# Thèse de Doctorat de l'université Paris VI
# Pierre et Marie Curie

Spécialité

**Systèmes Informatiques**

présentée par

**Mlle. Jocelyne Elias**

pour obtenir le grade de

**Docteur de l'université Pierre et Marie Curie**

## Allocation dynamique de la bande passante dans les réseaux à qualité de service

Présentée et soutenue publiquement le 3 Juillet 2006 devant le jury composé de

**Jury**

| | | |
|---|---|---|
| **Dominique GAITI** | **Directeur** | **Prof. à l'Université de Troyes** |
| **Guy PUJOLLE** | **Directeur** | **Prof. à l'Université Pierre et Marie Curie** |
| **Tijani CHAHED** | **Rapporteur** | **Prof. à l'INT** |
| **Walid DABBOUS** | **Rapporteur** | **Directeur de Recherche à l'INRIA, Sophia-Antipolis** |
| **Luigi FRATTA** | **Examinateur** | **Prof. à Politecnico di Milano** |
| **Fabio MARTIGNON** | **Examinateur** | **Prof. à l'Université des études de Bergamo** |
| **Samir TOHME** | **Examinateur** | **Prof. à l'Université de Versailles Saint-Quentin-en-Yvelines** |

# Doctor of Science Thesis
# Pierre and Marie Curie University (Paris VI)

Specialization

**COMPUTER SCIENCE**

presented by

**Ms. Jocelyne ELIAS**

Submitted in partial satisfaction of the requirements for the degree of

**DOCTOR OF SCIENCE of the Pierre and Marie Curie University**

# Dynamic Resource Provisioning in Quality of Service Networks

## Committee in charge:

| | | |
|---|---|---|
| **Dominique GAITI** | **Advisor** | **Prof. at Troyes University** |
| **Guy PUJOLLE** | **Advisor** | **Prof. at Pierre and Marie Curie University** |
| **Tijani CHAHED** | **Reviewer** | **Prof. at INT** |
| **Walid DABBOUS** | **Reviewer** | **Research Director at INRIA, Sophia-Antipolis** |
| **Luigi FRATTA** | **Examinator** | **Prof. at Politecnico di Milano** |
| **Fabio MARTIGNON** | **Examinator** | **Prof. at University of Bergamo** |
| **Samir TOHME** | **Examinator** | **Prof. at UVSQ University** |

*To my family.*

# Résumé

Des algorithmes efficaces de gestion dynamique des ressources sont nécessaires pour le développement et l'automation des réseaux à qualité de service. Le but principal de ces algorithmes est d'offrir des services qui répondent aux exigences des utilisateurs en terme de qualité de service tout en garantissant en même temps aux opérateurs une utilisation efficace des ressources du réseau.

Dans cette thèse, nous proposons un nouveau modèle de service qui assure pour chaque flux une bande passante garantie ; de plus, le réseau individualise périodiquement la capacité libre et propose des contrats à court terme où cette capacité libre est allouée et garantie exclusivement aux utilisateurs qui peuvent l'exploiter pour transmettre à un débit supérieur à celui spécifié dans leur contrat.

Pour implémenter notre modèle de service nous proposons une architecture de gestion dynamique de la bande passante pour les réseaux à qualité de service. Nous développons une série d'algorithmes efficaces pour l'allocation dynamique de la bande passante qui prennent explicitement en considération les statistiques du trafic et les profils des utilisateurs pour augmenter les revenus du réseau et les bénéfices des utilisateurs.

En plus, nous proposons un modèle mathématique pour le problème d'allocation dynamique de la bande passante disponible qui permet de maximiser les revenus du réseau. La solution de ce modèle permet d'obtenir des limites supérieures sur les performances qui peuvent être atteintes avec n'importe quel algorithme "online" d'allocation de la bande passante.

Nous démontrons, à travers les résultats numériques et considérant des scénarios réels, que les algorithmes proposés pour l'allocation dynamique de la bande passante sont toujours supérieurs à l'allocation statique en termes de trafic total admis et revenus "extra" du réseau. En plus, ces résultats numériques montrent que les algorithmes proposés s'approchent, dans plusieurs scénarios, des performances idéales fournies par notre modèle mathématique.

## Mots-clés :

Allocation Dynamique de la Bande Passante, Qualité de Service, Utilisation efficace des ressources, Revenus Extra du réseau, Modèle Mathématique, Modèle de service.

# Abstract

Efficient dynamic resource provisioning algorithms are necessary to the development and automation of Quality of Service (QoS) networks. The main goal of these algorithms is to offer services that satisfy the QoS requirements of individual users while guaranteeing at the same time an efficient utilization of network resources.

In this thesis we introduce a new service model that provides per-flow bandwidth guarantees, where users subscribe for a guaranteed rate; moreover, the network periodically individuates unused bandwidth and proposes short-term contracts where extra-bandwidth is allocated and guaranteed exclusively to users who can exploit it to transmit at a rate higher than their subscribed rate.

To implement this service model we propose a dynamic provisioning architecture for intra-domain Quality of Service networks. We develop a series of efficient bandwidth allocation algorithms that take explicitly into account traffic statistics and users' utility functions to increase users' benefit and network revenue.

Further, we propose a mathematical formulation of the extra-bandwidth allocation problem that maximizes network revenue. The solution of this model allows us to obtain an upper bound on the performance achievable by any online allocation algorithm.

We demonstrate through simulation in realistic network scenarios that the proposed dynamic allocation algorithms are superior to static provisioning in providing resource allocation both in terms of total accepted load and network revenue, and they approach, in several network scenarios, the ideal performance provided by the mathematical model.

## Key Words:

Dynamic Bandwidth Allocation, Quality of Service, Efficient Resource Utilization, Network Extra-Revenue, Mathematical Model, Service Model.

# List of Publications


[1] A. Capone, J. Elias, F. Martignon, and G. Pujolle, "Dynamic Resource Allocation in Communication Networks," in *Proceedings of Networking 2006*, Coimbra, Portugal, 15-19 May, 2006, also published in Springer LNCS, Vol. 3976, pp. 892-903, 2006.

[2] A. Capone, J. Elias, F. Martignon, and G. Pujolle, "Distributed Dynamic Resource Allocation in Quality of Service Networks," in *Springer LNCS*, Vol. 3383, January, 2006.

[3] A. Capone, J. Elias, F. Martignon, and G. Pujolle, "Distributed Dynamic Bandwidth Provisioning in Quality of Service Networks," in *Proceedings of Third EuroNGI Workshop on QoS and Traffic Control*, ENS, Paris, France, 7-9 December, 2005.

[4] A. Capone, J. Elias, F. Martignon, and G. Pujolle, "Distributed Dynamic Resource Allocation in Quality of Service Networks," in *Second EuroNGI Workshop on New Trends in Network Architectures and Services*, Como, Italy, 13-15 July, 2005.

[5] J. Elias and D. Gaïti, "Contrôle de MPLS par l'utilisation des Systèmes Multiagents," in *DNAC-PARIS'04*, Paris, France, Novembre-Décembre, 2004.

[6] J. Elias, D. Gaïti, and G. Pujolle, "Optimisation du Protocole MPLS par l'utilisation des Systèmes Multiagents," in *Proceedings of 6èmes Journées Doctorales Informatique et Réseau (JDIR'04)*, Lannion, FranceTélécom R&D, France, 2-4 Novembre, 2004.

# Acknowledgements

I would like to thank my advisors Professors Guy Pujolle and Dominique Gaïti for their support and guidance during my PhD studies at "Laboratoire d'Informatique de Paris 6" (LIP6), "Pierre et Marie Curie" University. I will always appreciate their cheerful encouragement over these three years.

I would also like to thank Professors Tijani Chahed, Walid Dabbous, Luigi Fratta, Fabio Martignon and Samir Tohmé for accepting to be part of my Thesis Jury.

I would like to express my sincere thanks to Mr. Charles Tabet at Scientific National Research Council in Lebanon (CNRS-Lebanon) for supporting me to accomplish my PhD studies. I would also like to thank all the members of CNRS-Lebanon for attributing me during three years a 'Bourse' for PhD studies.

I am particularly grateful to Prof. Antonio Capone, Prof. Luigi Fratta and Mr. Fabio Martignon for my research visit at ANTLab Laboratory at Politecnico di Milano. In particular, I would like to thank Mr. Fabio Martignon for collaborating with me on several valuable ideas.

I want to thank all of my colleagues and friends at LIP6. Especially, I thank our secretary, Laetitia Jacquey, for her patience and willingness to provide us, when it is possible, all that we need.

This thesis is dedicated to my family, to my parents on both sides and to all my village. There is a long list of people to thank. All contributed in one way or another to this thesis. To them all - I am eternally thankful.

# Table of contents

# Résumé de la Thèse

## 1. La problématique

Des algorithmes efficaces de gestion dynamique des ressources sont nécessaires pour le développement et l'automation des réseaux à qualité de service. Dans les réseaux de communications, l'allocation de la bande passante se fait principalement d'une manière statique, sur des échelles de temps de l'ordre des heures aux mois. Cependant, les ressources du réseau gérées d'une manière statique peuvent devenir insuffisantes ou considérablement sous-utilisées quand les statistiques du trafic changent significativement [1].

Un point clef pour le déploiement des réseaux à qualité de service est le développement des solutions qui sont capables d'estimer dynamiquement les statistiques du trafic et d'allouer les ressources du réseau d'une manière efficace, en satisfaisant les exigences des utilisateurs et en maximisant en même temps l'utilisation des ressources et les revenus du réseau.

L'allocation dynamique de la bande passante dans les réseaux à qualité de service a récemment attiré l'attention des chercheurs et un nombre de solutions a été proposé dans la littérature [1, 2, 3, 4, 5, 6, 7, 8, 9]. Ces solutions et un état de l'art sur l'allocation dynamique de la bande passante sont présentés dans la section 3.

## 2. Les contributions de la thèse

Cette thèse propose un nouveau modèle de service qui assure pour chaque flux une bande passante garantie ; en plus, le réseau individualise périodiquement la capacité libre et propose des contrats à court terme où cette capacité libre est allouée et garantie exclusivement aux utilisateurs qui ont l'intention de payer plus pour pouvoir transmettre à un débit supérieur à celui spécifié dans leur contrat.

Pour implémenter notre modèle de service nous proposons une architecture de gestion dynamique de la bande passante pour les réseaux à qualité de service. Cette architecture est composée de routeurs de bord (les routeurs qui se trouvent à la frontière du réseau) et de routeurs de coeur (les routeurs qui se trouvent à l'intérieur du réseau).

Les routeurs de coeur mesurent la disponibilité de la bande passante sur les liens locaux et envoient périodiquement cette information concernant l'état du réseau aux routeurs d'entrée utilisant des messages de contrôle semblables à ceux utilisés dans [1]. En plus, si les routeurs de coeur détectent que certains seuils ont été dépassés, c'est-à-dire s'ils détectent une congestion persistante sur quelques liens, ils notifient immédiatement les routeurs à l'entrée afin de résoudre la situation de congestion.

Les routeurs d'entrée sont responsables d'estimer dynamiquement le nombre effectif de connexions actives et aussi leur débit actuel, comme proposé dans [8, 9]. Basés sur ces mesures et sur l'information sur l'état du réseau communiquée par les routeurs de coeur, les routeurs d'entrée allouent d'une manière efficace la capacité du réseau.

Pour ce but, nous développons une série d'algorithmes efficaces pour l'allocation dynamique de la bande passante. Ces algorithmes prennent explicitement en considération les statistiques du trafic et les profils des utilisateurs et leurs intentions de demander de la bande passante en plus, basés sur leur fonction d'utilité.

En plus, nous concevons et développons un modèle mathématique pour allouer la bande passante disponible et non utilisée dans le réseau. Ce modèle a pour but de maximiser les revenus du réseau. La solution du modèle mathématique permet d'obtenir une limite supérieure sur les performances qui pourront être atteintes avec n'importe quel algorithme "online" d'allocation dynamique de la bande passante.

Nous démontrons à travers les résultats numériques décrits dans le chapitre 7 que les algorithmes proposés pour l'allocation dynamique de la bande passante donnent toujours de meilleures performances par rapport à l'allocation statique en termes de trafic total accepté et revenus du réseau. En plus, les résultats obtenus dans le chapitre 8 montrent que les algorithmes proposés s'approchent des performances idéales fournies par notre modèle mathématique.

Les principales contributions de cette thèse sont alors les suivantes :

– La définition d'un nouveau modèle de service qui prend en compte les profils des utilisateurs et les statistiques du trafic offert au réseau.
  Les profils des utilisateurs sont représentés, dans ce travail, par les fonctions d'utilité de la bande passante. La communauté de recherche définit la fonction d'utilité comme étant une abstraction des préférences de l'utilisateur [10, 11]. Le concept de la fonction d'utilité peut être utilisé, d'une part, pour fournir une information à propos de la quantité des ressources dont une application a besoin, et d'autre part, pour supporter la détermination d'une solution adéquate pour le problème d'allocation de la bande passante [12].
  Les statistiques du trafic sont calculées par les moniteurs de trafic installés aux routeurs d'entrée. Le rôle de ces moniteurs est de mesurer le trafic entrant et sonder l'état de chaque connexion c'est-à-dire de savoir durant chaque intervalle de temps si la connexion est active ou non et en plus combien de trafic elle est en train de générer au réseau.
– La proposition d'un ensemble d'algorithmes heuristiques qui prennent en compte les statistiques du trafic mesurées en ligne, l'utilisation des liens et les fonctions d'utilité des utilisateurs pour allouer la capacité du réseau, répondant aux exigences de ces utilisateurs et maximisant l'utilité du réseau et parsuite, les bénéfices de l'opérateur.

Ces heuristiques étendent l'algorithme de partage équitable "max-min fair share" qui est introduit dans [13]. Le but de cet algorithme est d'allouer la capacité du réseau d'une façon équitable et indépendamment des caractéristiques diverses des applications du réseau.

– L'illustration d'un modèle mathématique pour le problème d'allocation dynamique de la bande passante ; la solution de ce modèle permet de maximiser les revenus du réseau et d'obtenir des limites supérieures sur les performances obtenues par n'importe quel algorithme "online" d'allocation dynamique de la bande passante. En plus, ce modèle mathématique se base sur le problème de maximisation des revenus du réseau qui est adressé par un grand nombre de chercheurs [1, 2, 3, 6, 7, 14, 15].

Le reste de ce résumé est organisé comme suit : la section 3 dresse un état de l'art sur l'allocation dynamique de la bande passante. Ensuite, la section 4 introduit notre modèle de service et l'architecture proposée pour la gestion dynamique des ressources, en plus, cette même section décrit le modèle mathématique et l'ensemble des heuristiques proposées pour l'allocation dynamique de la bande passante. La section 5 donne une discussion des résultats obtenus et finalement, la section 6 termine le résumé par les conclusions et les perspectives futures de notre recherche.

## 3. Allocation dynamique de la bande passante : Algorithmes et Architectures

Dans [13], un algorithme du partage équitable max-min est introduit pour allouer la bande passante disponible dans le réseau équitablement entre toutes les connexions qui partagent le même lien. Cette allocation suppose que les caractéristiques de toutes les connexions sont identiques. Essentiellement, ce mécanisme de contrôle de flux maximise la bande passante allouée à chaque application indépendamment des différences entre ces applications. La notion du partage équitable max-min est étendue par [6] pour prendre en compte les fonctions d'utilité des utilisateurs. Les auteurs de [6] ont introduit le critère "utility max-min fairness" pour maximiser l'utilité minimale reçue par une application qui partage les ressources en contrainte du réseau.

Dans ce travail, nous étendons l'algorithme de partage équitable max-min pour allouer la bande passante disponible sous forme de contrats à court terme aux utilisateurs qui ont l'intention de transmettre au-delà de la bande passante spécifiée dans leur contrat SLA (Service Level Agreement).

Des mécanismes de contrôle de flux qui sont basés sur la notion d'utilité sont adressés par un grand nombre de chercheurs [7, 14, 15] afin d'allouer la capacité du réseau de manière à maximiser les revenus de ce dernier.

Dans [7, 14], les auteurs considèrent exclusivement les fonctions d'utilité concaves pour lesquelles existent des solutions de la théorie étendue et des algorithmes comme les conditions de Karush-Kuhn-Tucker et le théorème de dualité. Les auteurs dans [15] ont élaboré le problème d'allocation distribuée de la bande passante pour les utilisateurs qui présentent des fonctions d'utilité générales (concaves et sigmoïdales) pour prendre en compte le cas où se présentent les diverses applications. En plus, ils ont démontré que le fait d'appliquer des algorithmes de contrôle de flux développés pour les fonctions d'utilité concaves dans un cas plus réel, celui des fonctions d'utilité générales, peut mener à une instabilité et à une grande congestion du réseau.

Dans notre modèle de service, nous ne nous limitons pas aux fonctions d'utilité concaves ; comme dans [15], nous permettons aux utilisateurs d'avoir des fonctions d'utilité générales qui est le cas naturellement dans le contexte d'applications diverses. Cependant, pour résoudre le problème d'allocation de la bande passante disponible dans le réseau, nous distinguons deux cas : 1) tous les utilisateurs ont des fonctions d'utilité concaves et 2) les utilisateurs ont des fonctions d'utilité générales.

Dans le premier cas, nous proposons une formulation mathématique du problème de maximisation du revenu total du réseau tandis que, dans le deuxième cas, nous introduisons deux algorithmes heuristiques d'allocation qui prennent en compte les statistiques du trafic calculées par les routeurs d'entrée.

La problématique de concevoir des architectures pour la gestion dynamique des ressources dans les réseaux à qualité de service a récemment attiré l'attention de plusieurs chercheurs due à son potentiel de réaliser une utilisation efficace des ressources tout en satisfaisant à toutes les exigences des utilisateurs en terme de qualité de service [1, 2, 3, 4, 5].

Dans [1, 3], les auteurs proposent deux architectures de gestion dynamique des ressources qui s'appliquent au bord et au coeur du réseau, respectivement. La première architecture a pour but de partager les ressources au niveau des liens d'entrée et de dimensionner la capacité sur les liens de sortie. La deuxième architecture comporte deux catégories d'algorithmes : ceux qui s'appliquent localement au niveau des noeuds à l'intérieur du réseau et un qui s'applique sur le coeur du réseau d'une façon globale.

L'algorithme appliqué au niveau du noeud adopte un mécanisme d'auto-adaptation pour ajuster les poids des services dans un ordonnanceur de paquets équitable et pondéré (Weighted Fair Queuing (WFQ)). L'algorithme appliqué au niveau du réseau permet de réduire la bande passante aux routeurs d'entrée immédiatement après la réception d'un alarme de congestion des algorithmes appliqués au niveau des noeuds. En plus, cet algorithme réaligne périodiquement la bande passante pour établir une allocation max-min modifiée entre les trafics agrégés.

Le travail discuté dans [1] a des objectifs qui sont similaires aux nôtres, cependant, le modèle de service considéré dans [1, 3] est différent de notre modèle et les statistiques du trafic ne sont pas prises en compte dans leur procédure d'allocation dynamique de la bande passante. En plus, dans notre travail, nous adoptons une architecture distribuée tandis qu'eux implémentent leurs architectures d'une façon centralisée.

Une architecture à base de politiques est présentée dans [4] où les auteurs proposent une approche basée sur les mesures (measurement-based) pour l'adaptation dynamique de la qualité de service dans les réseaux DiffServ. Cette architecture est composée d'un PDP

(Policy Decision Point), d'un ensemble de PEP (Policy Enforcement Point) installés aux routeurs d'entrée et de moniteurs de trafic qui sont implémentés dans les noeuds au coeur du réseau. Quand les moniteurs détectent des changements significatifs dans la bande passante disponible, ils contactent directement le PDP qui change dynamiquement les politiques sur les trafics d'entrée "in-profile" et "out-of-profile" en se basant sur l'état actuel du réseau qui est estimé utilisant l'information collectée des moniteurs de trafic. Cependant, cette architecture, même si elle performe une adaptation dynamique de la qualité de service des applications multimédia, elle ne prend pas en compte les fonctions d'utilité des utilisateurs et leur intention de payer plus pour transmettre du trafic "out-of-profile", augmentant ainsi les revenus du réseau.

