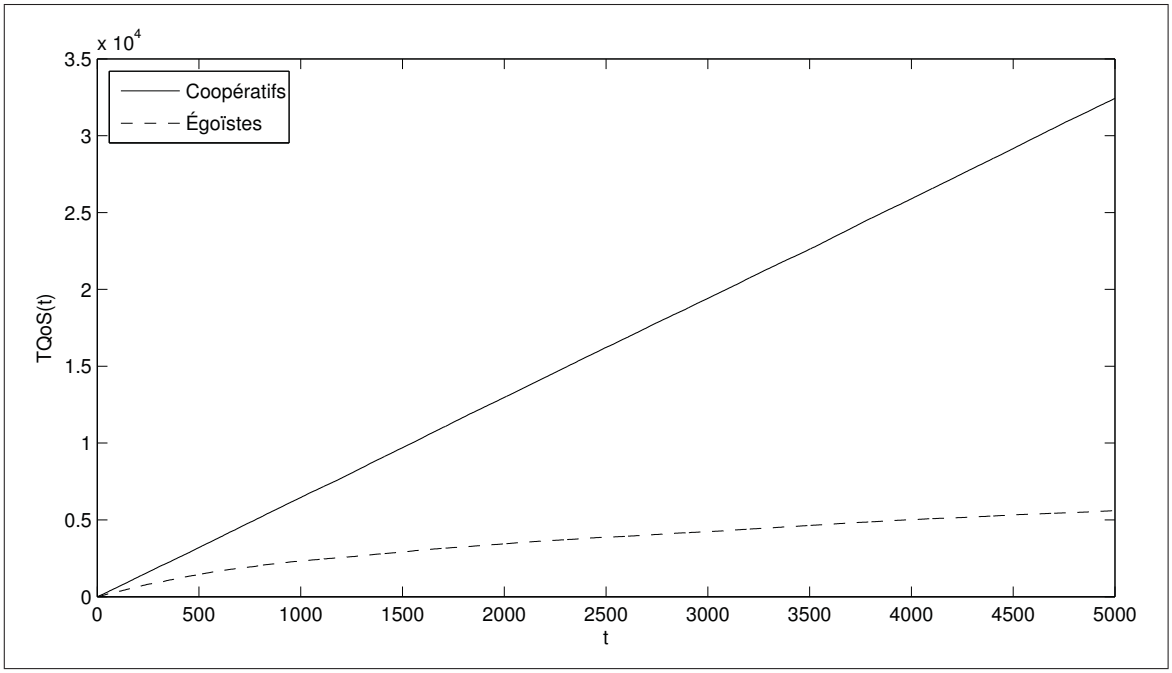

Figure 5.6 Somme totale des QoS si surchargé - 5M-15m

La figure 5.7 montre que très rapidement, les VPN égoïstes n'obtiennent plus de mesure de QoS pour des demandes en ressources supérieures d'au moins 25% de leur SLA, c'est-à-dire qu'ils ne sont plus capables de faire des offres assez élevées pour être sélectionnés par la fonction de sélection cherchant à maximiser la somme des offres.

Cette dernière information est confirmée par la figure 5.8. On observe qu'après à peine 1000 itérations, les VPN égoïstes n'ont quasiment plus aucune ressource. L'ensemble de la monnaie virtuelle a été complètement capitalisé par les VPN coopératifs.

