Plan de connaissance pour les réseaux sémantiques : application au contrôle d'admission

Doreid Ammar

N° d'ordre : 257 - 2012 Année 2012

THESE DE L'UNIVERSITE DE LYON

Délivrée par

L'UNIVERSITE CLAUDE BERNARD LYON 1

ECOLE DOCTORALE

DIPLOME DE DOCTORAT

(arrêté du 7 août 2006)

soutenue publiquement le 7 décembre 2012

par

Monsieur AMMAR Doreid

TITRE :

Plan de connaissance pour les réseaux sémantiques : application au contrôle d'admission

JURY :

Monsieur BEGIN Thomas	Codirecteur de thèse
Monsieur BEYLOT André-Luc	Rapporteur
Monsieur FRIEDMAN Timur	Membre
Madame GUERIN-LASSOUS Isabelle	Directrice de thèse
Monsieur NOIRIE Ludovic	Membre
Monsieur PANSIOT Jean-Jacques	Membre
Monsieur TURLETTI Thierry	Rapporteur

Remerciements

À mes directeurs de thèse,
Isabelle Guérin Lassous et Thomas Begin.

Aux rapporteurs de ma thèse,
André-Luc Beylot et Timur Friedman.

Aux membres de mon jury de thèse,
Ludovic Noirie, Jean-Jacques Pansiot et Thierry Turletti.

À ma famille,
et plus particulièrement à mes parents.

À mes amis,
et plus particulièrement à l'ensemble des membres de l'équipe RESO.

Au lecteur,
qui par essence justifie la rédaction de ce document.

Table des matières

Abstract **1**

Résumé **3**

1 Introduction **5**
1.1 Contexte . . . 5
1.2 Plan de connaissance . . . 6
1.3 Objectif initial . . . 8
1.4 Démarche . . . 9
1.5 Contributions . . . 9
1.6 Organisation de la thèse . . . 10

2 Etat de l'art **13**
2.1 Internet . . . 14
2.1.1 Architecture actuelle de l'Internet . . . 14
2.1.2 Contrôle des ressources réseau . . . 14
2.1.3 Nouvelles architectures de l'Internet . . . 15
2.1.4 Prediction de performances des chemins réseau . . . 16
2.1.5 Gestion de la connaissance . . . 16
2.1.6 Discussions . . . 16
2.2 Contrôle d'admission . . . 17
2.2.1 Classification du contrôle d'admission . . . 18
2.2.2 Contôle d'admission basé sur des mesures (MBAC) . . . 19
2.2.3 Comparaison des solutions MBAC . . . 21
2.3 Conclusions . . . 23

3 KBAC **27**
3.1 Origine de la solution KBAC . . . 28
3.1.1 Comportement du trafic sur un lien réseau . . . 28
3.1.2 Quel facteur ? . . . 30
3.1.3 Modélisation du comportement d'un lien réseau par un modèle de type file d'attente . . . 31
3.2 Nouvelle solution KBAC . . . 35
3.2.1 Algorithme de mesure . . . 35
3.2.2 Construction du plan de connaissance . . . 36
3.2.3 Algorithme de décision . . . 37
3.2.4 Anticiper le risque d'inondation du nombre de mesures . . . 39
3.2.5 Diversité des points de fonctionnement . . . 39
3.2.6 Cohérence temporelle . . . 40
3.3 Conclusions . . . 41

4 Evaluation de performances **43**
4.1 Scénarios considérés . . . 44
4.1.1 Trafic initial . . . 45
4.1.2 Les flux VBR . . . 52
4.2 Elements de comparaison pour le contrôle d'admission . . . 53
4.2.1 Solutions existantes . . . 53
4.2.2 Oracle . . . 55
4.3 Calibrage des contrôles d'admission . . . 56
4.3.1 Calibrage des contrôles d'admission considérés . . . 57
4.4 Estimation du débit crête des flux entrants . . . 58
4.5 Performances . . . 58
4.5.1 Cas d'une source Poisson . . . 59
4.5.2 Cas d'une source PPBP . . . 61
4.5.3 Cas d'une trace réelle (Trace 1) . . . 62
4.5.4 Cas d'une trace réelle (Trace 2) . . . 63
4.6 Conclusions . . . 65

5 Conclusions et Perspectives **67**
5.1 Principales contributions . . . 67
5.2 Réponses aux questions soulevées . . . 69
5.3 Perspectives et nouveaux défis . . . 70
5.3.1 Evaluation des performances de la solution KBAC pour le cas d'un taux de perte toléré. . . . 70
5.3.2 Etendre la topologie à un réseau. . . . 73
5.3.3 Classification des flux. . . . 75

A Notions de Base **79**

Publications **81**

References **83**

Table des figures

1.1 Croissance du trafic Internet mondial 5
1.2 Briques fonctionnelles du réseau sémantique 7

2.1 Contrôle d'admission 18
2.2 Une classification du contrôle d'admission 18
2.3 Surdimensionnement du lien réseau 23
2.4 Mauvaise qualité d'expérience 23

3.1 Evolution du délai d'attente moyen des paquets dans le buffer du lien en fonction du taux d'utilisation. 28
3.2 Evolution du taux de perte moyen de paquets dans le buffer du lien en fonction du taux d'utilisation. 29
3.3 Evolution du délai d'attente moyen des paquets dans le buffer du lien en fonction du taux d'utilisation pour un segment de la trace réseau divisé en quatre intervalles de temps de taille identique. 30
3.4 Evolution du comportement de quelques facteurs de la trace en fonction du temps. 32
3.5 File d'attente monoserveur 33
3.6 Comportement qualitatif des modèles de type file d'attente. 34
3.7 Exemple d'un plan de connaissance comprenant les mesures, les points de fonctionnement et le modèle type file d'attente associé 37
3.8 Calcul du débit ajusté $\widehat{X}$ 39
3.9 Découpage de l'intervalle de débit de sortie du lien afin d'assurer la diversité des *points de fonctionnement* en S intervalles identiques . . 40
3.10 Nouvelle solution KBAC proposée. 41

4.1 Trafic transitant sur un lien de communication, modélisé par une source initiale et des flux VBR. 45
4.2 Evolution du taux d'utilisation en fonction du temps pour une source Poisson 46
4.3 Répartition des débits des arrivées pour une source Poisson 46
4.4 Processus PPBP. 47
4.5 Fonction d'autocorrélation pour une source Poisson, pour une source PPBP, pour une superposition de 100 flux CBR indépendants et pour une superposition de 20 flux On/Off Pareto. 47
4.6 Evolution du taux d'utilisation en fonction du temps pour une source PPBP 48
4.7 Répartition des débits des arrivées pour une source PPBP 48
4.8 Répartition des tailles des paquets pour une trace réelle (Trace 1) . . 49

4.9 Fonctions d'autocorrélations pour une trace réelle (Trace 1),pour une source Poisson, pour une source PPBP, pour une superposition de 100 flux CBR indépendants et pour une superposition de 20 flux On/Off Pareto sur différents intervalles de temps 50
4.10 Evolution du taux d'utilisation en fonction du temps pour une trace réelle (Trace 1) 50
4.11 Répartition des débits des arrivées pour une trace réelle (Trace 1) . . 50
4.12 Répartition des tailles des paquets pour une trace réelle (Trace 2) . . 51
4.13 Fonctions d'autocorrélation pour une trace réelle (Trace 2), pour une source Poisson, pour une source PPBP, pour une superposition de 100 flux CBR indépendants et pour une superposition de 20 flux On/Off Pareto sur différents intervalles de temps 52
4.14 Evolution du taux d'utilisation en fonction du temps pour une trace réelle (Trace 2) 52
4.15 Répartition des débits des arrivées pour une trace réelle (Trace 2) . . 52
4.16 Performances des solutions dans le cas où la source initiale est un processus de Poisson avec un délai d'attente de 10 ms comme critère de QoS 60
4.17 Performances des solutions dans le cas où la source initiale est un processus PPBP avec un délai d'attente de 20 ms comme critère de QoS 61
4.18 Performances des solutions dans le cas où la source initiale est un modélisée par une trace réelle (Trace 1) avec un délai d'attente de 10 ms comme critère de QoS 62
4.19 Performances des solutions dans le cas où la source initiale est modélisée par de la trace réelle (Trace 2) avec un délai d'attente de 20 ms comme critère de QoS 64

5.1 Topologie de Chen 73
5.2 Répartition des flux dans le réseau 74
5.3 Calcul du délai du flux Id 25 en fonction du taux d'utilisation 75

Liste des tableaux

2.1 Performances de la solution du contrôle d'admission, Enveloppes du Trafic Agrégé, soumis à trois paramétrages différents. 22
2.2 Solutions de contrôles d'admission existantes dans la littérature . . . 25
2.3 Solutions de contrôles d'admission existantes dans la littérature . . . 26

4.1 Synthèse des différents paramètres utilisés pour notre solution KBAC. 56
4.2 Synthèse des différents paramètres utilisés dans les solutions étudiées dans le cas du délai d'attente. 57
4.3 Performances des solutions dans le cas où la source initiale est un processus de Poisson tout au long de la durée de la simulation. 60
4.4 Performances des solutions dans le cas où la source initiale est un processus PPBP tout au long de la durée de la simulation. 61
4.5 Performances des solutions dans le cas où la source initiale est modélisée par une trace réelle (Trace 1) tout au long de la durée de la simulation. 63
4.6 Performances des solutions dans le cas dans où la source initiale est modélisée par une trace réelle (Trace 2) tout au long de la durée de la simulation. 64

5.1 Synthèse des différents paramètres utilisés dans les solutions étudiées dans le cas du taux de perte. 72
5.2 Table des flux 78

Abstract

Abstract

Over the last few years, new usages such as streaming or live video watching are increasingly representing a significant part of Internet traffic. Network operators face the challenge of satisfying the quality of experience expected by end-users while, in the same time, avoiding the over-provisioning of transmission links. Bandwidth management offers a wide spectrum of policies to overcome this issue. Possible options include congestion control, scheduling algorithms, traffic shaping and admission control.

The initial objective of this thesis was the design of a new architecture of traffic management and quality of service for admission control. More precisely, we introduce a novel data-driven method based on a time-varying model that we refer to as Knowledge-Based Admission Control solution (KBAC). Our KBAC solution consists of three main stages: (i) collect measurements on the on-going traffic over the communication link; (ii) maintain an up-to-date broad view of the link behavior, and feed it to a *Knowledge Plane*; (iii) model the observed link behavior by a mono-server queue whose parameters are set automatically and which predicts the expected QoS if a flow requesting admission were to be accepted. Our KBAC solution provides a probabilistic guarantee whose admission threshold is either expressed, as a bounded delay or as a bounded loss rate.

We run extensive simulations using various traffic conditions to assess the behavior of our KBAC solution in the case of a delay threshold. The results show that our KBAC solution leads to a good trade-off between flow performance and resource utilization. This ability stems from the quick and automatic adjustment of its admission policy according to the actual variations on the traffic conditions. On the other hand, our KBAC solution avoids the critical step of precisely calibrating key parameters.

Résumé

Résumé

Depuis quelques années, il y a un réel changement dans les usages des réseaux en termes d'applications véhiculées ainsi que dans leur nombre. On voit apparaître de plus en plus d'applications contraintes en termes de délai, comme par exemple la Téléphonie sur IP, ainsi que d'applications gourmandes en ressources comme par exemple le Streaming Vidéo. La croissance en volume de ces applications commence à poser des problèmes de congestion dans les réseaux filaires et sans fil. Les opérateurs réseaux doivent être capables d'absorber ces changements de trafic, de faire face à cette demande de plus en plus intensive en bande passante et de fournir une bonne qualité de service (QoS) aux applications. Cela nécessite des mécanismes intelligents en termes d'ordonnancement et de gestion des files d'attente, de contrôle d'admission, de contrôle de débit et/ou de routage.

L'objectif initial de cette thèse était d'aboutir à la conception d'une nouvelle architecture de traitement et de gestion du trafic et de la qualité de service pour le contrôle d'admission. Plus précisément nous présentons une nouvelle solution pour le contrôle d'admission qui repose sur l'élaboration en continu d'un plan de connaissance et sur la modélisation automatique du comportement d'un lien réseau par une file d'attente monoserveur. Notre solution doit permettre d'offrir une garantie probabiliste d'un paramètre de performance QoS qui peut être le délai d'attente moyen des paquets dans le buffer du lien ou le taux de perte.

Nous avons évalué les performances de notre nouveau contrôle d'admission par simulation en considérant diverses conditions possibles de trafic. Les résultats obtenus indiquent que la solution proposée permet d'atteindre un contrôle d'admission ni trop conservateur, ni trop permissif. En outre, notre solution offre l'avantage de se baser uniquement sur une connaissance acquise au cours du temps et permet ainsi de s'affranchir d'un paramétrage compliqué des paramètres comme c'est le cas pour les solutions classiques de contrôle d'admission.

1 Introduction

1.1 Contexte

Depuis quelques années, il y a un réel changement dans les usages des réseaux en termes d'applications véhiculées ainsi que dans leur nombre. On voit de plus en plus d'applications contraintes en termes de délai⋆, comme par exemple la téléphonie sur IP (*VoIP: Voice over Internet Protocol*), ainsi que d'applications gourmandes en ressources comme par exemple la vidéo à la demande (*VOD: Video on Demand*). La croissance en volume de ces applications commence à poser des problèmes de congestion dans les réseaux filaires et sans fil. Par exemple, un problème de congestion a été rencontré par AT&T lors de l'été 2010 dans ses réseaux d'accès sans fil dû à une utilisation intensive des iPhones et iPads [lex, 2010] Ces congestions sont fortement pénalisantes pour les flux nécessitant une qualité de service⋆ (*QoS: Quality of Service*) et qui, désormais, représentent une proportion grandissante et significative du trafic. Par exemple, les flux⋆ émis par les serveurs vidéos de Netflix sont devenus en moins de deux ans la principale source Internet sur les réseaux de backbone du territoire nord-américain avec près de 25% du trafic [san, 2011]. Or pour ce type de flux, il est crucial que le taux de perte⋆ et le délai des paquets IP restent sous un certain seuil.

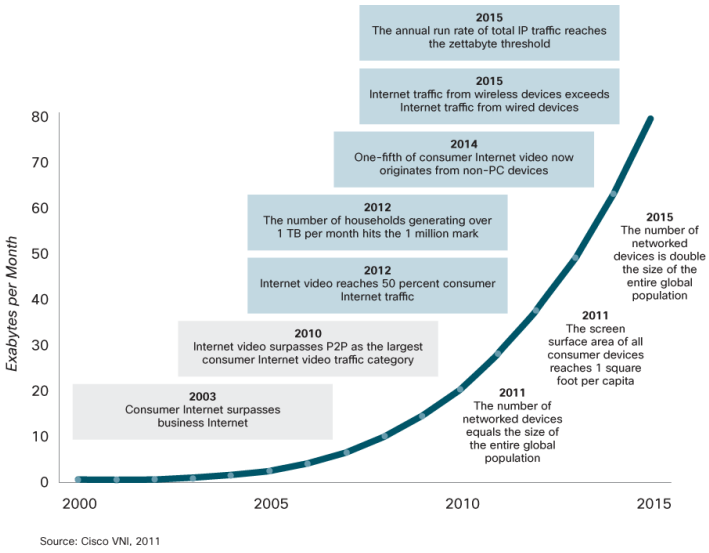


Figure 1.1: Prévisions *Cisco* liées à la croissance du trafic Internet mondial.

⋆ Divers paramètres réseau sont définis dans l'Annexe A

Le volume des données échangées sur les réseaux informatiques, filaires comme sans fil, est donc en constante augmentation, et à supposer que les capacités de transmissions des réseaux ne progressent pas à cette même vitesse, les opérateurs risquent un engorgement de leurs réseaux. De plus, avec une utilisation plus massive de la télévision sur IP (*IPVT: Internet Protocol Television*), de la vidéo à la demande (*VOD: Video on Demand*) et avec l'arrivée de la vidéo 3D et/ou très haute définition, le trafic Internet pourrait être confronté à une véritable explosion. Les prévisions *Cisco* liées à la croissance du trafic Internet mondial (figure 1.1) montrent que, d'ici 2015, le trafic Internet sera quatre fois plus grand dû à la multiplication des moyens permettant d'accéder au réseau [cis, 2012].

Les réseaux qu'ils soient filaires ou bien sans fil devront êtres capables d'absorber ces changements de trafic. C'est un défi pour les opérateurs réseaux de faire face à la demande de plus en plus intensive en bande passante et de fournir une bonne qualité de service aux applications. Les opérateurs réseaux devront améliorer l'utilisation de leurs ressources existantes et mettre en œuvre des nouveaux mécanismes intelligents de gestion de la bande passante*, tout en réduisant les coûts et la complexité liés au déploiement de nouvelles infrastructures.

Plusieurs mécanismes de gestion de bande passante ont été proposés dans la littérature afin de remédier à ce problème, tels que les mécanismes d'ordonnancement (*Scheduling*) et de gestion des files d'attente (*AQM: Active Queue Management*), de contrôle d'admission (*AC: Admission Control*), de contrôle de débit (*Data Rate Control*) et/ou de routage avec QoS (*QoS-Aware Routing*).

1.2 Plan de connaissance

Fournir de bonnes performances aux applications ou optimiser l'utilisation du réseau nécessitent des mécanismes intelligents en termes d'ordonnancement et de gestion des files d'attente, de contrôle d'admission, de contrôle de débit et/ou de routage avec QoS. Chacune de ces solutions va baser ses décisions sur des informations dont la localité peut être très différente d'une solution à une autre. Par exemple, la plupart des solutions d'ordonnancement ou de gestion des files d'attente n'utilisent que des informations locales présentes seulement dans le routeur. Certaines solutions pour le contrôle d'admission vont utiliser des informations locales ou alors provenant des chemins empruntés par les flux tandis que la majorité des solutions de routage avec QoS nécessitent la connaissance globale des liens du réseau.

Plan de connaissance (Knowledge Plane). Les performances de ces mécanismes sont souvent très dépendantes des caractéristiques du réseau et du type de trafic transitant. Une connaissance plus fine du réseau et de son utilisation permettrait très probablement d'améliorer les performances des solutions proposées. Cette connaissance peut reposer sur l'utilisation d'un plan de connaissance dans le réseau capable de construire et maintenir une vue haut-niveau afin de conseiller et de fournir des

services pour mieux contrôler et gérer le réseau [Clark et al., 2003]. Ce plan de connaissance définit un ensemble de mesures, diverses dans l'espace du réseau et dans le temps, dont les valeurs reflètent de façon collective le comportement d'un réseau.

Réseaux sémantiques. Cette thèse s'inscrit dans l'axe de recherche sur les réseaux sémantiques (*SemNet: Semantic Networking*) [Noirie et al., 2009] dans le cadre du laboratoire commun INRIA - Alcatel Lucent Bell Labs. L'objectif des réseaux sémantiques est de prendre en compte la sémantique des flux pour améliorer d'une part les performances des flux circulant dans un réseau et d'autre part de diminuer les coûts du réseau, en limitant le surdimensionnement requis aujourd'hui pour assurer la qualité de service (QoS).

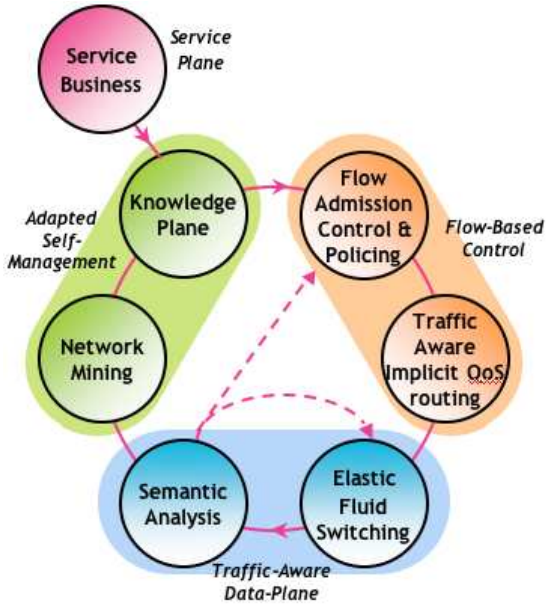


Figure 1.2: Briques fonctionnelles du réseau sémantique.

Le concept de réseau sémantique est défini comme l'association des trois idées suivantes (*cf.* figure 1.2) :

[1] *Flow-Based Networking* : L'idée est de traiter, contrôler et gérer le trafic par flux à tous les niveaux dans le réseau. L'objectif ici est de déterminer des mécanismes de contrôle par flux afin de garantir la qualité de service requise par chacun d'eux dans le réseau. Cette QoS peut être garantie par un contrôle d'accès adapté (*Flow Access Control*), une surveillance du respect des ressources par chaque flux (*Flow Policing*) et/ou un routage adéquat (*QoS-Aware Routing*).

[2] *Traffic-Aware Networking* : Le réseau découvre et analyse les différents flux de trafic transitant en son sein (*Semantic Analysis* et *Flow Monitoring*), et

utilise les informations qu'il a récupérées pour le traitement global des flux dans chaque nœud du réseau.

[3] *Self-Managed Networking* : L'idée générale est que le réseau utilise les informations qu'il aura récupérées par le *Traffic-Aware Networking* et les transforme en connaissances afin d'optimiser l'utilisation globale de ses ressources de manière autonome, pour minimiser les actions des opérateurs dans la configuration et la gestion du réseau. Le principal acteur de ce processus est le plan de connaissance (*Knowledge Plane*) qui permet de coordonner et d'optimiser la distribution des informations utiles pour la gestion du réseau.

Les spécificités apportées par les réseaux sémantiques sont l'information sur le trafic transitant dans le réseau en temps réel et la possibilité d'exploiter ces informations aussi en temps réel pour agir sur l'utilisation des ressources pour la gestion du réseau et de sa configuration. La figure 1.2 illustre les différentes briques fonctionnelles du réseau sémantique décrites ci-dessus.

Objectif de notre travail. La manipulation de flux occupe une position centrale dans les réseaux sémantiques. Cette thèse s'intéresse, plus particulièrement, à contrôler et gérer le trafic par flux de manière autonome en se basant sur un plan de connaissance.

1.3 Objectif initial

L'objectif initial de cette thèse était d'apporter des éléments à la conception d'une nouvelle architecture de traitement et de gestion du trafic et de la qualité de service pour les réseaux sémantiques. Plus précisément cette thèse devait répondre aux questions suivantes :

[Q_1] *Quelles informations doivent être apportées pour la gestion du trafic et la QoS pour les réseaux sémantiques ?*

[Q_2] *Quelle est la périodicité des mesures adaptée à chacune des informations identifiées afin de limiter les mauvaises décisions ou afin d'améliorer les performances ?*

[Q_3] *Est-il nécessaire d'agréger certaines informations ? Si oui, quelles sont les techniques d'agrégation à utiliser ?*

De manière implicite, cette thèse s'intéresse aux problèmes de la représentation et de l'utilisation des connaissances. Notamment, la représentation des connaissances choisie devra permettre un auto-apprentissage sur la base des connaissances qui devrait améliorer les performances du réseau.

1.4 Démarche

Pour répondre à ces questions clés et atteindre l'objectif visé, nous avons suivi la démarche suivante :

Comme première étape de notre recherche, nous avons réalisé une étude sur le plan de connaissance et un état de l'art sur les solutions proposées dans la littérature. L'étude de l'état de l'art nous a montré que peu de solutions ont été proposées pour le plan de connaissance et que toutes ces études restaient essentiellement conceptuelles et de haut niveau. Pour démarrer sur le plan de connaissance, il nous a donc semblé plus simple et plus efficace d'avoir une approche "bottom-up" : nous choisissons un problème de qualité de service bien spécifique et tentons de dégager un plan de connaissance efficace pour cette problématique. Notre choix s'est porté sur le contrôle d'admission (nous expliquons ce choix dans le chapitre 2).

Comme deuxième étape de notre recherche, nous avons réalisé une étude détaillée sur les solutions de contrôle d'admission proposées dans la littérature. Malheureusement, les solutions de contrôle d'admission proposées semblent ne pas souvent loin d'atteindre un degré suffisant de précision. En effet, le problème majeur réside dans la difficulté de calibrer les paramètres des solutions.

L'étude de l'état de l'art nous a permis de dégager un ensemble de connaissances importantes pour bâtir un algorithme de contrôle d'admission capable de manipuler les contraintes de QoS requises pas les applications et de garantir les performances globales atteintes par le réseau. Afin de pallier les limitations des solutions de contrôle d'admission proposées dans la littérature, nous nous sommes fixés comme objectif de développer un nouveau contrôle qui permet de s'affranchir d'un calibrage compliqué des paramètres. Cette nouvelle solution est basée uniquement sur une connaissance acquise au cours du temps. Dans le développement de cette solution réside les contributions majeures de cette thèse.

1.5 Contributions

À part les études de l'état de l'art, les analyses et les synthèses des travaux de la littérature que nous effectuons dans chaque étape de notre démarche de travail, les principales contributions de cette thèse s'articulent en trois volets :

[1] Plan de connaissance (Knowledge Plane). Nous proposons un plan de connaissance pour le contrôle d'admission qui définit un ensemble de mesures dont les valeurs reflètent de façon collective le comportement d'un lien réseau. Son établissement et son utilisation aboutissent à une meilleure gestion du réseau et permet de s'affranchir d'un paramétrage compliqué des paramètres comme c'est le cas pour les solutions classiques de contrôle d'admission.

[2] Notre solution KBAC (Knowledge-Based Admission Control). Nous avons conçu une nouvelle solution pour le contrôle d'admission qui repose sur l'élaboration en

temps continu d'un plan de connaissance et sur la modélisation automatique du comportement d'un lien réseau par un modèle de type file d'attente monoserveur. D'une part, notre solution offre l'avantage de se baser uniquement sur une connaissance acquise au cours du temps et permet ainsi de calibrer automatiquement ces paramètres. D'autre part, elle offre une garantie probabiliste du critère de performance QoS qui peut être le délai d'attente moyen des paquets dans la file ou le taux de perte.