Une structure générique de chargement est présentée dans [2] pour caractériser les schémas de chargement actuellement utilisés dans l'Internet. En plus, un algorithme de chargement dynamique et sensible à la congestion est proposé pour inciter les applications multimédia à adapter leur débit en fonction des conditions actuelles du réseau. Comme dans [2], nous prenons en compte les fonctions d'utilité des utilisateurs pour évaluer nos algorithmes d'allocation dynamique de la bande passante en terme de l'augmentation des revenus du réseau. Cependant, les auteurs dans [2] considèrent un modèle de service différent et se focalisent principalement sur l'idée de charger dynamiquement les services afin de performer une adaptation du débit basée sur les conditions du réseau.

L'idée de mesurer dynamiquement le nombre effectif de connexions actives avec leur débit actuel est une technique qui est adoptée dans plusieurs travaux de recherche [5, 8, 9]. Dans [5], les auteurs proposent une approche active de gestion de ressources (ARM) dans un environnement DiffServ. Le concept de base de ARM est de savoir effectivement à chaque moment quand un client est en train de transmettre et combien il est en train d'utiliser de sa bande passante allouée. La bande passante non utilisée par ces clients peut être réallouée à d'autres clients. Ce concept est en ligne avec nos objectifs ; cependant, différemment de notre travail, ARM ne garantit pas à l'utilisateur une bande passante minimale pendant la durée du contrat du fait que la bande passante non utilisée par un utilisateur est coupée et stockée dans une pile de bande passante disponible pour être utilisée pour admettre de nouvelles connexions en dépit de celles qui sont déjà admises.

# 4. Le modèle de service et l'architecture de gestion dynamique des ressources proposés

Dans cette section, nous commençons d'abord par la description de notre modèle de service, puis nous présentons notre architecture proposée pour la gestion dynamique des ressources, avec les messages de contrôle utilisés pour assurer l'interaction entre les éléments du réseau. Enfin, nous donnons une synthèse du modèle mathématique et des algorithmes conçus pour l'allocation dynamique de la bande passante.

## 4.1. Le modèle de service

Nous proposons un modèle de service qui, tout d'abord, fournit pour chaque flux une bande passante garantie ; en plus, il exploite périodiquement la bande passante disponible individualisée dans le réseau pour proposer aux utilisateurs qui ont l'intention de payer plus de la bande passante "extra" de court terme et garantie.

Dans ce processus, les utilisateurs peuvent être attribués des poids divers de sorte que la bande passante disponible sera allouée avec différentes priorités. Ces poids peuvent être spécifiés d'une manière statique (offline), tenant compte du contrat de l'utilisateur, ou adaptés "online" tenant compte, par exemple, da la fonction d'utilité de l'utilisateur.

Notre modèle de service est alors caractérisé par :

- Une garantie quantitative de bande passante qui est exprimée à travers la spécification du contrat de l'utilisateur.
- Une bande passante "extra" garantie de court terme : le réseau est monitoré en

ligne pour individualiser la bande passante non utilisée qui est ensuite allouée aux utilisateurs qui peuvent l'exploiter pour transmettre du trafic "extra" pour une durée spécifiée dans leur contrat. La durée du contrat dépend des besoins de l'utilisateur et peut durer sur plusieurs intervalles de temps consécutifs.

– Le poids qui exprime comment chaque utilisateur partage les ressources du réseau avec les autres utilisateurs durant la procédure d'allocation de la bande passante.

Dans ce présent travail, nous supposons, pour simplifier, que la durée des contrats est minimale, c'est-à-dire elle est égale à un seul intervalle de temps ; cela nous permet de dériver des limites supérieures sur les performances qui peuvent être obtenues par notre approche.

## 4.2. L'architecture de gestion dynamique des ressources

Pour implémenter notre modèle de service, nous considérons une architecture distribuée constituée par des routeurs de bord et des routeurs de coeur, comme illustré dans la Figure 1 ; les moniteurs de trafic sont installés aux routeurs de bord et de coeur pour mesurer en ligne le trafic entrant et l'utilisation de la capacité du réseau, respectivement.

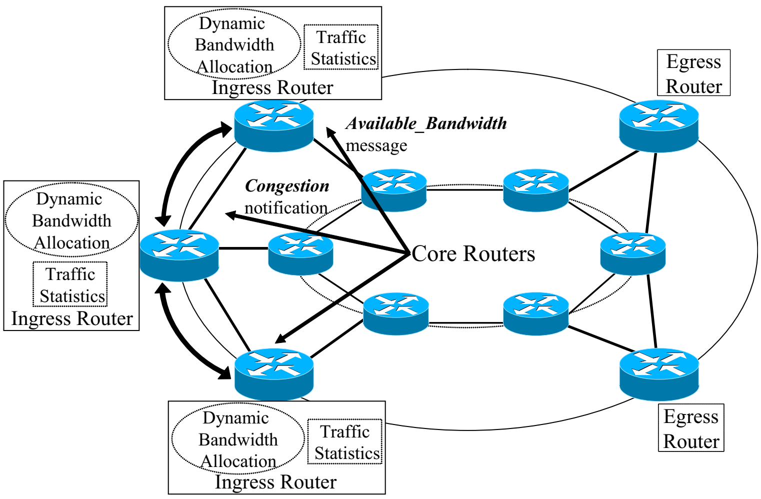



FIG. 1 – Notre architecture distribuée proposée pour l'allocation dynamique de la bande passante

Les moniteurs de trafic installés aux routeurs d'entrée ont pour but de mesurer le débit de transmission et aussi le débit offert pour chaque connexion entrant au réseau.

Les routeurs de coeur échangent des messages avec les routeurs d'entrée pour reporter l'utilisation de la capacité sur les liens ou pour notifier une situation de congestion.

Chaque routeur d'entrée collecte les mesures effectuées par les moniteurs de trafic, et échange périodiquement avec tous les autres routeurs d'entrée des messages de mise à jour des statistiques actuelles du trafic entrant.

Chaque intervalle de temps, par exemple chaque $T_u$ secondes, tous les routeurs d'entrée exécutent le même algorithme considérant plusieurs contraintes comme les statistiques du trafic mesurées en ligne, l'état actuel du réseau reporté par les routeurs de coeur et enfin les profils des utilisateurs pour allouer les ressources du réseau d'une manière dynamique et efficace.

A noter que les intervalles de temps, $T_u$, sont considérés de l'ordre de dizaines de secondes, parsuite, une synchronisation précise entre les routeurs d'entrée n'est pas aussi cruciale. Nous discutons en détails le "tradeoff" derrière le choix de l'intervalle de temps dans le chapitre 7.

Les messages échangés entre les éléments du réseau, et illustrés par des flèches sur la Figure 1, sont similaires aux messages de contrôle proposés par [1] pour reporter la persistance de la congestion ou la disponibilité de la bande passante sur certains liens. En plus, une partie des messages définis pour le protocole RNAP présenté dans [16] peuvent aussi être utilisés pour le même but.

## 4.3. Les Algorithmes d'allocation dynamique de la bande passante et le Modèle Mathématique

Cette section résume l'ensemble des heuristiques et le modèle mathématique proposés dans cette thèse pour l'allocation dynamique de la bande passante disponible dans le réseau. Ces algorithmes et le modèle mathématique sont discutés en détails dans les chapitres 5 et

6.

Nous proposons trois algorithmes heuristiques qui prennent en compte les statistiques du trafic mesurées en ligne, l'utilisation des liens et les fonctions d'utilité des utilisateurs, pour allouer la capacité du réseau, répondant aux exigences des utilisateurs et maximisant en même temps les revenus du réseau.

Ces trois algorithmes sont :

- Optimum Bandwidth Allocation Algorithm (OBA) : cet algorithme permet d'allouer la bande passante de sorte de maximiser en même temps les revenus du réseau et l'utilité des utilisateurs en tenant compte des fonctions d'utilité de ces derniers.
- Simple Dynamic Bandwidth Allocation (SDBA) : cet algorithme étend l'algorithme de partage équitable max-min pour allouer équitablement la bande passante non utilisée entre les connexions qui partagent le même lien et veulent transmettre à un débit supérieur à celui specifié dans leur contrat.
- Iterative Dynamic Bandwidth Allocation (IDBA) : cet algorithme constitue une version itérative et étendue de SDBA. IDBA distribue la bande passante disponible tenant compte du trafic offert par chaque connexion ; en effet, avec IDBA, une connexion ne peut pas être assignée une bande passante supérieure à son propre trafic offert pour éviter de gaspiller les ressources du réseau.

L'ensemble des algorithmes procède en deux étapes principales :

- Etape 1, la bande passante est distribuée à toutes les connexions actives en essayant de satisfaire leurs exigences à court terme qui sont estimés en fonction des statistiques collectées par les routeurs d'entrée.
- Etape 2, la bande passante disponible et inutilisée par les connexions (inactives et actives) est individualisée sur chaque lien. Cette bande est allouée avec garantie, durant chaque intervalle de temps, exclusivement aux connexions qui peuvent tirer profit d'elle comme elles ont déjà exploité complètement la bande passante spécifiée dans leur contrat de service.

La première étape est commune pour les trois algorithmes tandis que la seconde étape diffère pour chacun. Une description plus détaillée des algorithmes est donnée ultérieurement dans le chapitre 5.

En plus, pour obtenir une limite supérieure sur les performances qui peuvent être atteintes avec n'importe quel algorithme "online" d'allocation dynamique de la bande passante, nous développons un modèle mathématique qui s'appelle "Ideal Bandwidth Allocation" (IBA). Ce modèle résout le problème de maximisation des revenus du réseau, en supposant la connaissance à l'avance du trafic offert à ce dernier. Le modèle IBA est décrit en détails dans le chapitre 6.

# 5. Un extrait des résultats numériques

Pour évaluer les performances des algorithmes et du modèle mathématique proposés dans cette thèse, nous envisageons plusieurs topologies de réseau. Par contre, cette présente section se limite à élaborer une discussion sur les résultats obtenus dans le cas d'un simple scénario [1, 2] qui est illustré dans la Figure 2(a). Les diverses topologies considérées et les résultats numériques obtenus sont discutés en détails dans les chapitres 7 et 8.

Le scénario présenté dans la Figure 2(a) est constitué de deux routeurs de coeur, six routeurs de bord, et de 20 paires de source et destination. Les liens sont full-duplex et ont un délai de propagation égal à 1 ms. Les capacités des liens figurent à coté des liens sur la Figure.

Nous considérons 20 sources de trafic On/Off à durées de distribution exponentielle, la durée moyenne des périodes On est de 200 s et celle des périodes Off varie de 0 entre 150 s pour simuler différentes conditions de trafic offert au réseau et aussi pour varier le pourcentage de la bande passante inutilisée par les connexions lorsque ces dernières deviennent inactives.

Durant la période On, chaque source génère du trafic à un débit constant que nous appelons débit de crête. Six sources ont un débit de crête de 50 kb/s et un débit spécifié dans le contrat de 150 kb/s, 8 sources ont un débit de crête de 250 kb/s et un débit spécifié dans le contrat de 200 kb/s, le reste des sources ont comme débit de crête 1 Mb/s et comme débit spécifié dans le contrat 500 kb/s. La bande passante minimale requise par chaque source est égale à 10 kb/s. L'intervalle de temps , $T_u$, est égal à 20 s.

Nous supposons que tous les utilisateurs ont le même poids et le même type de fonction d'utilité qui est proposé dans [14], $U_k(x) = a \cdot log(b + x)$, où $a \geq 0$ et $0 \leq b \leq 1$. Nous assignons $U_k(x) = 0.5 \cdot log(1 + x)$ pour les sources qui ont comme débit spécifié dans le contrat 200 kb/s et $U_k(x) = 1.5 \cdot log(1 + x)$ pour celles qui ont comme débit spécifié dans le contrat 500 kb/s.

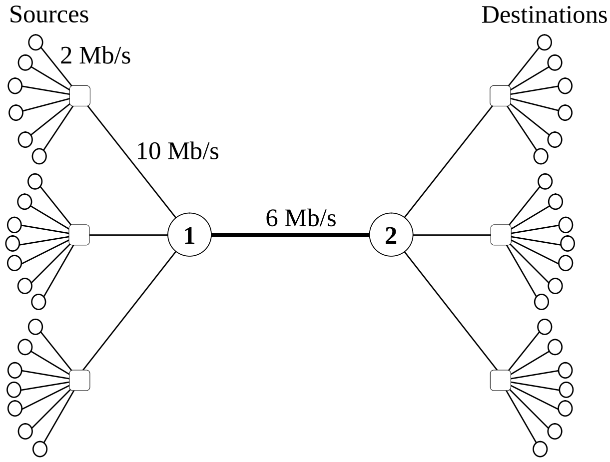
Les figures 2(b) et 2(c) montrent, respectivement, le trafic total moyen admis dans le réseau et le revenu extra total correspondant en fonction de la charge totale moyenne offerte au réseau.

Ces résultats vérifient que tous les algorithmes proposés, SDBA, IDBA, OBA et IBA, donnent de meilleures performances par rapport à la technique d'allocation statique de la bande passante.

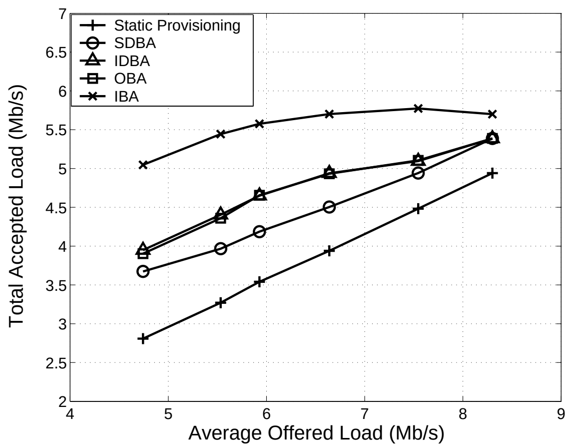
Nous pouvons observer que les meilleures performances sont données par IBA. Avec ce dernier, nous pouvons obtenir une amélioration de 17% pour le trafic total admis et 31% pour le revenu extra correspondant, par rapport à OBA.

En plus, nous constatons que IDBA et OBA présentent les mêmes performances en terme de trafic total admis, cependant OBA permet d'obtenir un revenu extra supérieur à celui donné par IDBA et SDBA. Cela est attendu parce que OBA distribue la bande passante disponible dans le réseau en tenant compte des fonctions d'utilité des utilisateurs, ce qui n'est pas le cas dans IDBA et SDBA.

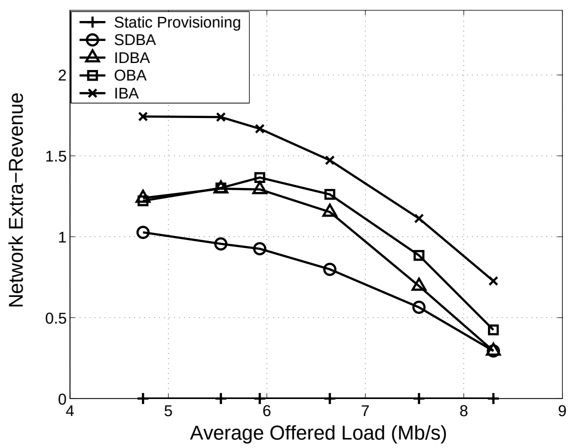
Nous ajoutons que les courbes de SDBA, IDBA, OBA et IBA suivent la même allure, ce qui prouve une uniformité entre ces différents algorithmes. En particulier, à fur et à mesure que le réseau devient surchargé, la courbe de OBA s'approche de celle de IBA .




(a) Scénario "Single-bottleneck"




(b) Trafic total moyen admis dans le réseau en fonction du trafic total moyen offert au réseau




(c) Revenus "extra" moyens du réseau en fonction du trafic total moyen offert au réseau

FIG. 2 – Performances obtenues avec nos algorithmes d'allocation dynamique de la bande passante dans le cas du scénario "Single-bottleneck"

# 6. Conclusion générale et Perspectives

Dans cette thèse, nous avons proposé un nouveau modèle de service où les utilisateurs ont une bande passante garantie, puis, le réseau individualise périodiquement la bande passante disponible et non utilisée pour être réallouée avec garantie sous forme de contrats de court terme aux utilisateurs qui peuvent l'exploiter de la meilleure façon selon leurs besoins et leur fonction d'utilité.

Nous avons décrit une architecture distribuée pour la gestion dynamique des ressources dans les réseaux à qualité de service et nous avons développé un ensemble d'algorithmes efficaces pour l'allocation dynamique de la bande passante. Ces algorithmes prennent en compte les statistiques du trafic satisfaisant les préférences des utilisateurs et augmentant les revenus du réseau. Nous avons aussi développé un modèle mathématique qui fournit une allocation optimale de la bande du point de vue de l'opérateur, maximisant les revenus du réseau.

Nous avons démontré, considérant des scénarios de réseau réels, que nos algorithmes d'allocation et notre modèle de service garantissent une meilleure utilisation de ressources et par la suite, une augmentation des revenus du réseau, en comparaison aux techniques d'allocation statiques. En plus, nous avons comparé les performances de nos algorithmes heuristiques aux limites idéales fournies par notre modèle mathématique conçu pour résoudre le problème d'allocation de la bande passante non utilisée. Cela nous a permis de spécifier des limites théoriques sur les performances obtenues par nos algorithmes et de juger combien sont loins nos algorithmes de ces limites.

## Perspectives

Dans cette thèse, nous avons étudié la problématique d'allocation dynamique de la bande passante considérant une stratégie de routage fixe. Nous avons supposé que chaque communication entre deux utilisateurs est établie en créant une session impliquant un chemin qui reste fixe tout le long de la durée de la conversation. En plus, la méthode du choix du chemin de la session, c'est-à-dire l'algorithme du routage, n'est pas considérée.

Dans [7, 17], les auteurs ont adressé le problème de chargement, contrôle du flux et routage dans un réseau présentant uniquement un trafic élastique.

Pour cette raison, nous envisageons de prendre en compte le routage pour étendre notre travail de recherche. Cette perspective devra être analysée profondément afin d'aboutir à augmenter encore les revenus du réseau tout en garantissant aux utilisateurs leurs différentes exigences en terme de qualité de service.

D'autre part, une réalisation flexible et efficace d'une allocation dynamique de la bande passante dans les réseaux sans fil présentent un nombre de points techniques critiques. Donc une future perspective de notre travail sera consacrée à étudier les points critiques dans les réseaux sans fil et étendre notre modèle de service, en tenant compte du routage, dans ce type de réseau.

# Chapter 1

# Introduction

The Internet was originally designed to support mainly best-effort services like file transfer, electronic email and remote login. However, with the emerging and widespread use of multimedia applications, Internet users are now exchanging through the network real-time traffic like voice and video. Multimedia applications usually require Quality of Service (QoS) guarantees from the network, in terms of throughput, packet loss, delay, and jitter.

Several approaches have been proposed to support QoS in communications networks, such as network resource reservation [18, 19], admission control [20, 21, 22, 23], scheduling mechanisms [24], service differentiation at network nodes [25] and dynamic bandwidth provisioning [1, 2, 4, 5, 6, 7, 17].

Efficient dynamic resource provisioning mechanisms are necessary to the development and automation of Quality of Service networks. In communication networks, resource allocation is performed mainly in a static way, on time scales on the order of hours to months. However, statically provisioned network resources can become insufficient or considerably under-utilized if traffic statistics change significantly [1].

A key challenge for the deployment of Quality of Service networks is the development of solutions that can dynamically track traffic statistics and allocate network resources efficiently, satisfying the QoS requirements of users while aiming at maximizing, at the same time, resource utilization and network revenue.

Therefore, in this thesis, we focus on the issue of dynamic bandwidth allocation; we propose and evaluate a new service model that considers users' requirements. Moreover, we develop a broad set of dynamic bandwidth allocation algorithms that achieve a high network utilization and satisfy users' preferences.

## 1.1 Thesis Contribution

In this thesis, we first propose a new service model that provides quantitative per-flow bandwidth guarantees, where users subscribe for a guaranteed transmission rate. Moreover, the network periodically individuates unused bandwidth and proposes short-term contracts where extra-bandwidth is allocated and guaranteed exclusively to users who are willing to pay for it to transmit at a rate higher than their subscribed rate.

To implement this service model we propose a distributed provisioning architecture composed by core and edge routers; core routers monitor bandwidth availability and periodically report this information to ingress routers using signalling messages like those defined in [1, 2]. Moreover, if persistent congestion is detected, core routers notify immediately ingress routers.

Ingress routers perform a dynamic tracking of the effective number of active connections, as proposed in [8, 9], as well as of their actual sending rate. Based on such information and that communicated by core routers, ingress routers allocate network resources.