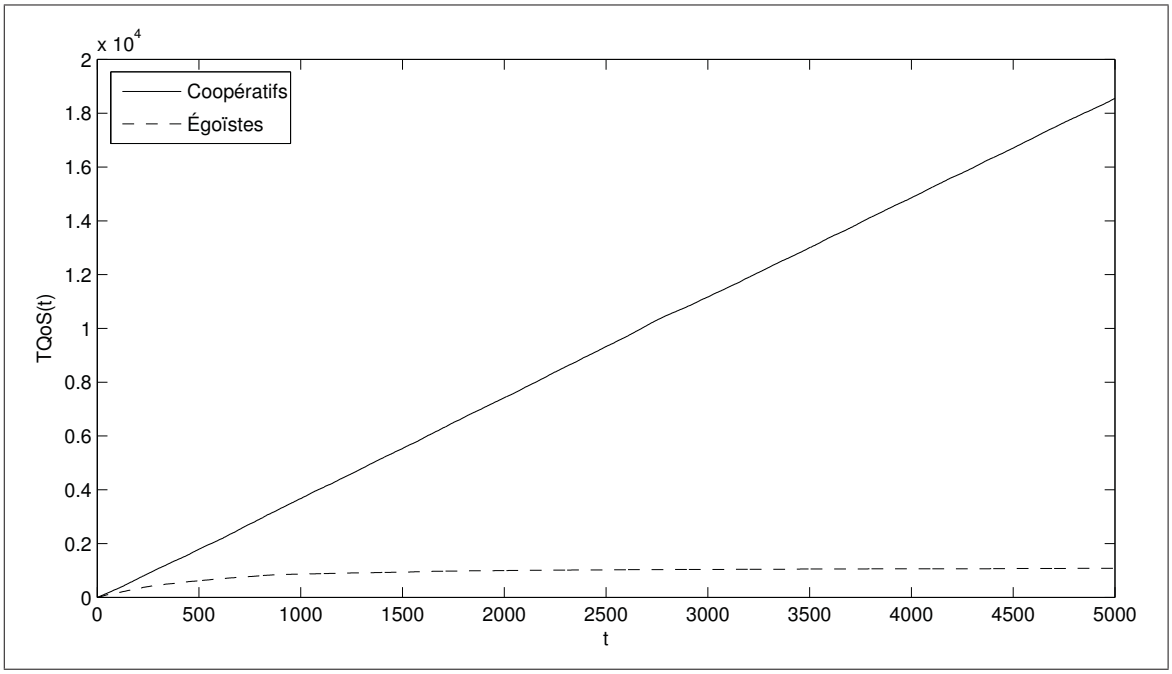

Figure 5.7 Somme totale des QoS si surchargé de plus de 25% - 5M-15m

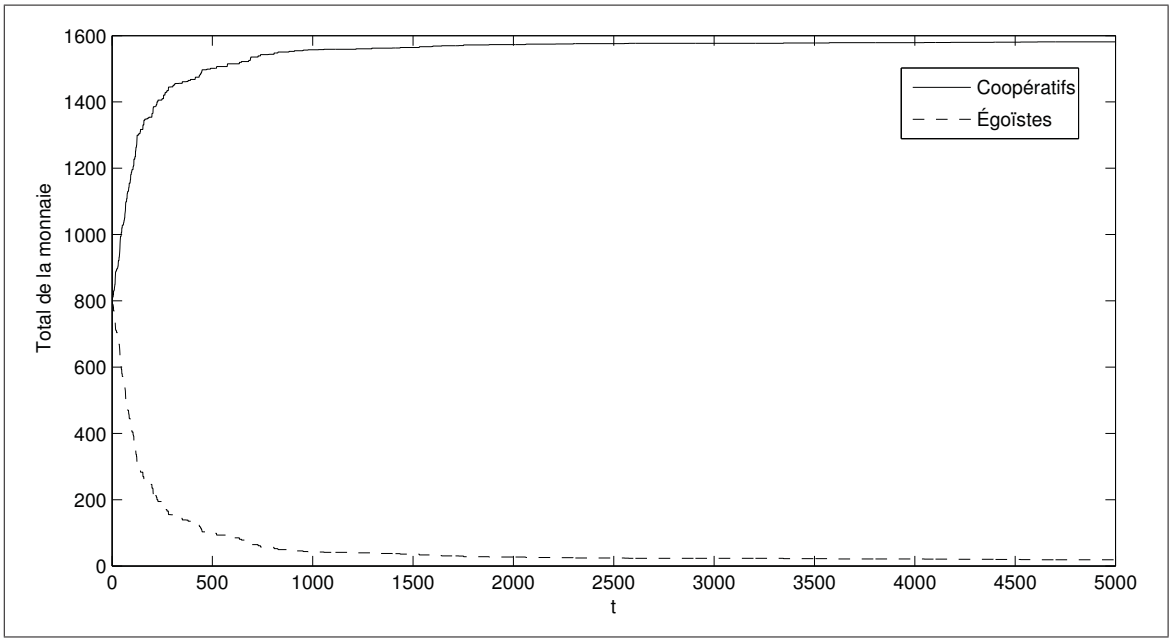

Figure 5.8 Répartition de l'argent virtuel - 5M-15m

5.2.3 Simulation avec demandes déséquilibrées 15M-5m

La troisième simulation est simplement une variante de la deuxième. Sur 40 VPN, 20 seront en état de surcharge ; 15 utiliseront l'algorithme OL_{max} alors que 5 utiliseront l'algorithme OL_{min}. L'ensemble des paramètres de la simulation est résumé dans le tableau 5.5.

Tableau 5.5 Ensemble des paramètres de la simulation avec demandes déséquilibrées 15M-5m

Durée (Nombre d'itérations)	5000
Bande passante ISP (MO)	600
Nombre de VPN	40
… dont coopératifs	20
… dont égoïstes	20
Nombre de VPN en surcharge	20
Monnaie initiale par VPN	40
Prix par unité minimum	0.8
Prix par unité maximum	1.2
Utilisation sans surcharge du SLA	$[50\%;100\%]$
Utilisation avec surcharge du SLA … avec 25% de chance … avec 75% de chance	 $[100\%;150\%]$ $[100\%;300\%]$

Comme la simulation précédente, la bande passante offerte est encore une fois limitée, mais l'inversion du nombre d'utilisations des OL_{max} et OL_{min} comme algorithme de surcharge amplifie l'effet de façon significative. La figure 5.9 présente tout à fait les mêmes caractéristiques que la figure 5.6, issue de la deuxième simulation, mais de façon plus abrupte.

Rien de surprenant au niveau de la répartition de l'argent virtuel. La figue 5.10 montre que celle-ci est encore une fois à l'avantage des VPN coopératifs. En 500 itérations, la quasi-totalité de l'argent virtuel se trouve en possession des VPN coopératifs. On peut facilement statuer que plus la demande est importante face à l'offre, plus les VPN coopératifs s'en sortent face aux VPN égoïstes. Bien sûr, moins d'offres ou plus de demandes diminuent le nombre de mesures de QoS, car même en étant coopératif, il se peut qu'il n'y ait tout simplement pas assez de ressources disponibles.