[3] Evaluation des performances dans un cadre plus réaliste. Les décisions prises par un contrôle d'admission sont toujours étroitement liées au trafic agrégé circulant sur le lien. Afin de tester notre solution dans un cadre plus réaliste, nous avons évalué les performances de notre nouveau contrôle d'admission par simulation en considérant diverses conditions possibles de trafic. Pour cela, nous avons implanté dans le simulateur réseaux ns-3 un générateur de trafic qui permet de rejouer des traces réelles ainsi que deux autres générateurs qui permettent de simuler des sources théoriques, notamment, un processus de Poison et un processus PPBP (Poisson Pareto Burst Process) [Zukerman et al., 2003] qui présente un fort degré d'autosimilarité.

Afin de comparer les performances de notre solution, nous avons implanté trois autres solutions pour le contrôle d'admission [Floyd, 1996,Jamin et al., 1997,Qiu and Knightly, 2001] ainsi qu'une procédure permettant de déterminer le nombre maximal de flux pouvant être acceptés, et donc d'atteindre le taux d'utilisation maximal, tout en satisfaisant le critère de QoS (*i.e.,* ni faux positifs ni faux négatifs).

1.6 Organisation de la thèse

Pour détailler l'ensemble des contributions et la démarche de ce travail, ce manuscrit de thèse est organisé comme suit :

Après ce chapitre introductif, le **Chapitre 2** a comme objectif la présentation d'un état de l'art couvrant les pincipales solutions proposées dans la littérature pour le plan de connaissance et le contrôle d'admission. Ainsi, dans un premier temps, nous décrivons quelques études sur la gestion de la connaissance et sur la prédiction de la performance. Dans un second temps, nous décrivons les solutions de contrôle d'admission existantes. Nous nous intéressons plus particulièrement à la méthode du contrôle d'admission basé sur des mesures (MBAC). Nous concluons ce chapitre en soulevant la difficulté de calibrer les paramètres des solutions du contrôle d'admission proposées dans la littérature.

Notre nouvelle solution pour le contrôle d'admission fait l'objet du **Chapitre 3**. La première section de ce chapitre et consacré à la description de l'origine de la solution. Nous détaillons ensuite notre nouvelle solution KBAC (Knowledge-Based Admission Control). Dans un premier temps, nous décrivons l'algorithme de mesure ainsi que les mesures collectées. Dans un second temps, nous décrivons la construction et la modélisation du plan de connaissance. Nous nous intéressons plus partic-

ulièrement à l'agrégation des mesures collectées. Enfin, nous décrivons l'algorithme de décision.

Le **Chapitre 4** est consacré à l'évaluation des performances de notre solution KBAC. Nous menons notre évaluation sous l'angle de la simulation en utilisant le simulateur réseaux ns-3 (*Network Simulator 3*). Ce chapitre illustre le cas d'étude à partir duquel nous avons mené notre évaluation. Premièrement, nous détaillons les diverses conditions de trafic utilisées : un processus de Poisson, un processus PPBP [Zukerman et al., 2003] et des traces collectées sur un réseau réel. Ensuite, nous comparons notre solution avec trois autres solutions de contrôle d'admission [Floyd, 1996, Jamin et al., 1997, Qiu and Knightly, 2001] ainsi qu'une procédure permettant de déterminer le nombre maximal de flux pouvant être acceptés, et donc d'atteindre le taux d'utilisation maximal, tout en satisfaisant le critère de QoS.

Pour finir, le **Chapitre 5** est consacré aux conclusions générales obtenues de l'ensemble de ces travaux et aux perspectives de cette thèse.

2 Etat de l'art

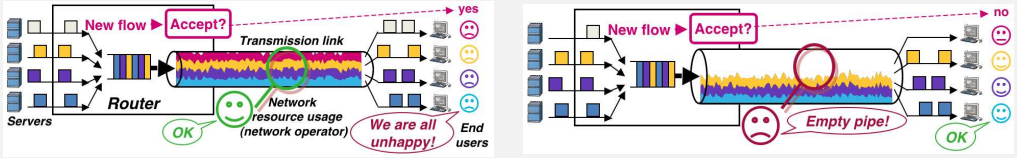
2.1 Internet
- 2.1.1 Architecture actuelle de l'Internet
- 2.1.2 Contrôle des ressources réseau
- 2.1.3 Nouvelles architectures de l'Internet
- 2.1.4 Prediction de performances des chemins réseau
- 2.1.5 Gestion de la connaissance
- 2.1.6 Discussions

2.2 Contrôle d'admission
- 2.2.1 Classification du contrôle d'admission
- 2.2.2 Contôle d'admission basé sur des mesures (MBAC)
- 2.2.3 Comparaison des solutions MBAC

2.3 Conclusions

Le travail présenté dans ce chapitre a été publié à "36th Annual IEEE Conference on Local Computer Networks (LCN)" en 2011 [2]. De plus, il a été publié à "15th Colloque Francophone sur l'Ingénierie des Protocoles, CFIP" en 2011 [7]. En outre, un rapport de recherche a été déposé à "Institut National de Recherche en Informatique et en Automatique (INRIA)" en 2012 [9].

Résumé. L'état de l'art sur le plan de connaissance montre que toutes les études proposées restent essentiellement conceptuelles et de haut niveau. Pour démarrer sur le plan de connaissance, nous choisissons un problème de qualité de service bien spécifique et tentons de dégager un plan de connaissance efficace pour cette problématique. Notre choix s'est porté sur le contrôle d'admission.



L'état de l'art sur les solutions de contrôle d'admission montre que le problème majeur des solutions de contrôle d'admission proposées dans la littérature réside dans la difficulté de calibrer correctement les paramètres des solutions.

2.1 Internet

Objectif. Dans l'Internet, l'abstraction "paquet" a été choisie comme entité de base de l'ensemble des opérations liées à l'acheminement de l'information. L'Internet, tel que nous le connaissons aujourd'hui, est par essence un réseau de données à commutation par paquets. La commutation par paquets assure un meilleur partage de la bande passante et elle constitue une solution plus simple, plus efficace et moins onéreuse que d'autre approches comme la commutation de circuits. Toutefois, cette commutation par paquets semble être mal adaptée aux besoins des services en temps réel (telle que la téléphonie sur IP) car les délais de bout-en-bout des paquets peuvent être très variables (voir certains paquets peuvent être perdus). Afin de répondre à ces besoins, il semble intéressant à mettre en œuvre des mécanismes intelligents de gestion de la qualité de service. L'utilisation d'un plan de connaissance peut être une option possible afin d'assurer une meilleure qualité de service aux applications réseaux. Cependant, l'intégration d'un plan de connaissance dans un réseau nécessite de redéfinir une nouvelle architecture de l'Internet dans laquelle la notion de flux est nativement considérée.

Nous détaillons dans cette section l'architecture actuelle de l'Internet ainsi que les nouvelles architectures proposées. Ensuite, nous présentons de nouvelles solutions de gestion des ressources réseaux qui s'appuient sur ces nouvelles architectures.

2.1.1 Architecture actuelle de l'Internet

L'architecture actuelle de l'Internet est composée de trois plans [Greenberg et al., 2005] :

[1] Un plan de données (*data plane*) qui traite et achemine les paquets de manière individuelle et indépendante.

[2] Un plan de contrôle (*control plane*) qui implémente les algorithmes distribués de routage à travers les éléments de réseau.

[3] Un plan de gestion (*management plane*) qui surveille le réseau et configure les mécanismes du plan de données et les protocoles du plan de contrôle.

2.1.2 Contrôle des ressources réseau

L'architecture classique de l'Internet implique un traitement par paquet. Or le contrôle des ressources réseau peut être réalisé en trois échelles de temps [Noirie et al., 2009] qui correspondent à des niveaux d'agrégation différents : paquets, agrégats de paquets (ou flux) et les agrégats de flux. Ces trois échelles de temps doivent être idéalement considérées car elles affectent toutes qualité perçue par l'utilisateur et elle impactent la simplicité et l'efficacité avec lesquelles le réseau peut être opéré. Or les protocoles Internet actuels ne tiennent pas compte des niveaux des agrégats de paquets et des agrégats de flux. Par exemple, TCP s'occupe du contrôle

des ressources à l'échelle paquet et dans les routeurs traditionnels, les paquets individuels sont traités de manière indépendante, sans tenir compte du comportement du flux. Cela peut donc affecter la qualité de réception au niveau de l'application car des paquets importants peuvent être éliminés ou retardés. Par conséquent, la gestion des agrégats de paquets (flux) et des agrégats de flux est une option possible afin de pallier ces limitations.

2.1.3 Nouvelles architectures de l'Internet

Des études, relativement récentes, ont été élaborées pour proposer de nouveaux plans de fonctionnement du réseau afin de réaliser un contrôle des ressources du réseau par flux. Notamment, certains travaux [Clark et al., 2003] militent pour l'utilisation d'un plan de connaissance dans le réseau capable de construire et maintenir une vue haut-niveau du réseau afin de conseiller et de fournir des services au réseau.

Pour équiper le réseau avec un plan de connaissance, de nouvelles architectures ont été proposées pour relever les défis dans la gestion du réseau. La principale proposition concerne l'architecture 4D [Greenberg et al., 2005]. Cette architecture propose une méthode centralisée pour la gestion des décisions. La décision logique est assurée par des serveurs qui assurent le contrôle sur tous les éléments du réseau. Cette architecture est composée de quatre plans :

[1] Un plan de découverte (*discovery plane*) qui est responsable de la découverte des composants physiques du réseau.

[2] Un plan de données (*data plane*) qui traite les paquets individuellement et qui, en plus, peut collecter des mesures destinées au plan de découverte.

[3] Un plan de décision (*decision plane*) qui prend toutes les décisions nécessaires pour contrôler le réseau. Il gère l'accessibilité, la répartition de charge, la sécurité et la configuration d'interface. Les règles émises par le plan de décision sont destinées au plan de donnée.

[4] Un plan de diffusion (*dissemination plane*) qui rassemble les informations sur l'état du réseau et les transmet au plan de décision et, inversement, il transmet au plan de données les informations de gestion créées par le plan de décision.

Tessesract [Yan et al., 2007] présente un système expérimental basé sur l'architecture 4D. Il fournit le contrôle direct des ressources à travers deux services, le service de diffusion qui permet de découvrir et communiquer avec les nœuds voisins (il peut également fournir une connexion logique entre les éléments de décisions) et de véhiculer la connaissance locale et de voisinage aux autres éléments du réseau et le service de configuration des nœuds qui fournit une table de consultation qui contient les règles du plan de décision et les actions de contrôle correspondantes.

L'architecture 4D a été utilisée dans d'autres solutions, comme présenté juste après.

2.1.4 Prediction de performances des chemins réseau

Dans iPlane [Madhyastha et al., 2006a], *V. Madhyastah et al.* présentent un service évolutif fournissant des prédictions de performances sur les chemins du réseau. Ces prédiction concerne la latence, le taux de perte et la bande passante [Madhyastha et al., 2006b]. IPlane propose une méthode centralisée de la gestion de la connaissance [Madhyastha et al., 2006a, Madhyastha et al., 2009]. L'idée est de construire un "Atlas" des routes à partir des nœuds PlanetLab, Traceroutes et Route Views. Cet "Atlas" contient des prédictions sur des paramètres comme la latence, le taux de perte et la bande passante des chemins entre deux hôtes. IPlane est limité en nombre de participants et son "Atlas" ne peut pas contenir les informations sur tous les chemins du réseau.

2.1.5 Gestion de la connaissance

Ji Li propose dans [Li, 2007, Li, 2008] une solution centralisée pour la gestion de la connaissance. C'est une structure d'agents pour rassembler les connaissances entre les différentes régions du réseau. Cette structure est basée sur deux points principaux :

[1] l'architecture de base pour chaque région qui est responsable de l'organisation des agents

[2] et le partage de la connaissance et la propagation des demandes entre les différentes régions du réseau.

Les informations utiles identifiées sont la topologie statique du réseau, les politiques du réseau et les performances dynamiques du réseau. Ces dernières incluent des informations sur les ressources comme la latence, la bande passante et/ou le taux de pertes.

2.1.6 Discussions

Etat de l'art. Peu de solutions ont été proposées pour le plan de connaissance. Notamment aucune recommandation n'est donnée sur ce qui doit être mesuré, à quelle échelle de temps, comment l'information doit être agrégée, sur quelle partie de réseau, etc. De plus, aucune information sur la nature du trafic n'est intégrée (comme les différentes classes de trafic considérées et les informations associées) et sur les besoins des applications. Enfin, aucun de ces travaux ne s'intéresse au problème de la mise à jour des informations nécessaires et à l'impact éventuel d'informations obsolètes.

Pour conclure, toutes ces études restent essentiellement conceptuelles, de haut niveau et ne répondent pas à nos besoins.

Comment démarrer ? Pour démarrer sur le plan de connaissance, nous avons décidé d'avoir une approche "bottom-up" : nous choisissons un problème de qualité de

service bien spécifique et tentons de dégager un plan de connaissance efficace pour cette problématique.

Quel mécanisme de qualité de service ? Plusieurs mécanismes de qualité de service étaient envisageable dans le travail de cette thèse :

[1] Ordonnancement et gestion des files d'attente (*Scheduling and Active Queue Management*).

[2] Routage avec QoS (*QoS-Aware Routing*).

[3] Contrôle de débit (*Data Rate Control*).

[4] Contrôle d'admission (*Admission Control*).

Dans le cadre du projet sur les réseaux sémantiques [Noirie et al., 2009] plusieurs thématiques avaient été abordées (ou étaient en cours) avant le début de ma thèse. D'une part, plusieurs travaux sur l'ordonnancement et la gestion des files d'attente et le contrôle de débit avaient été réalisés [Carra et al., 2009, Divakaran et al., 2009, Divakaran, 2010]. D'autre part, des travaux sur le routage avec QoS avaient été lancés un an avant le début de ma thèse. En outre, dégager un plan de connaissance pour l'ordonnancement et la gestion des files d'attente nécessite une connaissance approfondie sur les architectures des routeurs, alors que dégager un plan de connaissance pour le routage avec QoS nécessite une connaissance sur l'intégralité du réseau. Nous avons donc décidé de nous concentrer sur le contrôle d'admission, car d'une part ce thème n'avait pas été traité au sein du projet, et d'autre part, avec le contrôle d'admission, il était possible de se concentrer sur l'étude d'un plan de connaissance sur un lien réseau (contrairement au routage avec QoS), ce qui constitue une première étape pour ce travail sur le plan de connaissance.

La section suivante est un état de l'art sur les solutions de contrôle d'admission.

2.2 Contrôle d'admission

Principe. Le contrôle d'admission (noté AC pour *Admission Control* par la suite) vise à limiter le nombre de flux circulant dans un réseau afin de maintenir un niveau d'utilisation des ressources du réseau en deçà d'un certain seuil permettant d'offrir de bonnes performances aux flux, et notamment de satisfaire les critères de performances exigés par les flux avec QoS (*cf.* figure 2.1). En effet, si l'utilisation des liens demeure basse ou modérée, il est généralement admis que les performances des flux seront satisfaits. Deux critères de performances sont généralement considérés dans les solutions existantes de contrôle d'admission : le taux de paquets perdus et le délai passé à attendre dans le buffer du lien avant transmission.

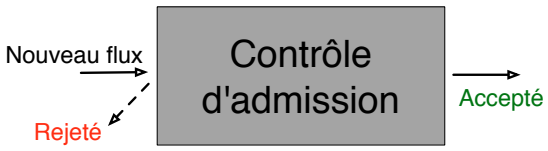


FIGURE 2.1 – Contrôle d'admission

2.2.1 Classification du contrôle d'admission

Plusieurs classifications du contrôle d'admission ont été proposées dans la littérature. Certains travaux classent les contrôles d'admission selon leur nature [Lima et al., 2007] (distribuée ou centralisée) et d'autres selon la technologie utilisée [Mase, 2004] (IntServ/DiffSev, protocoles de signalisation, contrôle de bout-en-bout). Une classification générale du contrôle d'admission souvent adoptée est celle proposée par *Statovci-Halimi* dans [Statovci-Halimi, 2008] (*cf.* figure 2.2) :

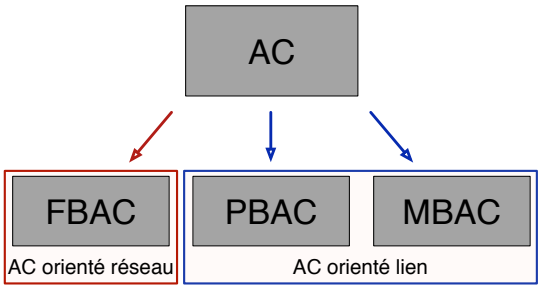


FIGURE 2.2 – Une classification du contrôle d'admission [Statovci-Halimi, 2008].

2.2.1.1 Contrôle d'admission orienté réseau

Ce type de contrôle d'admission permet d'admettre un flux sur un réseau et pas seulement sur un lien. L'approche la plus développée et la plus étudiée est le contrôle d'admission aux extrémités du réseau.

Contrôle d'admission aux extrémités du réseau (FBAC). Les solutions de contrôle d'admission aux extrémités du réseau (*FBAC : Feedback-Based Admission Control*) sont basées sur la notion de « feedback ». Elles utilisent généralement des paquets de tests envoyés sur le chemin à emprunter par le flux qui demande à entrer dans le réseau afin d'évaluer la charge et les capacités offertes sur ce chemin [Breslau et al., 2000b].

Ces approches sont qualifiées de solutions actives car elles injectent du trafic de contrôle dans le réseau pour réaliser le contrôle d'admission. Ce trafic peut, d'une part, avoir un impact sur les performances des flux existants et, d'autre part, ne donne une indication sur l'état du chemin que sur un intervalle de temps réduit, correspondant au temps d'envoi des paquets de test.

2.2.1.2 Contrôle d'admission orienté lien

Cette approche permet d'admettre un flux sur un seul lien réseau. Elle est décomposée en deux types :

Contrôle d'admission basé sur la description du trafic (PBAC). Les solutions de contrôle d'admission basé sur la description (*a priori*) du trafic (*PBAC : Parameter-Based Admission Control*) supposent la connaissance de descripteurs de trafic pour tous les flux désirant entrer dans le réseau, mais également pour ceux déjà entrés afin d'estimer la charge courante dans le réseau et d'en déduire la possibilité ou non d'accepter de nouveaux flux [Jamin et al., 1997]. Ces solutions caractérisent très souvent le flux entrant par son débit moyen et/ou son débit crête.

Cependant, déterminer des descripteurs de trafic qui caractérisent avec précision les profils des flux peut s'avérer être une tâche coûteuse et difficile pour l'utilisateur et l'opérateur. On pourrait également envisager de caractériser par un descripteur de trafic unique l'agrégation de flux résultant du multiplexage réalisé dans le réseau afin de réduire le nombre de descripteurs à maintenir, mais, en pratique, cette opération se révèle difficile à réaliser avec précision.

Contrôle d'admission basé sur des mesures (MBAC). Les solutions de contrôle d'admission basé sur des mesures (*MBAC : Measurement-Based Admission Control*) s'affranchissent des problèmes de précision et d'extensibilité rencontrés dans les solutions PBAC, car elles ne nécessitent pas une connaissance par flux, mais sur l'agrégation du trafic qui traverse le lien de communication. Les solutions MBAC, comme leur nom l'indique, utilisent des mesures sur les liens pour estimer la bande passante résiduelle offerte par le réseau. Il s'agit donc d'une solution basée sur des mesures passives, contrairement aux solutions FBAC, et qui ne nécessite pas la connaissance de descripteurs de trafic pour les flux déjà admis, contrairement aux solutions PBAC.

2.2.1.3 Discussion

Les avantages du contrôle d'admission basé sur des mesures (MBAC) en font une approche intéressante pour le contrôle d'admission et c'est à cette approche que nous nous intéressons dans toute la suite de ce chapitre.

2.2.2 Contôle d'admission basé sur des mesures (MBAC)

Principe. Les solutions de contrôle d'admission basé sur des mesures (MBAC) sont généralement pensées pour fonctionner à l'échelle d'un lien. Le mécanisme des solutions proposées jusqu'ici peut se décomposer en deux parties :

[1] Des opérations de mesure sur le trafic déjà admis permettent de le caractériser et d'en déduire un certain nombre de métriques (e.g., le taux d'utilisation du lien).

[2] Un algorithme, composé d'une (ou plusieurs) opération(s) de test, permet d'évaluer si, étant donné le débit du flux cherchant à entrer, supposé connu, le réseau peut correctement « absorber » ce nouveau flux tout en maintenant un niveau de performances suffisant à tous les flux.

Dans le cas le plus simple, l'algorithme peut se réduire à s'assurer que le débit du flux cherchant à entrer est inférieur à bande passante résiduelle* du lien [Jamin et al., 1997]. Les solutions existantes pour le contrôle d'admission basé sur les mesures diffèrent par la nature des mesures, par les hypothèses faites sur le trafic entrant et par l'algorithme d'admission.

Etendre la solution MBAC sur la globalité du réseau. Afin d'étendre la solution MBAC sur la globalité du réseau, il suffit de répéter ce contrôle d'admission sur chacun des liens traversés par le flux. En outre, pour simplifier la tâche des opérateurs réseaux, le contrôle d'admission peut être appliqué uniquement sur les liens correspondant à des goulots d'étranglement (*bottleneck links*) dans les réseaux d'accès et dans le cœur du réseau.

2.2.2.1 Solutions existantes

Plusieurs solutions de contrôle d'admission basé sur des mesures ont été proposées dans la littérature. Les auteurs de [Guerin et al., 1991] ont été les premiers à proposer la notion de *capacité équivalente* utilisée dans plusieurs solutions de contrôle d'admission. La capacité équivalente (du trafic) d'un lien, notée $C(\epsilon)$, est telle que, en régime stationnaire, le taux d'arrivée sur ce lien excède $C(\epsilon)$ avec une probabilité au plus de ϵ. Dans les contrôles d'admission utilisant la notion de capacité équivalente d'un lien, l'algorithme d'admission s'assure que la capacité équivalente actuelle du lien à laquelle s'ajouterait le flux cherchant à entrer n'excède pas la bande passante résiduelle du lien. La formule de la capacité équivalente donnée dans [Guerin et al., 1991] repose sur l'hypothèse que le taux d'arrivée des flux suit une distribution Normale ainsi que sur un modèle sans buffer. Floyd, dans [Floyd, 1996], a proposé une autre formule pour la capacité équivalente en se basant sur les bornes de "Hoeffding". Dans [Georgoulas et al., 2008], les auteurs se basent aussi sur la formule de la capacité équivalente donnée dans [Guerin et al., 1991] mais intègrent un facteur d'admission (APF - Admission Policy Factor) dans leur algorithme d'admission afin de moduler le niveau de rigueur de l'admission que l'opérateur désire appliquer en termes de taux de perte sur les paquets. Ces trois solutions nécessitent une mesure du taux d'utilisation ou du débit courant du lien pour fonctionner. Dans [Jamin and Danzig, 1997], les auteurs intègrent la contrainte de délai pour certains flux dans leur contrôle d'admission. Pour cela, l'algorithme nécessite, en plus d'une mesure du taux d'utilisation courant du lien, une mesure du délai (d'attente) maximal enregistré sur ce lien. Leur algorithme d'admission se décompose en deux parties : un test relatif au taux d'utilisation courant du lien et un test pour le délai. Les auteurs de [Qiu and Knightly, 2001] ont cherché à améliorer les travaux de [Jamin and Danzig, 1997] en proposant une autre mesure du taux d'utilisation du lien afin de mieux

caractériser le trafic circulant sur ce lien ainsi qu'un algorithme d'admission plus flexible que celui de [Jamin and Danzig, 1997] où les paramètres de QoS peuvent être directement contrôlés sans fixer au préalable des valeurs seuils comme dans [Jamin and Danzig, 1997]. Notons que toutes les solutions citées précédemment ont été pensées et évaluées en supposant une connaissance du débit crête des flux entrant, soit parce qu'il est donné, soit parce qu'il peut être déduit des paramètres connus d'un seau à jetons appliqué en entrée du lien.

L'ensemble des algorithmes du contrôle d'admission décrits précédemment ont besoin de déterminer une échelle de temps pour la mesure de la bande passante utilisée/restante dans le système. Cette échelle a été le sujet de plusieurs études car elle a un impact non négligeable sur la précision du contrôle d'admission. *Grossglauser et al.* proposent dans [Grossglauser and Tse, 2003] la notion d'échelle de temps critique pour un MBAC. Cette échelle de temps est représentée par $\tilde{T} = T_h/\sqrt{n}$, avec T_h, la durée moyenne des flux et n le nombre de flux sur un lien. *Eun et Shroff* décrivent dans [Eun and Shroff, 2003a] une méthode pour calculer l'échelle de temps à utiliser, appelée DTS (Dominant Time Scale), pour caractériser différentes mesures de qualité de service. Les auteurs s'intéressent ici à la probabilité de surcharge d'un lien. La solution combine une approche analytique et une approche par mesures afin de déterminer, en temps borné, l'échelle de temps nécessaire.

Dans [Milbrandt et al., 2004, Milbrandt et al., 2007b, Milbrandt et al., 2007a], les auteurs présentent une solution hybride, dénommée EBAC pour *Experience-Based Admission Control*, qui combine les solutions classiques PBAC et MBAC. Cette solution intègre un facteur, noté *overbooking factor*, qui définit le rapport entre le débit moyen et le débit crête des descripteurs de trafic. Cette solution nécessite une marge de sécurité pour garantir le critère de QoS.

La table 2.2 et la table 2.3 illustrent des solutions de contrôle d'admission existantes dans la littérature. Elles montrent les paramètres de mesures collectés, l'échelle de temps de mesure, l'algorithme de décision, le type de trafic utilisé et le critère de QoS cherché.