For this purpose, we develop a broad set of efficient dynamic bandwidth allocation algorithms that take into account online measured traffic statistics as well as users' profile and willingness to acquire extra-bandwidth based on their bandwidth utility function.

Further, we propose a novel and flexible mathematical model for the extra-bandwidth allocation problem. This model assumes the exact knowledge of the future traffic offered to the network; it allows to maximize the total network revue, and provides theoretical upper bounds to the performance achievable by any online dynamic bandwidth allocation algorithm.

We evaluate by simulation the performance of our proposed bandwidth allocation algorithms in realistic network scenarios. Numerical results show that our algorithms and service model allow to achieve better performance than statically provisioned networks both in terms of total accepted load and network revenue. Furthermore, simulation results show that the proposed algorithms approach, in several network scenarios, the ideal performance provided by the mathematical model.

In summary, this thesis makes the following main contributions:

- The definition of a new service model that takes into account users' profile and traffic statistics.

- The proposition of a broad set of efficient heuristic algorithms for dynamic bandwidth allocation that aim at maximizing both users utility and network revenue.

- The illustration of a mathematical model for the bandwidth allocation problem; the solution of this model maximizes the network revenue and allows to obtain an upper bound on the performance achievable by any online allocation algorithm.

## 1.2 Thesis Structure

The thesis is structured as follows:

Chapter 2 discusses the work related to the dynamic bandwidth allocation problem and compares our approach to the literature.

Chapter 3 introduces the proposed service model that allows to maximize at the same time users' utility and total network revenue. To implement such service model, this Chapter further proposes a dynamic provisioning architecture and a set of control messages that assure communication between network elements.

Chapter 4 presents a precise statement of the bandwidth allocation problem. Further, a utility-based definition of the network extra-revenue is provided.

Chapter 5 proposes a set of heuristic bandwidth allocation algorithms tailored to our proposed service model. These heuristics take into account online traffic statistics and users' profiles to distribute efficiently extra-bandwidth to users who pay for it to transmit extra-traffic.

Chapter 6 introduces a novel and flexible mathematical programming model that provides an optimal bandwidth allocation capable of accommodating the traffic offered to the network.

Chapter 7 discusses simulation results that show the efficiency of our dynamic bandwidth allocation algorithms in providing resource allocation both in terms of total accepted load and network revenue compared to static provisioning.

Chapter 8 presents numerical results showing that the proposed algorithms approach, in several network scenarios, the ideal performance provided by the mathematical model.

Finally, the conclusions and directions for future work are presented in Chapter 9.

# Chapter 2

# Dynamic Bandwidth Allocation: Algorithms and Architectures

This Chapter provides the background required for the understanding of the bandwidth allocation problem. Dynamic bandwidth allocation algorithms are essential to guarantee Quality of Service (QoS) and fairness to connections competing for available network resources. Several algorithms have been proposed to allocate network capacity equally among all connections bottlenecked at the same link. However, these algorithms can not guarantee to service providers a high network revenue, since the available bandwidth is allocated with no regard to differences between network applications.

Therefore, to improve network revenue, many algorithms have been introduced to allocate available bandwidth taking into account users utility.

Recently, pricing-based algorithms have also been proposed to address the bandwidth allocation problem. The basic idea is to perform a congestion-dependent pricing, so that network applications are incited to adapt their transmission rates according to network conditions. Dynamic source tracking mechanisms have also been adopted to improve the network revenue.

Several dynamic provisioning architectures are designed to support QoS in communications networks, implementing various bandwidth allocation algorithms. However, these architectures are generally implemented in a centralized way and do not take into account users' utility functions and traffic statistics in the bandwidth allocation procedure.

Admission control mechanisms are complementary to dynamic bandwidth provisioning algorithms; they can be implemented in provisioning architectures at the edge routers to ensure that the network will not be overloaded.

The focus of this thesis is on proposing dynamic bandwidth allocation algorithms that allow to satisfy the QoS requirements of individual users while guaranteeing at the same time an efficient utilization of network resources.

The structure of this Chapter is the following: Section 2.1 introduces the notion of bandwidth utility function and dynamic pricing. Section 2.2 presents the work related to the fair share bandwidth allocation, then, it surveys the most notable utility and pricing-based bandwidth allocation algorithms and, finally, it describes related work on the dynamic tracking of active sources in the network. Section 2.3 describes briefly different dynamic bandwidth provisioning architectures for intra-domain quality of service networks.

## 2.1 Utility Functions and Pricing

The networking research community considers utility functions as an abstraction of a user's preference [10, 11]. The concept of utility function can be used to provide information about the amount of resources needed by each application and also to support the determination of a solution for the bandwidth allocation problem [12]. Utility functions establish a common ground that allows to relate the performance of different applications and to obtain the optimal bandwidth allocation solution.

Breslau and Shenker introduce in [10, 11] several types of bandwidth utility functions, for both adaptive and rigid applications, which are presented in some detail in the following.

### Utility Function Types

In a multi-application network environment, different types of utility functions (concave, step, s-shaped, linear, etc.) are envisioned and have been qualitatively described in [10, 11].

Applications can be classified based on their nature. Applications that generate data independently of the current network state are called *real-time* applications. These applications can be further classified into real-time applications with hard requirements and less stringent requirements, respectively. The latter ones are called adaptive real-time applications.

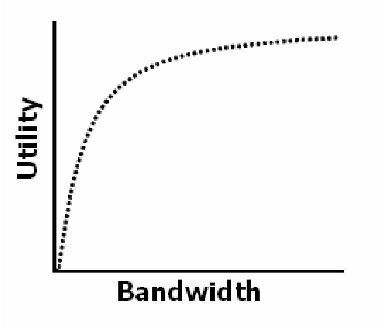
On the other hand, applications that can adapt their data generation rate according to the network conditions are called *elastic* applications.

Traditional data applications like file transfer, electronic mail, and remote terminal are examples of elastic applications. These applications are rather tolerant to network delay and present a decreasing utility improvement like that illustrated in Figure 2.1(a).

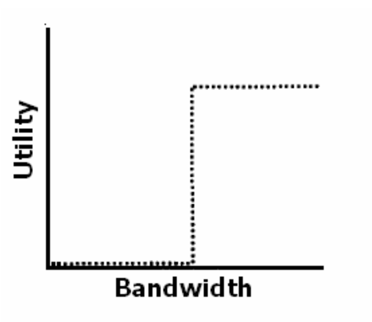
On the other hand, some applications are characterized by hard real-time requirements (e.g. guaranteed bandwidth, very low delay and jitter). Examples of such applications are link emulation, IP telephony, and other applications that expect circuit-switched-like service.

For applications with hard real-time requirements, the typical utility function is represented in Figure 2.1(b): while the minimum guaranteed bandwidth is met, the application performance is constant, but as soon as the allocated bandwidth drops below such minimum, the performance falls sharply to zero.

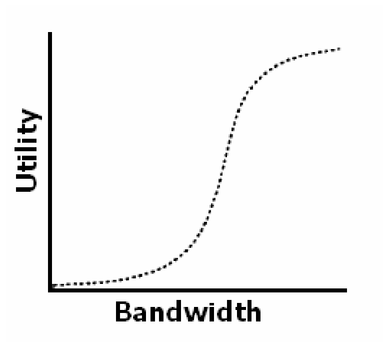
Traditionally, video and audio applications have been designed with hard real-time requirements. However, recent experiments on the Internet have shown that most of these applications can be implemented to be rather tolerant of occasional delay-bound violations and dropped packets. However, such applications still need a guaranteed bandwidth since they generate data independently of the network conditions (like, for example, congestion events). Thus, the performance degrades rapidly as soon as the allocated bandwidth be-



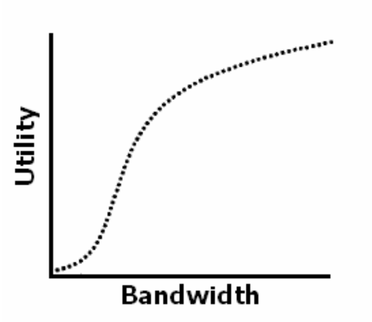

(a) Elastic Utility




(b) Hard Real-time Utility




(c) Delay-Adaptive Utility




(d) Rate-Adaptive Utility

Figure 2.1: Examples of bandwidth utility functions: (a) Elastic utility for data transfer applications, (b) Hard real-time utility for audio and video applications, (c) utility for delay-adaptive real-time applications and (d) utility for rate-adaptive real-time applications

comes smaller than the application transmission rate. For delay-adaptive audio and video applications, the utility function curves can look like that in Figure 2.1(c).

Another class of real-time applications is represented by rate-adaptive applications. These applications do not require hard bandwidth guarantees and hence they can adapt their transmission rate in response to network congestion. The utility function for rate-adaptive applications has the shape shown in Figure 2.1(d).

### Pricing

The pricing can represent an important component of bandwidth utility functions [2, 26]. Dynamic pricing mechanisms can take advantage of application adaptivity to increase network availability, revenue, and user-perceived utility [2].

In a network with QoS support, pricing of network services based on the level of service, usage, and congestion provides an incentive for applications to adapt their sending rates according to network conditions. Increasing the price during congestion gives the application an incentive to back-off its sending rate and at the same time allows an application with more stringent bandwidth and QoS requirements to maintain a high quality by paying more.

A number of pricing schemes are used in the Internet today like access-rate-dependent charge, volume-dependent charge, or the combination of the both. These charging schemes are described in [26].

## 2.2 Dynamic Bandwidth Allocation Algorithms

### 2.2.1 Max-Min Fair Allocation Algorithms

Several algorithms have been proposed to allocate network resources fairly among competing connections. A max-min fair allocation algorithm is introduced in [13] to allocate available bandwidth equally among all connections bottlenecked at the same link. In essence, this allocation mechanism attempts to maximize the bandwidth allocated to each application, with no regard to differences in applications.

In our work we extend the max-min fair allocation algorithm to perform a periodical allocation of unused bandwidth, through short-term contracts, to users who are willing to transmit more than their subscribed rate.

### 2.2.2 Utility and Pricing-based Dynamic Bandwidth Allocation Algorithms

An extension to the max-min fair allocation algorithm that takes into account users' bandwidth utility function has been proposed in [6], where the *utility max-min fairness* criterion is introduced to maximize the minimum utility perceived by any application sharing the constrained resources in the network. However, the *utility max-min fairness* criterion does not allow to maximize network revenue.

Further, the *max-min fairness* criterion has been examined in [7], and the alternative *proportional fairness* criterion is introduced to maximize the overall utility of rate allocations assuming each connection has a logarithmic utility function.

In [27], the authors have investigated the bandwidth sharing problem, proposing the *minimal potential delay* criterion. This criterion allows to distribute network bandwidth resources, among competing flows, minimizing the total potential delay. Moreover, it is shown in [27] that the *minimal potential delay* criterion is intermediate between the *max-min* and *proportional fairness*, penalizing long routes less severely than this latter.

Several works [7, 14, 15, 17, 28, 29, 30, 31, 32] have studied utility-based bandwidth allocation problems by exploiting the elasticity of network services. Most of these works use a utility and pricing framework that attempts to obtain the optimal bandwidth allocation that maximizes the total network revenue using the price as a control signal.

In [7, 14], the authors deal exclusively with concave utility functions for the solution of which there exist extensive theories and algorithms such as the Karush-Kuhn-Tucker conditions and duality theorem.

Kelly has suggested in [7] that the problem of bandwidth allocation should be posed as one of achieving maximum aggregate utility for the users. These users are assumed to be of elastic traffic, and can adjust their rates based on the feedback from the network. Further, in [7], pricing is used to decompose the overall problem into subproblems for the network and for the individual users.

Authors in [14] have also investigated the problem of achieving the optimal bandwidth allocation that maximizes the aggregate utility of the users, using only the information available at the end hosts. The users are assumed of elastic traffic, but differently from [7], in [14], the users adjust their rates based on their estimates of network congestion level.

On the other hand, the authors in [15] have addressed the problem of allocating transmission data rates distributedly to users who have concave as well as sigmoidal utility functions, to take into account the behavior of different applications. Moreover, it is demonstrated in [15] that applying rate control algorithms developed for concave utility functions in a more realistic setting can lead to instability and high network congestion.

In our service model, we do not restrict the utility functions to be concave; in line with [15], we allow users to have more general utility functions that arise naturally in the context of various applications. However, to solve the problem of extra-bandwidth allocation, we distinguish between two cases: 1) all users have concave utility functions; or 2) users have concave as well as non concave utility functions. In the first case, we propose a mathematical formulation of the bandwidth allocation problem that maximizes the total network revenue, while in the second case we introduce two heuristic online measurement-based bandwidth allocation algorithms.

In [2], a generic pricing structure is presented to characterize the pricing schemes currently used in the Internet, and a dynamic, congestion-sensitive pricing algorithm is introduced to provide an incentive for multimedia applications to adapt their transmission rates according to network conditions. Moreover, a service model is introduced in [2], where the network operator can create different tradeoffs between blocking admissions and raising congestion prices to motivate the rate and service adaptation of applications to the varying network conditions, technologies and platforms. Upon congestion, the network adjusts congestion price periodically on the time scale of a minute or longer, encouraging the adaptation-capable applications to adapt their transmission rates or select a different service class.

As in [2], we take into account users bandwidth utility functions to evaluate our proposed allocation algorithms based on the increased network revenue that is achieved. However, the authors in [2] consider a different service model than that proposed in our work and focus mainly on the issue of dynamic pricing to perform rate adaptation according to network conditions.

### 2.2.3 Dynamic Source Tracking Mechanisms

The idea of measuring dynamically the effective number of active connections as well as their sending rates is a well accepted technique [5, 8, 9] for improving the total network revenue.

A fast rate computation (FASTRAC) algorithm is introduced in [8], where it is shown that such algorithm differs from other schemes in achieving max-min fairness automatically without requiring complex per-connection measurements, constraint information, and complex control parameter set tuning. Moreover, in [8], the authors describe techniques for dynamically changing the effective number of connections in the rate computation. Their dynamic source tracking mechanism is used when the number of connections is large and it enables connections that are active to quickly grab unused bandwidth that is left over by connections that are silent.

In our bandwidth provisioning architecture, we also perform a dynamic tracking of active connections and their actual sending rate, among that offered to the network. This allows us to allocate the bandwidth left unused by idle and active connections to those that want to transmit more than their subscribed rate, thus increasing network revenue.

## 2.3 Dynamic Bandwidth Provisioning Architectures

Dynamic bandwidth provisioning in Quality of Service networks has recently attracted a lot of research attention due to its potential to achieve efficient resource utilization while providing the required quality of service to network users [1, 2, 3, 4, 5].

In [1], the authors introduce a dynamic provisioning architecture and a set of dynamic node and core provisioning algorithms for interior nodes and core networks, respectively.

The node provisioning algorithm prevents transient violations of service level agreements (SLA) by predicting the onset of service level violations based on a multiclass virtual queue measurement technique, and by adjusting the service weights of weighted fair queueing schedulers at core routers. Persistent service level violations are reported to the core provisioning algorithm, which dimensions traffic aggregates at the network ingress edge.

The core provisioning algorithm is designed to address the problem of provisioning DiffServ traffic aggregates by taking into account fairness issues across different traffic aggregates and also within the same aggregate whose packets take different routes through a core IP network. The core provisioning algorithm has two functions: to reduce edge bandwidth immediately after receiving a Congestion Alarm signal from a node provisioning module, and to provide periodic bandwidth realignment to establish a modified max-min bandwidth allocation for traffic aggregates.

In [3], the authors propose a set of dynamic provisioning algorithms that operate at the edge routers of a differentiated services core network. These edge mechanisms include: 1) ingress dynamic link sharing, which augments class based queueing techniques with bandwidth utility functions so that dynamic link sharing can be used to distribute bandwidth among traffic conditioners located at edge routers; and 2) egress dynamic capacity dimensioning, which formulates bandwidth dimensioning at egress links to peering/transit networks taking into account measured core network traffic conditions.

The work discussed in [1] has similar objectives to our research. However, the service model considered in [1, 3] differs from our proposed model and traffic statistics are not taken into account in the allocation procedure. Moreover, in our work we suggest a distributed architecture implementation, while in these works only a centralized scheme is considered.

A policy-based architecture is presented in [4], where a measurement-based approach is proposed for dynamic Quality of Service adaptation in DiffServ networks. The proposed architecture is composed of one Policy Decision Point (PDP), a set of Policy Enforcement Points that are installed in ingress routers and bandwidth monitors implemented in core routers. When monitors detect significant changes in available bandwidth they inform the PDP which changes dynamically the policies on in-profile and out-of-profile input traffics based on the current state of the network estimated using the information collected by the monitors. However, this scheme, while achieving dynamic QoS adaptation for multimedia applications, does not take into account the users utility function and their eventual willingness to be charged for transmitting out-of-profile traffic, thus increasing network revenue.

In [5], the authors propose an active resource management approach (ARM) for a differentiated services environment. The basic concept behind ARM is that by effectively knowing when a client is sending packets and how much of its allocated bandwidth is being used at any given time, the unused bandwidth can be reallocated without loss of service. This concept is in line with our research objectives. Differently from our work, however, ARM does not guarantee to the user a minimum subscribed bandwidth throughout the contract duration since unused bandwidth is sent to a pool of available bandwidth and it can be used to admit new connections in the network, in spite of those already admitted.

As we have mentioned previously, admission control mechanisms are complementary to resource provisioning algorithms for guaranteeing quality of service to network users.

Shenker in [10] states that the network service model can be augmented with admission control, which is the ability to turn some flows away when the network is overloaded. In other words, the network should turn away additional flows when admitting them would lead to violating its quantitative service commitments. Therefore, in our work, we assume the existence of admission control algorithms at the edge [20, 21, 22] of the network that cooperate with our proposed bandwidth allocation algorithms operating inside the network.

In the following, we present a brief review of two classes of admission control algorithms proposed in the literature.

In [20], the authors propose an endpoint admission control scheme which avoids the complexities of per-flow state in network routers. The basic idea is that the end hosts (or edge routers) probe the network by sending packets at the rate that a new flow would like to reserve. The flow is admitted if the resulting loss rate of probing packets is sufficiently low. Alternatively, the admission control test can be performed based on available bandwidth measurements, avoiding the negative effect of losses.

The scheme described in [21] is part of a more general admission control scheme that bases pricing on ECN (Explicit Congestion Notification) marks. All packets are treated identically: data and probe packets are indistinguishable, as are best-effort and real-time packets, and they are marked upon congestion. Flows can send as much traffic as they wish, but must pay for those packets that are marked. In this setting, admission control is a service offered by third-parties (or the network itself) to provide a guaranteed price for the flow (as opposed to guaranteeing the level of service, in traditional IntServ).

# Chapter 3

# Service Model and Dynamic Provisioning Architecture

A key challenge for the deployment of quality of service networks is the development of solutions that can dynamically track traffic statistics and allocate network resources efficiently, satisfying the QoS requirements of users while aiming at maximizing, at the same time, resource utilization and network revenue.

The contribution of this Chapter is as follows. We introduce a novel service model that provides quantitative per-flow bandwidth guarantees. Moreover, the network periodically individuates unused bandwidth and proposes short-term contracts where extra-bandwidth is allocated and guaranteed exclusively to users who are willing to pay for it to transmit at a rate higher than their subscribed rate. We implement our proposed service model considering a distributed provisioning architecture and a set of control messages that assure the communication between ingress and core routers. Note that a centralized implementation of our service model can be devised as well; the extension is straightforward and therefore is not discussed in this thesis.

The structure of this Chapter is the following: in Section 3.1, we introduce our proposed service model. In Section 3.2, we present a distributed provisioning architecture which implements such service model by performing the dynamic bandwidth allocation algorithms described in Chapter 5. Finally, in Section 3.3, we present the signalling messages used to assure the interaction between network elements.

## 3.1 Service Model

We propose a service model that, first, provides a quantitative per-flow bandwidth guarantee and then exploits the unused bandwidth individuated periodically in the network to propose short-term guaranteed extra-bandwidth to users who are willing to pay more to get a higher bandwidth.

In this process, different weights can be assigned to network users to allocate extra-bandwidth with different priorities; such weights can be set statically offline, based on the service contract proposed to the user, or can be adapted online based, for example, on the user bandwidth utility function.

Our proposed service model is therefore characterized by:

- A quantitative bandwidth guarantee, expressed through the specification of user's subscribed rate.

- Short-term guaranteed extra-bandwidth: the network is monitored online to individuate unused bandwidth that is allocated to users who can exploit it to transmit extra-traffic for the duration specified in the contract. The contract duration depends on the user request and can span on several consecutive update intervals.