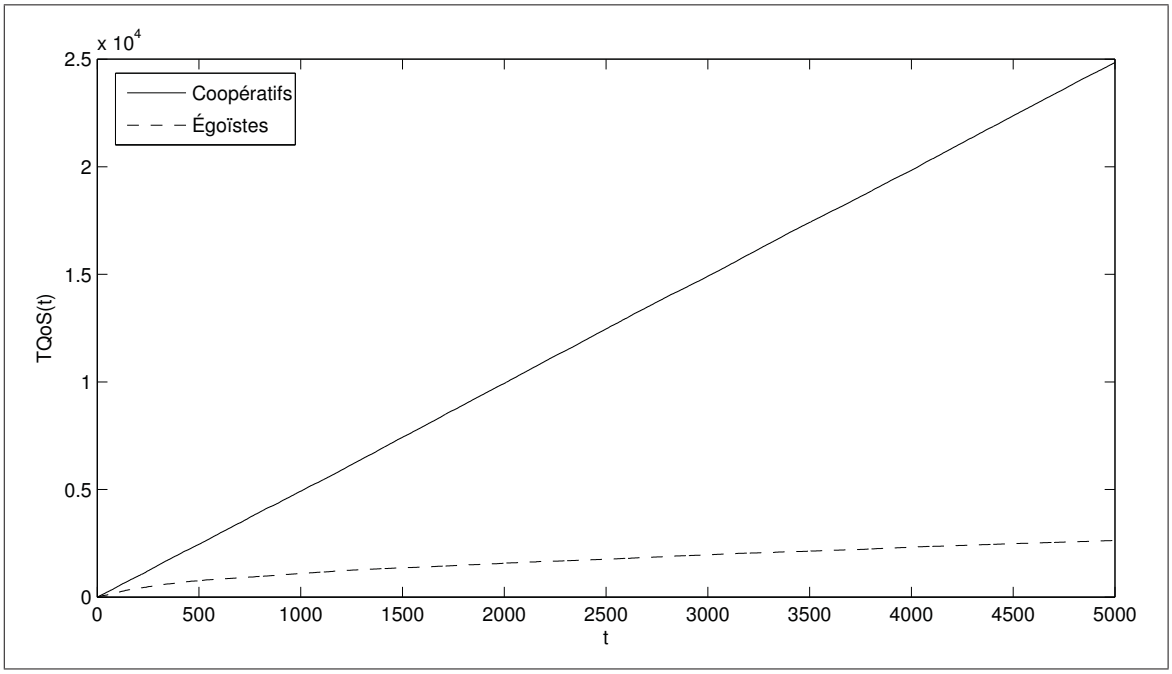

Figure 5.9 Somme totale des QoS si surchargé - 15M-5m

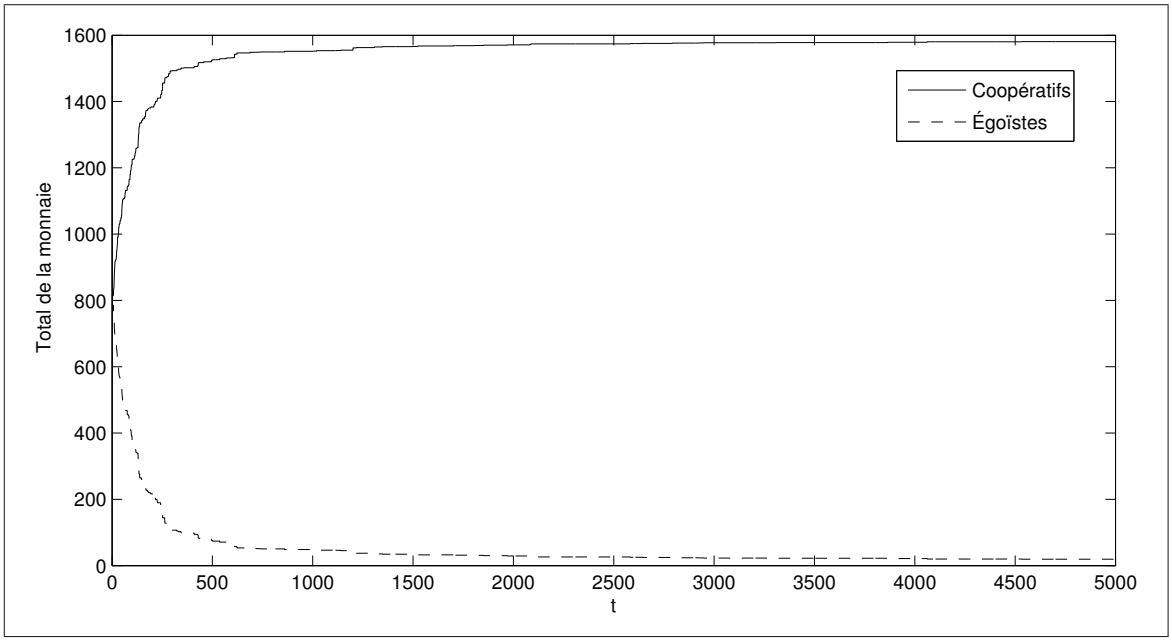

Figure 5.10 Répartition de l'argent virtuel - 15M-5m

CONCLUSION

L'utilisation d'un VPN dans les entreprises est aujourd'hui très répandue. Elle permet d'établir une liaison sécuritaire et fonctionnelle entre divers lieux physiques d'une entreprise et de les réunir sous un même réseau virtuel. Une communication sécurisée est alors possible entre les succursales ainsi que les employés travaillant à l'extérieur.

Les fournisseurs d'accès à Internet (ISP) peuvent jouer un rôle clé dans l'établissement de tel VPN. En effet, faisant partie intégrante de l'architecture d'accès à Internet, toutes les données transitant sur Internet d'une entreprise passent forcément par lui. Il a donc une position stratégique.

Cette position lui permet de vendre, avec une certaine garantie, de la bande passante qui sera utilisée par le VPN pour permettre de faire transiter les données. Un contrat de vente, le SLA, permet d'établir le montant de celle-ci ainsi que la qualité de service (QoS) exigée.

Afin d'optimiser au plus près ses bénéfices, les ISP vendent plus de ressources qu'ils n'en possèdent réellement, en espérant que leurs clients n'utiliseront pas tous ensemble leur bande passante réservée dans leur SLA.

Dans le but d'optimiser d'une part les bénéfices des ISP et d'autre part d'améliorer la qualité de services offerte à leurs clients, nous avons proposé une solution inédite qui tire avantage d'une science en pleine effervescence, les mécanismes d'incitation.

Un mécanisme d'incitation est un ensemble de règles, telles les règles d'un jeu, où l'ensemble des participants est obligé d'adhérer. Ces règles sont conçues afin de mettre en relief un certain comportement. Ce comportement, appelé stratégie dominante, permet à la fois d'avoir, pour quiconque le respectant, un bien-être maximal en fonction des conditions. De plus, plus le nombre de participants l'utilise, plus le bien-être général est grand.