2.2.3 Comparaison des solutions MBAC

2.2.3.1 Travaux de comparaison existants

Quelques travaux ont cherché à comparer ces différentes solutions entre elles. Dans [Jamin et al., 1997], une comparaison de trois solutions de contrôle d'admission basé sur des mesures a été réalisée sous simulation. Les résultats obtenus montrent, entre autres, qu'une version simplifiée de la solution de [Jamin and Danzig, 1997] (n'intégrant pas la contrainte de délai) obtient une meilleure utilisation du lien au prix d'un petit taux de perte comparé à la solution de [Floyd, 1996] qui, elle, n'induit aucune perte de paquet. Ces mêmes auteurs étendent leurs travaux dans [Breslau et al., 2000a] avec une comparaison de six solutions de contrôle d'admission basé sur des mesures réalisées sous simulation. En faisant varier les paramètres utilisés dans les tests d'admission des solutions étudiées, ils montrent que toutes ces solutions atteignent le même ensemble de valeurs sur le compromis taux d'utilisation

du réseau et taux de pertes sur les flux. Les auteurs concluent sur le fait que la difficulté réside dans les valeurs à donner aux paramètres des contrôles d'admission afin d'obtenir un compromis taux d'utilisation - taux de pertes donné car ces valeurs n'aboutissent pas au final aux performances réellement attendues sur le réseau et les flux et qu'elles sont donc, par conséquent, difficiles à déterminer *a priori*. Dans [Nevin et al., 2008], les auteurs comparent trois solutions de contrôle d'admission à un contrôle d'admission idéal en considérant diverses conditions possibles de trafic. Les résultats montrent que ces trois solutions violent le critère de QoS. En outre, *Moore* compare dans [Moore, 2004] plusieurs solutions de contrôles d'admission sous une plateforme expérimentale [Moore and Crosby, 1999]. Les résultats montrent que seule la solution de [Qiu and Knightly, 2001] atteint des résultats acceptables.

2.2.3.2 Notre Etude de Comparaison

Pour vérifier le problème de paramétrage des solutions classiques de contrôle d'admission, nous avons implanté trois algorithmes [Floyd, 1996, Jamin and Danzig, 1997, Qiu and Knightly, 2001] dans le simulateur ns-3 (*Network Simulator 3*) et nous avons mené une série de simulations en utilisant des paramétrages différents.

Scénarios. Nous avons considéré un lien de 10 Mb/s avec une taille de buffer de 60 ms. Le lien est déjà soumis à un trafic initial auquel vont tenter de venir s'ajouter des flux VBR de débits 64 kb/s et de durée moyenne 120 s. Les arrivées des flux VBR suivent un processus de Poisson. Dans tous les cas, le trafic initial émet à un débit moyen de 4,5 Mb/s.

Résultats. Dans cette section, nous présentons uniquement les résultats obtenus pour une solution de contrôle d'admission [Qiu and Knightly, 2001], dénommée Enveloppes du Trafic Agrégé (*Aggregate Traffic Envelopes*), avec le délai d'attente comme critère de QoS. Nous détaillons les résultats de performances dans le Chapitre 4.

TABLE 2.1 – Performances de la solution du contrôle d'admission, Enveloppes du Trafic Agrégé, soumis à trois paramétrages différents.

	Enveloppes du Trafic Agrégé		
	$\alpha_E = 0{,}01$	$\alpha_E = 1{,}3$	$\alpha_E = 3{,}62$
Nombre de flux acceptés	400	310	156
Pourcentage de violation	55,11%	0%	0%

La table 2.1 récapitule les résultats obtenus dans le cas de la solution Enveloppes du trafic Agrégé. Nous avons testé cette solution en utilisant trois paramétrages différents (*i.e.,* $\alpha_E = 0{,}01$, $\alpha_E = 1{,}3$ et $\alpha_E = 3{,}62$), où α_E et une constante spécifiant

le degré de confiance (se reporter à la section 4.2.1 du chapitre 4 pour plus de détails). La table 2.1 illustre, d'une part, le nombre de flux acceptés et, d'autre part, le pourcentage de violation du délai d'attente moyen qu'on cherche à garantir (10 ms pour ces scénarios). Les résultats montrent que les performances de cette solution de contrôle d'admission changent suivant le choix du paramètre α_E. Pour α_E = 0,01, ce contrôle d'admission accepte trop de flux (400 flux) et la qualité de service expérimentée par les flux se dégrade (la violation du critère de QoS est de 55,11%). En outre, pour $\alpha_E = 1{,}3$, ce contrôle d'admission accepte 310 flux et ne viole jamais le critère de QoS. Enfin, pour $\alpha_E = 3{,}62$, ce contrôle d'admission n'accepte pas assez de flux (156 flux) et le lien de communication est sont quasiment vide.

2.3 Conclusions

L'état de l'art sur le plan de connaissance nous montre que peu de solutions ont été proposées et que toutes ces études restent essentiellement conceptuelles et de haut niveau. Pour démarrer sur le plan de connaissance, nous choisissons un problème de qualité de service bien spécifique et tentons de dégager un plan de connaissance efficace pour cette problématique. Notre choix s'est porté sur le contrôle d'admission.

Plusieurs solutions de contrôle d'admission ont été proposées dans la littérature. Malheureusement, ces solutions s'avèrent loin d'atteindre un degré suffisant de précision. En effet, le problème majeur réside dans la difficulté de calibrer les paramètres des solutions. Soit on accepte trop de flux et la qualité de service expérimentée par les usagers se dégrade (*cf.* figure 2.3), soit on n'accepte pas assez de flux et les liens de communications sont quasiment vides (*cf.* figure 2.4).

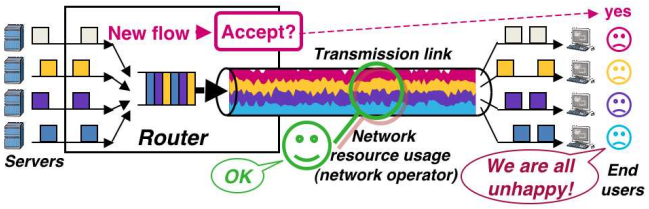


FIGURE 2.3 – Surdimensionnement du lien réseau.

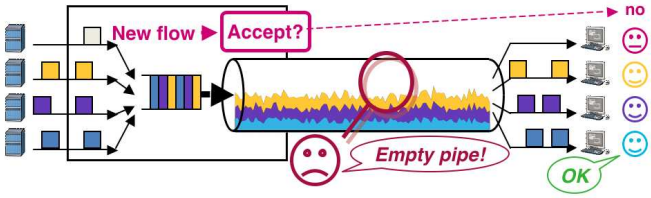


FIGURE 2.4 – Mauvaise qualité d'expérience.

Face à cette situation, nous nous fixons comme objectif de développer un nouveau contrôle d'admission basé sur un plan de connaissance qui pallie toutes ces limitations. Cette nouvelle solution est détaillée dans le chapitre 3.

TABLE 2.2 – Solutions de contrôles d'admission existantes dans la littérature

Trafic	Paramètres de mesures	Échelle de temps de mesures	Algorithme de décision	Critère de QoS
Georgoulas et al. [Georgoulas et al., 2008] Measurement-based admission control for real-time traffic in IP Differentiated Services Networks [Georgoulas et al., 2008] An integrated bandwidth allocation and admission control framework for the support of heterogeneous real-time traffic in class-based IP networks				
trafic agrégé (distribution Gaussienne) trafic temps réelle (VoIP et Vidéoconférence) Source ON-OFF	$C_{est} = M + r + \alpha'_{PLR}\sqrt{\sigma^2}$ C_{est} : bande passante estimée M : débit moyen mesuré r : débit crête du nouveau flux σ : variation sur le débit α'_{PLR} : paramètre de tuning $\alpha'_{PLR} = \sqrt{-2\ln(P_r - \ln(2\Pi)}$ P_r : taux de perte visé	fenêtre glissante $w = \max(DTS, w')$ DTS : "*Dominant Time Scale*" [Eun and Shroff, 2003b] w' : temps moyen inter-paquets	$C_{est} \times APF \leq C$ $\Rightarrow$ flux admis $C_{est} \times APF > C$ $\Rightarrow$ flux rejeté C : capacité nominale du lien APF : facteur d'admission	taux de perte
$APF = (N_{ref}/T_{ref}) \times \sqrt{\frac{-2\ln(P_r - \ln(2\Pi))}{-2\ln(P_r - \ln(2\Pi))}}$, APF paramètre pour ajuster le niveau de rigueur				
Qiu et al. [Qiu and Knightly, 2001] Measurement-based admission control with aggregate traffic envelopes				
trafic agrégé (traces réelles MPEG)	débit maximal du trafic agrégé $R_k^1 = \max_{t-T+k \leq s \leq t} \sum_{u=s-k+1}^{s} a_u$	$a_\tau = A[t\tau, (t+1)\tau]$	$\max_{k=1,\ldots,T} \{k\tau(\bar{R}_k + r + \alpha_E\sigma_k - C)\} \leq C \times D$ et $\bar{R}_T + r + \alpha_E\sigma_T \leq C$ $\Rightarrow$ flux admis	taux de perte et délai
α_E : degré de confiance ; r : débit crête du nouveau flux ; D : délai d'attente maximal ; C : Capacité du lien				
Floyd [Floyd, 1996] Comments on measurement-based admissions control for controlled-load services				
trafic audio Source ON-OFF	capacité équivalente $\hat{C}_H = \hat{r} + \sqrt{\frac{\ln(1/\epsilon)\sum_{i=1}^{n} r_i^2}{2}}$		$\hat{C}_H + r \leq C$ $\Rightarrow$ flux admis	taux de perte
r : débit crête du nouveau flux ; r_i : débit crête du i^e flux ; ϵ : propabilité de violation de la capacité équivalente ; $\hat{r}$: débit moyen du trafic agrégé				

TABLE 2.3 – Solutions de contrôles d'admission existantes dans la littérature

Trafic	Paramètres	Échelle de temps de mesures	Algorithme de décision	Critère de QoS
Jamin et al. [Jamin and Danzig, 1997] A measurement-based admission control algorithm for integrated services packet networks				
trafic audio Source ON-OFF Source ON-OFF Pareto	R : charge existante mesuré r : débit crête du nouveau flux $\widehat{D}$: délai mesuré ϑ : paramètre de tuning b_i : variabilité d'un flux		$R + r \leq C\vartheta$ $\Rightarrow$ flux admis $\widehat{D} + \frac{b_i}{C} \leq D$ $\Rightarrow$ flux admis	taux de perte et délai
C : capacité du lien ; D : delai d'attente maximal				
Nevin et al. [Nevin et al., 2008] Robustness Study of MBAC Algorithms				
trafic audio ON-OFF	S : nombre de flux λ : débit d'un flux $\widehat{m}$: durée d'un flux		*Oracle* $S = \lambda \times \widehat{m}$	taux de perte et délai
comparaison de trois solutions de contrôle d'admission [Floyd, 1996, Jamin and Danzig, 1997, Qiu and Knightly, 2001] à un *Oracle*				
Grossglauser et al. [Grossglauser and Tse, 2003] A time-scale decomposition approach to measurement-based admission control				
Trace vidéo *Star Wars*	M_t : nombre maximal des flux qui peuvent être admis par un AC à un temps donné N_t : nombre actuelle des flux dans le système	Echelle de temps critique $\tilde{T}_h = T_h/\sqrt{n}$ T_h : durée moyenne d'un flux n : nombre de flux sur un lien réseau	$M_t \geq N_t + 1$ $c - (N_t + 1)\hat{\mu}_t > \alpha_q \hat{\sigma}_t^H \sqrt{N_t + 1}$ $\Rightarrow$ flux admis	taux de perte
$c - (N_t + 1)\hat{\mu}_t$: bande passante restante après avoir accépté un nouveau flux $\alpha_q \hat{\sigma}_t^H \sqrt{N_t + 1}$: bande passante requise pour accommoder la flucturation rapide dans l'echelle du temps				

KBAC

3

3.1 Origine de la solution KBAC

3.1.1 Comportement du trafic sur un lien réseau

3.1.2 Quel facteur ?

3.1.3 Modélisation du comportement d'un lien réseau par un modèle de type file d'attente

3.2 Nouvelle solution KBAC

3.2.1 Algorithme de mesure

3.2.2 Construction du plan de connaissance

3.2.3 Algorithme de décision

3.2.4 Anticiper le risque d'inondation du nombre de mesures

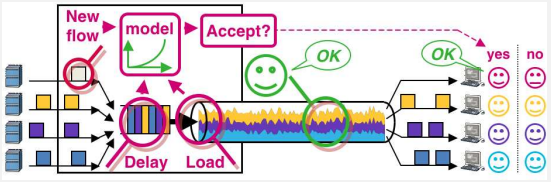
3.2.5 Diversité des points de fonctionnement

3.2.6 Cohérence temporelle

3.3 Conclusions

Le travail présenté dans ce chapitre a été publié à "37th Annual IEEE Conference on Local Computer Networks (LCN)" en 2012 [1]. De plus, il a été publié à "Algotel" en 2012 [6]. En outre, un rapport de recherche a été déposé à "INRIA" en 2012 [8] et deux démonstrations ont été présentées en 2012 à "Demonstrations of the IEEE Conference on Local Computer Networks (LCN-Demos)" [4] et à "Alcatel-Lucent Bell Labs Open Days" [5].

Résumé. Notre nouvelle solution KBAC (*Knowledge-Based Admission Control*) repose sur (*i*) l'élaboration en temps continu d'un plan de connaissance (*ii*) et sur la modélisation du comportement d'un lien réseau par un modèle de type file d'attente.



Notre solution KBAC offre l'avantage de se baser uniquement sur une connaissance acquise au cours du temps et permet ainsi de s'affranchir d'un calibrage compliqué de paramètres.

3.1 Origine de la solution KBAC

3.1.1 Comportement du trafic sur un lien réseau

Objectif initial. Le comportement du trafic réseau varie fortement en fonction du type d'application véhiculée et de la nature du trafic transitant. Ce comportement change au cours du temps et varie suivant l'emplacement du lien de communication considéré. Ce comportement volatile peut engendrer : (*i*) des périodes de silence pendant lesquelles il n'y a aucun trafic qui circule sur le lien ; (*ii*) des périodes quasiment stables avec des arrivées régulières des paquets ; (*iii*) et des périodes de pics avec une très forte variabilité sur le temps des interarrivées des paquets.

Fournir de bonnes performances aux applications ou optimiser l'utilisation du réseau nécessitent des mécanismes intelligents capables d'absorber le comportement volatile du trafic réseau. Par conséquent, afin de mieux comprendre le comportement du trafic sur un lien réseau, nous avons rejoué par simulation une trace réelle [Sass, 2004] en l'injectant en entrée d'un lien de communication et nous avons observé deux paramètres de performances, à savoir le temps d'attente dans le buffer et le taux de perte des paquets.

Scénarios considérés. Sous le simulateur réseau ns-3, nous avons considéré un lien de 10 Mb/s avec une taille de buffer de 20 ms. Le lien est soumis à une trace réelle injectée en entrée du lien. Cette trace a été collectée par l'Université de Stuttgart [Sass, 2004], le dimanche 31 Octobre 2004 entre 18 heures et 22 heures, sur un lien du réseau "SelfNet". Pour chaque fenêtre de temps de longueur 2 s, nous mesurons le taux d'utilisation du lien ainsi que d'autres paramètres de performance de QoS, comme notamment le délai d'attente moyen des paquets dans le buffer du lien ou le taux de perte moyen de paquets.

Evaluation. La figure 3.1 montre les résultats obtenus dans le cas du délai.

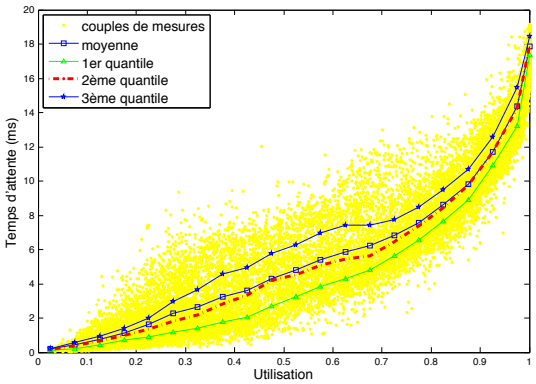


FIGURE 3.1 – Evolution du délai d'attente moyen des paquets dans le buffer du lien en fonction du taux d'utilisation.

La figure 3.1 montre des couples de mesures qui lient un délai mesuré à un taux d'utilisation mesuré. Cette figure illustre l'évolution du temps d'attente moyen passé par les paquets dans le buffer du lien en fonction du taux d'utilisation. Premièrement, nous remarquons que les couples de mesures collectées sont très variables et forment un large faisceau de performances (par exemple, pour un taux d'utilisation de 0,6 on retrouve plusieurs valeurs du délai qui varient entre quelques millisecondes et 10 ms). D'autre part, la valeur moyenne et la valeur médiane (2^e quartile) des couples de mesures pris à des intervalles réguliers sur le taux d'utilisation forment des courbes avec une allure assez lisse qui semblent suivre un comportement proche de celui d'un modèle de type file d'attente. Malheureusement, ces deux méthodes de calcul de la valeur moyenne ne peuvent pas êtres considérées comme une référence du comportement du lien car elles ne sont pas capables d'absorber le comportement volatile des couples de mesures. Cette grande variabilité sur les mesures est soulevée par le calcul du 1^{er} quartile (le 1^{er} quartile sépare les 25% inférieures des couples de mesures) et du 3^e quartile (le 3^e quartile sépare les 75% inférieures des couples de mesures). Cette dispersion de l'écart entre les quartiles empêche de prédire avec précision la performance du lien pour une valeur de taux d'utilisation donnée.

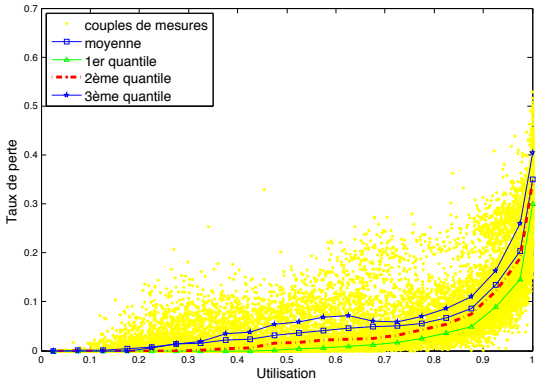


FIGURE 3.2 – Evolution du taux de perte moyen de paquets dans le buffer du lien en fonction du taux d'utilisation.

La figure 3.2 illustre l'évolution du taux de perte moyen de paquets dans le buffer du lien en fonction du taux d'utilisation. Les résultats montrent le comportement volatile des couples de mesures qui lient un taux de perte mesuré à un taux d'utilisation mesuré. Ces mesures sont très variables et forment un large faisceau de performances (par exemple, pour un taux d'utilisation de 0,6 on retrouve plusieures valeurs du taux de perte dans la file qui varient entre zéro et 0,2).

Nouvel objectif. Face à ces observations, nous nous fixons comme objectif de chercher, si c'est possible, un facteur qui expliquerait ce comportement volatile des couples de mesures collectés.

3.1.2 Quel facteur ?

Afin de mieux comprendre le comportement des paramètres de performances nous analysons plus en détail des segments de la trace réseau dans le but d'identifier le facteur qui pourrait expliquer la variabilité observée dans les paramètres de performances.

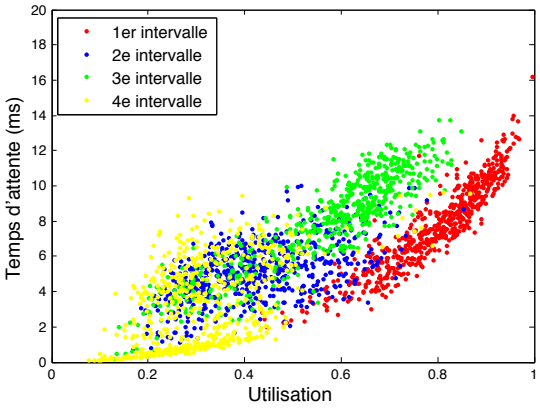


FIGURE 3.3 – Evolution du délai d'attente moyen des paquets dans le buffer du lien en fonction du taux d'utilisation pour un segment de la trace réseau divisé en quatre intervalles de temps de taille identique.

La figure 3.3 illustre l'évolution du délai d'attente moyen des paquets dans le buffer du lien en fonction du taux d'utilisation pour un segment de la trace réseau d'une durée de 30 minutes. Nous divisons de nouveau ce segment en quatre intervalles de temps de taille identique (7,5 minutes par intervalle). Nous représentons les couples de mesures de ces intervalles par quatre couleurs différentes. Nous observons que les points rouges collectés sur les 7,5 premières minutes forment un faisceau de performances étroit (par exemple, pour un taux d'utilisation de 0,6 le délai d'attente dans le buffer du lien varie entre 2 ms et 4 ms). De plus, le comportement des points verts, collectés sur l'avant-dernier intervalle de temps, concerne le même taux d'utilisation que les points rouges mais avec une valeur plus grande du délai. Par exemple, le délai varie entre 6 ms et 10 ms pour un taux d'utilisation de 0,6. En revanche, les autres intervalles des couples de mesures forment un large faisceau de performances qui souligne le comportement volatile des paramètres de performances.

Les facteurs étudiés. Nous étudions l'évolution du comportement de quelques facteurs de la trace qui pourraient impacter le comportement des couples de mesures présentés dans la figure 3.3. Dans ce manuscrit, nous nous limitons uniquement à cinq facteurs : (1) le taux d'utilisation du lien réseau ; (2) le coefficient de variation de l'utilisation ; (3) le temps de service (temps de transmission pour ce scénario) des paquets ; (4) le coefficient de variation sur les temps d'interarrivées entre les paquets

successifs ; (5) et le coefficient de variation sur les débits des arrivés des paquets en entrée du lien.

La figure 3.4 montre les résultats obtenus. Elle illustre l'évolution au cours du temps des facteurs étudiés suivant les quatre intervalles de temps décrits précédemment (le temps est divisé en quatre intervalles de temps de taille 7,5 minutes). Premièrement, les résultats du taux d'utilisation du lien réseau montrent des niveaux d'utilisation élevés dans le premier et le troisième intervalle de temps. Ce comportement explique la concentration des couples de mesures (rouges et verts) à des niveaux d'utilisation élevé comparé aux autres intervalles. En revanche, ce premier facteur n'apporte pas les explications nécessaires sur la largeur des faisceaux de performances. De plus, il ne justifie pas le fait que la valeur du délai est plus grande pour les points verts comparé aux points rouges. Deuxièmement, le coefficient de variation de l'utilisation du lien pourrait être le facteur qui explique la dispersion des couples de mesures. Nous remarquons que le coefficient de variation de l'utilisation est un peu plus significatif pour le deuxième et le dernier intervalle, ce qui justifie la largeur des faisceaux de performances de la figure 3.3. En outre, le coefficient de variation sur les débits des arrivés des paquets en entrée du lien semble également capable d'expliquer la largeur des faisceaux de performances pour le dernier intervalle. Cependant, en ce qui concerne le deuxième et le troisième intervalles, les niveaux de coefficient de variation sont quasiment similaires. Toutefois, la largeur des faisceaux de performances (*cf.* figure 3.3) n'est pas la même. Par conséquent, ce facteur n'est pas un paramètre significatif qui peut expliquer le comportement du trafic. Enfin, les autres facteurs sur le temps de service des paquets et le coefficient de variation sur les temps d'interarrivées entre les paquets successifs n'apportent pas d'explications sur le comportement volatile des paramètres de performances.

Conclusion partielle. Pour conclure, nous n'avons pas pu trouver un facteur, parmi ceux étudiés, qui peut clairement expliquer le comportement du trafic réseau sur un lien réseau. Il semble que ce soit plutôt la combinaison de plusieurs paramètres qui reflètent l'évolution possible du comportement du lien.

3.1.3 Modélisation du comportement d'un lien réseau par un modèle de type file d'attente

Objectif. En se basant sur les résultats obtenus précédemment, nous choisissons de modéliser le comportement d'un lien réseau par un modèle de type file d'attente. Ce modèle de type file d'attente est approximativement capable de décrire le comportement d'un routeur ou d'un lien réseau.

Il s'agit de proposer une abstraction mathématique du système réel permettant de concentrer, dans un modèle, les comportements et les paramètres reproduisant aux mieux le fonctionnement étudié du lien réseau.

Notons que dans la suite de ce chapitre nous considérons le débit en sortie de lien à la place du taux d'utilisation.

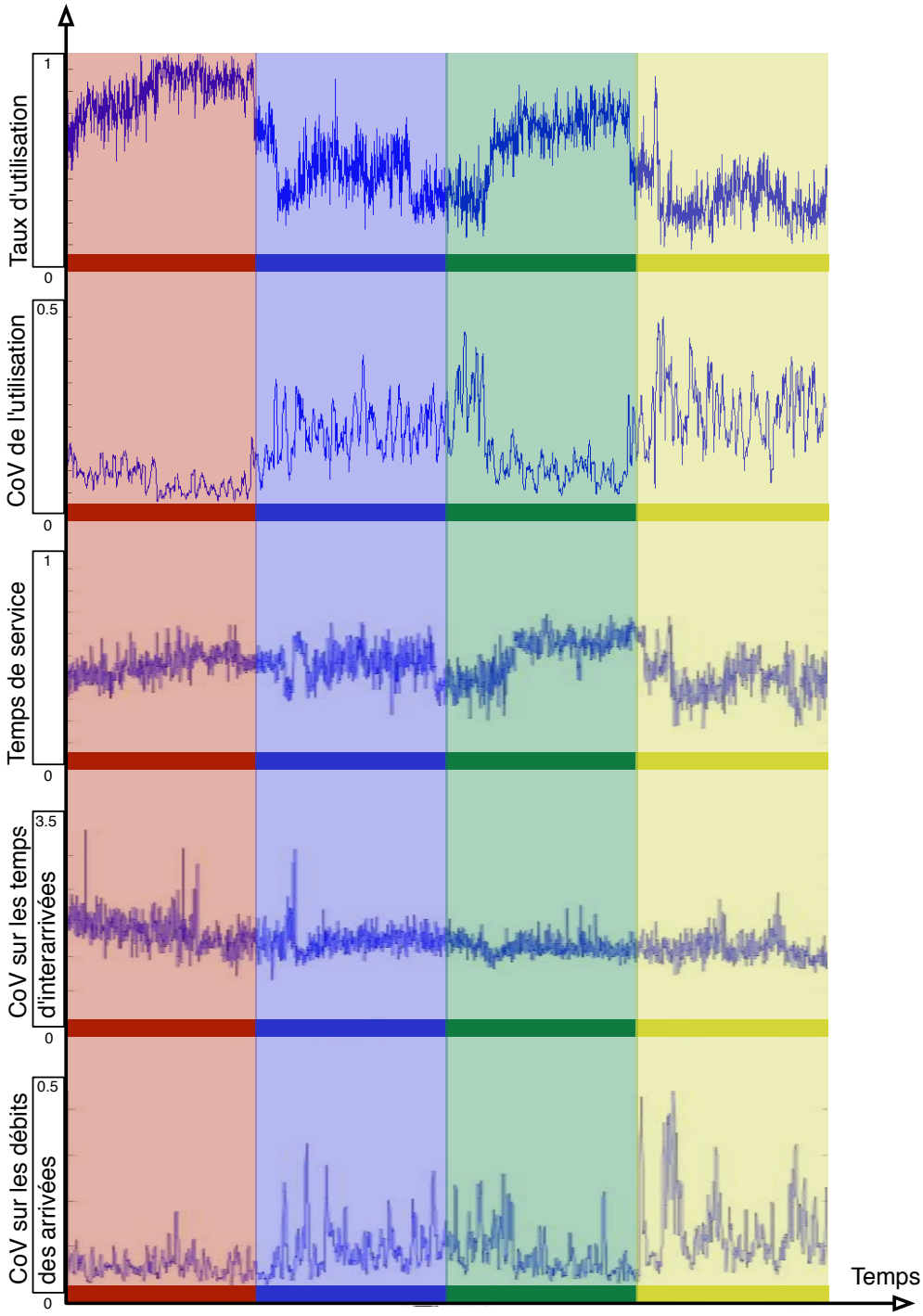


FIGURE 3.4 – Evolution du comportement de quelques facteurs de la trace en fonction du temps.