- A weight that expresses how each user shares network resources with other users in the extra-bandwidth assignment procedure.

Throughout this thesis, we assume for simplicity that the duration of such contracts is minimal (i.e. equal to one bandwidth update interval); this allows us to derive upper bounds to the performance that can be achieved by our proposed architecture.

## 3.2 Architecture

To implement our service model we assume a distributed architecture constituted by core and edge routers, as shown in Figure 3.1.

Traffic monitors are installed on ingress and core routers to perform online measurements on the incoming traffic flows and network capacity utilization, respectively.

Traffic monitors on ingress routers can measure the transmission rate as well as the offered rate of each connection entering the network.

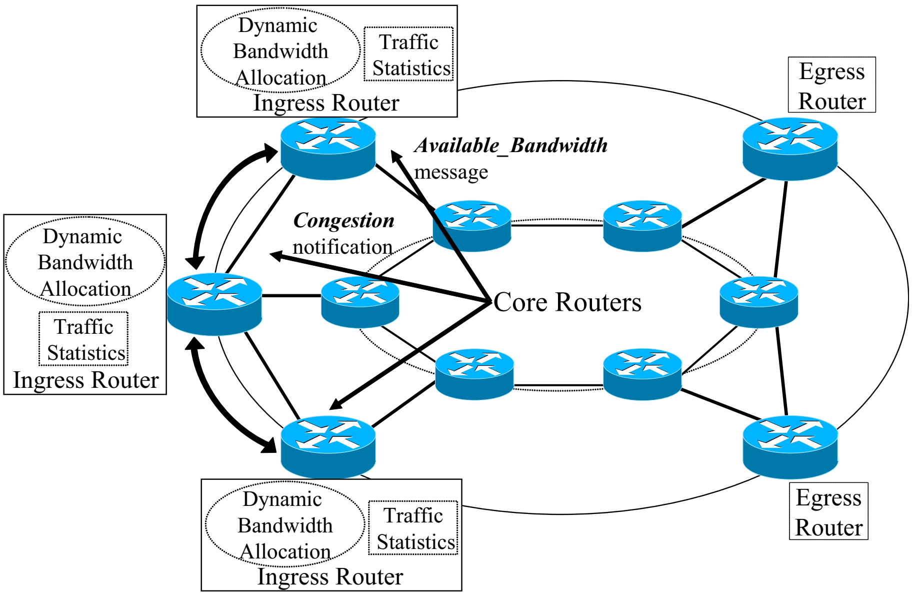



Figure 3.1: The proposed distributed architecture that supports dynamic bandwidth allocation

Core routers exchange messages with ingress routers to report links utilization or to notify a congestion situation. In this latter case, core routers alert the ingress routers to invoke immediately bandwidth re-allocation to solve this situation.

Each ingress router collects the measurements performed by traffic monitors and exchanges periodically update messages with all other ingress routers to report the current incoming traffic statistics.

Every update interval (i.e. $T_u$ seconds), all ingress routers execute the same algorithm considering different constraints like traffic statistics gathered online, network state information reported by core routers, and users' bandwidth utility functions to allocate network resources dynamically and efficiently.

Note that, as we consider update intervals on the order of tens of seconds, precise synchronization between ingress routers is not crucial. We discuss in more details the tradeoffs in the setting of the update interval in Chapter 7.

## 3.3 Control Messaging

The messages exchanged between network routers, illustrated with arrows in Figure 3.1, are similar to the control messages proposed in [1] to report persistent congestion or resource availability. Further, a subset of the messages defined in the RNAP protocol [16] can be used for these purposes.

*Available_Bandwidth* messages are sent by core routers to ingress routers when unused bandwidth is detected on some links. These messages can be exchanged either periodically or only when significant changes are perceived in the network. On the other hand, *Congestion notification* messages are sent immediately by core routers to ingress routers when some local thresholds are violated. In this case, the bandwidth allocation module is invoked directly by all ingress routers to re-allocate network resources accordingly resolving the congestion situation.

# Chapter 4

# Network Model

Dynamic bandwidth allocation in Quality of Service (QoS) networks is a challenging problem that has recently been the focus of many researchers. Maximization of total network revenue and satisfaction of users' QoS requirements are two important aspects to consider in resolving such problem.

This Chapter makes the following contributions. First, we describe the network model and we define the notation for the full comprehension of the problem. Then, we perform a connections' classification taking into account the online traffic statistics collected by ingress routers. This classification is fundamental to our heuristic algorithms, in that it allows to exploit the unused bandwidth left over by *idle* and *Non-greedy* connections to achieve high network revenue. Finally, we provide a definition of the network extra-revenue that is expressed in terms of users' bandwidth utility functions. This definition constitutes one of the two performance metrics used to evaluate the performance of our bandwidth allocation algorithms.

This Chapter is organized as follows. Section 4.1 presents a precise statement of the bandwidth allocation problem. Section 4.2 performs a connections' classification based on online measured traffic statistics. Finally, Section 4.3 introduces the network extra-revenue definition.

## 4.1 Problem Statement

Let us model the network as a directed graph $G = (N, L)$ where nodes represent routers and directed arcs represent links. Each link $l \in L$ has associated the capacity $C_l$. A set of $K$ connections is offered to the network. Each connection is represented by the notation $(s_k, d_k, sr_k, r\_min_k)$, for $k = 1, \ldots, K$, where $s_k$, $d_k$, $sr_k$ and $r\_min_k$ represent respectively the connections source node, destination node, subscribed rate and the minimum bandwidth the application requires to work correctly. Let $a_k^l$ be the routing matrix: $a_k^l = 1$ if connection $k$ is routed on link $l$, $a_k^l = 0$ otherwise. We assume that a communication between a user pair is established by creating a session involving a path that remains fixed throughout the user pair conversation duration. The session path choice method (i.e., the routing algorithm) is not considered in this thesis. We denote the interval between two successive allocations performed by the algorithm as the *update interval*, whose duration is $T_u$ seconds.

The goal is to determine $r_k^n$, the amount of bandwidth allocated to each source $k$ in the $n-th$ update interval.

At the beginning of the $n-th$ update interval, each ingress router computes the transmission rate, $b_k^{n-1}$, and the offered load, $Ol_k^{n-1}$, averaged over the last $T_u$ seconds, for all connections $k \in K$ that enter the network through it.

This information is then sent to all other ingress routers using control messages as described in the previous Section, so that all ingress routers share the same information about current traffic statistics and perform simultaneously the same allocation procedure.

Moreover, based on this information, connections are classified according to their traffic statistics. Such classification is taken into account in the bandwidth allocation process to distribute network resources and to achieve high network revenue.

## 4.2 Connections Classification

The $K$ connections offered to the network are first classified according to the criteria specified in the following.

Connections having $b_k^{n-1} < r\_min_k$ are considered *idle*;

All other active connections are further classified as *greedy* if they used a fraction greater than $\gamma$ of their subscribed rate $sr_k$ (i.e. if $b_k^{n-1} > \gamma \cdot sr_k$), otherwise they are classified as $non - greedy$.

Greedy connections are assigned extra-bandwidth by the network according to their traffic statistics and utility functions, thus contributing to increase network revenue, defined in the following.

We denote by $K_i$, $K_{ng}$ and $K_g$ the sets of idle, non-greedy and greedy connections, respectively.

Such classification constitutes a basic prerequisite for all the proposed dynamic bandwidth allocation algorithms.

## 4.3 Network Revenue Definition

We define, in line with [2], the average network revenue as the total charge paid to the network for all the extra-bandwidth utilization, averaged over all the bandwidth update intervals. In this computation we consider only network revenue generated by greedy users that are assigned extra-bandwidth by our proposed dynamic allocation algorithms. The revenue deriving from the static subscription of guaranteed rates ($sr_k$) is not considered since we focus on the extra network revenue that dynamic allocation can generate.

As already stated, we do not consider the pricing component of bandwidth utility functions, and we assume that network revenue can be measured as the extra-utility perceived by network users. Furthermore we assume, in line with [7], that the utilities are additive so that the aggregate utility of rate allocation is given by the sum of the utilities perceived by all network users.

Using the notation introduced in the previous Section, the average network extra-revenue can be obtained averaging over all the update intervals $i$ the quantity:

$$\sum_{k \in K_g} U_k(b_k^i) - U_k(sr_k) \tag{4.1}$$

where $sr_k$ and $b_k^i$ represent, respectively, the subscribed rate and the average transmission rate in the $i-th$ update interval for all connections $k \in K_g$, and $U_k(x)$ models the perceived utility of the $k-th$ user for an allocation of $x$ bandwidth units.

# Chapter 5

# Dynamic Bandwidth Allocation Algorithms

This Chapter provides three measurement-based bandwidth allocation algorithms that take into account the traffic statistics measured online, links utilization and users' utility functions to allocate network resources dynamically and efficiently. First, we introduce the Optimum Bandwidth Allocation (OBA) algorithm that allows to achieve the highest network revenue taking into account users' bandwidth utility functions. OBA assumes that the utility function is known or can be estimated for every connection. Then extending the max-min fair share algorithm introduced in [13], we propose two heuristic algorithms, with increasing performance and complexity. The first algorithm is simple and named Simple Dynamic Bandwidth Allocation (SDBA) algorithm, while the second algorithm iterates over the first taking into account sources' offered loads, avoiding to waste bandwidth and thus increasing network revenue. This latter is called Iterative Dynamic Bandwidth Allocation (IDBA) algorithm.

Initially, OBA, SDBA and IDBA need to perform the connections' classification described in Section 4.2 and then proceed in two main steps, where the first step is in common, while the second step differs for each algorithm. These two steps are described in the following Section.

The structure of this Chapter is as follows. Section 5.1 outlines the two main steps of the proposed algorithms and then describes the implementation of these latters. Section 5.2 discusses the convergence property of the algorithms.

## 5.1 Proposed Dynamic Bandwidth Allocation algorithms

All the proposed algorithms proceed in two main steps:

- In step one, bandwidth is allocated to all active connections trying to match their near-term traffic requirements that are predicted based on the statistics collected by ingress routers.
- In step two, the spare bandwidth as well as the bandwidth left unused by idle and active connections is individuated on each link. Such available extra-bandwidth is allocated with guarantee during each update interval exclusively to connections that can take advantage of it since they are already fully exploiting their subscribed rate.

### Step 1

Using the definitions and notation introduced previously we perform the following assignments:

- *Idle* connections are assigned their minimum required transmission rate, i.e. $r_k^n = r\_min_k, \forall k \in K_i$.
- *Non-greedy* connections are assigned a bandwidth that can accommodate traffic growth in the current update interval while, at the same time, save unused bandwidth that can be re-allocated to other users. Several techniques have been proposed

in the literature to predict the near-term transmission rate of a connection based on past traffic measurements; however, the focus of this thesis is not to find out the best traffic predictor.

In this work we only rely on the last measured value, $b_k^{n-1}$, and we propose the following simple bandwidth allocation: $r_k^n = \min\{2 \cdot b_k^{n-1}, sr_k\}, \forall k \in K_{ng}$. In this regard we are currently studying more efficient traffic predictors that can allow improved bandwidth allocation. We study the impact of this choice in Chapter 8, through the performance of the mathematical model.

- *Greedy* connections are assigned in this step their subscribed rate $sr_k$, and they further take part to the allocation of extra-bandwidth performed in Step 2, since they are already exploiting all their subscribed rate.

### Step 2

After having performed the allocations described in Step 1, each algorithm individuates on each link $l$ the residual bandwidth $R_l$, i.e. the spare bandwidth as well as the bandwidth left unused by *idle* and *non-greedy* connections. $R_l$ is hence given by the following expression:

$$R_l = C_l - (\sum_{k \in K_i \cup K_{ng}} r_k^n \cdot a_k^l + \sum_{k \in K_g} sr_k \cdot a_k^l ), \forall l \in L \tag{5.1}$$

where the first summation represents the total bandwidth allocated in Step 1 to idle and non-greedy connections, while the second summation represents the bandwidth allocated to greedy connections. Such extra-bandwidth is distributed exclusively to greedy users who can exploit it to transmit at a rate higher than their subscribed rate.

To this aim, we propose in the following three heuristics for the bandwidth allocation problem and in the next Chapter, we further illustrate a mathematical model which provides an upper bound to the network revenue that can be obtained by any dynamic bandwidth allocation scheme.

**Optimum Bandwidth Allocation Algorithm**

If we assume that the utility function $U_k(x)$ is known or can be estimated for every greedy source $k$, then we can illustrate the *Optimum Bandwidth Allocation* (OBA) problem formulation. The decision variable $f_k^n$, $k \in K_g$, represents the amount of extra-bandwidth that is allocated to each greedy connection during the $n-th$ update interval. We maximize the total network extra-revenue considering the following mathematical model:

$$Maximize \sum_{k \in K_g} U_k(sr_k + f_k^n) - U_k(sr_k) \tag{5.2}$$

$$s.t. \sum_{k \in K_g} f_k^n \cdot a_k^l \leq R_l, \forall l \in L \tag{5.3}$$

$$sr_k + f_k^n \leq Ol_k^{n-1}, \forall k \in K_g \tag{5.4}$$

$$f_k^n \geq 0, \forall k \in K_g \tag{5.5}$$

The objective function (5.2) is the total network extra-revenue.

Constraint (5.3) represents capacity constraints expressed for each link of the graph.

Constraint (5.4) imposes that, for every update interval $n$, the total load allocated to each greedy source $k$ does not exceed the total load offered to the network by $k$, thus avoiding to waste extra-bandwidth.

If sources offered load is unknown or difficult to estimate, we can consider an alternate formulation to the OBA problem by simply dropping constraint (5.4).

If all users utility functions are differentiable and strictly concave, then the objective function (5.2) is differentiable and strictly concave. Since the feasible region (5.3), (5.4) and (5.5) is compact, a maximizing value of $f_k^n$ exists and can be found using Lagrangian methods. Further, if the objective function can be approximated using piece-wise linear concave functions, the problem can be solved using standard linear programming (LP) techniques. Finally, when all these assumptions do not hold, the OBA model can not be applied. For this reason we further develop two other heuristic algorithms, detailed in the following, that do not rely on any assumption concerning the users utility functions.

### Simple Dynamic Bandwidth Allocation Algorithm

When neither the utility functions nor the offered loads can be easily measured for all sources, we propose the Simple Dynamic Bandwidth Allocation algorithm (SDBA) detailed in Table 5.1, which is an extended version of the max-min fair allocation algorithm introduced in [13].

The simple rationale behind SDBA is the following: the extra-bandwidth is allocated equally to all greedy sources bottlenecked at the same link, starting from the most congested bottleneck link. SDBA takes as input the set $K_g$ of greedy connections, the link set $L$ with the residual capacity on each link $l$, $R_l$, and the routing matrix $a_k^l$, and produces as output the amount of extra-bandwidth $f_k^n, k \in K_g$ that is assigned to each greedy connection during the $n-th$ update interval, so that finally $r_k^n = sr_k + f_k^n, \forall k \in K_g$.

### Iterative Dynamic Bandwidth Allocation Algorithm

Note that, since SDBA does not take into account sources' offered load, it can allocate to a source more bandwidth than it can use, thus wasting it. For this reason, when utility functions are unknown while sources' offered load can be measured online at ingress routers, we propose the Iterative Dynamic Bandwidth Allocation (IDBA) algorithm detailed in Table 5.2.

IDBA takes as input the offered load $Ol_k^{n-1}, k \in K_g$, besides all the inputs considered in SDBA, to compute the amount of extra-bandwidth $f_k^n, k \in K_g$ that is assigned to each greedy connection during the $n-th$ update interval.

For every update interval $n$, all greedy connections, $k \in K_g$, are marked as *Non-served*

Table 5.1: Pseudo-code specification of the Simple Dynamic Bandwidth Allocation algorithm (SDBA)

| | Algorithm SDBA($K_g$, $R_l$, $a_k^l$) |
|---|---|
| (1) | initiate all $f_k^n = 0,\ \forall\ k \in K_g$ |
| (2) | remove from the link set $L$ all links $l \in L$ that have a number of connections crossing them $n_l$ equal to 0 |
| (3) | for every link $l \in L$, calculate $F_l = R_l/n_l$ |
| (4) | identify the link $\alpha$ that minimizes $F_\alpha$ i.e. $\alpha \mid F_\alpha = min_k(F_k)$ |
| (5) | set $f_k^n = F_\alpha,\ \forall\ k \in K_\alpha$, where $K_\alpha \subseteq K_g$ is the set of greedy connections that cross link $\alpha$ |
| (6) | for every link $l$, update the residual capacity and the number of crossing greedy connections as follows: $R_l = R_l - \sum_{k \in K_\alpha} f_k^n \cdot a_k^l$ <br> $n_l = n_l - \sum_{k \in K_\alpha} a_k^l$ |
| (7) | remove from set $L$ link $\alpha$ and those that have $n_l = 0$ |
| (8) | if $L$ is empty, then stop; else go to Step (3) |

at the beginning of the first iteration of IDBA. For every successive iteration, a greedy connection can be either marked as *Served*, when its total assigned rate $(sr_k + f_k^n)$ is at least equal to its offered load, or *Non-served* otherwise. Let $K_g^{NS}$ and $K_g^S$ denote the set of greedy *Non-served* and *Served* connections respectively.

IDBA iterates over the two following steps until either all sources become *Served* or the extra-bandwidth on the bottleneck links is completely allocated:

- First, the extra-bandwidth individuated in the network is allocated fairly among the *Non-served* greedy connections using SDBA($K_g^{NS}$, $R_l$, $a_k^l$).

- Second, for every *Non-served* greedy connection, $k \in K_g^{NS}$ if $sr_k + f_k^n > Ol_k^{n-1}$ then $r_k^n$ is set equal to $Ol_k^{n-1}$ and the connection is classified as *Served*; the difference, $sr_k + f_k^n - Ol_k^{n-1}$, is redistributed in the next iteration to the remaining *Non-served* greedy connections.

Table 5.2: Pseudo-code specification of the Iterative Dynamic Bandwidth Allocation algorithm (IDBA)

| | Algorithm IDBA($K_g$, $Ol_k^{n-1}$, $R_l$, $a_k^l$) |
|---|---|
| (1) | initiate $K_g^{NS} = K_g$ |
| (2) | run $SDBA(K_g^{NS}, R_l, a_k^l)$, then set $f_k^n,\ k \in K_g^{NS}$ |
| (3) | set $N_{g,b}^{NS} = 0$, |
| | where $N_{g,b}^{NS}$ represents the number of bottlenecked connections |
| | among the set of greedy *Non-served* connections, $K_g^{NS}$ |
| (4) | for every connection $k \in K_g^{NS}$, |
| |     $if(sr_k + f_k^n < Ol_k^{n-1})$ |
| |         increase $N_{g,b}^{NS}$ by 1 |
| |     $end\ if$ |
| | end for |
| (5) | $if(N_{g,b}^{NS} = \vert K_g^{NS}\vert)$ |
| |     Mark all connections as served; hence $K_g^{NS} = \emptyset$ |
| | $end\ if$ |
| (6) | $if(N_{g,b}^{NS} < \vert K_g^{NS}\vert)$ |
| |     for every connection $k \in K_g^{NS}$, |
| |         $if(sr_k + f_k^n \geq Ol_k^{n-1})$ |
| |             $r_k^n = Ol_k^{n-1}$ |
| |             $f_k^n = r_k^n - sr_k$ |
| |             mark $k$ as served; hence $K_g^{NS} = K_g^{NS} - \{k\}$ |
| |                 for every link $l$, update the residual capacity and |
| |                 the number of crossing greedy connections as follows: |
| |                     $R_l = R_l - \sum_k f_k^n \cdot a_k^l$ |
| |                     $n_l = n_l - \sum_k a_k^l$ |
| |                 end for |
| |         $end\ if$ |
| |     end for |
| | $end\ if$ |
| (7) | $if(K_g^{NS} = \emptyset)$, then stop; |
| | *else* go to Step (2) |
| | $end\ if$ |

To take into account users weights in the SDBA and IDBA algorithms, it is sufficient to substitute $n_l$ in Tables 5.1 and 5.2 with $w_l$, which is defined as the sum of the weights of all greedy connections that are routed on link $l$.

It should be clarified that our proposed algorithms can temporarily present some limitations in bandwidth allocation, since the bandwidth allocated to a user can at most double from an update interval to the successive one. This could affect the performance of users that experience steep increases in their transmission rate. In Chapter 7, we evaluate numerically this effect showing at the same time how it is counterbalanced by increased network revenue in all the considered network scenarios under several traffic load conditions.