Face à n'importe quel algorithme centralisé, notre solution est dans un premier temps lieu plus efficace pour un ISP louant des ressources, et dans un deuxième temps distribué entre

les clients de celui, laissant à chacun le pouvoir et l'opportunité de gérer le plus finement possible ses offres et demandes en ressource (bande passante). Ceci est un avantage certain face à tout autre modèle de partage. De plus, il permet à chacun d'améliorer son propre bien-être en laissant la possibilité d'effectuer ses propres calculs de la fonction d'évaluation de l'algorithme d'allocation.

Nos simulations portaient principalement sur deux catégories de VPN. La première correspondait aux VPN coopératifs alors que la deuxième était composée exclusivement de VPN égoïstes. Nos simulations étaient destinées à démontrer que la coopération entre les VPN est utile afin d'améliorer son propre bien-être, c'est-à-dire avoir le meilleur QoS possible.

Les résultats de ces simulations ont mis clairement en évidence l'efficacité de notre modèle. Dans tous les cas, le fait d'être coopératif améliore le nombre de mesures de bon QoS en tout temps. De plus, le fait d'avoir beaucoup de demande face à peu d'offres affecte considérablement les VPN égoïstes alors que les VPN coopératifs s'en sortent bien mieux en conservant tout de même un bon taux de mesure de QoS.

Au final, nous avons réussi à proposer un modèle aux propriétés très intéressantes pour l'ISP et les opérateurs de VPN. L'allocation des ressources étant optimisée, l'ISP peut vendre plus de ressources tout en améliorant le service à ses clients. Les différents calculs d'allocation n'ont plus lieux d'être, car ceux-ci sont relégués aux VPN de façon distribuée. Les opérateurs de VPN peuvent gérer eux-mêmes les ressources dont ils disposent et ainsi peuvent prévoir facilement leurs futurs besoins. Chacun peut de plus proposer sa propre fonction d'allocation. Même en se privilégiant, le VPN ne peut qu'améliorer la qualité de service pour l'ensemble des VPN.