Principe. Un modèle de file d'attente monoserveur (*cf.* figure 3.5) est une entité constituée d'une file d'attente (buffer) et d'un serveur unique. Les clients arrivent de l'extérieur, patientent éventuellement dans le buffer, reçoivent un service, puis quittent la file.

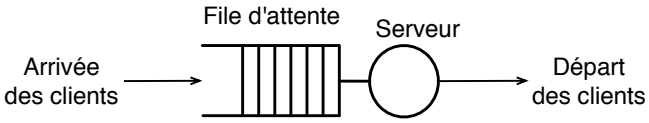


FIGURE 3.5 – File d'attente monoserveur.

Afin de spécifier complètement une file monoserveur, il faut caractériser le processus d'arrivée des clients, la distribution du temps de service, la taille du buffer, ainsi que la structure de la discipline de service de la file d'attente (dans notre étude, la discipline de service de la file est *FIFO*).

Modèles de type file d'attente. Dans la suite, nous présentons successivement les files *M/M/1*, *M/M/1/K*, *M/GI/1*, *M/GI/1/K*, *GI/M/1*, *GI/M/1/K*, *GI/GI/1* et *GI/GI/1/K*.

[1] *M/M/1* :
Une file de type *M/M/1* représente un système de file d'attente avec un processus d'arrivée des clients dans la file qui suit un processus de Poisson et un temps de service d'un client modélisé par une variable aléatoire ayant une distribution exponentielle.

[2] *M/M/1/K* :
Une file de type *M/M/1/K* est un système de file d'attente *M/M/1* avec une taille finie K pour le buffer du lien.

[3] *M/GI/1* :
Une file *M/GI/1* représente un système de file d'attente où le processus d'interarrivée des clients est modélisé par une variable aléatoire ayant une distribution exponentielle et le temps de service d'un client est distribué selon une variable aléatoire générale.

[4] *M/GI/1/K* :
Une file de type *M/GI/1/K* est un système de file d'attente *M/GI/1* avec une taille finie K pour le buffer du lien.

[4] *GI/M/1* :
Une file *GI/M/1* peut être vue comme un « duale » d'une file *M/GI/1*, puisque les temps d'interarrivée et de service sont respectivement exponentiels et généraux pour une file *M/GI/1*, généraux et exponentiels pour une file *GI/M/1*.

[4] *GI/M/1/K* :
Une file de type *GI/M/1/K* est un système de file d'attente *GI/M/1* avec une taille finie K pour le buffer du lien.

[5] *GI/GI/1* :
Dans une file *GI/GI/1* le processus d'interarrivée ainsi que le temps de service sont distribués selon une variable aléatoire générale.

[6] *GI/GI/1/K* :
Une file de type *GI/GI/1/K* est un système de file d'attente *GI/GI/1* avec une taille finie K pour le buffer du lien.

Il existe aussi d'autres modèles de files d'attente (par exemple les files *M/M/C*, les files *M/GI/C*, etc.), ainsi que des extensions possibles des modèles des files, comme par exemple les techniques d'identification à des lois de type « phase », que nous ne présentons pas dans cette thèse car ils modélisent des comportements inutiles dans notre cas d'étude.

Analyse des modèles de type file d'attente. Nous analysons les différents modèles de type file d'attente décrits précédemment afin de choisir un modèle fiable, simple et capable de modéliser adéquatement le comportement d'un lien réseau.

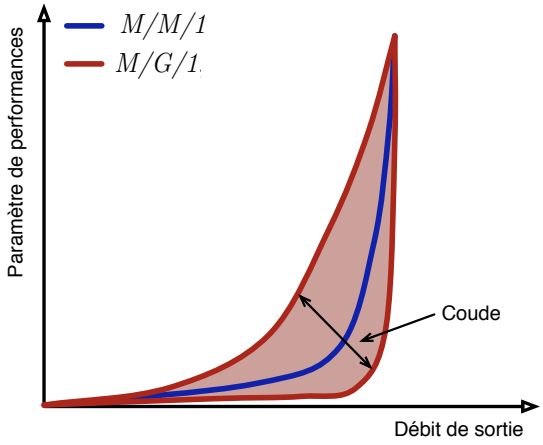


FIGURE 3.6 – Comportement qualitatif des modèles de type file d'attente.

[1] *M/M/1* et *M/M/1/K* :
Ces systèmes de file d'attente sont les plus simples à résoudre. En revanche, ils n'arrivent pas à bien modéliser le comportement du trafic réseau transitant sur un lien. La figure 3.6 montre que ces files d'attente ne sont pas flexibles et n'arrivent pas à couvrir la variabilité des paramètres de performances (*cf.* figure 3.3).

[2] *M/GI/1* et *M/GI/1/K* :
Ces modèles de files peuvent reproduire le comportement d'un lien réseau et sont simples à résoudre (comparé aux systèmes de file d'attente de type *GI/GI/1* et *GI/GI/1/K*). Ces modèles peuvent englober un large spectre de performances en faisant varier le coefficient de variation sur la distribution du temps d'interarrivées entre les paquets successifs.

[3] *GI/M/1* et *GI/M/1/K* :
Ces modèles sont capables reproduire le comportement d'un lien réseau et peuvent également englober un large spectre de performances. Cependant, ils sont plus compliqués à résoudre.

[4] *GI/GI/1* et *GI/GI/1/K* :
Ces systèmes de type file d'attente sont certainement capables de modéliser adéquatement le comportement du trafic transitant sur un lien réseau. En revanche, ces systèmes sont, la plupart du temps, très difficiles à résoudre (voir parfois impossible à résoudre).

Choix de modèles de type file d'attente. Pour conclure, nous avons décidé de considérer uniquement les modèles de type d'attente monoserveur car dans un routeur réseau un seul paquet peut être transmis à la fois. Dans la suite de notre travail, notre choix s'est porté sur les modèles de file d'attente moneserveur de type *M/GI/1* et *M/GI/1/K*.

3.2 Nouvelle solution KBAC

Principe. Notre solution de contrôle d'admission repose (*i*) sur l'élaboration en continu d'un plan de connaissance (*ii*) et sur la modélisation du comportement d'un lien réseau par une file monoserveur.

KBAC vs MBAC. Contrairement aux solutions MBAC (*cf.* section 2.2.2 du chapitre 2), notre solution comprend une étape supplémentaire, le plan de connaissance, qui s'interpose entre l'algorithme de mesure et l'algorithme de décision. Nous détaillons maintenant chacune de ces étapes.

3.2.1 Algorithme de mesure

Fenêtre de temps de mesure. L'algorithme de mesure surveille continuellement l'activité du lien de communication de façon à collecter des données de mesure. Ces données sont mesurées sur une courte fenêtre de temps de mesure W_T. Ces mesures à court terme prennent en compte le comportement "instantané" du trafic transitant.

Points de mesure. Pour chaque fenêtre de temps de longueur W_T, nous mesurons le débit moyen en sortie du lien du trafic transitant, noté par X (paquets/ms), avec un autre paramètre de performance QoS, noté par P. Ce paramètre de QoS peut être un délai d'attente moyen des paquets dans le buffer du lien ou un taux de perte des paquets. Les valeurs mesurées de X et P sont rassemblées en une paire de mesures. Nous nommons ces couples de mesures, (X, P), points de mesure.

3.2.2 Construction du plan de connaissance

Principe. Un plan de connaissance (*cf.* section 1.2 du chapitre 1) définit un ensemble de mesures, diverses dans l'espace du réseau et dans le temps, dont les valeurs reflètent de façon collective le comportement d'un réseau. Sa constitution doit permettre une meilleure gestion du réseau. Ici, nous en proposons une définition pensée pour le contexte du contrôle d'admission sur un lien de communication.

Points de fonctionnement. Notre plan de connaissance comprend un ensemble de k points, que l'on désigne comme points de fonctionnement. Chacun de ces points correspond à un couple de valeurs associant (i) un débit moyen en sortie du lien et (ii) un délai d'attente moyen des paquets dans le buffer du lien (respectivement, un taux de perte). Ces points sont des agrégats (*clusters*) dont les coordonnées seront régulièrement mises à jour par la collecte de nouveaux points de mesure. Pour cela, nous mesurons, sur toutes les fenêtres de temps de longueur W_T, quels ont été le débit et le délai d'attente (respectivement le taux de perte) observés sur le lien. Puis, toutes les T_{kp} secondes (c'est-à-dire une fois collectées 100 nouvelles mesures pour $W_T = 200$ ms et $T_{kp} = 20$ secondes), nous calculons les nouvelles coordonnées des points de fonctionnement.

Dans nos expériences, nous limitons le nombre de points de fonctionnement à 10 ($k = 10$).

Clusterisation. Pour mener l'étape de clusterisation, nous nous appuyons sur l'algorithme *k-means*. Au final, ce plan de connaissance permet de produire un ensemble de k points de fonctionnement représentant une courbe de performances du lien.

La figure 3.7 illustre ce principe pour un exemple avec un lien de 10 Mb/s soumis à un trafic issu d'une trace réelle. Cette figure représente les nouveaux points de mesures collectés et les nouvelles cordonnées des points de fonctionnement calculés. Elle montre aussi la courbe de performances du lien. Premièrement, nous remarquons que l'allure de cette courbe est conforme à nos attentes et aux résultats issus de la théorie des files d'attente. Ces observations sont vérifiées par le comportement qualitatif suivant : à mesure que le débit moyen en sortie progresse, le délai d'attente (respectivement le taux de perte) s'accroît. Notons que selon la nature du lien et du trafic soumis en entrée du lien, les caractères quantitatifs de ces courbes de performances diffèrent (*cf.* section 3.1.2).

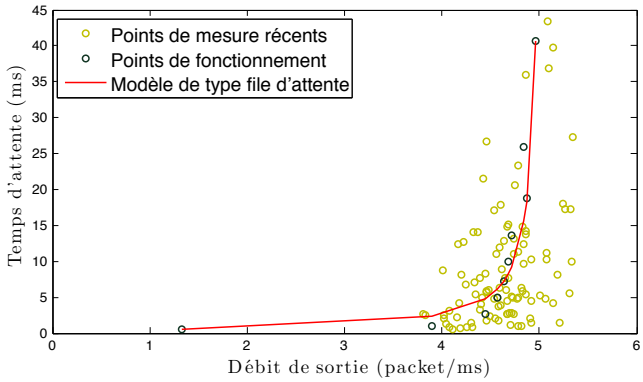


FIGURE 3.7 – Exemple d'un plan de connaissance comprenant les mesures, les points de fonctionnement et le modèle type file d'attente associé.

Modélisation du plan de connaissance. Nous nous appuyons sur une méthode rapide et automatique pour associer un modèle de type file d'attente au comportement du lien réseau tel qu'il est décrit par les points de fonctionnement [Begin et al., 2010]. Cette méthode de modélisation, nommée HLM (*High Level Modeling*) permet la modélisation rapide et automatique de systèmes opérationnels de type « boîte noire » pour lesquels on dispose uniquement de mesures. Le principe de la méthode HLM consiste à rechercher parmi un ensemble présupposé de modèles génériques, si une fois correctement calibré, l'un de ces modèles permet de reproduire le comportement d'entrée/sortie du système considéré tel qu'il est décrit par les mesures.

Enfin, cette méthode délivre une file monoserveur qui reproduit au plus près les performances observées du lien (*cf.* figure 3.7). Selon la section 3.1.3, nous limitons la recherche du modèle aux files de type *M/GI/1* (respectivement *M/GI/1/K*) lorsqu'on s'intéresse au délai d'attente (respectivement au taux de perte). La figure 3.7 montre un exemple de la modélisation du plan de connaissance par une file monoserveur. Nous observons qu'une file de type *M/GI/1*, avec un taux de service de 5,01 paquets/ms et un coefficient de variation de 2,02, reproduit adéquatement le comportement du lien réseau tel qu'il est décrit par les points de fonctionnement.

3.2.3 Algorithme de décision

Prédiction de la performance. La décision de notre contrôle d'admission d'accepter ou de rejeter un nouveau flux se base sur une prédiction de la performance. Cette prédiction repose sur l'exploitation du modèle de type file d'attente trouvé. Nous réalisons une projection de charge (*capacity planning*) sur ce nouveau modèle afin d'évaluer quelles seraient les performances du lien réseau si la charge soumise en entrée du lien venait à augmenter. La résolution du modèle pour ce niveau de charge nous permet d'estimer le risque que, une fois le nouveau flux accepté, la nouvelle valeur du délai d'attente (respectivement du taux de perte) dépasse le seuil exigé par la QoS des flux.

Lorsqu'un flux cherche à entrer sur un lien de communication en demandant un débit crête r, la prédiction de la performance, notée $\widehat{P}$ est calculé de la manière suivante :

$$\widehat{P} = f_P(\widehat{X} + r) \tag{3.1}$$

où f_p définit l'évolution de P en fonction du débit du lien, et $\widehat{X}$ reflète le débit ajusté du trafic transitant. Notons que nous utilisons une valeur ajustée du débit afin d'éviter le comportement volatile de la valeur de X, mesurée sur une petite fenêtre de temps W_T. Nous expliquerons plus tard la façon dont la valeur de $\widehat{X}$ est calculée.

Test d'admission. Un nouveau flux est admis si :

$$\widehat{P} + \alpha\widehat{\sigma}_p < P^* \tag{3.2}$$

où P^* est une constante représentant la contrainte de la QoS (cette constante peut être un temps si on considère le délai d'attente comme critère de QoS ou un taux dans le cas du taux de perte), $\widehat{\sigma}_p$ represente l'écart-type de $\widehat{P}$, fourni par le modèle type file d'attente associé à la courbe de performances, et α est une constante spécifiant la probabilité de la violation de la performance.

Dans la forme actuelle de notre solution KBAC, nous considérons que $\widehat{\sigma}_p = \widehat{P}$. A savoir, le calcul de l'écart-type de $\widehat{P}$, $\widehat{\sigma}_p$, est une tâche complexe à réaliser. Ce calcul peut être parfois possible à obtenir mais, la plupart du temps, il représente une tâche impossible [Gross and Harris, 1985]. Dans ces conditions, afin de simplifier le calcul de $\widehat{\sigma}_p$, nous considérons que la moyenne est égale à l'écart type, ce qui est vrai, si on suppose une distribution exponentielle de moyenne $\widehat{P}$.

Nous détaillons à présent les paramètres énumérés ci-dessus :

3.2.3.1 Choix de la valeur de α

Contrôle d'admission probabiliste. Notre solution KBAC garantit le critère de QoS de manière probabiliste. Plus précisément, la valeur de α est définie de sorte que les flux transitant sur le lien de communication ne dépassent pas la cible de QoS, avec une probabilité Q. Nous définissons α en utilisant l'inégalité de Tchebychev [All,]. Dans notre étude, nous définissons la valeur de α à 1.7, de sorte que $Q = 0.75$.

3.2.3.2 Calcul du débit ajusté $\widehat{X}$

Nous rappelons que $\widehat{X}$ reflète le débit ajusté du trafic transitant utilisé dans la prédiction de la performance, $\widehat{P}$, en utilisant l'équation 3.1. Le débit ajusté $\widehat{X}$ est déterminé sur les M dernières fenêtres de mesure de longueur W_T comme suit :

$$\widehat{X} = \frac{1}{M}\sum_{m=1}^{M} X_m + \sum_{m=1}^{M} \frac{m}{M} \times \sum_{f=1}^{F_m} r_m^f \tag{3.3}$$

où X_m correspond au débit moyen en sortie de lien obtenu sur la m^e fenêtre de mesure, F_m représente le nombre des flux acceptés sur la m^e fenêtre de mesure, et r_m^f correspond au débit crête estimé du f^e flux sur la m^e fenêtre de mesure.

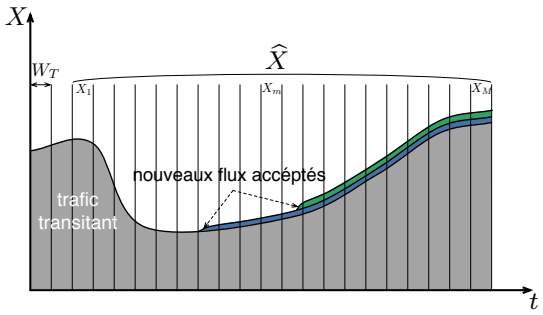


FIGURE 3.8 – Calcul du débit ajusté $\widehat{X}$.

La figure 3.8 illustre le calcul du débit ajusté en sortie de lien. Ce type de calcul permet d'avoir une valeur plus lisse du débit en sortie que le débit d'un seul point de mesure X. En particulier cette méthode utilise une pondération des débits crêtes des nouveaux flux acceptés auquel s'ajoute la valeur moyenne du débit de sortie.

La valeur de $\widehat{X}$ doit être régulièrement mise à jour. Dans nos expériences, le calcul du débit de sortie ajusté est répété sur chaque fenêtre de mesure W_T. C'est cette valeur qui est utilisée dans notre algorithme de décision.

Il faut aussi noter que la valeur de $\widehat{X}$ peut aussi être mise à jour à l'intérieur d'une fenêtre de mesure. A chaque fois qu'un nouveau flux est accepté, la valeur de $\widehat{X}$ est alors modifiée et prend comme valeur le débit de sortie ajusté courant auquel s'ajoute le débit crête du flux entrant, r, pour être $\widehat{X} + r$. De cette manière, cette mise à jour permet de tenir compte des événements de type *"Flash Crowd"* (*i.e.*, plusieurs nouveaux flux qui arrivent dans une fenêtre de mesure W_T).

3.2.4 Anticiper le risque d'inondation du nombre de mesures

Comme décrit précédemment, notre méthode, pour opérer, maintient en temps réel une vue mise à jour du comportement du lien. Cette connaissance est obtenue grâce aux points de mesure. Compte tenu du nombre énorme de points de mesure collectés (*e.g.*, 300 nouveaux points de mesure par minute pour $W_T = 200$ ms), notre solution KBAC peut connaître rapidement un problème d'inondation du nombre de points de mesure lors du calcul des points de fonctionnement.

Afin d'éviter une explosion du nombre de mesures dans le calcul des agrégats, nous limitons le nombre total de points de mesure à n. Nous rappelons que ces points de mesure sont utilisés pour calculer les points de fonctionnement.

3.2.5 Diversité des points de fonctionnement

Néanmoins, la limitation du nombre de points de mesure peut causer une perte d'information du fait que les n points considérés risquent d'appartenir à la même zone de débit. Plus précisément, il y a une forte probabilité que les points de mesure collectés, pendant une période de temps, aient, quasiment, la même valeur de débit, et donc ils appartiennent à la même zone de débit.

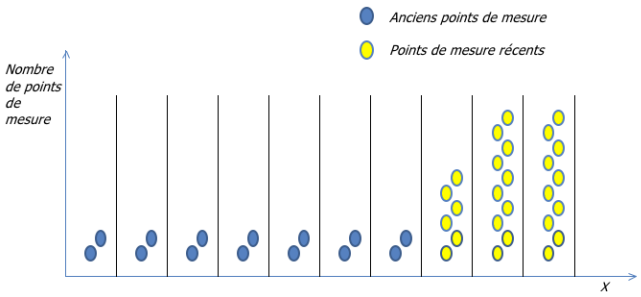


FIGURE 3.9 – Découpage de l'intervalle de débit de sortie du lien afin d'assurer la diversité des *points de fonctionnement* en S intervalles identiques.

Pour répondre à cette problématique, nous découpons l'intervalle de débit $[0, X_{max}]$ en S intervalles de même longueur (*cf.* figure 3.9). Chaque point de mesure appartient à l'un de ces intervalles. Après chaque fenêtre de mesure W_T, on remplace le plus ancien point de mesure par le dernier point de mesure collecté, tout en assurant qu'il y a au moins n_s points de mesure dans chaque intervalle de débit.

Cette méthode nous permet d'anticiper le risque d'inondation du nombre de mesures, tout en assurant la diversité des points de fonctionnement qui est un point critique dans notre solution pour bien découvrir le modèle de file d'attente qui régira notre contrôle d'admission. La figure 3.7 illustre la diversité des points de fonctionnement. Bien que les points de mesure les plus récents se localisent au plus haut niveau du débit, les points de fonctionnement sont largement distribués et couvrent un éventail plus large.

Notons que notre solution KBAC nécessite une période d'apprentissage afin d'assurer une distribution assez large des points de fonctionnement (la diversité des points de fonctionnement est ultime pour la méthode HLM afin d'avoir une modélisation adéquate du comportement du lien réseau). Cette période d'apprentissage dure généralement moins de quelques minutes pour un lien actif.

Dans nos expériences, nous limitons le nombre de points de mesures à 1000 ($n = 1000$) et nous choisissons $S = 6$ et $n_s = 20$.

3.2.6 Cohérence temporelle

Plus globalement, notre solution repose sur l'idée que les performances qui ont été observées dans le passé (et intégrées au plan de connaissance) sont un élément utile pour estimer les performances futures d'un système. Plus formellement, nous faisons l'hypothèse de cohérence suivante :

$$\forall (t_1, t_2) \in [t, t + T_{kp}]^2,\ X_{t_1} = X_{t_2} \Rightarrow P_{t_1} \simeq P_{t_2} \tag{3.4}$$

où X_{t_1} (*resp.*, X_{t_2}) représente le niveau de débit moyen en sortie du lien au temps t_1 (*resp.*, t_2) et P_{t_1} (*resp.*, P_{t_2}) représente le paramètre de performance (délai d'attente associé ou taux de perte) observé sur le lien au temps t_1 (*resp.*, t_2).

Cette relation paraît naturelle, toutefois, il est clair que plus X_{t_1} et X_{t_2} seront éloignés dans le temps, plus cette relation risquera de s'altérer, en particulier si, entretemps, le profil statistique du trafic (e.g., à cause de sa sporadicité ou son auto-corrélation) a largement varié (*cf.* figure 3.3). Notre procédure de construction des points de fonctionnement tient compte de cette dernière remarque car, progressivement, les mesures les plus anciennes sont retirées et remplacées par des mesures récentes de l'activité du lien.

3.3 Conclusions

Dans ce chapitre, nous avons présenté un nouveau contrôle d'admission basé sur (*i*) l'élaboration en continu d'un plan de connaissance (*ii*) et sur la modélisation du comportement d'un lien réseau par une file monoserveur.

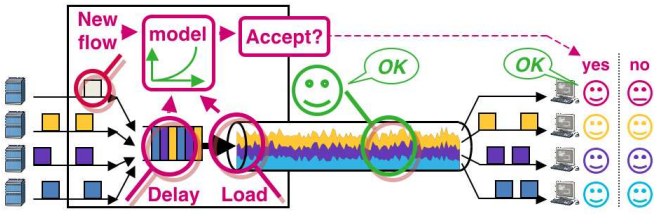


FIGURE 3.10 – Nouvelle solution KBAC proposée.

Notre solution (*cf.* figure 3.10) se distingue des solutions concurrentes par sa simplicité d'implantation et de mise œuvre. Ainsi, aucun calibrage a priori n'est requis pour son utilisation. Le plan de connaissance et sa modélisation se construisent automatiquement à partir de mesures « réelles » collectées sur le réseau, ce qui permet de s'affranchir des étapes de calibrage souvent difficiles.

4 Evaluation de performances

4.1 Scénarios considérés
- 4.1.1 Trafic initial
- 4.1.2 Les flux VBR

4.2 Elements de comparaison pour le contrôle d'admission
- 4.2.1 Solutions existantes
- 4.2.2 Oracle

4.3 Calibrage des contrôles d'admission
- 4.3.1 Calibrage des contrôles d'admission considérés

4.4 Estimation du débit crête des flux entrants

4.5 Performances
- 4.5.1 Cas d'une source Poisson
- 4.5.2 Cas d'une source PPBP
- 4.5.3 Cas d'une trace réelle (Trace 1)
- 4.5.4 Cas d'une trace réelle (Trace 2)

4.6 Conclusions

Le travail présenté dans ce chapitre a été publié à "37th and 36th Annual IEEE Conference on Local Computer Networks (LCN)" en 2012 [1] et en 2011 [2], à "Algotel" en 2012 [6] et à "CFIP" en 2011 [7]. De plus, une implémentation d'un nouveau générateur de trafic, publié à "SIMUTools" en 2011 [3], à été soumis au simulateur réseau ns-3 en 2011 [10]. En outre, deux rapports de recherche ont été déposés à "INRIA" en 2012 [8, 9].