## 5.2 Convergence Property

The analysis of our algorithms' convergence property follows closely from [33]. It is shown in [33] that a distributed algorithm needs at least M iterations to stabilize toward max-min allocation in a descending order starting from the most congested bottleneck link, where M is the number of distinct bottleneck links in the network.

In practice, the convergence upper-bound can be improved by network engineering. One approach is to use a centralized implementation of the bandwidth allocation algorithm. This effectively removes the M factor from consideration.

Since SDBA is an extended version of the max-min fair share algorithm introduced in [13], the convergence speed of SDBA depends on the set of bottleneck links and how connections are routed in the network sharing such bottleneck links. Various traffic sources can send traffic over the same congested links, a situation that arises frequently in communication networks. In the extreme case, when all the sources have portions of traffic over all the congested links, these sources are only constrained by the most congested bottleneck link. In this case, SDBA takes one round to finish, and the allocation is done with respect to the capacity of the most congested bottleneck link.

As we have previously described, IDBA iterates over SDBA taking into account sources' offered load to avoid wasting bandwidth, then the convergence speed of IDBA is slightly slower than that of SDBA.

# Chapter 6

# Mathematical Model

This Chapter introduces a novel and flexible mathematical model, Ideal Bandwidth Allocation (IBA), that maximizes the total network revenue and provides bounds to the performance achievable by any online dynamic bandwidth allocation algorithm.

Such model assumes the exact knowledge of the future traffic offered to the network. The bandwidth allocations are performed optimizing the operation of the network loaded with the actual present and future traffic. No practical bandwidth allocation scheme can perform better.

This mathematical model extends the optimization problem related to utility maximization introduced in [7] and its solution can be obtained using Lagrangian methods. Further, if the objective function can be approximated using piece-wise linear concave functions, the problem can be solved using standard linear programming (LP) techniques.

IBA allows us to gauge the impact of the traffic predictor used to assign bandwidth to *non-greedy* connections (Section 5.1), as well as of the granularity in bandwidth assignment (i.e. the sensitivity to the parameter $T_u$) on the performance of our proposed dynamic bandwidth allocation algorithms.

The structure of this Chapter is as follows. In Section 6.1, we first illustrate the network model, then we define some notations needed for our mathematical model. In Section 6.2, we describe in details the Ideal Bandwidth Allocation (IBA) algorithm that needs the knowledge of arrival time, duration and offered load of all the connections offered to the network.

## 6.1 Problem Statement

Hereafter, we recall the network model described in Section 4.1.

Let us model the network as a directed graph $G = (N, L)$ where nodes represent routers and directed arcs represent links. Each link $l \in L$ has associated the capacity $C_l$. A set of $K$ connections is offered to the network. Each connection is represented by the notation $(s_k, d_k, sr_k, r\_min_k)$, for $k = 1, \ldots, K$, where $s_k$, $d_k$, $sr_k$ and $r\_min_k$ represent respectively the connections source node, destination node, subscribed rate and the minimum bandwidth the application requires to work correctly. Let $a_k^l$ be the routing matrix: $a_k^l = 1$ if connection $k$ is routed on link $l$, $a_k^l = 0$ otherwise. We assume that a communication between a user pair is established by creating a session involving a path that remains fixed throughout the user pair conversation duration. The session path choice method (i.e., the routing algorithm) is not considered in this thesis.

Each connection $k$ is further characterized by its arrival time, $t_k$ and its duration $\tau_k$. Given $K$ connections (Figure 6.1 shows an example for $K = 4$), the time interval from the arrival of the first connection and the ending time of the last connection is subdivided in a set $I$ of $2K - 1$ time intervals. In each time interval, $t$, the number of active connections remains constant. This number changes by one from interval to interval: it increases if a new connection arrives or becomes active, and decreases if a connection ends or becomes inactive. We assume that the traffic offered by connection $k$ in each time interval $t$, $Ol_k^t$, is known.

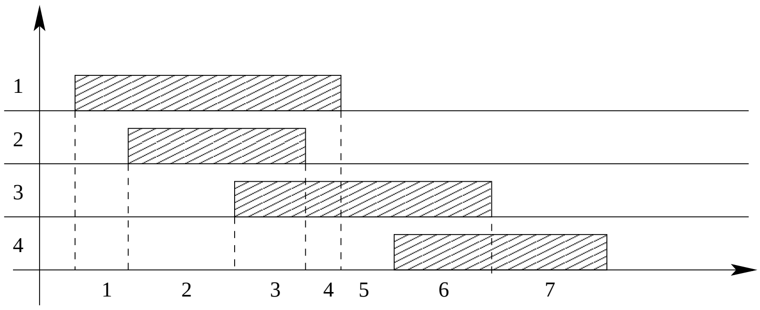



Figure 6.1: Arrival time and duration of the connections offered to the network

## 6.2 Ideal Bandwidth Allocation (IBA)

In each time interval $t$, all connections that have $Ol_k^t < sr_k$ are classified, in accordance with Section 4.2, as *non-greedy*, and they are assigned their offered load $Ol_k^t$. All other connections are classified as *greedy*; let $K_g^t$ be the set of connections that are *greedy* in time interval $t$. Let $R_l^t$ be the residual bandwidth available on each link $l$ in time interval $t$ after having allocated bandwidth to all *non-greedy* connections.

Based on the above definitions and notations, we establish the mathematical formulation of the IBA model. The decision variable $r_k^t$, $k \in K_g^t$, represents the amount of bandwidth that is allocated to each *greedy* connection during the time interval $t$. We maximize the total network extra-revenue considering the following mathematical model:

$$Maximize \sum_{t \in I} \sum_{k \in K_g^t} U_k(r_k^t) - U_k(sr_k) \tag{6.1}$$

$$s.t. \;\; sr_k \le r_k^t \le Ol_k^t, \forall k \in K_g, t \in I \tag{6.2}$$

$$\sum_{k \in K_g^t} r_k^t \cdot a_k^l \le R_l^t, \forall l \in L, t \in I \tag{6.3}$$

The objective function (6.1) is the total network extra-revenue.

Constraint (6.2) imposes that, for every time interval $t$, the total load allocated to each *greedy* source $k$ does not exceed the total load offered to the network by $k$, thus avoiding to waste extra-bandwidth.

Finally, constraint (6.3) represents capacity constraints expressed for each link of the graph.

# Chapter 7

# Numerical Results

The main goal of our bandwidth allocation algorithms is to offer services that satisfy the QoS requirements of individual users while guaranteeing at the same time an efficient utilization of network resources.

In this Chapter we compare the performance of the proposed dynamic bandwidth allocation algorithms illustrated in Chapter 5 with a static provisioning strategy, referring to different network scenarios to cover a wide range of possible environments. The simulation tool we used was J-Sim simulator version 1.3 [34]. Furthermore, to solve the utility maximization problem in OBA and IBA, we use AMPL (A Modeling Language for Mathematical Programming) [35].

Note that our proposed service model is novel and different from all those referred in Chapter 2. Therefore, since it is impossible to compare our approach to the dynamic provisioning schemes presented in the literature, in this Chapter we performed a comparison with the static provisioning technique, which consists in assigning to each source $k$ its subscribed rate $sr_k$.

## 7.1 Performance Metrics

We consider the following performance metrics: the average accepted load and network extra-revenue.

The average accepted load is obtained averaging the total load accepted in the network over all the bandwidth update intervals, while the average network extra-revenue has been defined in Section 4.3. We recall, hereafter, the definition of network extra-revenue:

$$\sum_{k \in K_g} U_k(b_k^n) - U_k(sr_k)$$

where $sr_k$ and $b_k^n$ represent, respectively, the subscribed rate and the average transmission rate in the $n-th$ update interval for all greedy connections $k \in K_g$, and $U_k(x)$ models the perceived utility of the $k-th$ user for an allocation of $x$ bandwidth units.

## 7.2 Network Scenarios

We refer to four network scenarios ranging from a single-bottleneck topology to a complex core network topology to evaluate the performance of our proposed bandwidth allocation algorithms. All numerical results have been calculated over long-lived data exchanges, achieving very narrow 95% confidence intervals [36].

### 7.2.1 Single-Bottleneck Scenario

In this scenario we gauge the effectiveness of the proposed traffic-based bandwidth allocation algorithms. We consider, in line with [1, 2], the scenario illustrated in Figure 7.1, that consists of a single-bottleneck with 2 core nodes, 6 access nodes, 40 end nodes (20 source-destination pairs) and traffic conditioners at the edge. Each ingress conditioner is configured with one profile for each traffic source, and drops out-of-profile packets. All links are full-duplex and have a propagation delay of 1 ms. The capacity of the link connecting the two core nodes is equal to 6 Mb/s, that of the links connecting the access nodes to core nodes is equal to 10 Mb/s, and that of the links connecting the end nodes to access nodes is 2 Mb/s. The buffer size of each link can contain 50 packets.

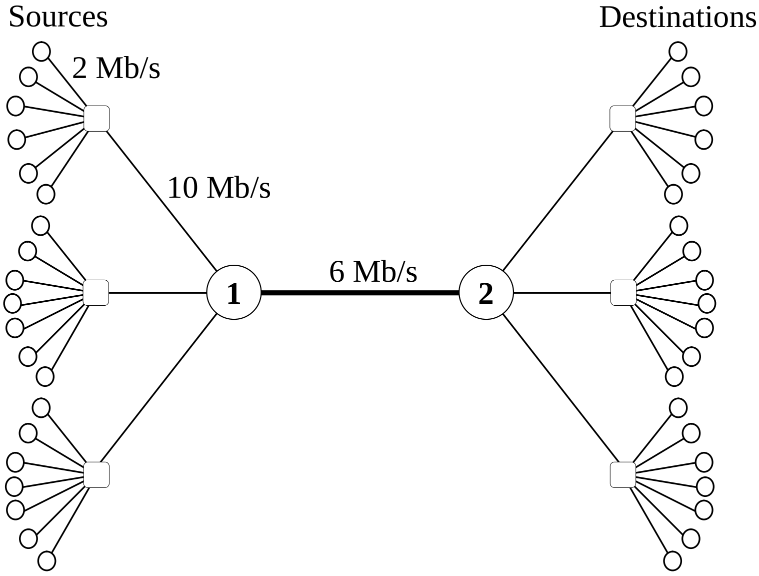



Figure 7.1: Network topology with a single bottleneck

We use 20 Exponential On-Off traffic sources; the average On time is set to 200 s, and the average Off time is varied in the 0 to 150 s range to simulate different traffic load conditions while varying at the same time the percentage of bandwidth left unused by every connection. During On times each source transmits with a constant rate that we refer to hereafter as the source's peak rate.

Six sources have a peak rate of 50 kb/s and a subscribed rate of 150 kb/s, 8 sources have a peak rate of 250 kb/s and a subscribed rate of 200 kb/s, while the remaining six sources have a peak rate of 1 Mb/s and a subscribed rate of 500 kb/s; the minimum bandwidth required by each source, $r\_min_k$, is equal to 10 kb/s. The algorithm updating interval, $T_u$, is set to 20 s and $\gamma$ is set to 0.9.

We assume, for simplicity, that all users have the same weight $w_k$ and the same type of utility function proposed in [14], $U_k(x) = a \cdot log(b + x)$, where $a \geq 0$ and $0 \leq b \leq 1$.

In this scenario, we assume that all users have the same utility function considering $a = 0.5$ and $b = 1$ (see Figure 7.2).

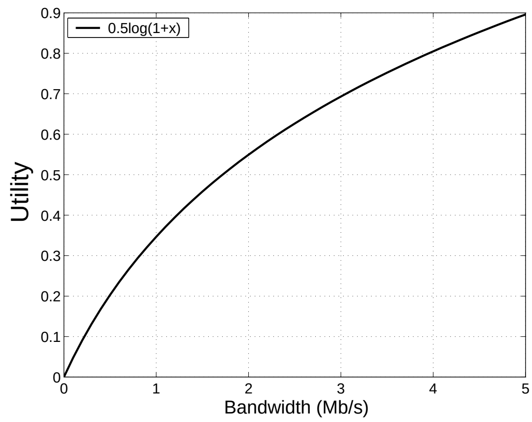



Figure 7.2: Utility of an elastic application as a function of bandwidth

Note that a realistic characterization of network applications is outside the scope of this thesis. The specification of the utility function allows us exclusively to gauge the extra network revenue that can derive from the deployment of our proposed bandwidth allocation algorithms.

Figures 7.3 and 7.4 show, respectively, the average total load accepted in the network and the corresponding total extra-revenue as a function of the average total load offered to the network.

It can be observed that the best performance is achieved by the Optimum Bandwidth Allocation, OBA, and Iterative Dynamic Bandwidth Allocation, IDBA. This is mainly due to the fact that we considered the same utility function for all users, as we will show in the following. SDBA achieves lower performance, but it still outperforms static bandwidth allocation both in terms of total accepted load and network revenue.

The maximum network extra-revenue is achieved when the average Off time of Exponential sources is equal to 80 s, corresponding to an offered load approximately equal to

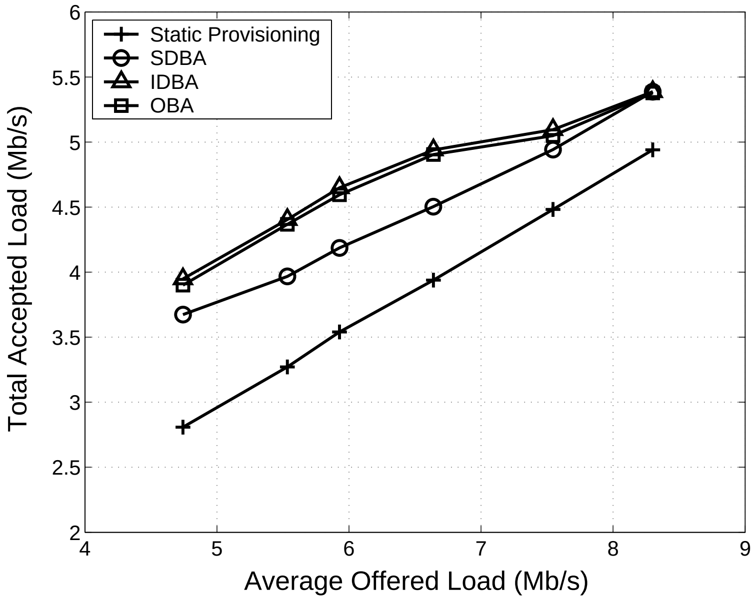



Figure 7.3: Average total accepted load versus the average load offered to the network of Figure 7.1 (all sources have the same utility function)

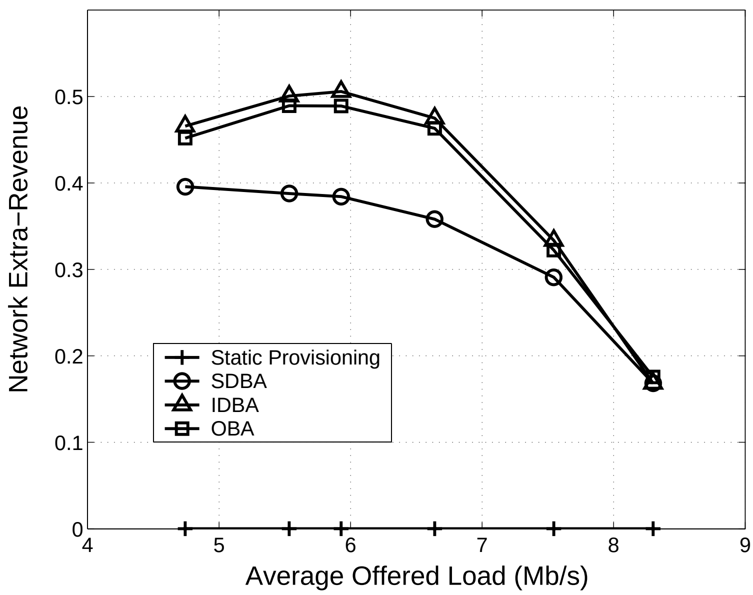



Figure 7.4: Average network extra-revenue versus the average load offered to the network of Figure 7.1 (all sources have the same utility function)

6 Mb/s. In this situation, in fact, the average number of idle connections (i.e. 6) is sufficiently high to exalt our dynamic allocation algorithms that reallocate unused bandwidth to active users who can take advantage of it, sending extra-traffic and generating network extra-revenue. With lower Off time values (i.e. with higher offered loads) the total revenue slightly decreases as less connections are idle, in average, and consequently less bandwidth is available for re-allocation.

To investigate the impact on the performance of the update interval duration, we have considered, in the single-bottleneck scenario, different values for $T_u$, viz. 40 s and 60 s. Figures 7.5 and 7.6 show, respectively, the average total load accepted in the network and the corresponding total extra-revenue as a function of the total load offered to the network for $T_u = 40$ s. Furthermore, Figures 7.7 and 7.8 show the same performance metrics for $T_u = 60$ s.

The average increase in the total accepted load, expressed as a percentage of the traffic admitted in the static allocation case, is of 14% for $T_u = 40$ s and 11% for $T_u = 60$ s, while for $T_u = 20$ s it was 24% (see Figure 7.3). These results allow to gauge the trade-off between performance improvement and overhead resulting from a more frequent execution of the allocation algorithms.

We observe that an update interval value of 20 s leads to higher network revenue than other values. This is consistently observed across the whole range of offered loads below 90% of the maximum offered load. When the offered load increases beyond 90% of the maximum offered load, the system becomes overloaded and the impact of the update interval value becomes negligible. Note that the best performance is achieved with $T_u = 20$ s; therefore, in the following, we choose an update interval of $20s$ in all network scenarios.