Ainsi, notre modèle convient à tous, en proposant énormément d'avantages face aux autres systèmes de gestion de ressources utilisés aujourd'hui.

Notre approche ouvre plusieurs perspectives de recherche future. En effet, comme notre modèle propose une nouvelle solution, son étude n'a pas été exhaustive. Plusieurs champs restent donc à être étudiés.

Premièrement, nous avons démontré que la pertinence de notre modèle face à des opérateurs de VPN solitaire. Il serait intéressant de démontrer que notre modèle reste pertinent face à des opérateurs de VPN pouvant former des collusions afin d'améliorer leurs situations ou de détériorer celles d'autres participants.

Un deuxième axe de recherche serait d'améliorer la fonction de paiement. Nous avons utiliser la fonction de paiement de VCG classique, mais il en existe d'autres qui possèdent moins d'inconvénients, par exemple en utilisant la valeur de Shapley qui permet une redistribution des paiements proche d'un système à budget équilibré[3].

Enfin, il serait pertinent d'établir des tests réels de nos simulations, afin de valider de manière concrète notre modèle. Bien que sensiblement proches de la réalité, beaucoup de paramètres de nos simulations ont été choisis de manière empirique.

RÉFÉRENCES BIBLIOGRAPHIQUES

[1] Nadarajah ASOKAN, Phillipe A. JANSON, Michael STEINER et Michael WAIDNER : The state of the art in electronic payment systems. *Computer*, 30:28–35, 1997.

[2] Gilles BRASSARD, David CHAUM et Claude CRÉPEAU : Minimum disclosure proofs of knowledge. *J. Comput. Syst. Sci.*, 37(2):156–189, 1988.

[3] Ruggiero CAVALLO : Optimal decision-making with minimal waste : strategyproof redistribution of VCG payments. *In Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, AAMAS '06, pages 882–889, New York, NY, USA, 2006. ACM.

[4] Yu CHENG, Ramy FARHA, Myung Sup KIM, Alberto LEON-GARCIA et James WON-KI HONG : A generic architecture for autonomic service and network management. *Computer Communications*, 29(18):3691–3709, 2006.

[5] Edward H. CLARKE : Multipart pricing of public goods. *Public Choice*, 18:19–33, 1971.

[6] Stephano D'AMIANO et Giovanni DI CRESCENZO : Methodology for digital money based on general cryptographic tools. *Proceedings of the EURO-CRYPT'24*, pages 156–170, 1994.

[7] George B. DANTZIG et Mukund N. THAPA : *Linear programming 1 : introduction*. Springer-Verlag New York, Inc., 1997.

[8] Sanjeeb DASH : On the complexity of cutting-plane proofs using split cuts. *Operations Research Letters*, 38(2):109–114, 2010.

[9] Shahar DOBZINSKI et Noam NISAN : Limitations of VCG-based mechanisms. *In STOC '07 : Proceedings of the thirty-ninth annual ACM Symposium on Theory of Computing*, pages 338–344, New York, NY, USA, 2007. ACM.

[10] Robert A. DOYLE et Steve BASKA : History of auctions : From ancient Rome to todays high-tech auctions. *Auctioneer*, Novembre 2002.

[11] Joan FEIGENBAUM et Scott SHENKER : Distributed algorithmic mechanism design : Recent results and future directions. *In Proceedings of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 1–13. ACM Press, 2002.

[12] Theodore GROVES : Incentives in teams. *Econometrica*, 41(4):617–31, Juillet 1973.