Résumé. Nous avons évalué les performances de notre nouvelle solution de contrôle d'admission KBAC par simulation en considérant diverses conditions possibles de trafic. De plus, afin d'apprécier les performances de notre solution, nous avons implanté trois autres solutions de contrôle d'admission ainsi qu'une procédure permettant de déterminer le nombre maximal de flux pouvant être acceptés, et donc d'atteindre le taux d'utilisation maximal, tout en satisfaisant au critère de QoS (*i.e.*, ni faux positifs ni faux négatifs). D'après nos résultats d'expériences, notre solution permet d'atteindre un contrôle d'admission ni trop conservateur, ni trop permissif.

4.1 Scénarios considérés

Quel objectif ? L'évaluation de performances d'un contrôle d'admission peut traiter de différents aspects. On peut considérer le surcoût en CPU ou en mémoire pour les nœuds du réseau, la simplicité de configuration, la pertinence/qualité des décisions prises, etc. Il nous a semblé plus pertinent dans cette thèse de s'intéresser surtout au couplage entre la configuration d'un contrôle d'admission et la pertinence de ses décisions. Plus précisément, nous souhaitons mettre en lumière la capacité des contrôles d'admission à atteindre le taux d'utilisation maximal du lien tout en respectant une contrainte donnée sur la qualité de service.

Simulateur réseau ns-3. Nous évaluons les performances des contrôles d'admission sous l'angle de la simulation car elle nous permet d'évaluer des scénarios variés. Notre choix s'est porté sur le simulateur réseau ns-3 [ns3, 2008] (*Network simulator 3*).

Topologie simulée. Nous considérons un lien de capacité C égale à 10 Mb/s. La taille de la file à l'entrée de ce lien est fixée à 60 ms (soit 393 paquets si nous considérons que tous les paquets sont de longueur 190 octets). La discipline de service de la file est FIFO (*First In First Out*), les pertes surviennent lorsqu'un paquet arrivant trouve la file pleine (*Drop-Tail*) et le taux d'erreur (*Bit Error Rate*) est supposé nul.

Pourquoi un lien ? Notre évaluation de performance est effectuée sur un lien réseau car notre solution de contrôle d'admission est pensée pour fonctionner à l'échelle d'un lien.

L'objectif de nos scénarios est d'évaluer les performances de notre solution KBAC sur un lien de communication. Afin d'étendre la solution KBAC sur la globalité du réseau, il suffit de répéter ce contrôle d'admission sur chacun des liens traversés par le flux.

Trafic transitant. Le lien est soumis à une source initiale à laquelle vont tenter de venir s'ajouter des flux VBR (figure 4.1). Par soucis de généralité, nous avons considéré plusieurs possibilités pour modéliser la source initiale : un processus Poisson, un processus PPBP [Zukerman et al., 2003] qui présente un fort degré d'autosimilarité et deux traces collectées sur des réseaux réels. Notons que ce trafic initial n'est pas soumis au contrôle d'admission. Il peut correspondre, par exemple, à un trafic prioritaire, à un trafic de réseau privé virtuel (*VPN : virtual private network*) soumis à un contrôle d'accès faible ou nul, ou à des flux déjà acceptés.

Dans notre étude, le trafic de fond sur ce lien n'est donc pas seulement une agrégation de flux individuels acceptés (et ayant le même profil) comme c'est le cas dans les autres études, mais un trafic de fond initial (dont on peut maîtriser les caractéristiques) auquel s'ajoutent les flux individuels acceptés par le contrôle d'admission. Cette approche permet de tester notre solution de contrôle d'admission

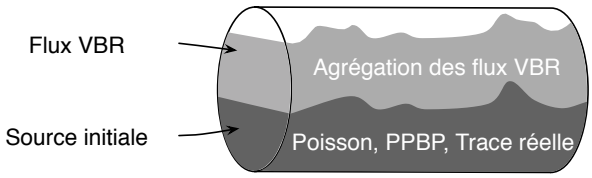


FIGURE 4.1 – Trafic transitant sur un lien de communication, modélisé par une source initiale et des flux VBR.

sous une diversité de trafic peu explorée dans les études précédentes et répondant à certaines propriétés statistiques des réseaux IP (comme par exemple, le degré d'autocorrélation, dépendance à long terme dans le trafic, etc.). Nous détaillons maintenant comment nous modélisons chacun des deux processus impliqués dans le trafic transitant.

4.1.1 Trafic initial

Clairement, les décisions prises par un contrôle d'admission sont toujours étroitement liées au trafic agrégé circulant sur le lien et précédant l'arrivée du nouveau flux à accepter (ou refuser). Par la suite, nous appellerons ce trafic agrégé, le trafic de fond. Il n'existe pas actuellement de modèle universel pour représenter le trafic de fond. Ceci s'explique par le fait que les profils des flux et par conséquent le profil du trafic agrégé circulant dans un réseau varie fortement selon le type de réseau et l'emplacement du lien considéré. Dans les travaux de comparaison existants sur le contrôle d'admission, comme [Jamin et al., 1997, Breslau et al., 2000a, Nevin et al., 2008], le trafic de fond transitant sur un lien à l'arrivée d'un nouveau flux correspond à l'agrégation des flux précédemment acceptés ayant tous le même profil. Seule l'étude de [Breslau et al., 2000a] considère une agrégation de flux ayant des profils hétérogènes mais tous ces flux sont lissés en entrée par le même seau à jetons. Contrairement à ces travaux, notre étude considère diverses conditions possibles du trafic de fond. Plus précisément, nous avons choisi de représenter le trafic initial par une source Poisson, une source PPBP, une trace réelle collectée à l'Université de Stuttgart [Sass, 2004], notée Trace 1, et une autre trace réelle collectée à l'Université de Brescia [II,], notée Trace 2.

4.1.1.1 Cas d'une source Poisson

Le trafic de fond est composé d'une source Poisson avec une intensité moyenne de 2,5 Mb/s émettant des paquets de taille 1500 octets.

Propriétés d'une source Poisson. Afin de représenter les propriétés de chaque source, nous montrons, d'une part, l'évolution du taux d'utilisation en fonction du temps, et d'autre part, la répartition des débits des arrivées des paquets.

Premièrement, la figure 4.2 illustre l'évolution du taux d'utilisation en fonction du temps pour une source Poisson avec un taux d'utilisation moyen du lien de $0,25$.

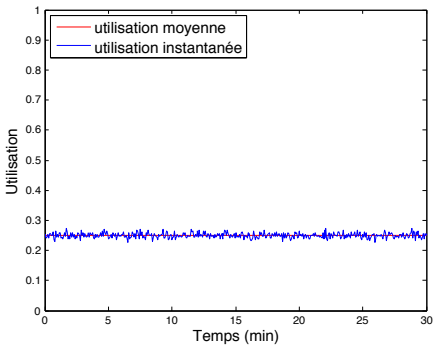


FIGURE 4.2 – Evolution du taux d'utilisation en fonction du temps pour une source Poisson.

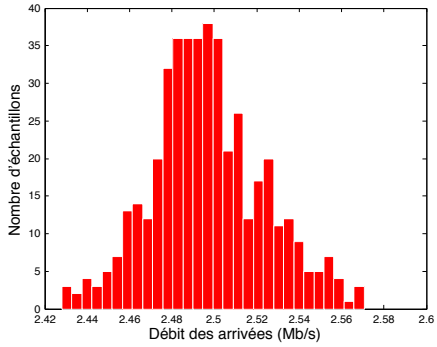


FIGURE 4.3 – Répartition des débits des arrivées pour une source Poisson.

Les résultats montrent que les taux d'utilisation « instantanés » mesurés toutes les 4 s subissent des déviations très faibles (typiquement entre 0,24 et 0,26) autour de la valeur moyenne du taux d'utilisation (*i.e.,* 0,25). D'autre part, la figure 4.3 illustre la répartition des débits des arrivées pour une source Poisson. Elle montre que les débits des arrivés pour cette source sont quasiment stables.

4.1.1.2 Cas d'une source PPBP

Principe. Le processus PPBP (*Poisson Pareto Burst Process*) [Zukerman et al., 2003] représente le comportement d'une infinité de sources On-Off indépendantes avec des durées ON distribuées selon une loi de Pareto. Le processus PPBP est étroitement lié au modèle de file d'attente $M/GI/\infty$ proposé par Cox [Cox and Isham, 1980, Cox, 1984].

La figure 4.4 illustre le processus PPBP décrit ci-dessus. Dans le modèle PPBP, les arrivées des flux suivent un processus de Poisson avec débit λ_p, et la durée de vie de chaque flux suit une distribution de Pareto caractérisée par le paramètre de Hurst H, typiquement entre 0,5 et 0,9, et une moyenne T_{on}. Chaque arrivée de flux est modélisée par un flux CBR de débit binaire constant r.

En se basant sur la loi de Little [Allen, 1990], le nombre moyen des arrivées est

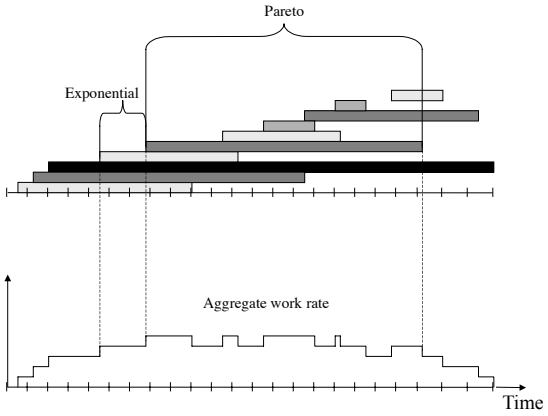


FIGURE 4.4 – Processus PPBP (extrait de la thèse de T. Neame [Zukerman et al., 2003]).

donné par :

$$E[n] = T_{on} \times \lambda_p \tag{4.1}$$

Puisque chaque rafale donne naissance à un flux avec un débit binaire constant, il est alors facile de calculer le débit total du trafic PPBP, noté par λ :

$$\lambda = T_{on} \times \lambda_p \times r \tag{4.2}$$

Paramétrage utilisé. Dans nos expériences, nous choisissons une durée moyenne de $T_{on} = 200$ ms et un paramètre de *Hurst*, $H = 0{,}5$. Chaque source génère du trafic CBR avec un débit fixe de 0,1 Mb/s. L'intensité moyenne du trafic PPBP est de 2,5 Mb/s émettant des paquets de taille 1500 octets.

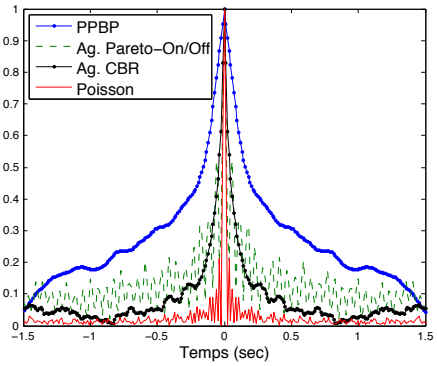


FIGURE 4.5 – Fonction d'autocorrélation pour une source Poisson, pour une source PPBP, pour une superposition de 100 flux CBR indépendants et pour une superposition de 20 flux On/Off Pareto.

Propriétés d'une source PPBP. Les propriétés statistiques de ces processus, notamment leur degré d'autocorrélation, diffèrent largement.

La figure 4.5 illustre le degré d'autocorrélation pour quatre processus différents ayant tous la même intensité. Les résultats montrent que le degré d'autocorrélation est nul pour une source Poisson, modéré pour l'agrégation de 100 flux CBR indépendants ou pour l'agrégation de 20 flux On/Off Pareto indépendants et important pour une source PPBP.

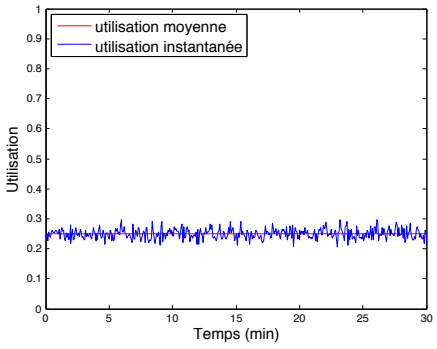


FIGURE 4.6 – Evolution du taux d'utilisation en fonction du temps pour une source PPBP.

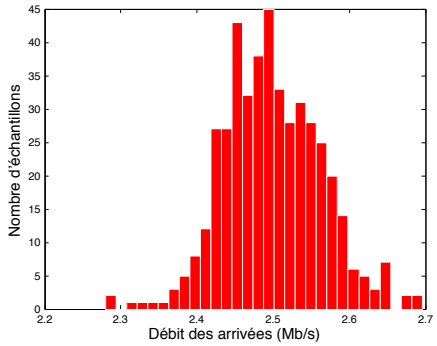


FIGURE 4.7 – Répartition des débits des arrivées pour une source PPBP.

La figure 4.6 illustre l'évolution du taux d'utilisation en fonction du temps pour une source PPBP. Elle montre que le taux d'utilisation « instantané » varie fortement autour de la valeur moyenne du taux d'utilisation (typiquement il varie entre 0,22 et 0,27).

D'autre part, la figure 4.7 illustre la répartition des débits des arrivées pour une source PPBP. Les résultats montrent que les débits des arrivés pour cette source subissent des variations modérées.

4.1.1.3 Cas d'une trace réelle (Trace 1)

La première trace réelle, notée Trace 1, a été collectée par l'Université de Stuttgart [Sass, 2004], le dimanche 31 Octobre 2004 entre 18 heures et 22 heures, sur un lien 100 Mb/s du réseau "SelfNet".

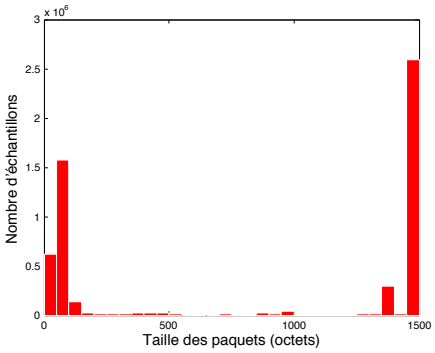


FIGURE 4.8 – Répartition des tailles des paquets pour une trace réelle (Trace 1).

La figure 4.8 représente la répartition des tailles des paquets pour cette trace réelle (Trace 1). Les résultats montrent que presque la moitié des paquets sont de petites tailles (typiquement moins de 100 octets), et l'autre moitié sont de grandes tailles (*i.e.,* 1500 octets).

Dans nos expériences, nous ajustons la trace à un lien de 10 Mb/s en faisant le passage à l'échelle de telle sorte que son débit moyen de paquets transmis est égal à 2,5 Mb/s.

Propriétés de la trace réelle (Trace 1). Pour le cas des traces réelles, nous choisissons de représenter les fonctions d'autocorrélations sur différents intervalles de temps afin de mettre en évidence l'évolution du comportement de la trace en fonction du temps. Plus précisément, nous calculons toutes les 2 minutes une fonction d'autocorrélation de la trace. Par conséquent, nous obtenons, après une durée de 30 minutes, 15 fonctions d'autocorrélations. Dans la suite, nous montrons uniquement trois fonctions d'autocorrélations par trace.

La figure 4.9 illustre le degré d'autocorrélation de la trace comparé à quatre autres processus (*cf.* figure 4.5) ayant tous la même intensité. Les résultats montrent que le degré d'autocorrélation de la trace subit trois comportements différents. D'une part, le premier graphique de gauche de la figure 4.9 montre que le degré d'autocorrélation est quasiment nul (typiquement le degré d'autocorrélation est proche de celui d'une source Poisson). D'autre part, le degré d'autocorrélation est modéré pour les deux autres graphiques de la figure 4.9. Ces résultats mettent en valeur la forte dynamicité dans le trafic qui change de nature en fonction du temps.

La figure 4.10 illustre l'évolution du taux d'utilisation en fonction du temps pour le cas de la trace réelle Trace 1. Elle montre que le taux d'utilisation « instan-

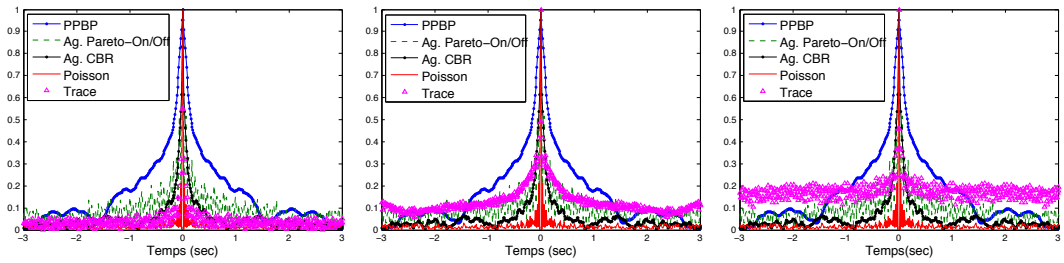


FIGURE 4.9 – Fonctions d'autocorrélations pour une trace réelle (Trace 1), pour une source Poisson, pour une source PPBP, pour une superposition de 100 flux CBR indépendants et pour une superposition de 20 flux On/Off Pareto sur différents intervalles de temps.

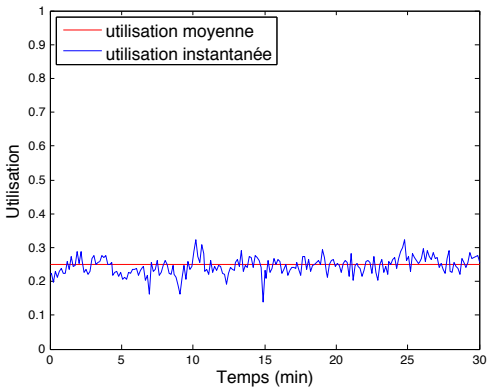


FIGURE 4.10 – Evolution du taux d'utilisation en fonction du temps pour une trace réelle (Trace 1).

tané » varie entre 0,2 et 0,28. Ces variations subissent des comportements différents en fonction du temps.

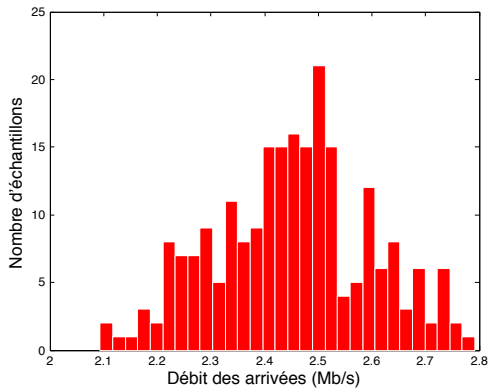


FIGURE 4.11 – Répartition des débits des arrivées pour une trace réelle (Trace 1).

D'autre part, la figure 4.11 illustre la répartition des débits des arrivées pour le cas de la trace réelle Trace 1. Les résultats montrent que les débits des arrivés pour cette source subissent des variations modérées qui diffèrent des comportements de la figure 4.3 et de la figure 4.7.

4.1.1.4 Cas d'une trace réelle (Trace 2)

La deuxième trace réelle, notée Trace 2, a été collectée par l'Université de Brescia [II,] sur trois jours consécutifs en Septembre/Octobre 2009, sur un lien de 100 Mb/s du réseau du campus.

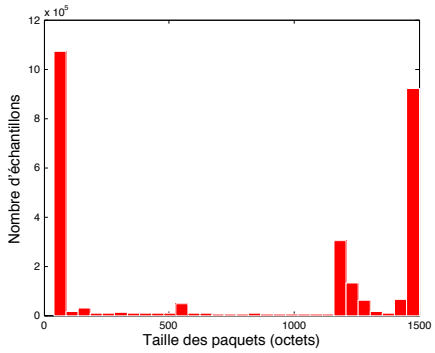


FIGURE 4.12 – Répartition des tailles des paquets pour une trace réelle (Trace 2).

La figure 4.12 représente la répartition des tailles des paquets pour cette trace réelle (Trace 2). Les résultats obtenus concordent avec ceux qui sont illustrés dans la figure 4.8. Nous remarquons que la majorité des paquets sont répartis entre des petites et des grandes tailles.

Dans nos expériences, nous ajustons la trace à un lien de 10 Mb/s en faisant le passage à l'échelle de telle sorte que son débit moyen de paquets transmis est égal à 2,5 Mb/s.

Propriétés de la trace réelle (Trace 2). La figure 4.13 illustre le degré d'autocorrélation de la trace. Les résultats montrent que le degré d'autocorrélation de la trace est modéré pour le premier graphique de gauche de la figure 4.13, important pour le graphique du milieu et beaucoup plus important pour le dernier graphique.

Enfin, la figure 4.14 illustre l'évolution du taux d'utilisation en fonction du temps pour le cas de la trace réelle Trace 2. Elle montre que le taux d'utilisation « instantané » varie entre 0,21 et 0,28. En outre, la figure 4.15 illustre la répartition des débits des arrivées pour le cas de la trace réelle Trace 2. Les résultats montrent que les débits des arrivés pour cette source subissent des variations modérées, et la plupart des débits des arrivées se trouvent entre 2,25 Mb/s et 2,45 Mb/s.

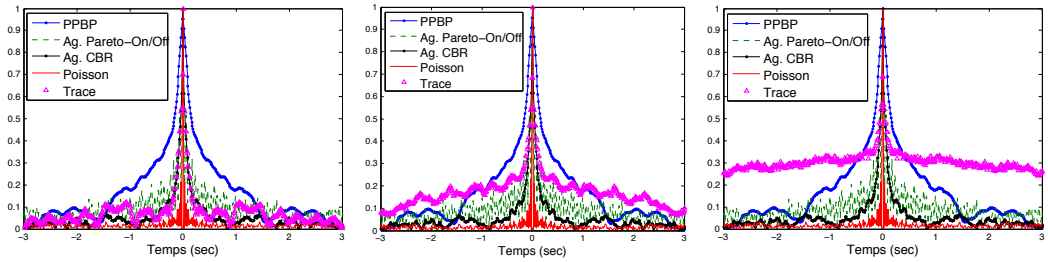


FIGURE 4.13 – Fonctions d'autocorrélations pour une trace réelle (Trace 2), pour une source Poisson, pour une source PPBP, pour une superposition de 100 flux CBR indépendants et pour une superposition de 20 flux On/Off Pareto sur différents intervalles de temps.

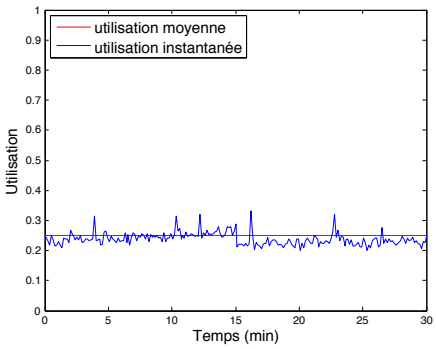


FIGURE 4.14 – Evolution du taux d'utilisation en fonction du temps pour une trace réelle (Trace 2).

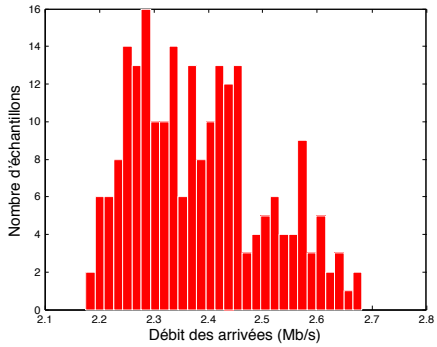


FIGURE 4.15 – Répartition des débits des arrivées pour une trace réelle (Trace 2).

4.1.2 Les flux VBR

Les flux entrants soumis au contrôle d'admission sont de type VBR. Ils représentent des flux audio avec un débit de 64 Kb/s et des paquets transportant 190 octets.

Les instants de départs de ses paquets sont déterminés comme suit : avec une

probabilité p, le départ du prochain paquet est prévu t_p millisecondes après le paquet précédent, et avec une probabilité $q = 1 - p$, le départ du prochain paquet survient t_q millisecondes plus tard. Dans l'ensemble, le débit moyen de chaque flux VBR (en paquet/s) est donné par :

$$\overline{r} = \frac{p}{t_p} + \frac{1-p}{t_q} \tag{4.3}$$

Dans nos expériences, nous choisissons $p = 0{,}95$, $t_q = 28 \times t_p$ et une taille de paquet constante égale à 190 octets. De plus, chaque flux VBR génère des paquets avec un débit moyen $\overline{r}$ de 64 kb/s et un coefficient de variation de 2,5 (nous rappelons qu'il est de 0 pour un flux CBR et 1 pour une source Poisson).

Les flux VBR arrivent aléatoirement selon un processus de Poisson avec un débit constant, noté γ. Leurs durées sont tirées suivant une distribution exponentielle avec une moyenne d_{vbr}. Notons que si aucun contrôle d'admission devait être exécuté, le débit cumulé des flux VBR serait égal à :

$$\Lambda_{vbr} = \overline{n}.\overline{r} \tag{4.4}$$

où $\overline{n} = d_{vbr}.\gamma$ (Loi de Little [Allen, 1990]) représente le nombre moyen des flux VBR sur le lien de communication (sans aucune politique de contrôle d'admission). Nous choisissons $d_{vbr} = 120$ s et $\gamma = 0{,}717$ arrivées par seconde. De ce fait, nous avons : $\Lambda_{vbr} = 5{,}5$ Mb/s.

La nécessité de mettre en place une politique de contrôle d'admission. Comme nous avons décrit précédemment, la source initiale a un débit moyen de 2,5 Mb/s. Le débit d'émission de la source initiale et des flux VBR serait de 8 Mb/s si aucune politique de contrôle d'admission n'était appliquée. Avec un tel niveau de charge, la qualité de service ne peut pas être garantie car l'acceptation de tous les flux VBR qui cherchent à entrer sur le lien de communication conduit à un niveau élevé d'utilisation, ce qui induit des retards sur les paquets allant jusqu'à 55 ms (ce résultat est obtenu par simulation en rejouant les scénarios décrits précédemment dans le cas de la trace réelle (Trace 1) sans mettre aucune politique de contrôle d'admission en place).

Par conséquent le but du contrôle d'admission est de limiter le nombre de flux VBR de manière à maintenir le débit du trafic à un "bon" niveau afin d'éviter que les paquets subissent un temps d'attente excessif dans le tampon.