In the same scenario of Figure 7.1 we then considered different utility functions for some sources. More specifically, the 8 connections having $sr_k$ equal to 200 kb/s have associated the utility function equal to $0.5 \cdot log(1 + x)$ while the 6 connections having $sr_k$ equal to 500 kb/s have associated the utility function of $1.5 \cdot log(1 + x)$. These utility functions are

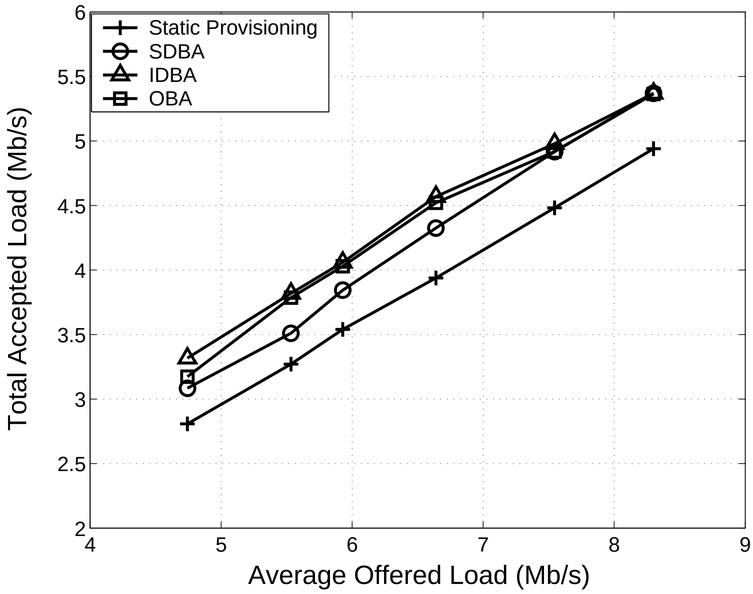



Figure 7.5: Average total accepted load as a function of the average load offered to the network of Figure 7.1 ($T_u = 40s$)

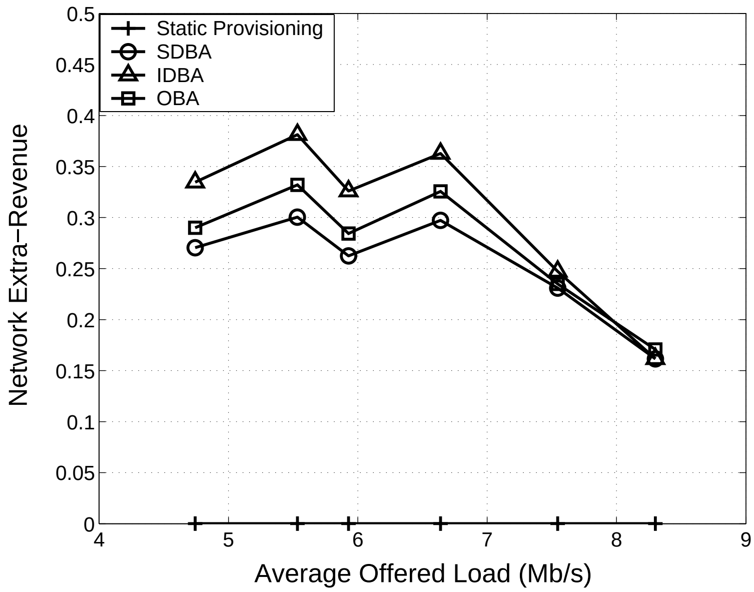



Figure 7.6: Average network extra-revenue as a function of the average load offered to the network of Figure 7.1 ($T_u = 40s$)

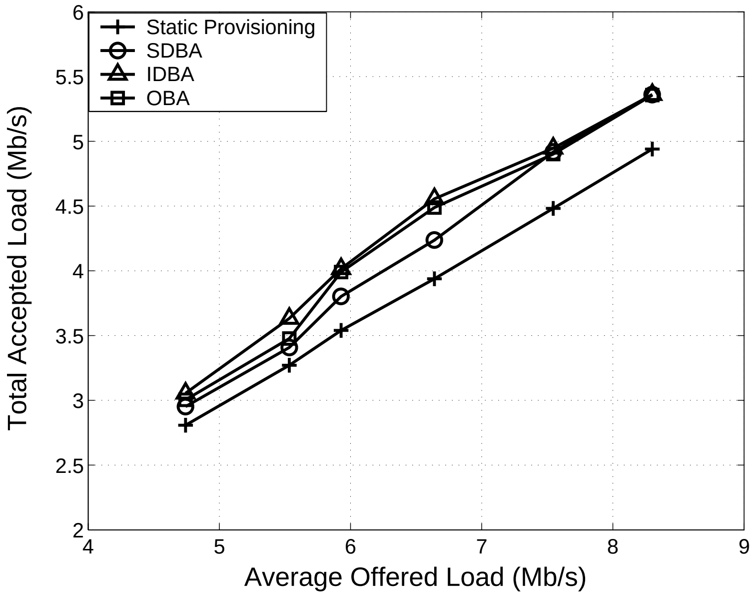



Figure 7.7: Average total accepted load as a function of the average load offered to the network of Figure 7.1 ($T_u = 60s$)

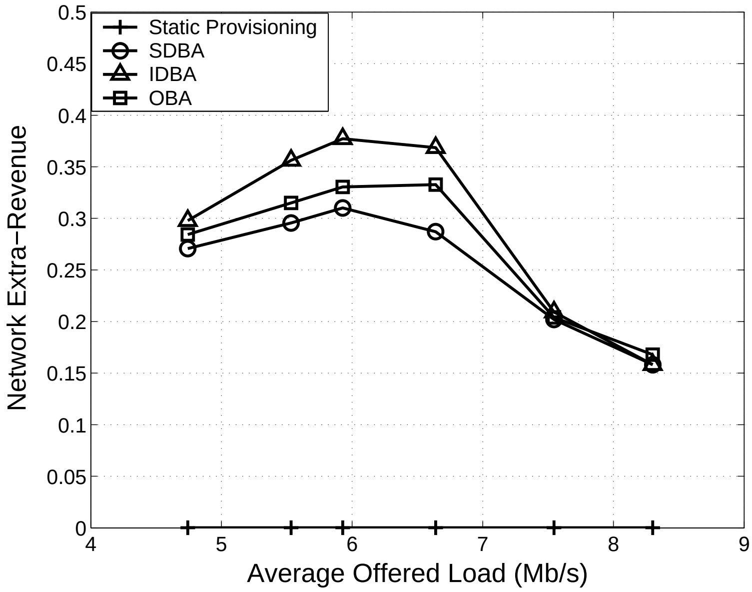



Figure 7.8: Average network extra-revenue as a function of the average load offered to the network of Figure 7.1 ($T_u = 60s$)

plotted in Figure 7.9.

Figures 7.10 and 7.11 show the total accepted load and network extra-revenue achieved by the various allocation algorithms. In this scenario, OBA achieves the better performance in terms of network revenue. This is expected since it distributes network extra-bandwidth taking into account users' utility functions, differently from IDBA, SDBA and static provisioning.

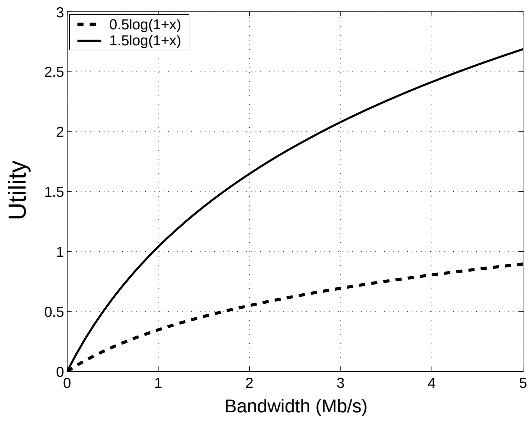



Figure 7.9: Utility of an elastic application as a function of bandwidth

Finally, in the same scenario of Figure 7.1 we fixed the average Off time of Exponential sources to 100 s while maintaining the average On time equal to 200 s, and we varied the peak rate of all sources scaling them by a factor $\alpha$, with $0.25 \leq \alpha \leq 1.5$. We considered the same utility functions as in the previous scenario. Figures 7.12 and 7.13 show, respectively, the average total accepted load and the average total network extra-revenue in this scenario.

At very low load the static provisioning technique achieves slightly higher performance than dynamic allocation algorithms. This is due to the fact that in this situation static provisioning is in effect sufficient to accommodate all incoming traffic; on the other hand, dynamic provisioning algorithms need some time (in the worst case up to $T_u$ seconds) to track the transition of sources from the idle to the active state. For all other traffic loads the advantage of the proposed dynamic bandwidth allocation algorithms with respect to

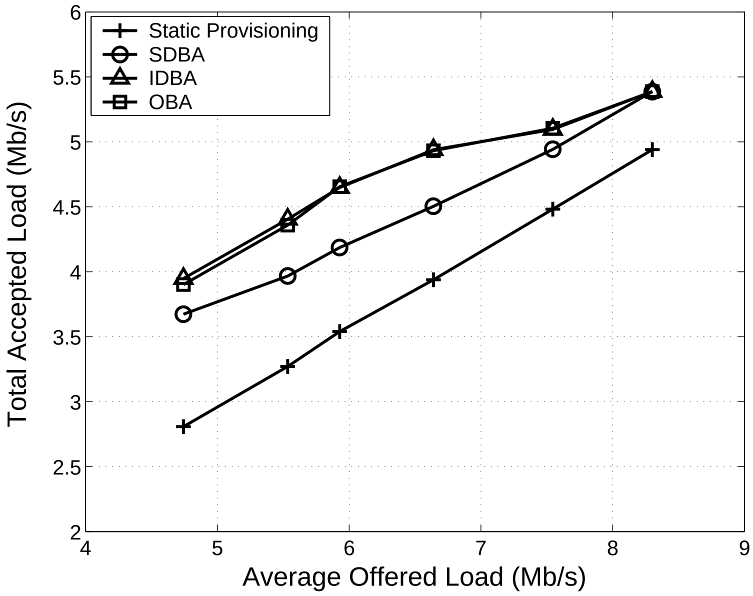



Figure 7.10: Average total accepted load, sources have different utility functions: $U_i(x) = 0.5 \cdot log(1+x)$ and $U_j(x) = 1.5 \cdot log(1+x)$

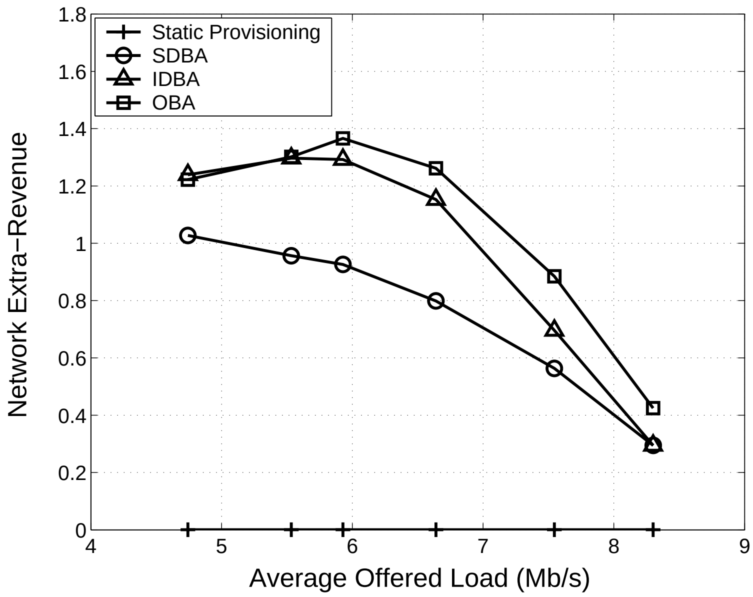



Figure 7.11: Average network extra-revenue, sources have different utility functions: $U_i(x) = 0.5 \cdot log(1+x)$ and $U_j(x) = 1.5 \cdot log(1+x)$

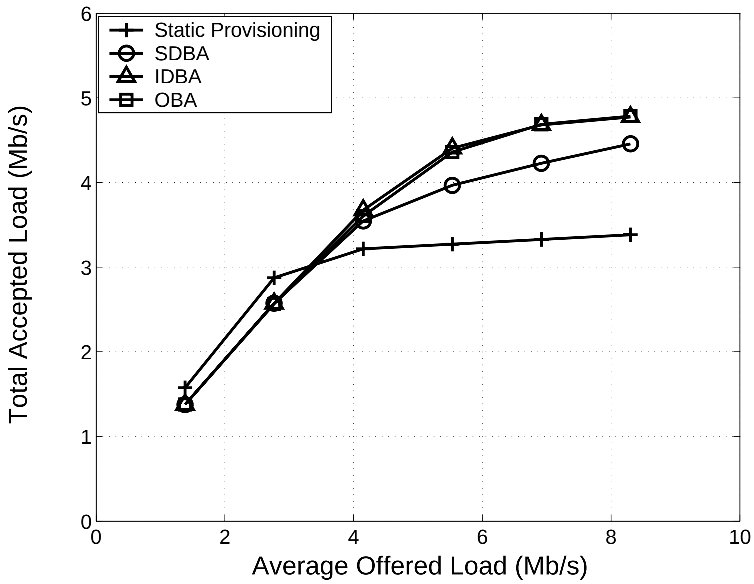



Figure 7.12: Average total accepted load using dynamic bandwidth allocation versus the average offered load in the Single-bottleneck topology of Figure 7.1

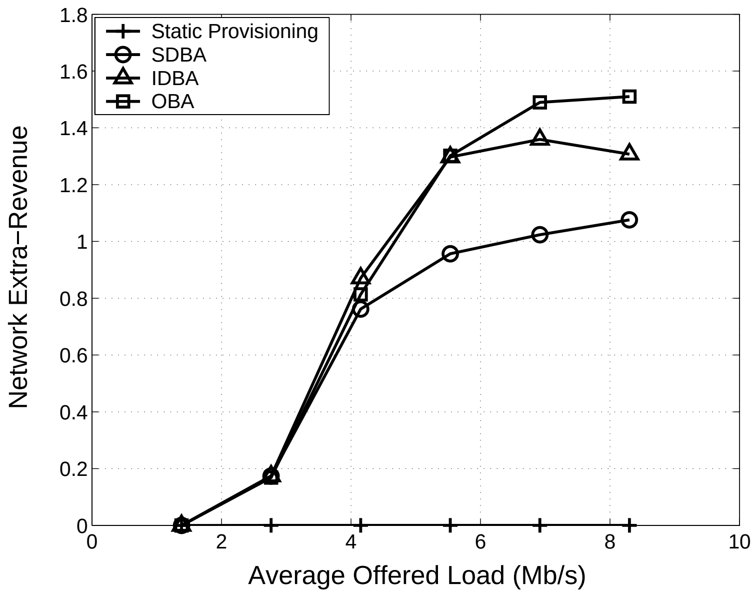



Figure 7.13: Average total network extra-revenue using dynamic bandwidth allocation versus the average offered load in the Single-bottleneck topology of Figure 7.1

static allocation is evident both in terms of accepted load and network revenue. More specifically, OBA achieves the best performance in terms of network revenue especially for high offered loads; note that IDBA achieves almost the same performance as SDBA for medium offered loads.

### 7.2.2 Simple Core Network Scenario

A more realistic scenario is shown in Figure 7.14. It comprises 6 nodes and 8 bidirectional links, all having a capacity equal to 2 Mb/s and propagation delay of 1 ms. In this topology, 6 Exponential On-Off traffic sources are considered, and their source and destination nodes are indicated in the Figure. Table 7.1 reports the peak rate, the subscribed rate, the utility function and the path for all the connections. Sources S1, S2 and S3 are associated the utility function $0.5 \cdot log(1+x)$ while the remaining sources are associated the utility function $1.5 \cdot log(1 + x)$. All other parameters are set as in the previous scenarios. Note that, with such paths choice, various connections compete for network capacity with different connections on different links.

In this scenario, the performance of IDBA almost overlaps that of OBA. This is expected because the number of connections entering the network (i.e. 6) is quite small to gauge the potential of OBA. Further, all the proposed algorithms (SDBA, IDBA and OBA) outperform static allocation, as shown in Figures 7.15 and 7.16, thus proving the benefit of the proposed schemes. These results verify that our allocation algorithms allow service providers to increase network capacity utilization and consequently network extra-revenue with respect to static provisioning techniques.

### 7.2.3 Multi-Bottleneck Scenario

We then consider the network topology shown in Figure 7.17, originally proposed in [1]. It comprises 8 core nodes and 7 bidirectional links, all having the same propagation delay,

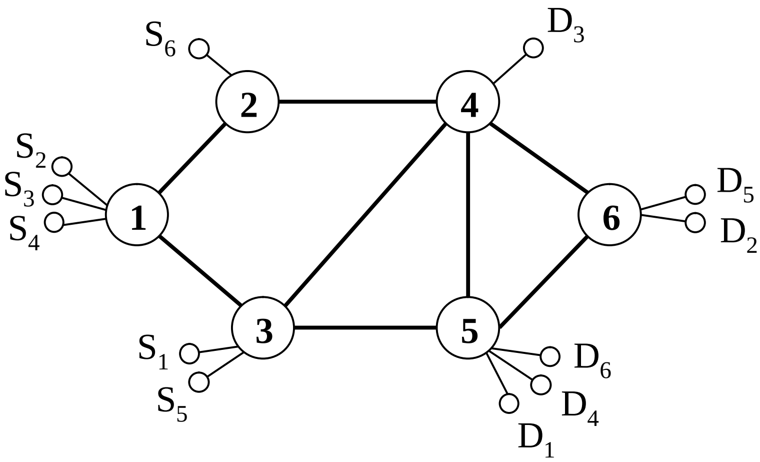



Figure 7.14: Network topology with a larger number of links

Table 7.1: Peak rate, subscribed rate, utility function and path for the connections in the network scenario of Figure 7.14

| Connection | Peak Rate (kb/s) | Subscribed Rate (kb/s) | Utility Function | Path |
|---|---|---|---|---|
| 1 | 100 | 300 | 0.5log(1+x) | 3-4-5 |
| 2 | 500 | 400 | 0.5log(1+x) | 1-2-4-6 |
| 3 | 500 | 400 | 0.5log(1+x) | 1-3-4 |
| 4 | 1000 | 400 | 1.5log(1+x) | 1-3-5 |
| 5 | 1000 | 400 | 1.5log(1+x) | 3-4-6 |
| 6 | 1000 | 400 | 1.5log(1+x) | 2-4-5 |

equal to 1 ms. The capacities are given next to the links in the Figure. The three links highlighted in the Figure represent three bottlenecks in this network topology.

Twelve Exponential On-Off traffic sources are considered, and their source and destination nodes are indicated in the Figure. Table 7.2 reports the peak rate, the subscribed rate, the utility function and the path for all the connections. All other parameters are set as in the previous scenarios. Also in this scenario, various connections compete for network capacity with different connections on different links.

Even though OBA and IDBA achieve the same total accepted load, it can be observed

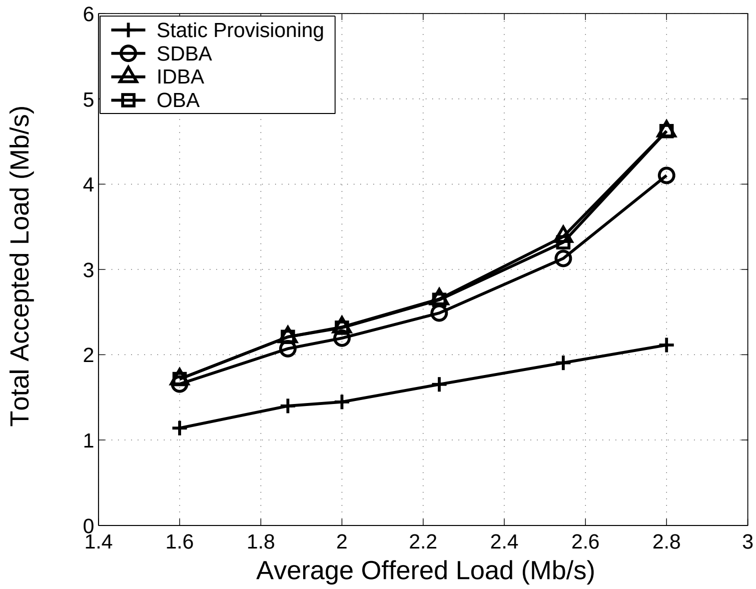



Figure 7.15: Average total accepted load versus the average load offered to the network of Figure 7.14

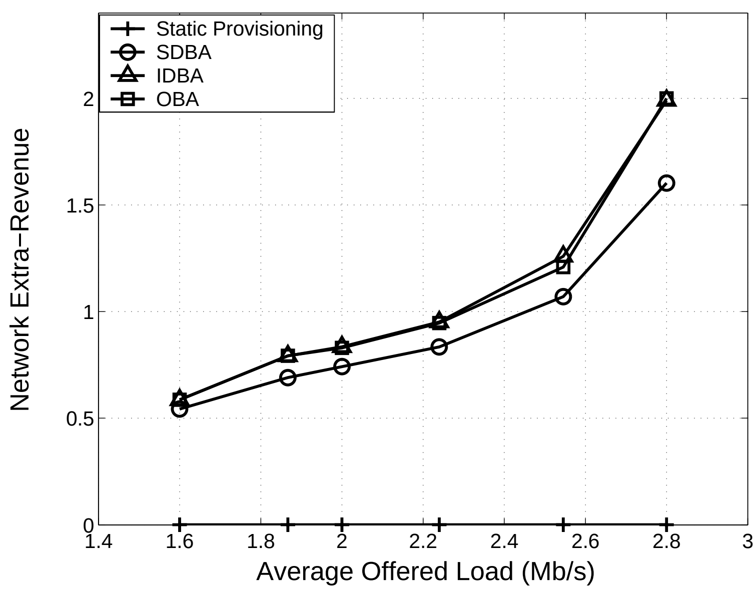



Figure 7.16: Average network extra-revenue versus the average load offered to the network of Figure 7.14

in Figures 7.18 and 7.19 that OBA outperforms IDBA in network revenue, especially for high network loads, where it achieves almost 18% higher revenue than IDBA.