[13] Leonid HURWICZ : The design of mechanisms for resource allocation. *American Economic Review*, 63(2):1–30, Mai 1973.

[14] Juncheng JIA, Qian ZHANG, Qin ZHANG et Mingyan LIU : Revenue generation for truthful spectrum auction in dynamic spectrum access. *In MobiHoc '09 : Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, pages 3–12, New York, NY, USA, 2009. ACM.

[15] John F. Nash JR. : Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36(1):48–49, Janvier 1950.

[16] R. M. KARP : Reducibility among combinatorial problems. *In* R. E. MILLER et J. W. THATCHER, éditeurs : *Complexity of Computer Computations*. Plenum Press, 1972.

[17] Mark KLEIN, Gabriel A. MORENO, David C. PARKES, Daniel PLAKOSH, Sven SEUKEN et Kurt WALLNAU : Handling interdependent values in an auction mechanism for bandwidth allocation in tactical data networks. *In Proceedings of the 3rd international workshop on Economics of networked systems*, pages 73–78, New York, NY, USA, 2008. ACM.

[18] Patrick MAILLÉ et Bruno TUFFIN : Why VCG auctions can hardly be applied to the pricing of inter-domain and ad hoc networks. *In Proceedings of the 3rd European next generation internet conference*, pages 36–39. IEEE Computing Society Press, Mai 2007.

[19] Eric S MASKIN et John G RILEY : Optimal auctions with risk averse buyers. *Econometrica*, 52(6):1473–1518, Novembre 1984.

[20] Alfred J. MENEZES, Paul C. van OORSCHOT et Scott A. VANSTONE : *Handbook of Applied Cryptography*. CRC-Press, 1997.

[21] Roger B MYERSON : Mechanism design by an informed principal. *Econometrica*, 51(6):1767–97, Novembre 1983.

[22] Noam NISAN et Amir RONEN : Computationally feasible VCG mechanisms. *In Proceedings of the 2nd ACM conference on Electronic commerce*, EC '00, pages 242–252, New York, NY, USA, 2000. ACM.

[23] Noam NISAN, Tim ROUGHGARDEN, Eva TARDOS et Vijay V. VAZIRANI : *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA, 2007.

[24] Ariel D. PROCACCIA et Moshe TENNENHOLTZ : Approximate mechanism design without money. *In EC '09 : Proceedings of the tenth ACM conference on Electronic*

commerce, pages 177–186, New York, NY, USA, 2009. ACM.

[25] Jorge-Arnulfo QUIANÉ-RUIZ, Philippe LAMARRE, Sylvie CAZALENS et Patrick VALDURIEZ : Managing virtual money for satisfaction and scale up in p2p systems. *In Proceedings of the 2008 international workshop on data management in peer-to-peer systems*, pages 67–74, New York, NY, USA, 2008. ACM.

[26] Ahmad QUTTOUM, Hadi OTROK et Zbigniew DZIONG : ARMM : an autonomic resource management mechanism for virtual private networks. *In Proceedings of the 7th IEEE conference on Consumer communications and networking conference*, CCNC'10, pages 194–199, Piscataway, NJ, USA, 2010. IEEE Press.

[27] J. ROSENBERG, H. SCHULZRINNE, G. CAMARILLO, A. JOHNSTON, J. PETERSON, R. SPARKS, M. HANDLEY et E. SCHOOLER : SIP : Session Initiation Protocol. RFC 3261, Internet Engineering Task Force, 2002.

[28] Adam SMITH : *An Inquiry into the Nature and Causes of the Wealth of Nations*. History of Economic Thought Books. McMaster University Archive for the History of Economic Thought, 1776.

[29] Hal R. VARIAN : *Introduction à la microéconomie*. De Boeck, 2006.

[30] William VICKREY : Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16(1):8–37, 1961.

[31] John von NEUMANN et Oskar MORGENSTERN : *Theory of games and economic behavior*. Princeton University Press, Princeton, 1944.