4.2 Elements de comparaison pour le contrôle d'admission

4.2.1 Solutions existantes

Dans notre étude, nous considérons deux solutions de contrôle d'admission basé sur des mesures : (1) la solution appelée *Somme Mesurée* [Jamin and Danzig, 1997, Jamin et al., 1997] car son test d'admission basé sur le taux d'utilisation est simple ; (2) la solution *Enveloppes du Trafic Agrégé* [Qiu and Knightly, 2001] car elle intègre

une caractérisation plus fine de la variation du trafic en considérant plusieurs échelles de temps. Notons que dans leur forme originale, toutes ces solutions supposent que des flux de débit crête r connu cherchent à entrer dans un lien de communication avec une capacité de transmission C.

4.2.1.1 Somme Mesurée (S.M.)

Jamin et al. présentent dans [Jamin et al., 1997] un contrôle d'admission qui se base sur une mesure de la charge existante sur un lien, notée R. Lorsqu'un flux cherche à entrer sur un lien en demandant un débit crête r, l'algorithme vérifie la condition suivante :

$$R + r \leq \nu C \tag{4.5}$$

où ν est un paramètre permettant de fixer l'utilisation maximale du lien attendue. Si la condition est vérifiée alors le flux est accepté.

La mesure sur la charge courante du lien est réalisée sur une fenêtre de mesure T et est répétée sur chaque fenêtre de mesure. Cette fenêtre de mesure est elle-même découpée en périodes d'observation de durée identique. Le débit moyen du trafic sortant du lien est calculé sur chaque *période d'observation* et conservé en mémoire. A la fin d'une fenêtre de mesure, la charge courante est considérée comme étant le maximum des débits moyens obtenus sur les périodes d'observation constituant cette fenêtre de mesure. C'est cette valeur qui est utilisée dans le test d'admission, s'il doit être appliqué, lors de la fenêtre de mesure suivante. Il faut aussi noter que la valeur de la charge courante peut également être modifiée à l'intérieur d'une fenêtre de mesure. C'est le cas lorsque le débit moyen calculé sur une *période d'observation* est supérieur à la charge courante utilisée dans la fenêtre de mesure associée ou lorsqu'un nouveau flux est accepté. La charge courante du lien est alors modifiée à l'intérieur de la fenêtre de mesure et prend comme valeur soit le débit moyen qui vient d'être calculé sur la *période d'observation*, soit la charge courante auquel s'ajoute le débit du flux entrant. Il faut noter que les débits moyens calculés sur les périodes d'observation sont toujours conservés en mémoire et ce sont ces valeurs qui sont utilisées pour déterminer la charge courante à la fin d'une fenêtre de mesure.

Jamin et al. introduisent un test de délai à leur contrôle d'admission. Cette solution rejette un flux qui cherche à entrer sur un lien si l'acceptation de ce flux viole la contrainte suivante :

$$\widehat{D} + \frac{b_i}{C} < D, \tag{4.6}$$

où D est une constante représentant la contrainte du délai d'attente maximal, $\widehat{D}$ est le délai mesuré et b_i représente la variabilité d'un flux [Jamin et al., 1997]. Le délai mesuré, noté $\widehat{D}$, représente le délai d'attente maximal de chaque paquet calculé sur une fenêtre de mesure de taille T. La valeur de $\widehat{D}$ est mise à jour à la fin de chaque fenêtre de mesure. De plus, à chaque fois que le nouveau délai mesuré d'un paquet dépasse le délai d'attente maximal la valeur de $\widehat{D}$ devient λ fois ce nouveau délai mesuré. Enfin, la valeur de $\widehat{D}$ est également mise à jour, pour prendre la valeur de la partie gauche de l'équation 4.6, à chaque fois qu'un nouveau flux est admis.

4.2.1.2 Enveloppes du Trafic Agrégé (Env.)

L'opération de mesure du contrôle d'admission proposé dans [Qiu and Knightly, 2001] vise à caractériser le débit du trafic agrégé par des enveloppes sur le débit crête. Les mesures se font sur une fenêtre de mesure de longueur T découpée en périodes d'observation de durée identique. Au sein d'une fenêtre de mesure, les mesures des débits crêtes se font sur différentes échelles de temps : R_k^m correspond au débit maximal obtenu sur une échelle de temps k, égale à k périodes d'observation, dans la m^e fenêtre de mesure. A partir de là, l'estimateur du débit du trafic agrégé, ainsi que la variance sur ce débit, sont déterminés sur les M dernières fenêtres de mesure comme suit :

$$\overline{R}_k = \sum_{m=1}^{M} \frac{R_k^m}{M} \tag{4.7}$$

et

$$\sigma_k^2 = \frac{1}{M-1} \sum_{m=1}^{M} (R_k^m - \overline{R}_k)^2 \tag{4.8}$$

L'algorithme d'admission est sujet à deux tests : l'un à court terme qui vérifie qu'aucun paquet n'est trop retardé, et l'autre à long terme qui vérifie que la capacité du lien n'est pas violée avec le flux demandant à entrer. Dans cet article, nous limitons l'algorithme d'admission au seul test à long terme (i.e. en utilisant une seule période d'observation de longueur T) puisque nous cherchons uniquement à comparer les contrôles d'admission dans leur capacité à respecter un taux de perte donné. Dans ce cadre, un nouveau flux avec un débit crête de r est admis sur un lien de capacité C si :

$$\max_{k=1,\ldots,T} \{k\tau(\overline{R}_k + r + \alpha_E \sigma_k - C) \leq C \times D \tag{4.9}$$

et

$$\overline{R}_T + r + \alpha_E \sigma_T \leq C \tag{4.10}$$

où D est une constante représentant la contrainte du délai d'attente maximal et α_E est une constante spécifiant le degré de confiance que nous relions au taux de perte.

4.2.2 Oracle

Afin d'apprécier les performances de notre solution KBAC, nous avons implanté une procédure permettant de déterminer le nombre maximal de flux VBR pouvant être acceptés, et donc d'atteindre le taux d'utilisation maximal, tout en satisfaisant au critère de QoS (*i.e.*, ni faux positifs, ni faux négatifs). Cette procédure est un *oracle* dont les performances ne pourront être égalées par aucune solution. En effet, les décisions prises par l'oracle requièrent une connaissance, non seulement du passé et du présent, mais aussi des futurs flux, ce qui n'est pas évidemment possible.

Dans nos scénarios, étant donné le grand nombre de flux qui cherchent à entrer sur le lien de communication (plus de 1000 flux), une approche exhaustive qui prendra en considération toutes les combinaisons possibles des flux acceptés / rejetés

mènerait à environ $2^{1000} \simeq 10^{301}$ combinaisons possibles, et donc serait intraitable. Cependant, une approche itérative pour déterminer la séquence des flux acceptés par l'oracle suivant la politique *premier arrivé, premier servi* (si le flux ne viole pas le critère de QoS) est envisageable. C'est ce que nous avons considéré dans la suite de notre tavail.

Plus précisément, à l'itération (i), supposons que k flux ont été acceptés (certains d'entre eux peuvent encore être en cours de transmission) et j flux ont été refusés. Dès qu'un nouveau flux arrive, nous l'acceptons et nous continuons la simulation en cours d'exécution jusqu'à ce que ce flux se termine, en refusant tous les flux VBR ultérieurs. Une fois que le flux s'achève, nous vérifions si le critère de QoS pour ce flux, ainsi que pour tous les autres flux préalablement acceptés a été satisfait. Si c'est le cas, nous considérons ce flux comme acceptable par l'oracle et la valeur de k est incrémentée. Sinon, le flux ne fera pas partie de la combinaison des flux acceptés et j est incrémenté.

4.3 Calibrage des contrôles d'admission

Quel objectif ? Nous décrivons ici comment nous paramétrons chaque contrôle d'admission de façon à respecter un objectif sur la qualité de service. Dans notre travail, cet objectif sur la QoS est le délai maximal toléré, noté D^*.

Chaque contrôle d'admission a un paramètre de réglage pour ajuster le niveau de rigueur. D'une manière générale, plus le niveau de rigeur est élevé, plus le contrôle d'admission est conservateur et moins il y a de flux acceptés.

TABLE 4.1 – Synthèse des différents paramètres utilisés pour notre solution KBAC.

	KBAC (*Knowledge-Based Admission Control*)
Quantités mesurées	
Débit agrégé	$\widehat{X}$
Historique	20 dernières fenêtres de mesure ($M = 20$)
Délai d'attente des paquets	P
Historique	Une seule fenêtre de mesure
Fenêtre de mesure	$T = 200$ ms
Plan de connaissance	
Fenêtre de mesure	$T_{kp} = 20$ s
	$k = 10$, $S = 6$, $n_s = 20$ et $n = 1000$
Paramètres calibrés D^* : 10 ms ou D^* : 20 ms	$P^* = D^*$ et $\alpha = 1{,}7$

Calibrage de notre solution KBAC. Notre solution KBAC (*cf.* section 3.2.1) est simple à calibrer, il suffit tout simplement de choisir P^* (P^* est une constante représentant la contrainte de QoS) égale à l'objectif sur la QoS. Plus précisément, $P^* = D^*$ pour le cas d'un délai d'attente visé. La table 4.1 récapitule l'ensemble des valeurs des paramètres choisis pour notre solution KBAC

Nous détaillons par la suite la configuration des contrôles d'admission considérés (*cf.* section 4.2.1).

4.3.1 Calibrage des contrôles d'admission considérés

Nous décrivons ici comment nous paramétrons chaque contrôle d'admission de façon à respecter un objectif sur le délai maximal toléré, D^*.

4.3.1.1 Somme Mesurée

Pour la méthode *S.M.*, nous choisissons $D = D^*$. Vu qu'il n'y a pas de recommandations par les auteurs sur le choix de la valeur λ ($\lambda \geq 1$), nous considérons deux valeurs possibles, $\lambda = 1$ et $\lambda = 2$.

4.3.1.2 Enveloppes du Trafic Agrégé

Enfin, pour la méthode *Env.*, , nous choisissons $D = D^*$. Vu qu'il n'y a pas de recommandations par les auteurs sur le choix de de la valeur α_E, nous choisissons plusieurs valeurs possibles, à savoir, $\alpha_E = 0{,}01$, $\alpha_E = 1{,}3$, et $\alpha_E = 3{,}62$ (qui correspondent respectivement à $\Phi(\alpha_E)$ égal à 0,5, 0,9, et 0,9999).

TABLE 4.2 – Synthèse des différents paramètres utilisés dans les solutions étudiées dans le cas du délai d'attente.

	Somme Mesurée	Capacité Equivalente	Enveloppes
Quantités mesurées			
Débit agrégé	-	-	$\overline{R_k}(k = 1, \ldots t)$
Historique	-	-	20 dernières
	-	-	fenêtres de mesure
Écart-type	-	-	$\sigma_k(k = 1, \ldots t)$
Historique	-	-	20 dernières
		-	fenêtres de mesure
Délai estimé	$\widehat{D}$	-	-
Historique	Une seule	-	-
Fenêtre de mesure	$T = 4$ s	-	$T = 200$ ms
	période d'observation de 200 ms		
Paramètres calibrés			
D^* : 10 ms ou D^* : 20 ms	$D = D^*$	-	$D = D^*$
	$\lambda = 1$ ou $\lambda = 2$	-	$\alpha_E = 0{,}01$, $\alpha_E = 1{,}3$ ou $\alpha_E = 3{,}62$

La table 4.2 récapitule l'ensemble des valeurs des paramètres choisis pour les différentes solutions testées pour le cas du délai d'attente.

4.4 Estimation du débit crête des flux entrants

Travaux existants. Les travaux d'évaluation des contrôles d'admission présentés dans la littérature, supposent que le débit des flux entrant est déjà connu. La majorité de ces travaux supposent que le trafic est lissé en entrée par un seau à jetons dont les paramètres sont connus.

Notre étude Dans notre étude, les caractéristiques du flux entrant ne sont pas connues *a priori* mais découvertes par le contrôle d'admission. L'estimation du débit moyen des flux VBR injectés est réalisée après que 20 de leurs paquets soient entrés dans le lien.

Dans nos expériences, nous ne supposons aucune connaissance explicite sur les flux entrants. Dans certains cas, cette connaissance peut être obtenue via la signalisation et / ou l'utilisation d'un seau à jetons. Toutefois, les seaux à jetons peuvent être difficiles à paramétrer et peuvent induire une mauvaise qualité d'expérience pour le contrôle d'admission (car l'algorithme de décision utilise une valeur conservatrice de r). Dans ce travail, nous considérons plutôt une approche simple qui ne nécessite pas de signalisation car elle est uniquement basée sur les paquets de données. Nous détaillons ici la procédure que nous mettons en œuvre pour estimer le débit crête d'un nouveau flux entrant sur le lien de communication.

Estimation du débit crête. Pour estimer le débit crête d'un nouveau flux entrant, nous regardons les A premiers paquets de ce flux (notons que cette propriété implique qu'un flux peut être rejeté, même si ses premiers paquets ont été transmis). Nous utilisons ensuite une fenêtre glissante de longueur égale à a paquets pour calculer le débit crête.

Pour toutes les fenêtres de mesures possibles sur les premiers A paquets, nous calculons le débit moyen. Puis nous considérons que le débit crête correspond à la valeur maximale trouvée sur les $(A-a+1)$ fenêtres possibles. Dans ce travail, le débit crête d'un nouveau flux est calculé en considérons les 20 premiers paquets ($A = 20$) avec une fenêtre glissante de longueur égale à 5 paquets ($a = 5$). Notons que dans nos expériences, le débit maximal que peut atteindre un flux VBR correspond à 150 kb/s.

4.5 Performances

Dans la suite, nous limitons notre cadre expérimental pour le cas d'un délai maximal toléré (une autre option consisterait à considérer le taux de perte de paquets). Nous évaluons les performances de notre solution KBAC en utilisant le simulateur réseaux ns-3. Chaque simulation est exécutée pour une période de 30 minutes. Il est également intéressant de noter qu'on compare notre solution KBAC avec deux autres solutions (*i.e.,* Somme Mesuré et Enveloppes du trafic agrégé), ainsi que l'oracle.

Afin d'évaluer correctement le comportement de chaque contrôle d'admission, nous considérons plusieurs métriques de mesures :

[1] Des mesures à court terme du délai d'attente moyen pour chaque paquet calculé sur une fenêtre de mesure égale à 4 s ;

[2] Des mesures à long terme sur le pourcentage des flux acceptés ainsi que le pourcentage de violation qui représente le rapport du temps pendant lequel le seuil de QoS est violé. Ces mesures sont calculées tout au long de la durée de la simulation.

Nous rappelons que dans nos scénarios de simulations (Section 4.1), la source initiale est modélisé successivement par un processus de Poisson, une source PPBP, et par deux traces réelles provenant de réseaux différents.

Délai seuil. Dans notre travail, nous considérons deux valeurs différentes du délai seuil, à savoir $D^* = 10$ ms et $D^* = 20$ ms. Nous rappelons que la taille de la file à l'entrée du lien est fixée à 60 ms.

Dans un premier temps, nous avons fixé un délai seuil de 10 ms afin de se mettre dans une limite de congestion. Toutefois, dans certains scénarios, comme dans le cas d'une source initiale modélisée par un processus PPBP ou par une trace réelle (Trace 2), le délai d'attente de la source initiale, sans injecter les flux VBR, dépasse parfois 10 ms à cause de la grande variabilité de ces sources. Pour cela, nous considérons un délai seuil, $D^* = 20$ ms, pour le cas d'une source initiale modélisée par un processus PPBP ou par une trace réelle (Trace 2), et un délai seuil, $D^* = 10$ ms, pour le cas d'une source initiale modélisée par un processus de Poisson ou par une trace réelle (Trace 1).

La suite de cette section illustre les résultats obtenus.

4.5.1 Cas d'une source Poisson

Premièrement, nous présentons les résultats obtenus pour le cas d'une source initiale modélisée par un processus de Poisson.

La figure 4.16 représente l'évolution du temps moyen passé par les paquets dans le buffer du lien en fonction du temps. Premièrement, nous remarquons que l'oracle satisfait toujours le critère de QoS choisi en s'approchant très souvent de la valeur maximale tolérée. Les résultats de notre contrôle d'admission, nommé KBAC sur la figure, montrent qu'il maintient un temps d'attente moyen pour les paquets souvent compris ente 4 et 10 ms avec quelques dépassements du critère de QoS (environ 18% du temps). Il est également intéressant de noter qu'il présente un comportement qualitativement très proche de l'oracle. D'autre part, la solution de la Somme Mesurée, *M.S.*, présente un comportement excessivement conservateur. Enfin, les résultats de la solution des Enveloppes du Trafic Agrégé varient largement en fonction du réglage de son paramètre de tunning α_E. Pour $\alpha_E = 0{,}01$ (étiqueté comme *ENV.1*), ce contrôle d'admission viole presque constamment le critère de QoS. Pour

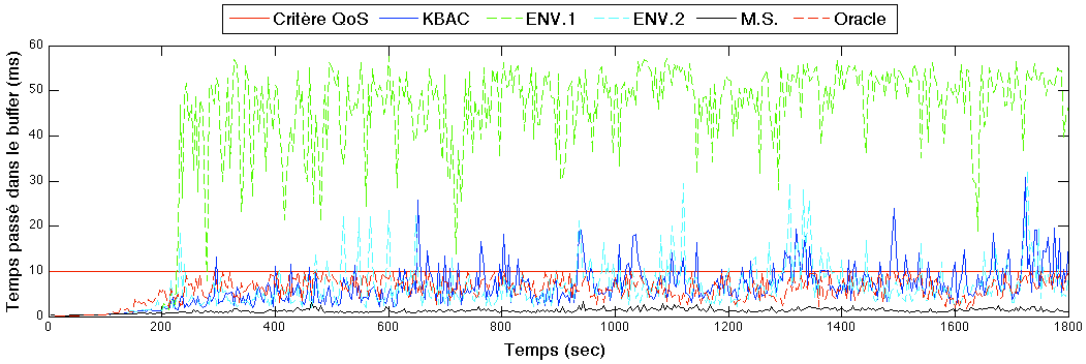


FIGURE 4.16 – Performances des solutions dans le cas où la source initiale est un processus de Poisson avec un délai d'attente de 10 ms comme critère de QoS.

α_E = 1,3 (étiqueté comme *ENV.2*), il produit un comportement proche de notre solution KBAC. Enfin, pour α_E = 3,62 (étiqueté comme *ENV.3*), qui n'est pas représenté dans la figure 4.16, ce contrôle d'admission conduit à un comportement excessivement conservateur.

TABLE 4.3 – Performances des solutions dans le cas où la source initiale est un processus de Poisson tout au long de la durée de la simulation.

	KBAC	**Somme Mesurée**		**Enveloppes du Trafic Agrégé**			**Oracle**
		$\lambda = 1$	$\lambda = 2$	$\alpha_E = 0.01$	$\alpha_E = 1.3$	$\alpha_E = 3.62$	
Nombre de flux acceptés	397	345	334	440	395	315	407
Pourcentage de flux acceptés	31,33%	27,23%	26,36%	34,73%	31,18%	24,86%	32,12%
Pourcentage de violation	18,22%	0%	0%	87,33%	14,89%	0%	0%

Ces observations sont confirmées par la Table 4.3. Tout d'abord, l'oracle permet d'accepter 407 flux. Deuxièmement, notre solution KBAC conduit à un nombre de flux acceptés (*i.e.,* 397 flux), qui est très proche au nombre des flux acceptés par l'oracle. Ensuite, lorsque nous regardons les autres solutions étudiées, le nombre de flux acceptés diffèrent largement. La solution de Somme Mesurée accepte un nombre de flux significativement plus faible que l'oracle (*i.e.,* 345 flux lorsque λ est fixé à 1 et 334 flux λ est fixé à 2). Enfin, pour la solution des Enveloppes du Trafic Agrégé, les résultats varient fortement selon le calibrage. Cette dernière solution accepte respectivement autour de 12 et 92 flux de moins que l'oracle pour α_E = 1,3 et 3,62, respectivement, tandis que pour α_E = 0,01, elle accepte un nombre extrêmement élevé de flux (*i.e.,* 440 flux) ce qui se traduit par un temps d'attente dans le buffer bien au-dessus du seuil d'admission (jusqu'à 75% du temps).

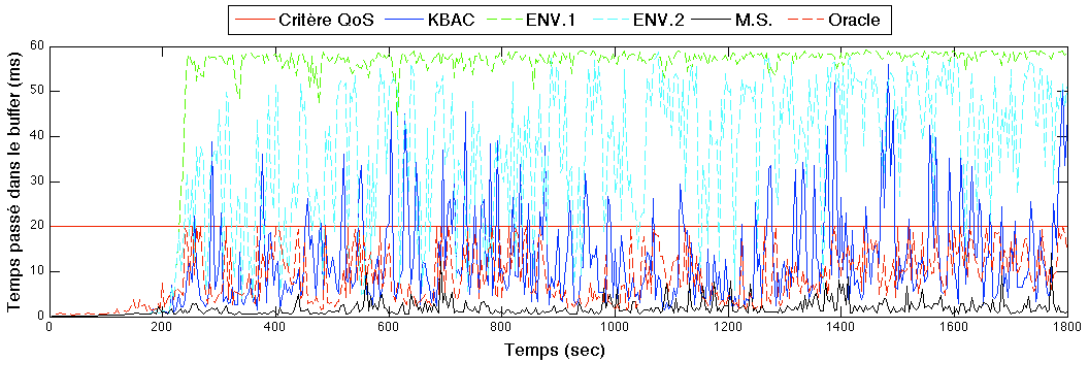


FIGURE 4.17 – Performances des solutions dans le cas où la source initiale est un processus PPBP avec un délai d'attente de 20 ms comme critère de QoS.

4.5.2 Cas d'une source PPBP

Deuxièmement, nous présentons les résultats obtenus pour le cas d'une source initiale modélisée par un processus PPBP.

La figure 4.17 représente l'évolution du temps moyen passé par les paquets dans le buffer du lien en fonction du temps avec un délai de 20 ms comme critère de QoS. Notre contrôle d'admission KBAC satisfait le critère de la QoS environ 82% du temps. Cependant, ces violations sont rencontrées pendant des périodes de temps relativement courtes (typiquement moins de 10 s, ainsi que l'amplitude de ces violations est généralement tolérable. D'autre part, la solution de la Somme Mesurée présente un comportement excessivement conservateur. Enfin, les résultats de la solution des Enveloppes du Trafic Agrégé viole presque constamment le critère de la QoS.

TABLE 4.4 – Performances des solutions dans le cas où la source initiale est un processus PPBP tout au long de la durée de la simulation.

	KBAC	**Somme Mesurée**		**Enveloppes du Trafic Agrégé**			**Oracle**
		$\lambda = 1$	$\lambda = 2$	$\alpha_E = 0.01$	$\alpha_E = 1.3$	$\alpha_E = 3.62$	
Nombre de flux acceptés	412	377	384	481	437	373	411
Pourcentage de flux acceptés	32,4%	29,6%	30,2%	37,8%	34,4%	29,3%	32,3%
Pourcentage de violation	18,66%	0%	0%	87,33%	74,67%	0%	0%

La table 4.4 représente les performances des solutions dans le cas d'un processus PPBP tout au long de la durée de la simulation. Tout d'abord, elle montre que l'oracle permet d'accepter 411 flux. Notre solution KBAC accepte quasiment le même nombre de flux que l'oracle. Ensuite, il montre que la solution de Somme Mesurée restreint trop le nombre de flux acceptés, impliquant une sous-utilisation des capacités des liens de transmissions de l'opérateur. Enfin, la solution des Enve-

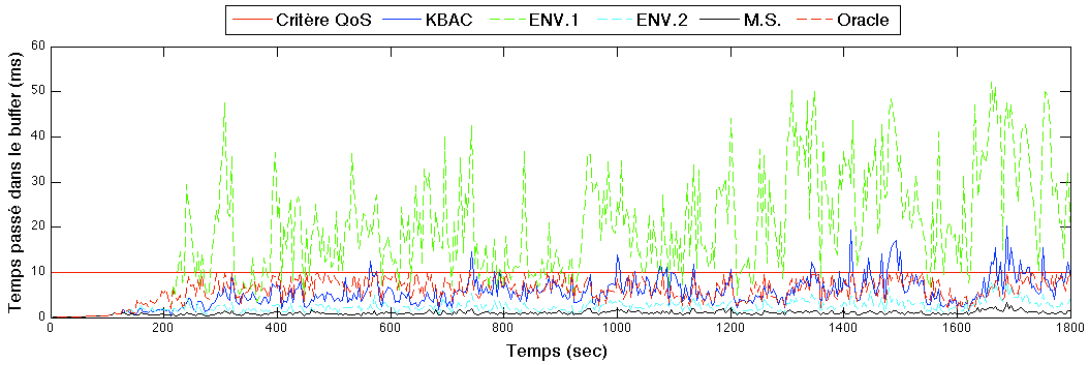


FIGURE 4.18 – Performances des solutions dans le cas où la source initiale est un modélisée par une trace réelle (Trace 1) avec un délai d'attente de 10 ms comme critère de QoS.

loppes du Trafic Agrégé accepte respectivement 26 et 70 flux de plus que l'oracle pour $\alpha_E = 0{,}01$ et 1,3, respectivement, tandis que pour $\alpha_E = 3{,}62$, elle accepte un nombre de flux significativement plus faible que l'oracle (*i.e.,* 38 flux de moins que l'oracle).

Discussions. Il faut noter que le calibrage du processus PPBP a été une étape difficile à réaliser afin de s'assurer que le délai d'attente de la source initiale modélisée par ce processus ne viole pas la contrainte de délai fixé.

D'autre part, il faut aussi noter que les performances de la solution KBAC ne sont pas aussi satisfaisantes lorsque la source initiale modélisée par un processus Poisson/PPBP émet des paquets de petite taille (*e.g.,* 190 octets). Nous n'avons pas encore identifié les raisons qui expliquent ces performances et nous comptons dans le futur investiguer plus en détail ce phénomène.

4.5.3 Cas d'une trace réelle (Trace 1)

Dans cette section, nous présentons les résultats obtenus pour le cas où la source initiale est modélisée par une trace réelle (Trace 1) collectée par l'Université de Stuttgart [Sass, 2004].