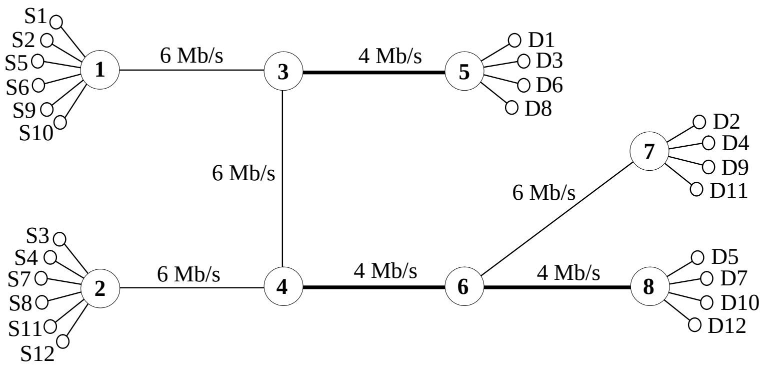



Figure 7.17: Network topology with multiple bottleneck links

Table 7.2: Peak rate, subscribed rate, utility function and path for the connections in the network scenario of Figure 7.17

| Connection | Peak Rate (kb/s) | Subscribed Rate (kb/s) | Utility Function | Path |
|---|---|---|---|---|
| 1 | 40 | 100 | 0.5log(1+x) | 1-3-5 |
| 2 | 40 | 100 | 0.5log(1+x) | 1-3-4-6-7 |
| 3 | 40 | 100 | 0.5log(1+x) | 2-4-3-5 |
| 4 | 40 | 100 | 0.5log(1+x) | 2-4-6-7 |
| 5 | 500 | 300 | 0.5log(1+x) | 1-3-4-6-8 |
| 6 | 500 | 300 | 0.5log(1+x) | 1-3-5 |
| 7 | 500 | 300 | 0.5log(1+x) | 2-4-6-8 |
| 8 | 500 | 300 | 0.5log(1+x) | 2-4-3-5 |
| 9 | 1000 | 300 | 1.5log(1+x) | 1-3-4-6-7 |
| 10 | 1000 | 300 | 1.5log(1+x) | 1-3-4-6-8 |
| 11 | 1000 | 300 | 1.5log(1+x) | 2-4-6-7 |
| 12 | 1000 | 300 | 1.5log(1+x) | 2-4-6-8 |

### 7.2.4 Complex Core Network Scenario

Finally, we considered the network topology proposed in [14] and illustrated in Figure 7.20. This network scenario is more complex than the previous ones and it allows to test our

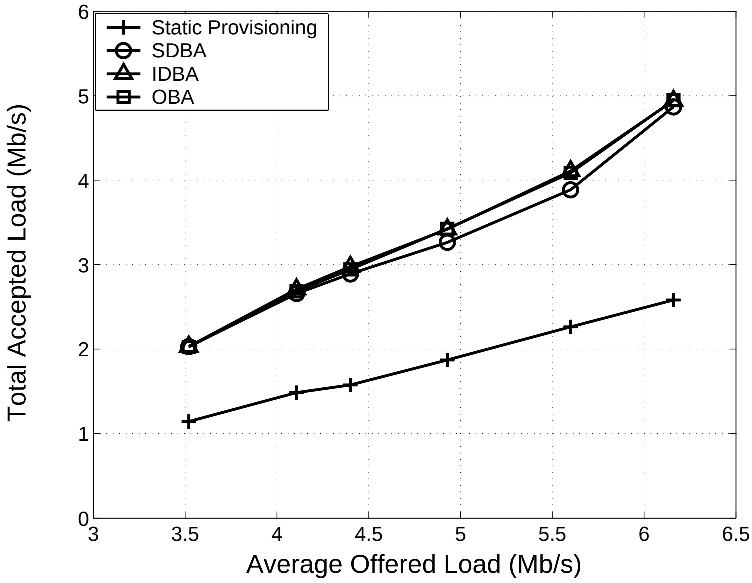



Figure 7.18: Average total accepted load as a function of the average load offered to the network of Figure 7.17

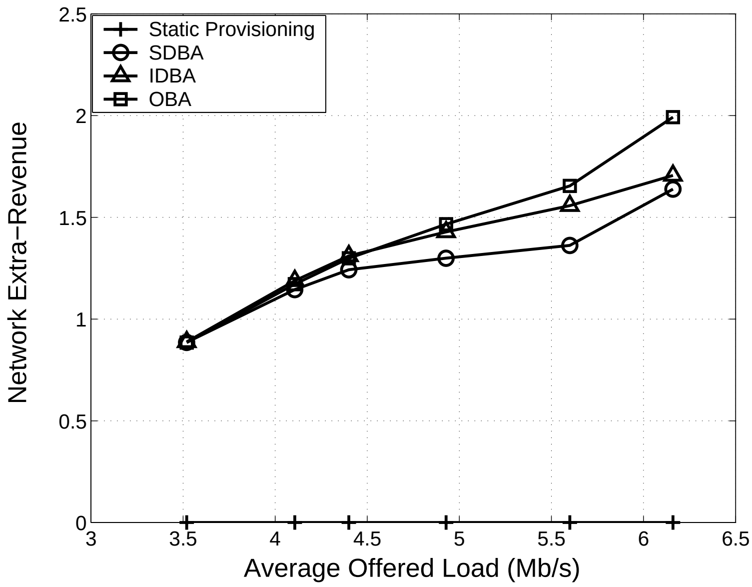



Figure 7.19: Average network extra-revenue as a function of the average load offered to the network of Figure 7.17

proposed allocation algorithms in a more realistic core network topology. It comprises 11 core nodes, 8 access nodes, 36 end nodes (18 source-destination pairs) and 28 bidirectional links, all having the same propagation delay, equal to 5 ms. The capacities are given next to the links in the Figure. Eighteen Exponential On-Off connections share the network. Table 7.3 reports for each connection its peak and subscribed rate, its utility function, as well as the path of the connection, which are the same as in [14].

Figures 7.21 and 7.22 show the performance of the considered bandwidth allocation algorithms as a function of the total load offered to the network. The results are in line with those achieved with the previous network scenarios and show that our proposed allocation algorithms allow to increase both total accepted traffic and network revenue with respect to a static allocation technique. Finally note that in this scenario the performance of IDBA almost overlaps that of OBA. This is expected since in this particular network topology, with the settings considered, few connections share capacity on each link and the utilization of enhanced allocation algorithms does not increase consistently network performance.

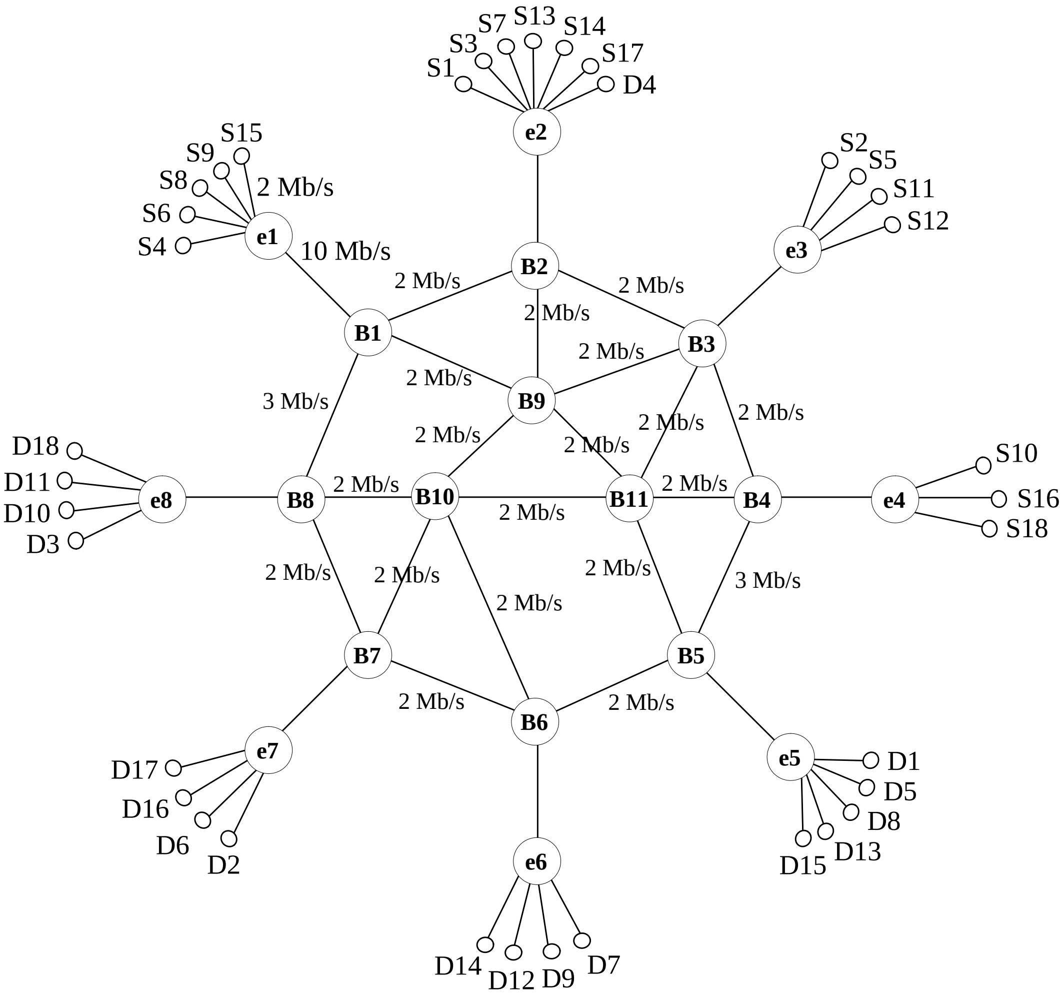



Figure 7.20: Complex core network topology

Table 7.3: Peak rate, subscribed rate, utility function and path for the connections in the network scenario of Figure 7.20

| Connection | Peak Rate (kb/s) | Subscribed Rate (kb/s) | Utility Function | Path |
|---|---|---|---|---|
| 1 | 100 | 300 | 0.5log(1+x) | e2-B2-B3-B4-B5-e5 |
| 2 | 100 | 300 | 0.5log(1+x) | e3-B3-B9-B10-B7-e7 |
| 3 | 100 | 300 | 0.5log(1+x) | e2-B2-B1-B8-e8 |
| 4 | 100 | 300 | 0.5log(1+x) | e1-B1-B2-e2 |
| 5 | 100 | 300 | 0.5log(1+x) | e3-B3-B4-B5-e5 |
| 6 | 100 | 300 | 0.5log(1+x) | e1-B1-B8-B7-e7 |
| 7 | 500 | 400 | 0.5log(1+x) | e2-B2-B9-B10-B6-e6 |
| 8 | 500 | 400 | 0.5log(1+x) | e1-B1-B9-B11-B5-e5 |
| 9 | 500 | 400 | 0.5log(1+x) | e1-B1-B8-B7-B6-e6 |
| 10 | 500 | 400 | 0.5log(1+x) | e4-B4-B11-B10-B8-e8 |
| 11 | 500 | 400 | 0.5log(1+x) | e3-B3-B2-B1-B8-e8 |
| 12 | 500 | 400 | 0.5log(1+x) | e3-B3-B4-B5-B6-e6 |
| 13 | 1000 | 500 | 1.5log(1+x) | e2-B2-B3-B4-B5-e5 |
| 14 | 1000 | 500 | 1.5log(1+x) | e2-B2-B9-B10-B6-e6 |
| 15 | 1000 | 500 | 1.5log(1+x) | e1-B1-B9-B11-B5-e5 |
| 16 | 1000 | 500 | 1.5log(1+x) | e4-B4-B5-B6-B7-e7 |
| 17 | 1000 | 500 | 1.5log(1+x) | e2-B2-B1-B8-B7-e7 |
| 18 | 1000 | 500 | 1.5log(1+x) | e4-B4-B11-B10-B8-e8 |

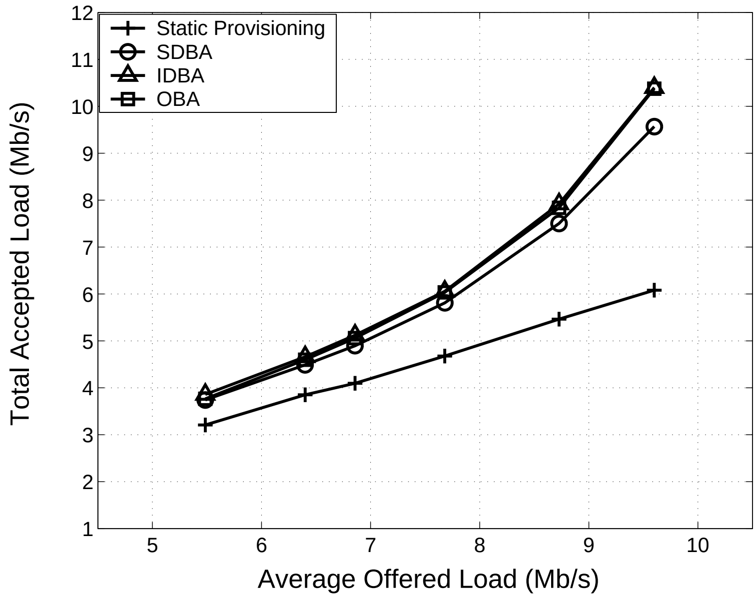



Figure 7.21: Average total accepted load in the complex core network of Figure 7.20 versus the average offered load

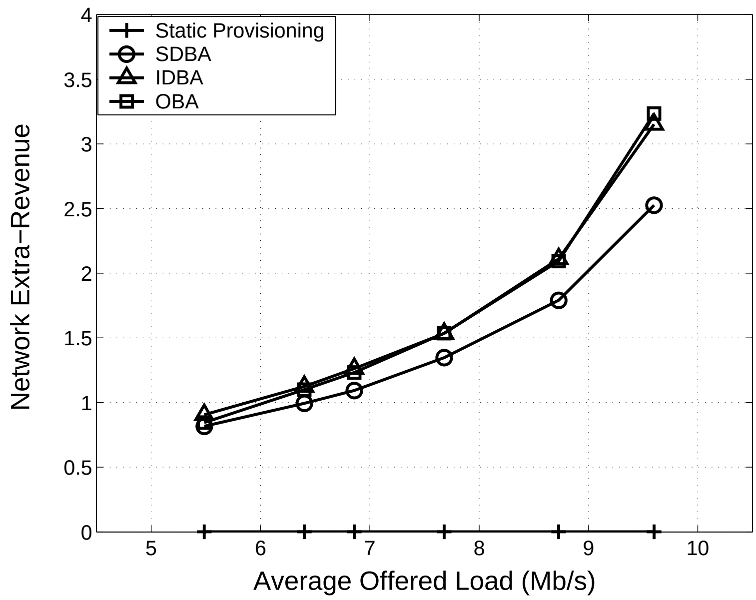



Figure 7.22: Average network extra-revenue in the complex core network of Figure 7.20 versus the average offered load

# Chapter 8

# Performance Upper Bound

The Ideal Bandwidth Allocation model (IBA) allows to optimize the total network revenue and provides bounds to the performance achievable by any online dynamic bandwidth allocation algorithm. Such model requires the knowledge of the future traffic offered to the network, the arrival time, the duration and the transmission rate of each connection offered to the network.

We implemented and solved the IBA model considering the same network scenarios and simulation tools described in the previous Chapter.

This Chapter provides a discussion on the numerical results obtained with IBA. These results show that the proposed dynamic bandwidth allocation algorithms approach, in several scenarios, the ideal performance provided by IBA.

## 8.1 Simulation results

We first considered the single-bottleneck topology illustrated in Figure 8.1(a). All traffic sources and parameters are the same as used in the previous Chapter for all network scenarios. Figures 8.1(b) and 8.1(c) show, respectively, the average total load accepted in the network and the corresponding total extra-revenue as a function of the load offered to the network. It can be observed that the best performance is achieved by the IBA model both in terms of total accepted load and network revenue. The average increase in the total accepted load and network extra-revenue, expressed as a percentage of those obtained with OBA, are of 17% and 31%, respectively. We can also observe that SDBA, IDBA, OBA and IBA exhibit the same decreasing trend. The performance of OBA well approaches the upper bounds provided by IBA for all values of the offered traffic.

We then considered the simple core network topology presented in Figure 8.2(a). Also in this scenario, IBA performs better than SDBA, IDBA and OBA, as shown in Figures 8.2(b) and 8.2(c). Numerical results show that IBA achieves in average 24% (respectively, 34%) higher total accepted load (respectively, network extra-revenue) than OBA. In this scenario, the performance of OBA approaches that of IBA; in particular, when the network is overloaded, the OBA performance overlaps that of IBA both in terms of total accepted load and network revenue.

A more generic scenario is presented in Figure 8.3(a). In Figures 8.3(b) and 8.3(c), we compare the performance of our proposed algorithms with the mathematical model as a function of the total load offered to the network. It can be observed that IBA provides an increase of 24% in terms of total accepted load and 32% in terms of network extra-revenue, compared to OBA.

Finally, we considered the network topology illustrated in Figure 8.4(a). We observe that IBA achieves higher total accepted load and network extra-revenue than OBA, as shown in Figures 8.4(b) and 8.4(c). Numerical results show that, the average increase in the total accepted load and network extra-revenue of IBA, with respect to those achieved by OBA, are of 27% and 44%, respectively.

Table 8.1 summarizes the average performance gain achieved by IBA with respect to OBA.

Table 8.1: Average performance gain achieved by IBA (Ideal Bandwidth Allocation) with respect to OBA (Optimum Bandwidth Allocation)

| Network Scenarios | Average increase in accepted load | Average increase in network revenue |
|---|---|---|
| Single-Bottleneck | 17% | 31% |
| Simple Core Network | 24% | 34% |
| Multi-Bottleneck | 24% | 32% |
| Complex Core Network | 27% | 44% |

## 8.2 Discussion

Note that, when Exponential sources are always active (the right points in all the graphs), the performance of OBA almost overlaps that of IBA, especially in the last three network scenarios. When the Off time of Exponential sources is greater than zero, the difference between IBA and OBA is more evident.

Two main factors impact on the performance of our proposed heuristic bandwidth allocation algorithms: traffic prediction and bandwidth allocation granularity. Let us recall that in the bandwidth allocation process *Non-greedy* sources are allocated the quantity $\min\{2 \cdot b_k^{n-1}, sr_k\}$, thus leading to a potential bandwidth wastage. On the other hand, since bandwidth allocation is performed only every $T_u$ seconds, traffic variations can be tracked only with such granularity, leading again to potential inefficiency in bandwidth allocation.

The behavior observed in these scenarios shows that the impact of traffic prediction is less remarkable with respect to that of the update interval $T_u$. In effect, when all sources are always active, no traffic changes occur, and $T_u$ has no impact on the performance of the allocation algorithms. Moreover, since in this case the gap between IBA and the heuristic algorithms is negligible, the bandwidth wasted due to the traffic predictor has little impact on the algorithms performance.

Since in these scenarios traffic prediction has even lower impact when sources are not always active, we can conclude that, practically, all the gap between IBA and heuristic algorithms is due to the bandwidth allocation granularity for every network load value.

On the other hand, in the single-bottleneck scenario, the number of sources that share the bottleneck link is more consistent than in the other network scenarios. Therefore, the impact of traffic prediction is not anymore negligible. This can be observed in Figures 8.1(b) and 8.1(c), where the gap between IBA and the heuristic algorithms is present for all offered loads.

Finally, we note that, also in this scenario, the performance of the proposed dynamic bandwidth allocation algorithms (in particular, OBA) well approaches that of IBA, especially for high network loads.