La figure 4.18 illustre les performances des solutions dans le cas de la trace réelle (Trace 1) avec un délai d'attente de 10 ms comme critère de QoS. Notre solution KBAC aboutit à des résultats très satisfaisants et elle garantit presque constamment le critère du délai d'attente visé. Il est également intéressant de noter que notre solution présente un comportement à peu près similaire à l'oracle. Plus précisément, les résultats de notre solution KBAC montrent que le critère de QoS est garanti plus de 91 % du temps. A l'inverse les deux autres solutions de contrôle d'admission que nous avons testées présentent un comportement excessivement conservateur pour la solution de la Somme Mesurée, *M.S.* et pour la solution des Enveloppes du Trafic Agrégé, *ENV.2* (pour $\alpha_E = 1{,}3$). En outre, pour le calibrage $\alpha_E = 0{,}01$ (étiqueté

comme *ENV.1*) pour la solution des Enveloppes du Trafic Agrégé, ce contrôle d'admission viole environ 73% du temps critère de QoS.

TABLE 4.5 – Performances des solutions dans le cas où la source initiale est modélisée par une trace réelle (Trace 1) tout au long de la durée de la simulation.

	KBAC	**Somme Mesurée**		**Enveloppes du Trafic Agrégé**			**Oracle**
		$\lambda = 1$	$\lambda = 2$	$\alpha_E = 0.01$	$\alpha_E = 1.3$	$\alpha_E = 3.62$	
Nombre de flux acceptés	357	246	241	400	310	156	379
Pourcentage de flux acceptés	28,18%	19,42%	19,02%	31,57%	24,47%	12,31%	29,91%
Pourcentage de violation	8,89%	0%	0%	73,11%	0%	0%	0%

La Table 4.5 apporte des résultats complémentaires sur la performance globale des solution de contrôle d'admission étudiées. L'oracle permet d'accepter presque 380 flux, notre solution accepte presque 360 flux, tandis que les solutions concurrentes subissent des comportements très variables en fonction du réglage de leurs paramètres de tunning (α_E et λ). Plusieurs observations peuvent être faites. Tout d'abord, la solution de la *Somme Mesurée* accepte toujours un nombre de flux significativement plus faible que l'oracle. A l'inverse, pour la solution des Enveloppes du Trafic Agrégé, les résultats varient fortement selon le calibrage. Cette solution accepte un nombre de flux plus faible que l'oracle pour $\alpha_E = 1{,}3$ et 3,62, tandis que pour $\alpha_E = 0{,}01$, cette dernière conduit à un nombre de flux acceptés qui est plus grand que le nombre de flux acceptés par l'oracle, ce qui se traduit par une violation du critère de QoS jusqu'à 55% du temps.

4.5.4 Cas d'une trace réelle (Trace 2)

Dans cette section, nous présentons les résultats obtenus pour le cas d'une source initiale modélisée par la trace réelle (Trace 2).

La figure 4.19 représente l'évolution du temps moyen passé par les paquets dans le buffer du lien en fonction du temps avec un délai de 20 ms comme critère de QoS. Les résultats sont proches de ceux qui sont présentés précédemment. Nous remarquons que notre solution KBAC satisfait le critère de QoS environ 91 % du temps. Il est également intéressant de noter que ces violations sont rencontrées pendant des périodes de temps relativement courtes (typiquement moins de 20 s), et l'amplitude de ces violations est généralement de taille moyenne tolérable (moins de 10 ms de violation de la cible D^*). Ce résultat met en évidence la capacité de notre solution KBAC à ajuster sa politique d'admission rapidement et automatiquement en fonction des variations des conditions de trafic. En revanche les deux autres solutions de contrôle d'admission que nous avons testées présentent un comportement différent de celui présenté dans la figure 4.18. D'une part, la solution de la Somme Mesurée, *M.S.* présente un comportement moins conservateur. D'autre part, en ce qui concerne la solution des Enveloppes du trafic Agrégé, et plus précisément pour $\alpha_E = 1{,}3$, les

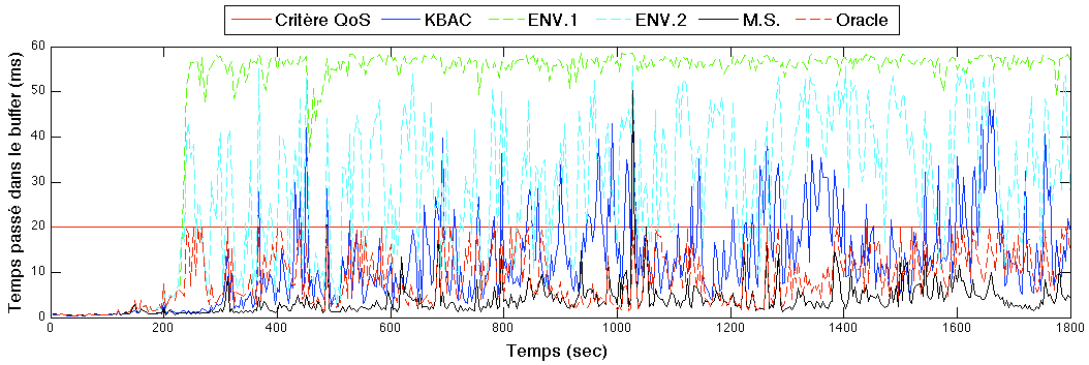


FIGURE 4.19 – Performances des solutions dans le cas où la source initiale est modélisée par de la trace réelle (Trace 2) avec un délai d'attente de 20 ms comme critère de QoS.

résultats montrent un comportement différent de celui qui est décrit précédemment dans la figure 4.18. Ce contrôle d'admission conduit à une forte violation du critère de QoS, alors que ce contrôle d'admission, avec le même paramétrage, conduit à un comportement conservateur dans le cas de la source initiale modélisé par la Trace 1. Enfin, pour $\alpha_E = 0{,}01$ (étiqueté comme *ENV.1*), les résultats concordent avec ceux qui sont présentés précédemment et conduisent à un niveau très élevé de temps d'attente des paquets dans le buffer. Ces résultats soulignent le problème du calibrage compliqué des solutions de contrôle d'admission existantes dans la littérature.

TABLE 4.6 – Performances des solutions dans le cas dans où la source initiale est modélisée par une trace réelle (Trace 2) tout au long de la durée de la simulation.

	KBAC	**Somme Mesurée**		**Enveloppes de trafic Agrégé**			**Oracle**
		$\lambda = 1$	$\lambda = 2$	$\alpha_E = 0.01$	$\alpha_E = 1.3$	$\alpha_E = 3.62$	
Nombre de flux acceptés	400	374	358	471	427	351	406
Pourcentage de flux acceptés	31,57%	29,52%	28,26%	37,17%	33,7%	27,7%	32,04%
Pourcentage de violation	19,44%	0,89%	0,67%	87,33%	66,22%	0,22%	0%

Nous passons maintenant à la Table 4.6. Les résultats montrent que l'oracle peut accepter jusqu'à 406 flux. Notre solution KBAC accepte 400 flux, qui est très proche du nombre de flux acceptés par l'oracle. La solution de la Somme Mesurée accepte environ 32 flux (respectivement, 73 flux) de moins que l'oracle pour $\lambda = 1$ (respectivement, $\lambda = 2$). Enfin, la solution des Enveloppes de trafic Agrégé accepte un nombre des flux extrêmement élevé pour α_E égale à 0,01 et 1,3, tandis qu'elle accepte moins de flux que l'oracle lorsque α_E est fixé à 3,62.

4.6 Conclusions

Dans ce chapitre, nous avons évalué les performances de notre nouvelle solution de contrôle d'admission KBAC pour le cas d'un délai maximal toléré. Nous avons réalisé ce travail sous l'angle de la simulation, en utilisant le simulateur réseau ns-3, tout en considérant diverses conditions possibles de trafic : un processus Poisson, un processus PPBP [Zukerman et al., 2003] et deux traces collectés sur des réseaux réels [Sass, 2004, II,].

Afin de comparer les performances de notre solution, nous avons implanté 2 autres algorithmes [Jamin and Danzig, 1997, Qiu and Knightly, 2001] ainsi qu'une procédure permettant de déterminer le nombre maximal de flux pouvant être acceptés, et donc d'atteindre le taux d'utilisation maximal, tout en satisfaisant au critère de QoS (*i.e.*, ni faux positifs ni faux négatifs). Cette procédure est un *oracle* dont les performances ne pourront être égalées par aucune solution. En effet, les décisions prises par l'oracle requièrent une connaissance non disponible à l'instant présent.

D'après nos résultats d'expériences, notre solution permet d'atteindre un contrôle d'admission ni trop conservateur, ni trop permissif. Elle permet ainsi de s'affranchir d'un calibrage compliqué des paramètres comme c'est le cas pour les solutions classiques de contrôle d'admission.

5 Conclusions et Perspectives

Depuis quelques années, il y a un réel changement dans les usages des réseaux en termes d'applications véhiculées ainsi que dans leur nombre. On voit de plus en plus d'applications contraintes en termes de délai, comme par exemple la téléphonie sur IP (*VoIP : Voice over Internet Protocol*), ainsi que d'applications gourmandes en ressources comme par exemple la vidéo à la demande (*VOD : Video on Demand*). La croissance en volume de ces applications commence à poser des problèmes de congestion dans les réseaux filaires et sans fil. Les opérateurs réseaux doivent êtres capables d'absorber ces changements de trafic, de faire face à cette demande de plus en plus intensive en bande passante et de fournir une bonne qualité de service (QoS) aux applications. Cela nécessite des mécanismes intelligents en termes d'ordonnancement et de gestion des files d'attente, de contrôle d'admission, de contrôle de débit et/ou de routage adéquat.

L'objectif principal de cette thèse était d'aboutir à la conception d'une nouvelle architecture de traitement et de gestion du trafic et de la qualité de service pour le contrôle d'admission.

Dans ce qui suit, nous résumons les principales contributions qui ont été détaillées dans cette thèse, ensuite, nous répondons aux questions soulevées dans le Chapitre 1 et, enfin, nous introduisons quelques perspectives intéressantes qui peuvent être explorées dans de futurs travaux.

5.1 Principales contributions

L'objectif que nous nous sommes fixés dans cette thèse est d'aboutir à la conception d'une nouvelle architecture de traitement et de gestion du trafic et de la qualité de service pour le contrôle d'admission. Pour atteindre cet objectif et après l'étude et l'analyse de l'état de l'art, un ensemble de verrous ont été identifiés. Dans le développement de la nouvelle architécture de contrôle d'admission que nous avons proposé pour lever ces verrous réside les contributions majeures de cette thèse. Ces contributions s'articulent en quatre volets :

[1] Le Plan de connaissance (Knowledge Plane). Le verrou majeur que nous avons identifié dans les solutions de contrôle d'admission proposées dans la littérature est la difficulté de calibrage des paramètres. Afin de remédier à ce problème, nous utilisons le concept de plan de connaissance. Le plan de connaissance définit un ensemble de mesures dont les valeurs reflètent de façon collective le comportement d'un lien réseau. Son établisement et son utilisation aboutissent à une meilleure gestion du réseau et permettent de s'affranchir d'un paramétrage compliqué des paramètres comme c'est le cas pour les solutions classiques de contrôle d'admission.

[2] Notre solution KBAC (Knowledge-based Admission Control). Nous avons proposé une nouvelle solution pour le contrôle d'admission qui repose sur l'élaboration en continu d'un plan de connaissance et sur la modélisation automatique du comportement d'un lien réseau par une file d'attente monoserveur. D'une part, notre solution offre une garantie probabiliste d'un paramètre de performance QoS qui peut être le délai d'attente moyen des paquets dans la file ou le taux de perte. D'autre part, elle offre l'avantage de se baser uniquement sur une connaissance acquise au cours du temps et permet ainsi de s'affranchir d'un calibrage compliqué des paramètres.

[3] Evaluation des performances dans un cadre plus réaliste. Les décisions prises par un contrôle d'admission sont toujours étroitement liées au trafic agrégé circulant sur le lien. Afin de tester notre solution dans un cadre plus réaliste, nous avons évalué les performances de notre nouveau contrôle d'admission par simulation en considérant diverses conditions possibles de trafic. Pour cela, nous avons implanté dans le simulateur réseaux ns-3 un générateur de trafic qui permet de rejouer des traces réelles ainsi que deux autres générateurs qui permettent de simuler des sources théoriques, notamment, un processus de Poison et un processus PPBP (Poisson Pareto Burst Process) [Zukerman et al., 2003] qui présente un fort degré d'autosimilarité.

Afin de comparer les performances de notre solution, nous avons implanté trois autres solutions pour le contrôle d'admission [Floyd, 1996, Jamin et al., 1997, Qiu and Knightly, 2001] ainsi qu'une procédure permettant de déterminer le nombre maximal de flux pouvant être acceptés, et donc d'atteindre le taux d'utilisation maximal, tout en satisfaisant le critère de QoS (*i.e.,* ni faux positifs ni faux négatifs).

[4] Contributions de code dans le simulateur réseau ns-3. Afin d'accomplir nos objectifs, un important travail de codage a été réalisé dans le simulateur réseau ns-3 [ns3, 2008] (voir plus de 10 000 lignes de code en C++). Ce travail a permis d'enrichir le simulateur ns-3 avec des nouveaux modules. Les contributions majeures du codage s'articulent en trois volets :

[1] Nous avons implémenté deux nouveaux générateurs de trafic : (i) un générateur de trafic qui permet de simuler une source théorique PPBP (Poisson Pareto Burst Process) [Zukerman et al., 2003], (ii) et un autre générateur de trafic qui permet de rejouer des traces réelles. Le générateur de trafic PPBP a été soumis au simulateur réseau ns-3 en 2011 [10] (environ 600 lignes de code en C++) et est actuellement en cours de validation.

[2] Nous avons implémenté dans le simulateur ns-3 des modules de collecte de mesures qui prennent en compte le comportement "temps réel" du trafic transitant. Les deux premiers moments de ces mesures (*i.e.,* moyenne et variance), comme par exemple le taux d'utilisation, le débit de sortie, le délai, le taux de perte et le temps de service, sont calculées sur des courtes fenêtres de temps mesure. Le choix de la taille de la fenêtre de mesure peut être simplement ajusté par l'utilisateur.

[3] Nous avons implémenté notre nouvelle solution KBAC ainsi que trois autres solutions de contrôle d'admission [Floyd, 1996, Jamin et al., 1997, Qiu and Knightly, 2001] dans le simulateur réseau ns-3.

Enfin, l'analyse des résultats à été effectuée sous *Matlab* avec des scripts d'environ 3000 lignes de code.

5.2 Réponses aux questions soulevées

Finalement nous répondons aux questions soulevées dans le premier chapitre :

[Q_1] *Quelles informations doivent être apportées pour la gestion du trafic et la QoS pour le contrôle d'admission ?*

— Le contrôle d'admission nécessite des opérations de mesures sur le trafic déjà admis afin de caractériser et d'en déduire un certain nombre de métriques qui décrivent le comportement du trafic transitant. Dans notre solution KBAC, nous collectons des *points de mesure* qui correspondent à des couples de valeurs associant :

[1] un débit moyen en sortie du lien ;

[2] un délai d'attente moyen des paquets dans le buffer du lien ou un taux de perte.

Dans notre travail, nous nous appuyons sur le *plan de connaissance* afin d'apporter la connaissance nécessaire sur le comportement du trafic transitant en se basant sur les informations collectées.

[Q_2] *Quelle est la périodicité de mesure adaptée à chacune des informations identifiées afin de limiter les mauvaises décisions ou afin d'améliorer les performances ?*

— Notre contrôle d'admission surveille continuellement l'activité du lien de communication de façon à collecter les informations identifiées. Afin de limiter les mauvaises décisions et d'améliorer les performances, nous utilisons deux périodicité différentes dans la collecte des informations :

[1] Une périodicité à court terme, destinée à la collecte des *points de mesure* sur chaque intervalle de temps de mesure de longueur 200 ms. Nous considérons cette périodicité afin de tenir en compte du comportement « instantané » du trafic transitant.

[2] Une périodicité à long terme, destinée à la découverte du plan de connaissance effectuée sur chaque intervalle de temps de mesure de longueur 20 s. Cette périodicité apporte la cohérence temporelle sur le comportement du trafic transitant, qui repose sur l'idée que les performances qui ont été observées dans le passé (et intégrées au plan

de connaissance) sont un élément utile pour estimer les performances futures d'un système.

[Q_3] *Est-il nécessaire d'agréger certaines informations ? Si oui, quelles sont les techniques d'agrégation à utiliser ?*

— Compte tenu de la quantité d'informations apportées au contrôle d'admission pour opérer, et afin d'anticiper le risque d'inondation du nombre de mesures, il est nécessaire d'agréger ces informations. De ce fait, nous choisissons de :

[1] limiter le nombre total de *points de mesures* tout en assurant leurs diversité ;

[2] agréger ces *points de mesures* en utilisant l'algorithme de clustering *k-means*.

5.3 Perspectives et nouveaux défis

Les travaux réalisés durant cette thèse et les résultats obtenus ouvrent la porte vers un ensemble de perspectives. Parmi celles-ci, nous pouvons citer :

[1] Evaluer les performances de notre solution KBAC pour le cas d'un taux de perte toléré.

[2] Etendre la topologie à un réseau, et s'attaquer aux méthodes de partage et de diffusion de la connaissance.

[3] Evaluer les performances de notre solution KBAC avec l'utilisation d'une classification des flux.

Nous détaillons maintenant ces perspectives.

5.3.1 Evaluation des performances de la solution KBAC pour le cas d'un taux de perte toléré.

Nous avons évalué dans le chapitre 4 les performances de notre solution KBAC pour le cas d'un délai maximal toléré. Une autre option consisterait à considérer le cas d'un taux de perte toléré. Par manque du temps, nous ne l'avons pas considéré jusqu'à présent. Toutefois, nous avons effectué le calibrage et l'évaluation des performances de trois solutions de contrôle d'admission existantes dans la littérature [Jamin et al., 1997, Floyd, 1996, Qiu and Knightly, 2001]. Nous présentons ces premières contributions ici.

5.3.1.1 Solutions existantes

Nous détaillons dans cette partie la solution de la Capacité Equivalente [Floyd, 1996]. Notons que les solutions de la Somme Mesurée [Jamin et al., 1997] et la solution des Enveloppes du Trafic Agrégé [Qiu and Knightly, 2001] ont été présentées dans le chapitre 4.

Capacité Equivalente (C.E.). Dans [Floyd, 1996], *Floyd* présente une solution de contrôle d'admission basée sur l'estimation de la capacité équivalente pour un ensemble de flux. Un nouveau flux est admis sur un lien si la somme de son débit crête, r, et de la capacité équivalente du lien, $C(\epsilon)$, est inférieure ou égale à la capacité nominale du lien, C. Plus formellement, cette condition s'exprime comme :

$$C(\epsilon) + r \leq C \tag{5.1}$$

Le point critique de cette méthode repose sur l'estimation de la capacité équivalente, $C(\epsilon)$. Dans notre cas d'étude, nous avons choisi la formule de la capacité équivalente donnée dans [Guerin et al., 1991] car elle est plus simple à évaluer dans notre cadre. La capacité équivalente proposée dans [Guerin et al., 1991] est une fonction affine du débit moyen du trafic agrégé, noté $\hat{r}$, et de son écart-type, σ :

$$C(\epsilon) = \hat{r} + \alpha.\sigma, \text{avec } \alpha = \sqrt{2\ln\frac{1}{\epsilon} + \ln\frac{1}{2\pi}}, \tag{5.2}$$

où ϵ est la probabilité attendue de violation de la capacité équivalente.

Afin de "lisser" la mesure du débit moyen du trafic agrégé, $\hat{r}$, l'auteur suggère de la définir comme une moyenne glissante exponentielle mise à jour après chaque fenêtre de mesure T, $\hat{r} = (1-\omega).\hat{r} + \omega.R$ où R correspond au débit moyen du trafic sortant du lien calculé sur la fenêtre de mesure T et ω est un réel compris entre 0 et 1. Comme rien n'était recommandé par les auteurs sur le calcul de σ, nous avons décidé de le déterminer à partir des M dernières mesures de R.

5.3.1.2 Calibrage des contrôles d'admission en fonction d'un taux de perte visé

Nous décrivons ici comment nous paramétrons chaque contrôle d'admission de façon à respecter un taux de perte maximal toléré, noté Pr^*.

Calibrage de notre solution KBAC. Pour notre solution KBAC, il suffit de choisir $P^* = Pr^*$.

Somme Mesurée. Pour la méthode *S.M.*, en suivant l'analyse proposée par les auteurs, nous avons choisi la valeur de ν comme étant le rapport du taux moyen d'arrivée sur le taux moyen de service produisant un taux de perte Pr^* dans une file *M/M/1/K*.

Capacité Equivalente. Pour la méthode *C. E.*, la valeur de ϵ représente la probabilité que le taux d'arrivée instantané du trafic agrégé modélisé par un processus Gaussien dépasse la capacité équivalente. Les auteurs ne fournissent qu'une plage de valeurs possibles concernant le choix de la valeur à donner à ϵ. En supposant que cette probabilité s'apparente à celle d'avoir la file pleine (ce qui serait le cas pour une taille de file égale à 1) et que les probabilités vues à l'arrivée des paquets dans la file

TABLE 5.1 – Synthèse des différents paramètres utilisés dans les solutions étudiées dans le cas du taux de perte.

	Somme Mesurée	Capacité Equivalente	Enveloppes
Quantités mesurées			
Débit agrégé	R	$\hat{r}$	$\overline{R_k}(k = 1, \dots \mathrm{t})$
Historique	Une seule fenêtre de mesure	Moyenne glissante exponentielle	20 dernières fenêtres de mesure
Écart-type	*Non mesurée*	σ	$\sigma_k(k = 1, \dots \mathrm{t})$
Historique	*Non mesurée*	20 dernières fenêtres de mesure	20 dernières fenêtres de mesure
Fenêtre de mesure	$T = 4$ s *période d'observation* de 200 ms	$T = 200$ ms	$T = 200$ ms
Paramètres calibrés			
$Pr^* : 10^{-2}$	$\nu = 0{,}9543$	$\alpha = 2{,}7152$	$\alpha_E = 2{,}325$
$Pr^* : 10^{-4}$	$\nu = 1{,}0045$	$\alpha = 4{,}0722$	$\alpha_E = 3{,}620$

sont celles à tout instant en régime stationnaire (ce qui serait le cas si les instants d'arrivées des paquets dans le lien suivent un processus de Poisson), la probabilité ϵ représente également la probabilité de rejet d'un paquet. Ainsi, nous avons fixé la valeur de ϵ égale à Pr^* et nous en avons dérivé la valeur de α.

Enveloppes du Trafic Agrégé. Enfin, pour la méthode *Env.*, en modifiant la valeur du degré de confiance, α_E, on détermine la probabilité qu'aucune perte ne soit observée pour l'ensemble des flux entrant. Nous approchons la valeur de cette probabilité à celle d'avoir un taux de perte égal à Pr^* pour tous les flux acceptés (ce qui est toujours le cas si le degré de confiance estimé est vérifié).

La table 5.1 récapitule l'ensemble des valeurs des paramètres choisis pour les différentes solutions testées pour le cas du taux de perte.

5.3.1.3 Evaluation de performances

Les premiers résultats obtenus [7, 9] indiquent que les performances de la solution de la Capacité Equivalente et la solution des Enveloppes du Trafic Agrégé semblent meilleures que la solution de la Somme Mesurée. Toutefois, aucune solution testée n'est pleinement satisfaisante pour un opérateur car bien que certains satisfont à l'objectif sur le taux de perte maximal toléré, elles restreignent trop le nombre de flux acceptés, impliquant ainsi une sous-utilisation des capacités des liens de transmissions de l'opérateur. Il est également intéressant de noter que ces solutions sont très difficiles à calibrer.

5.3.1.4 Travaux futurs

Comme prochaine étape de notre recherche, nous évaluerons les performances de notre solution KBAC pour le cas d'un taux de perte toléré. Dans notre solution KBAC, il faut considérer le cas d'une file d'attente monoserveur de type *M/GI/1/K* qui permet de reproduire le comportement du lien réseau. Afin d'accomplir cette tâche, il reste à implémenter la méthode de découverte des paramètres de cette file d'attente dans le simulateur réseau ns-3. Ensuite, une série de simulations doivent être lancés en considérant les scénarios de simulation illustrés dans le chapitre 4.

5.3.2 Etendre la topologie à un réseau.

Dans cette thèse, nous avons évalué les performances de notre solution ainsi que les solutions étudiées sur un seul lien réseau. Cependant, il serait intéressant d'étudier l'application de notre solution à un réseau complet.

Nous avons effectué une première phase de ce travail qui consiste à ajuster l'intensité du trafic de fond afin de se mettre en limite de congestion (lorsque les délais commencent à être significatifs sans être rédhibitoires) sur quelques liens dans le réseau afin que le contrôle d'admission soit utile. Nous détaillons maintenant cette première phase.

Topologie simulée. La topologie choisie est la topologie de Chen (*cf.* figure 5.1), réseau des Etats-Unis. Les liens sont symétriques avec des caractéristiques similaires dans les deux sens, la capacité et le délai de chaque lien sont fixés à 10 Mb/s et 10 ms respectivement. La taille des buffers en unité de temps est de 100 ms (*e.g.,* 125 ko sur un lien de 10 Mb/s) et la discipline de service est FIFO (*First In First Out*), et Drop-Tail.

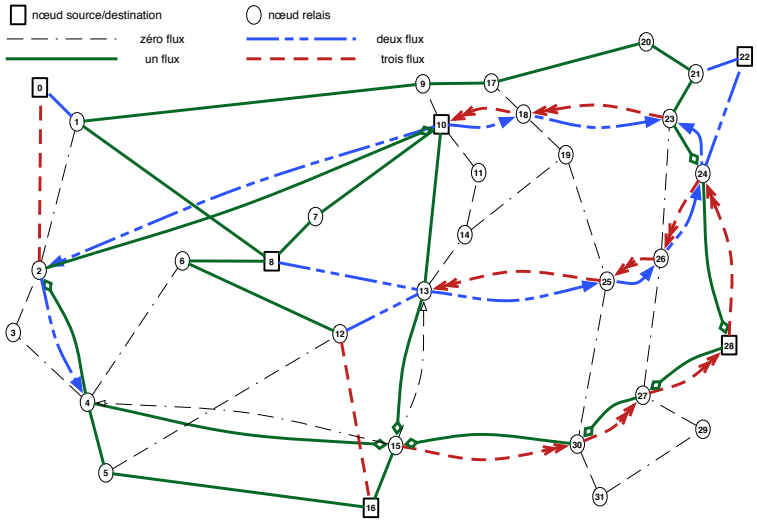


FIGURE 5.1 – Topologie de Chen.

Dans nos scénarios, le trafic de fond est modélisé par une distribution de Poisson de paramètre λ. Nous envisageons par la suite de ce travail de considérer un processus PPBP et des traces réelles.

Répartion de la charge du trafic de fond. Dans le but d'avoir une charge bien répartie sur l'ensemble du réseau, six nœuds sources/destinations ont été choisis (*i.e.,* les nœuds 0, 8, 10, 16, 22 et 28 de la figure 5.1). Les flux sont injectés par couple source/destination (à l'exception du couple [22 ; 28] afin d'avoir un trafic de fond mieux répartis dans le réseau), ce qui fait en total, 28 flux agrégés (donc 28 sources Poisson de paramètre λ) visant à représenter le trafic du fond. Nous utilisons le protocole de routage OSPF qui utilise un algorithme du plus court chemin *(SPF, Shortest Path First)*. La table 5.2 récapitule la table des flux.

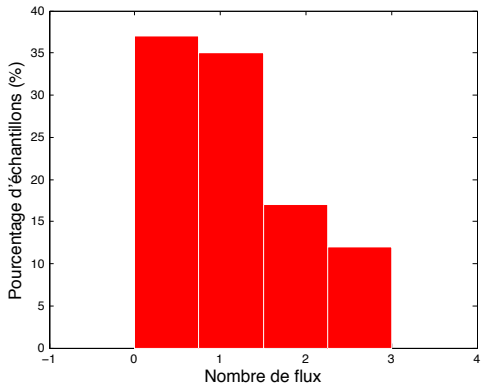


FIGURE 5.2 – Répartition des flux dans le réseau.

La figure 5.2 représente le taux d'utilisation de chaque lien en fonction de l'intensité du trafic agrégé, ainsi que le pourcentage de ces liens dans le réseau. Les résultats montrent que 12% des liens sont traversés par trois flux, 17% par deux flux, 34% par un seul flux et 37% par zéro flux. Le nombre des flux traversant chaque lien du réseau est également illustré dans la figure 5.1.

Évolution de performances du réseau. Nous présentons par la suite une étude de performances des différents flux en regardant le délai de bout-en-bout pour plusieurs intensités du trafic de fond. Le but ici est d'ajuster l'intensité du trafic de fond, λ, afin de se mettre en limite de congestion.

Choix de intensité du trafic de fond. Nous avons calculé les performances de chaque flux (*cf.* table 5.2) par simulation en regardant le délai de bout-en-bout. Le calcul des délais des différents flux dans le réseau montre que le flux Id 25 est le plus dégradé (pénalisé avec le plus grand délai) par la variation de l'intensité du trafic de fond.

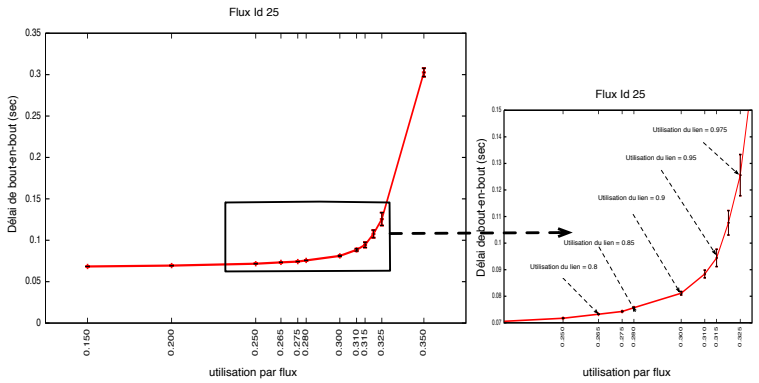


FIGURE 5.3 – Calcul du délai moyen de bout-en-bout du flux Id 25 en fonction du taux d'utilisation.

La figure 5.3 illustre l'évolution du délai moyen de bout-en-bout du flux Id 25 en fonction du taux d'utilisation effectuées avec un intervalle de confiance à 95% (6 réplications sur chaque scénario de simulation, un scénario correspond à une intensité du trafic de fond). Les résultats montrent une augmentation rapide dans le délai pour une valeur du taux d'utilisation supérieure à 0.28.

En se basant sur tous ces résultats, on peut considérer, une valeur du taux d'utilisation $\in [0.265; 0.280]$ comme une valeur seuil de l'intensité du trafic de fond afin de se mettre en limite de congestion. Cette valeur correspond à une valeur de $\lambda \in [2,6 \text{ Mb/s}; 2,8 \text{ Mb/s}]$.

5.3.2.1 Travaux futurs

La suite de ce travail est d'injecter les flux individuels qui seront soumis à in contrôle d'admission.

Nous avons envisagé, dans la section 4.1 du chapitre 4, la possibilité d'étendre notre solution KBAC, qui a été pensée pour fonctionner à l'échelle d'un lien, sur la globalité du réseau. Pour cela, il suffit de répéter ce contrôle d'admission sur chacun des liens traversés par le flux. D'autre part, afin de simplifier la tâche des opérateurs réseaux, le contrôle d'admission peut être appliqué uniquement sur les liens correspondant à des goulots d'étranglement (*bottleneck links*) dans les réseaux d'accès et dans le cœur du réseau. Il est intéressant de noter que certains travaux de la littérature, comme le travail de [Wehmuth and Ziviani, 2011], présentent des méthodes simples capables de localiser les liens correspondant à des goulots d'étranglement.

Ensuite, il serait intéressant d'étudier les méthodes de partage et de diffusion de la connaissance entre les nœuds du réseau.

5.3.3 Classification des flux.

Le contrôle d'admission vise à limiter le nombre de flux afin de satisfaire les critères de performances exigés par les flux avec QoS. Dans cette thèse, nous avons appliqué

notre contrôle d'admission sans prendre en compte les différentes classes d'applications des flux. Généralement, les applications réseau peuvent être classifiés en quatre catégories [Chen et al., 2004] :

[1] Applications temps réel interactives. Les applications temps réel interactives, comme par exemple la téléphonie sur IP et la visioconférence, sont très sensibles au délai de bout-en-bout, à la variation du délai (gigue) et à la perte d'informations. L'idéal pour ces applications temps réel interactives est de garantir un délai de bout-en-bout inférieur à 100 ms (respectivement, 400 ms) en local (respectivement, à longue distances), une gigue inférieure à 50 ms et un taux de perte inférieur à 3%.

[2] Streaming temps réel. Pour le streaming audio et vidéo, il est préférable de garantir un délai de bout-en-bout inférieur à 1 s. Pour les pertes, on est dans les mêmes limites des applications temps réel interactives (*i.e.,* taux de perte inférieur à 3%).

[3] Transfert des données interactives. Les applications de transfert des données interactives, comme la messagerie instantanée, sont beaucoup moins contraintes en terme de délai. Par exemple, un délai de bout-en-bout de 4 s peut être acceptable. De plus, ces applications ne sont pas affectées par les pertes de paquets grâce au contrôle de fiabilité utilisé dans le protocole TCP.

[4] Applications élastiques. Cette catégorie comprend diverses applications, comme par exemple le transfert des fichiers, le courrier électronique et la télécopie. La seule exigence pour les applications de cette catégorie est que les informations soient remises pratiquement sans erreur à l'utilisateur.

5.3.3.1 Travaux futurs

La classification des flux ouvre de nombreuses perspectives à court, moyen ou long terme pour une meilleure gestion de trafic et de la qualité de service.

Dans un premier temps, le réseau doit être capable d'extraire les informations nécessaires sur la sémantique des flux (comme par exemple, le débit crête du flux ainsi que le type d'application). Il est intéressant de noter qu'il existe des travaux de la littérature qui ont proposé des méthodes de classification des flux à partir de leurs 5 premiers paquets, comme par exemple la solution présentée dans [Jaber et al., 2012].

Dans un second temps, il serait intéressant d'appliquer notre nouvelle solution KBAC en prenant en compte ces diverses classes d'applications. Pour démarrer, il suffit de considérer deux catégories d'applications, les applications contraintes en terme de délai et les applications moins contraintes en terme de délai. On pourrait envisager que notre solution de contrôle d'admission admette trois régimes différents :

[1] des périodes avec une utilisation faible des ressources durant lesquelles le contrôle d'admission accepte tous les flux qui cherchent à entrer sur le lien de communication.

[2] des périodes de prévention de congestion où les flux qui sont moins contraints en terme de délai sont seulement contrôlés.

[3] des périodes de congestion où le contrôle d'admission contrôle tous les flux qui cherchent à entrer sur le lien de communication.

La classification des flux devrait permettre d'améliorer les performances des flux, et notamment de satisfaire les critères de performances exigés par les flux avec QoS grâce à un traitement différencié selon leurs caractéristiques.

TABLE 5.2 – Table des flux

Identifiant	Source	Destination	Chemin emprunté par le flux
Id 1	0	8	$(0 \to 1 \to 8)$
Id 2	0	10	$(0 \to 2 \to 10)$
Id 3	0	16	$(0 \to 2 \to 4 \to 5 \to 16)$
Id 4	0	22	$(0 \to 1 \to 9 \to 17 \to 20 \to 21 \to 22)$
Id 5	0	28	$(0 \to 2 \to 4 \to 15 \to 30 \to 27 \to 28)$
Id 6	8	0	$(8 \to 1 \to 0)$
Id 7	8	10	$(8 \to 7 \to 10)$
Id 8	8	16	$(8 \to 6 \to 12 \to 16)$
Id 9	8	22	$(8 \to 13 \to 25 \to 26 \to 24 \to 22)$
Id 10	8	28	$(8 \to 13 \to 15 \to 30 \to 27 \to 28)$
Id 11	10	0	$(10 \to 2 \to 0)$
Id 12	10	8	$(10 \to 7 \to 8)$
Id 13	10	16	$(10 \to 13 \to 12 \to 16)$
Id 14	10	22	$(10 \to 18 \to 23 \to 21 \to 22)$
Id 15	10	28	$(10 \to 18 \to 23 \to 24 \to 28)$
Id 16	16	0	$(16 \to 5 \to 4 \to 2 \to 0)$
Id 17	16	8	$(16 \to 12 \to 6 \to 8)$
Id 18	16	10	$(16 \to 12 \to 13 \to 10)$
Id 19	16	22	$(16 \to 12 \to 13 \to 25 \to 26 \to 24 \to 22)$
Id 20	16	28	$(16 \to 15 \to 30 \to 27 \to 28)$
Id 21	22	0	$(22 \to 21 \to 20 \to 17 \to 9 \to 1 \to 0)$
Id 22	22	8	$(22 \to 24 \to 26 \to 25 \to 13 \to 8)$
Id 23	22	10	$(22 \to 21 \to 23 \to 18 \to 10)$
Id 24	22	16	$(22 \to 24 \to 26 \to 25 \to 13 \to 12 \to 16)$
Id 25	26	0	$(28 \to 24 \to 23 \to 18 \to 10 \to 2 \to 0)$
Id 26	28	8	$(28 \to 24 \to 26 \to 25 \to 13 \to 8)$
Id 27	28	16	$(28 \to 24 \to 23 \to 18 \to 10)$
Id 28	28	10	$(28 \to 27 \to 30 \to 15 \to 16)$

A

Notions de Base

B

Bande passante (capacité). La bande passante d'un lien de communication désigne le débit maximal d'émission de données sur le lien réseau. Elle s'exprime généralement en kilobits, mégabits ou gigabits par seconde.

Bande passante résiduelle. La bande passante résiduelle ou disponible d'un lien de communication peut se définir comme le débit maximal qui peut être émis sur ce lien réseau sans dégrader aucun des flux présents dans ce lien.

C

Contrôle d'admission. Le contrôle d'admission vise à limiter le nombre de flux circulant dans un réseau afin de maintenir un niveau d'utilisation des ressources du réseau en deçà d'un certain seuil permettant d'offrir de bonnes performances aux flux, et notamment de satisfaire les critères de performances exigés par les flux avec QoS.

D

Délai de bout-en-bout. Le délai de bout-en-bout d'un paquet représente le temps d'acheminement d'un paquet d'un nœud émetteur vers un nœud destinataire. Il englobe trois aspects différents :

[1] Le délai de propagation, déterminé par la distance physique qui sépare la source de la destination

[2] Le délai de transmission, dépendant de la taille du paquet et des bandes passantes des liens traversés.

[3] Le délai d'attente, déterminé par le temps passé par un paquet à l'intérieur des buffers des liens successivement traversés.

F

Flux. Un flux représente une succession de paquets envoyés d'un nœud émetteur vers un nœud destinataire. Un flux regroupe tous les paquets ayant le même quintuplet [(i) adresse IP source ; (ii) adresse IP destination ; (iii) port source ; (iv) port destination ; (v) protocole de transmission].

G

Gigue (variation du délai). La gigue définit la variation de délai au moment de la réception entre deux paquets de données consécutifs d'un flux.

Q

Qualité de Service. La Qualité de Service (*QoS: Quality of Service*) désigne la capacité à véhiculer dans de bonnes conditions un type de trafic donné, en termes de disponibilité, débit, délai, taux de perte de paquets, etc.

T

Taux de Perte. Le taux de perte représente le pourcentage paquets perdus. Les pertes dans les réseaux peuvent être causées par les erreurs de transmission, la congestion, l'instabilité du routage, les défaillances des liens. La congestion est la cause la plus importante de pertes dans les réseaux filaires.

Publications

Conférences Internationales

[1] Doreid Ammar, Thomas Begin, Isabelle Guérin-Lassous and Ludovic Noirie. "KBAC: Knowledge-Based Admission Control," In *37th Annual IEEE Conference on Local Computer Networks (LCN 2012)*, Clearwater, Florida, USA, October 2012, pp. 218-225.

[2] Doreid Ammar, Thomas Begin, Isabelle Guérin-Lassous and Ludovic Noirie. "Evaluation and Comparison of MBAC Solutions," In *36th Annual IEEE Conference on Local Computer Networks (LCN 2011)*, Bonn, Germany, October 2011, pp. 215-218.

[3] Doreid Ammar, Thomas Begin and Isabelle Guérin-Lassous. "A new tool for generating realistic Internet traffic in NS-3," In *4th International ICST Conference on Simulation Tools and Techniques (SIMUTools)*, Barcelona, Spain, March 21-25, 2011.

Demonstrations

[4] Doreid Ammar, Julien Brochet, Thomas Begin, Isabelle Guérin-Lassous and Ludovic Noirie. "Knowledge-Based Admission Control: A Real-Time Performance Analysis," In *Demonstrations of the IEEE Conference on Local Computer Networks 2012 (LCN-Demos 2012)*, Clearwater, USA, October 2012.

[5] Doreid Ammar, Thomas Begin, Isabelle Guérin-Lassous and Ludovic Noirie. "Traffic-Aware Flow Admission Control," *Bell Labs Open Days 2012*, Alcatel-Lucent Bell Labs, Nozay, Villarceaux, France, 23-25 May 2012.

Conférences Nationales

[6] Doreid Ammar, Thomas Begin, Isabelle Guérin-Lassous and Ludovic Noirie. "Contrôle d'Admission Basé sur un Plan de Connaissance," *14èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel)*, La Grande Motte, France, 2012.

[7] Doreid Ammar, Thomas Begin, Isabelle Guérin-Lassous and Ludovic Noirie. "Contrôles d'admission basés sur des mesures : Evaluation et comparaison de solutions," *15th Colloque sur l'Ingénierie des Protocoles, CFIP11*, Sainte Maxime, France, 2011.

Rapports de Recherche

[8] Doreid Ammar, Julien Brochet, Thomas Begin, Isabelle Guérin-Lassous and Ludovic Noirie. "KBAC: Knowledge-Based Admission Control," *INRIA Research Report RR-7955*, May 2012.

[9] Doreid Ammar, Julien Brochet, Thomas Begin, Isabelle Guérin-Lassous and Ludovic Noirie. "Performance Evaluation of MBAC solutions," *INRIA Research Report RR-8080*, September 2012.

Logiciel

[10] Doreid Ammar. “Network Simulator 3: Poisson Pareto Burst Process,” - Available: http://codereview.appspot.com/4997043/

References

Bibliography

[All,]

[ns3, 2008] (2008). Network simulator3, http://www.nsnam.org/. online.

[lex, 2010] (2010). http://www.lexpansion.com/high-tech/at-t-sonne-la-fin-des-forfaits-mobiles-intern 233562.html. online.

[san, 2011] (2011). Global internet phenomena report: Spring 2011, http://www.sandvine.com/news/global_broadband_trends.asp. online.

[cis, 2012] (2012). http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html. online.

[Allen, 1990] Allen, A. O. (1990). *Probability, statistics, and queueing theory with computer science applications*. Academic Press Professional, Inc., San Diego, CA, USA.

[Begin et al., 2010] Begin, T., Brandwajn, A., Baynat, B., Wolfinger, B., and Fdida, S. (2010). High-level approach to modeling of observed system behavior. *Performance Evaluation*, 67(5).

[Breslau et al., 2000a] Breslau, L., Jamin, S., and Shenker, S. (2000). Comments on the Performance of Measurement-Based Admission Control Algorithms. In *Infocom*.

[Breslau et al., 2000b] Breslau, L., Knightly, E. W., Shenker, S., Stoica, I., and Zhang, H. (2000). Endpoint admission control: architectural issues and performance. *SIGCOMM Comput. Commun. Rev.*, 30.

[Carra et al., 2009] Carra, D., Avrachenkov, K., Alouf, S., Blanc, A., Nain, P., and Post, G. (2009). Passive Online RTT Estimation for Flow-Aware Routers using One-Way Traffic. Rapport de recherche RR-7124, INRIA.

[Chen et al., 2004] Chen, Y., Farley, T., and Ye, N. (2004). Qos requirements of network applications on the internet. *Inf. Knowl. Syst. Manag.*, 4(1):55–76.

[Clark et al., 2003] Clark, D. D., Partridge, C., Ramming, J. C., and Wroclawski, J. T. (2003). A knowledge plane for the internet. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 3–10, New York, NY, USA. ACM.

[Cox, 1984] Cox, D. R. (1984). Long-range dependence: A review. *H. A. David and H. T. David, editors, Statistics: An Appraisal*, pages 55–74.

[Cox and Isham, 1980] Cox, D. R. and Isham, V. (1980). Point Processes,. *Chapman and Hall*.

[Divakaran, 2010] Divakaran, D. M. (2010). *Dealing with Elephants in the Internet: Towards a Flow-Based Internet Architecture*. PhD thesis, ENS-Lyon, Université de Lyon, 46, allée d'Italie, 69364 Lyon cedex 07, France.

[Divakaran et al., 2009] Divakaran, D. M., Altman, E., Post, G., Noirie, L., and Vicat-Blanc Primet, P. (2009). From packets to XLFrames: sand and rocks for transfer of mice and elephants. In *IEEE INFOCOM 2009 Workshop on High-Speed Networks*, pages 1–6, Rio di Janerio, Brazil.

[Eun and Shroff, 2003a] Eun, D. Y. and Shroff, N. B. (2003). A measurement-analytic approach for qos estimation in a network based on the dominant time scale. *IEEE/ACM Trans. Netw.*, 11(2):222–235.

[Eun and Shroff, 2003b] Eun, D. Y. and Shroff, N. B. (2003). A measurement-analytic approach for qos estimation in a network based on the dominant time scale. *IEEE/ACM Trans. Netw.*, 11(2):222–235.

[Floyd, 1996] Floyd, S. (1996). Comments on Measurement-based Admissions Control for Controlled-Load Services. Technical report.

[Georgoulas et al., 2008] Georgoulas, S., Trimintzios, P., Pavlou, G., and Ho, K. (2008). An integrated bandwidth allocation and admission control framework for the support of heterogeneous real-time traffic in class-based ip networks. *Comput. Commun.*, 31(1):129–152.

[Greenberg et al., 2005] Greenberg, A., Hjalmtysson, G., Maltz, D. A., Myers, A., Rexford, J., Xie, G., Yan, H., Zhan, J., and Zhang, H. (2005). A clean slate 4d approach to network control and management. *SIGCOMM Comput. Commun. Rev.*, 35(5):41–54.

[Gross and Harris, 1985] Gross, D. and Harris, C. M. (1985). *Fundamentals of queueing theory (2nd ed.)*. John Wiley & Sons, Inc., New York, NY, USA.

[Grossglauser and Tse, 2003] Grossglauser, M. and Tse, D. N. C. (2003). A time-scale decomposition approach to measurement-based admission control. *IEEE/ACM Trans. Netw.*, 11(4):550–563.

[Guerin et al., 1991] Guerin, R., Ahmadi, H., and Naghshineh, M. (1991). Equivalent capacity and its application to bandwidth allocation in high-speed networks. *IEEE JSAC*, 9(7):968–981.

[II,] II, T. Brescia university. http://www.ing.unibs.it/ntw/tools/traces/.

[Jaber et al., 2012] Jaber, M., Cascella, R. G., and Barakat, C. (2012). Using host profiling to refine statistical application identification. In *INFOCOM'12*, pages 2746–2750.

[Jamin and Danzig, 1997] Jamin, S. and Danzig, P. B. (1997). A measurement-based admission control algorithm for integrated services packet networks. *IEEE/ACM Transactions on Networking*, 5(1):56–70.

[Jamin et al., 1997] Jamin, S., Shenker, S., and Danzig, P. B. (1997). Comparison of Measurement-based Admission Control Algorithms for Controlled-Load Service. In *Infocom*.

[Li, 2007] Li, J. (MIT-CSAIL-TR-2008-034, July 26, 2007). Agent organization and request propagation in the knowledge plane.

[Li, 2008] Li, J. (MIT-CSAIL-TR-2008-034, June 11, 2008). Agent organization in the knowledge plane.

[Lima et al., 2007] Lima, S. R., Carvalho, P., and Freitas, V. (2007). V.: Admission control in multiservice ip networks: Architectural issues and trends. *IEEE Computer Communications Magazine*, 45:114–121.

[Madhyastha et al., 2006a] Madhyastha, H., Isdal, T., Piatek, M., Dixon, C., Anderson, T., Krishnamurthy, A., and Venkataramani, A. (2006). iplane: an information plane for distributed services. In *OSDI '06: Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation*, pages 26–26, Berkeley, CA, USA. USENIX Association.

[Madhyastha et al., 2006b] Madhyastha, H. V., Anderson, T., Krishnamurthy, A., Spring, N., and Venkataramani, A. (2006). A structural approach to latency prediction. In *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 99–104, New York, NY, USA. ACM.

[Madhyastha et al., 2009] Madhyastha, H. V., Katz-Bassett, E., Anderson, T., Krishnamurthy, A., and Venkataramani, A. (2009). iplane nano: path prediction for peer-to-peer applications. In *NSDI'09: Proceedings of the 6th USENIX symposium on Networked systems design and implementation*, pages 137–152, Berkeley, CA, USA. USENIX Association.

[Mase, 2004] Mase, K. (2004). Toward scalable admission control for voip networks. *Communications Magazine, IEEE*, 42(7):42 – 47.

[Milbrandt et al., 2007a] Milbrandt, J., Menth, M., , and Junker, J. (2007). Improving Experience-Based Admission Control through Traffic Type Awareness. *Journal of Networks (JNW)*, Vol. 2, No.2.

[Milbrandt et al., 2007b] Milbrandt, J., Menth, M., , and Junker, J. (2007). Experience-Based Admission Control in the Presence of Traffic Changes. *Journal of Communications (JCM)*, Vol. 2, No.1.

[Milbrandt et al., 2004] Milbrandt, J., Menth, M., and Oechsner, S. (2004). Ebac - a simple admission control mechanism. In *ICNP 2004, 12th IEEE International Conference on Network Protocols*, Berlin, Germany.

[Moore and Crosby, 1999] Moore, A. and Crosby, S. (1999). An experimental configuration for the evaluation of cac algorithms. *SIGMETRICS Perform. Eval. Rev.*, 27(3):43–54.

[Moore, 2004] Moore, A. W. (2004). An implementation-based comparison of Measurement-Based Admission Control algorithms. *J. High Speed Netw.*, 13.

[Nevin et al., 2008] Nevin, A., Jiang, Y., and Emstad, P. J. (2008). Robustness study of mbac algorithms. In *ISCC*, pages 1040–1046. IEEE.

[Noirie et al., 2009] Noirie, L., Dotaro, E., Carofiglio, G., Dupas, A., Pecci, P., Popa, D., and Post, G. (2009). Semantic Networking: Flow-Based, Traffic-Aware, and Self-Managed Networking. *Bell Labs Technical Journal*, 14(2):22–38.

[Qiu and Knightly, 2001] Qiu, J. and Knightly, E. W. (2001). Measurement-Based Admission Control with Aggregate Traffic Envelopes. *IEEE/ACM Transactions on Networking*, 9(2):199–210.

[Sass, 2004] Sass, D. (2004). Internet traces of the"Selfnet" university dormitory network, Trace UST2, University of Stuttgart, Trace UST2.

[Statovci-Halimi, 2008] Statovci-Halimi, B. (2008). Support of ip multi-services through admission control. In *Innovations in NGN: Future Network and Services, 2008. K-INGN 2008. First ITU-T Kaleidoscope Academic Conference*, pages 407 –414.

[Wehmuth and Ziviani, 2011] Wehmuth, K. and Ziviani, A. (2011). Distributed assessment of network centrality. *CoRR*, abs/1108.1067.

[Yan et al., 2007] Yan, H., Maltz, D. A., Ng, T. S. E., Gogineni, H., Zhang, H., and Cai, Z. (2007). Tesseract: A 4d network control plane. In *NSDI*. USENIX.

[Zukerman et al., 2003] Zukerman, M., Neame, T. D., and Addie, R. G. (2003). Internet traffic modeling and future technology implications. In *Proceedings of INFOCOM*.