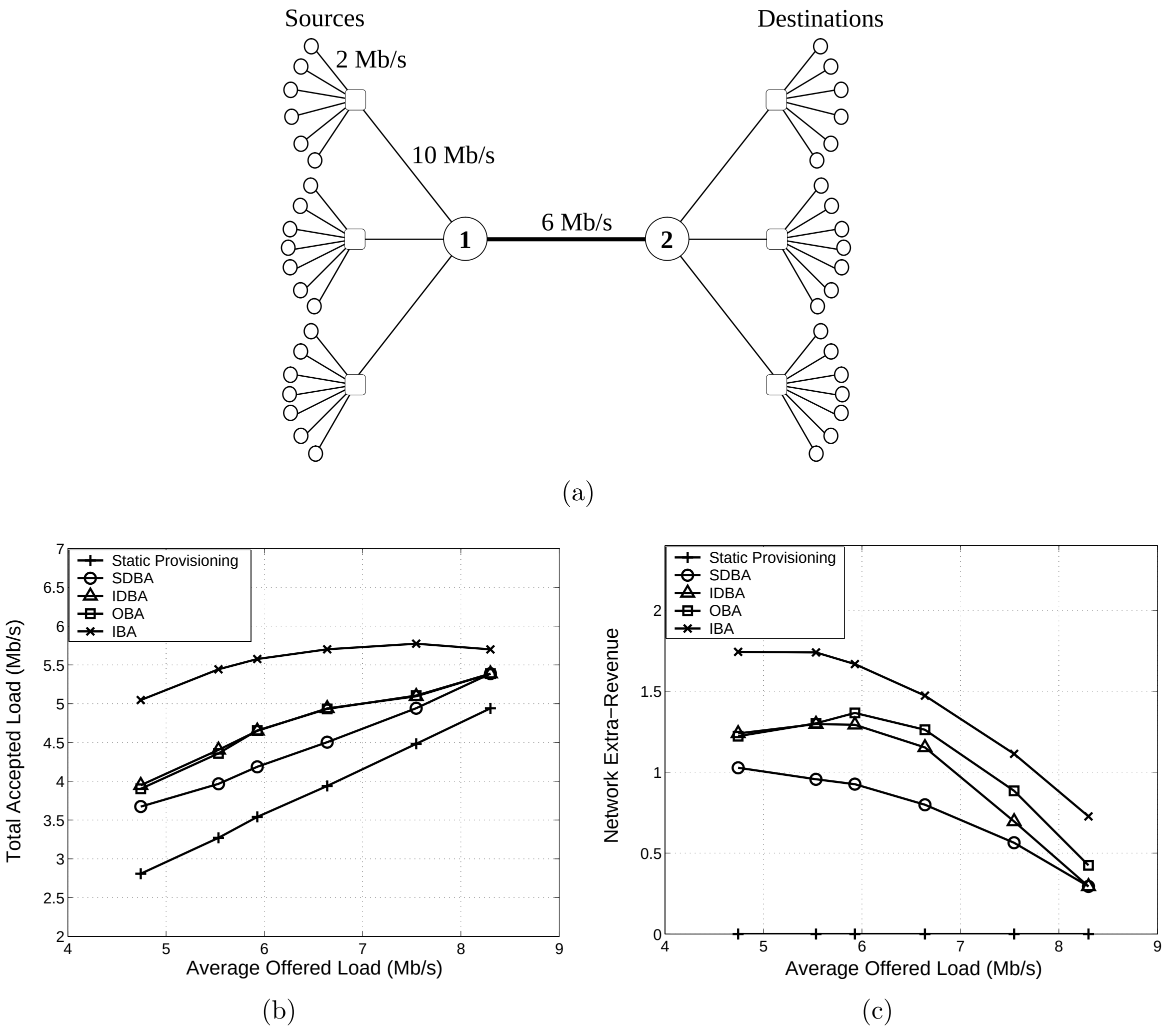



Figure 8.1: Performance upper bound for all the proposed bandwidth allocation algorithms, network topology with a single bottleneck (a). Average total accepted load (b) and network extra-revenue (c) versus the average load offered to the network.

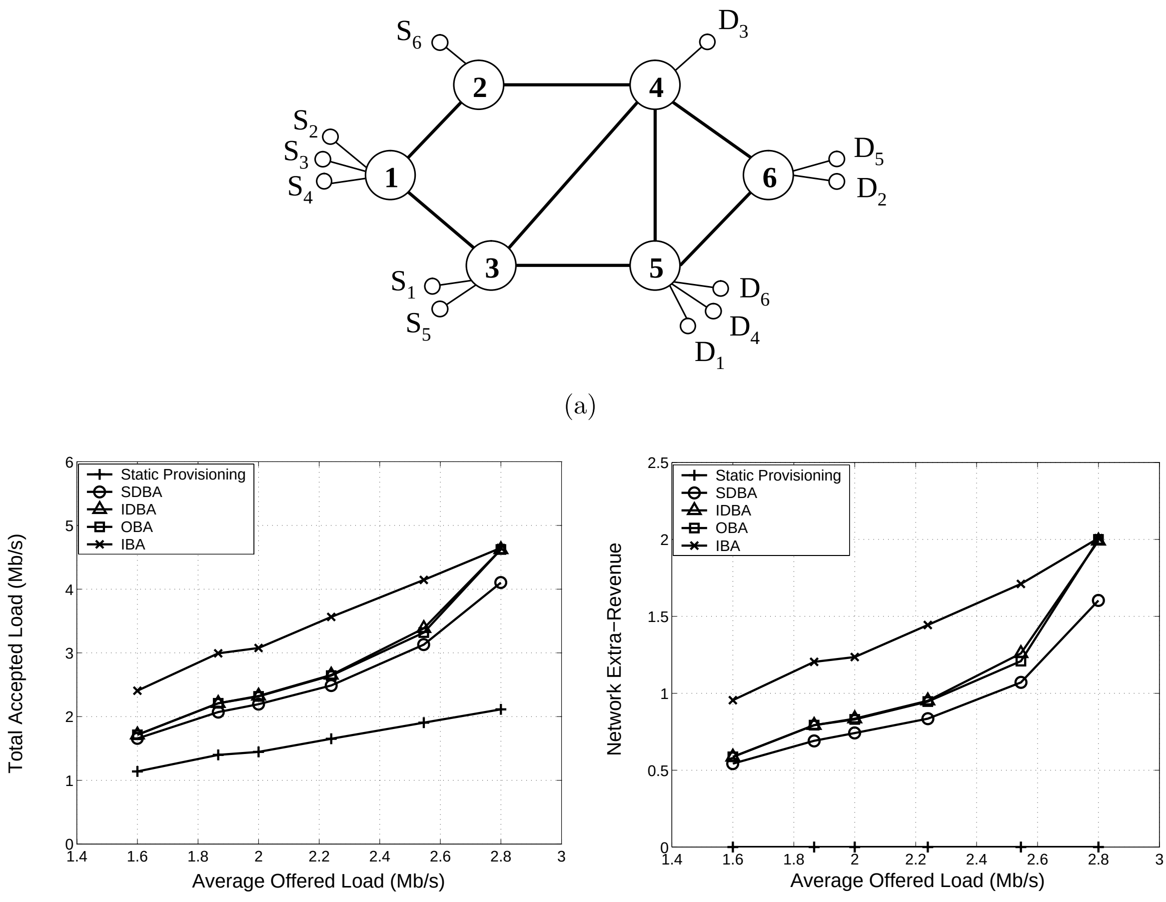
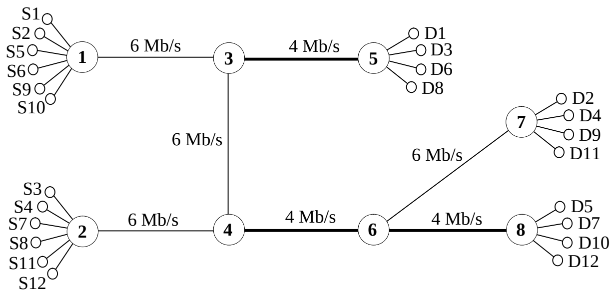
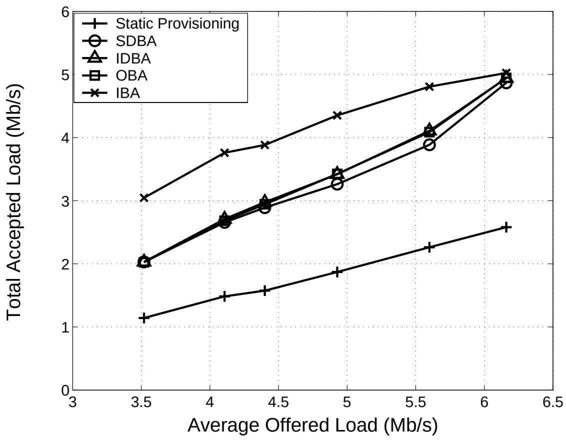


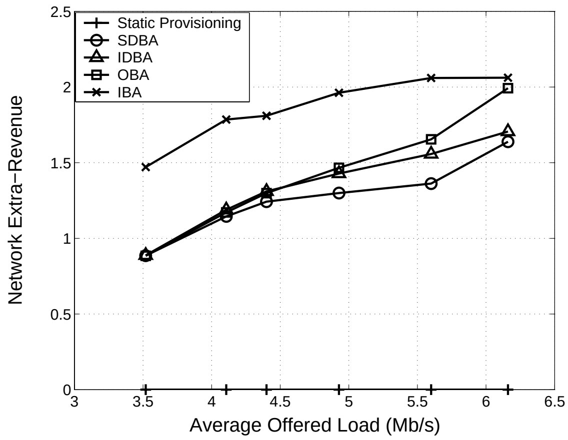

Figure 8.2: Performance upper bound, network topology with a larger number of links (a). Average total accepted load (b) and network extra-revenue (c) versus the average load offered to the network.




(a)




(b)




(c)

Figure 8.3: Performance upper bound, network topology with multiple bottleneck links (a). Average total accepted load (b) and network extra-revenue (c) as a function of the average load offered to the network.

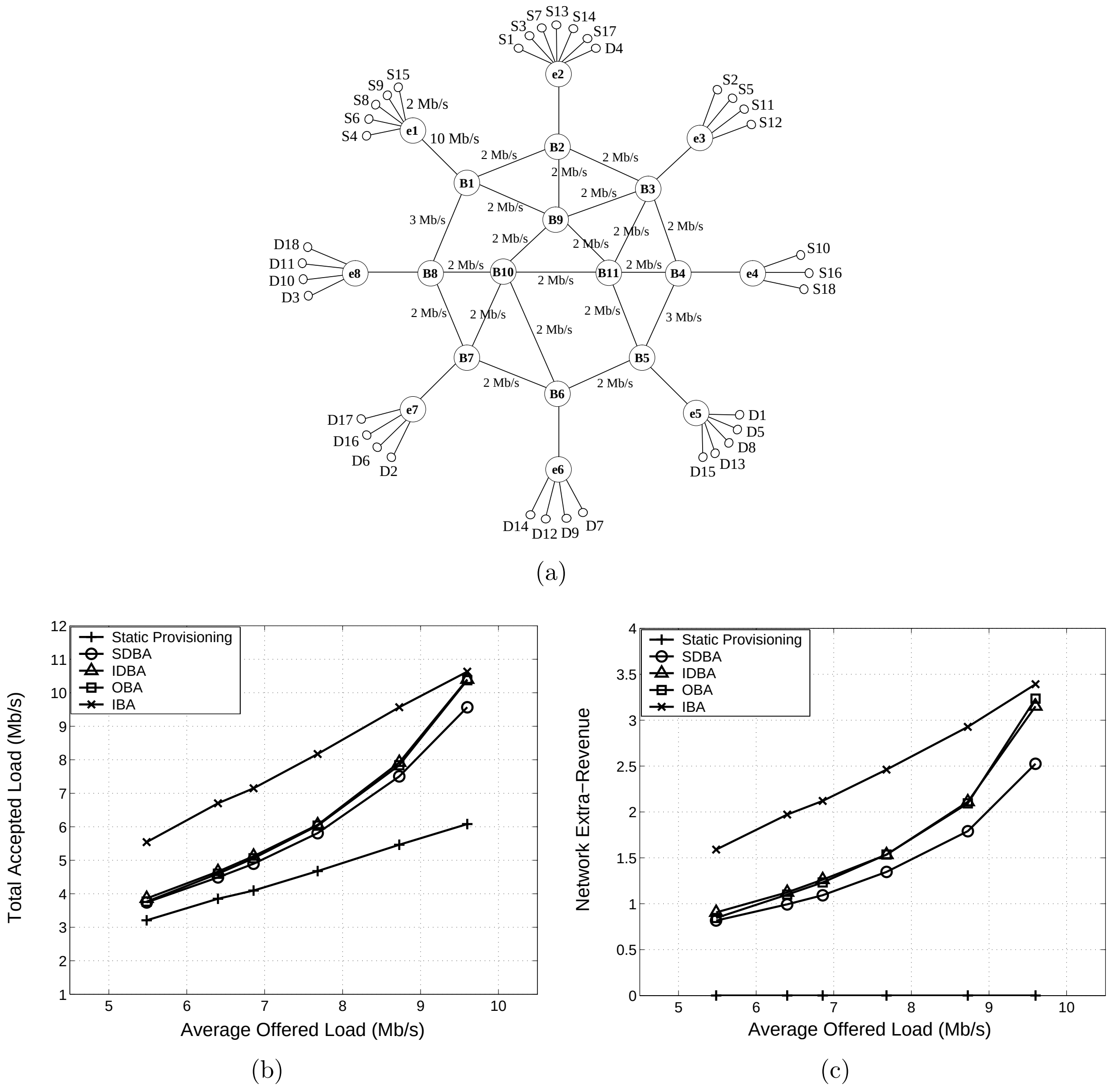



Figure 8.4: Performance upper bound, complex core network topology (a). Average total accepted load (b) and network extra-revenue (c) as a function of the average load offered to the network.

# Chapter 9

# Conclusion

## 9.1 Discussion and Concluding Remarks

Realizing flexible and efficient bandwidth allocation in communication networks is a challenging problem. This thesis has presented a number of contributions to address this problem. We have proposed a novel service model where users subscribe for guaranteed transmission rates, and the network periodically individuates unused bandwidth that is re-allocated and guaranteed with short-term contracts to users who can better exploit it according to their traffic needs and bandwidth utility functions. We have developed a set of efficient algorithms for dynamic bandwidth allocation. These algorithms take explicitly into account traffic statistics to increase the users perceived utility and the network revenue. Further, we have introduced a novel mathematical model that optimizes the total network revenue and provides bounds to the performance achievable by any online dynamic bandwidth allocation algorithm.

In Chapter 2, we presented the background required for the understanding of the bandwidth allocation problem, and the work related to this research. We compared our approach to existing schemes underlying the novelty of our approach.

In Chapter 3, we presented a new service model that, first, provides a quantitative per-flow bandwidth guarantee and then exploits the unused bandwidth individuated periodically in the network to propose short-term guaranteed extra-bandwidth to users who are willing to pay more to get a higher bandwidth. To implement our service model, we introduced a distributed dynamic resource provisioning architecture for quality of service networks and a set of control messages to provide interactions between network elements.

In Chapter 4, we presented a precise statement of the bandwidth allocation problem, then we provided a utility-based definition of the network extra-revenue and finally we performed a connections classification based on traffic statistics gathered at ingress routers. In Chapter 5, we proposed a broad set of heuristic dynamic bandwidth allocation algorithms tailored to our proposed service model, with increasing complexity and performance.

In Chapter 6, we introduced a novel mathematical model that assumes the exact knowledge of the future traffic offered to the network and extends the well known utility maximization problem studied in [7] to include connections arrival-times and durations. Our model allows to provide bounds to the performance achievable by any online dynamic bandwidth allocation algorithm.

Simulation results in Chapter 7 measured in realistic network scenarios showed that our allocation algorithms and service model allow to increase both resource utilization and network revenue with respect to static provisioning techniques. Further, in Chapter 8, we compared the performance of our bandwidth allocation algorithms to the bounds provided by the mathematical model. We have shown that these algorithms allow to increase consistently network revenue with respect to static provisioning schemes and approach, in several network scenarios, the upper bound provided by the mathematical model.

## 9.2 Future Research Issues

In this thesis we have addressed the problem of dynamic bandwidth allocation considering a fixed routing strategy. We have assumed that each communication between a user pair is established by creating a session involving a path that remains fixed throughout the user pair conversation duration. The session path choice method (i.e., the routing algorithm) has not been considered. In [7], the authors have addressed the problems of charging, rate control and routing for a communication network carrying elastic traffic. Therefore, we envisage to take into account the routing strategy to extend our work. The routing issue in dynamic bandwidth allocation should be studied carefully to achieve a higher network revenue while satisfying, at the same time, the individual users' requirements.

On the other hand, realizing flexible and efficient bandwidth allocation in wireless networks presents several technical challenges, including physical layer impairments, mobility and bandwidth mismatch between wireline and wireless networks. Physical layer impairments (e.g., co-channel interference, hidden terminals, path-loss, fast fading and shadowing) contribute toward time-varying error characteristics and time-varying channel capacity making the delivery of hard QOS guarantees unlikely. In addition, user mobility can trigger rapid degradation in the delivered service quality (e.g., during handoff). Hence, one important perspective of this thesis can be devoted to study the challenging points in wireless networks and extend our work to this area.

# References


[1] A. T. Campbell and R. R.-F. Liao, “Dynamic Core Provisioning for Quantitative Differentiated Services,” *IEEE/ACM Transactions on Networking*, pp. 429–442, vol. 12, no. 3, June 2004.

[2] H. Schulzrinne and X. Wang, “Incentive-Compatible Adaptation of Internet Real-Time Multimedia,” *IEEE Journal on Selected Areas in Communications*, pp. 417–436, vol. 23, no. 2, February 2005.

[3] A. T. Campbell and R. R.-F. Liao, “Dynamic Edge Provisioning for Core IP Networks,” in *Proceedings of IEEE/IFIP International Workshop on Quality of Service IWQOS*, Pittsburgh, USA, June 2000.

[4] T. Ahmed, R. Boutaba, and A. Mehaoua, “A Measurement-Based Approach for Dynamic QoS Adaptation in DiffServ Network,” *Journal of Computer Communications, Special issue on End-to-End Quality of Service Differentiation, Elsevier Science*, 2004.

[5] M. Mahajan, M. Parashar, and A. Ramanathan, “Active Resource Management for the Differentiated Services Environment,” *International Journal of Network Management*, pp. 149–165, vol. 14, no. 3, May 2004.

[6] Z. Cao and E. Zegura, “Utility Max-Min: An Application-Oriented Bandwidth Allocation Scheme,” in *Proceedings of IEEE INFOCOM*, New York, USA, March 1999.

[7] F. P. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, pp. 33–37, vol. 8, no. 1, January 1997.

[8] J. Aweya, M. Ouellette, and D. Y. Montuno, "A simple, scalable and provably stable explicit rate computation scheme for flow control in computer networks," *International Journal of Communication Systems*, pp. 593–618, vol. 14, no. 6, August 2001.

[9] J. Aweya, M. Ouellette, and D. Y. Montuno, "Design and stability analysis of a rate control algorithm using the Routh-Hurwitz stability criterion," *IEEE/ACM Transactions on Networking*, pp. 719–732, vol. 12, no. 4, August 2004.

[10] S. Shenker, "Fundamental Design Issues for the Future Internet," *IEEE Journal on Selected Areas in Communications*, pp. 1176–1188, vol. 13, no.7, September 1995.

[11] L. Breslau and S. Shenker, "Best-Effort versus Reservations: A Simple Comparative Analysis," in *Proceedings of ACM SIGCOMM*, pp. 3–16, September 1998.

[12] R. M. Salles and J. A. Barria, "Fair and efficient dynamic bandwidth allocation for multi-application networks," *Computer Networks*, pp. 856–877, vol. 49, December 2005.

[13] D. Bertsekas and R. Gallager, *Data Networks, 2nd Edition.* Prentice-Hall, 1992.

[14] R. J. La and V. Anantharam, "Utility-Based Rate Control in the Internet for Elastic Traffic," *IEEE/ACM Transactions on Networking*, pp. 272–286, vol. 10, no. 2, April 2002.

[15] J.-W. Lee, R. R. Mazumdar, and N. B. Shroff, "Non-Convex Optimization and Rate Control for Multi-Class Services in the Internet," *IEEE/ACM Transactions on Networking*, pp. 827–840, vol. 13, no. 4, August 2005.

[16] H. Schulzrinne and X. Wang, "RNAP: A resource negotiation and pricing protocol," in *International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'99)*, pp. 77–93, Basking Ridge, New Jersey, June 1999.

[17] F. Kelly, A. Maulloo, and D.Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, pp. 237–252, vol. 49, no. 3, Mar. 1998.

[18] J. Wroclawski, "The use of rsvp with ietf integrated services." RFC 2210, September 1997.

[19] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource reservation protocol (rsvp) - version 1 functional specification." RFC2205, September 1997.

[20] L. Breslau, E. Knightly, S. Shenker, I. Stoica, and H. Zhang, "Endpoint admission control: architectural issues and performance," in *Proceedings of ACM SIGCOMM*, pp. 57–70, Stockholm, Sweden, Sept. 2000.

[21] F. P. Kelly, P. B. Key, and S. Zachary, "Distributed admission control," *IEEE Journal on Selected Areas in Communications*, pp. 2617–2628, vol. 18, Dec. 2000.

[22] C. Cetinkaya and E. Knightly, "Egress admission control," in *Proceedings of IEEE INFOCOM*, pp. 1471–1480, Tel Aviv, Israel, Mar. 2000.

[23] S. Jamin, S. J. Shenker, and P. B. Danzig, "Comparison of measurement-based admission control algorithms for controlled-load service," in *Proceedings of IEEE INFOCOM*, April 1997.

[24] H. Zhang and S. Keshav, "Comparison of rate-based service disciplines," in *Proceedings of ACM SIGCOMM*, Zurich, Switzerland, September 1991.

[25] D. Black, S. Blake, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services." RFC 2475, December 1998.

[26] P. Reichl, S. Leinen, and B. Stiller, “A practical review of pricing and cost recovery for Internet services,” in *Proceedings of the 2nd Berlin Internet Economics Workshop (IEW’99)*, Berlin, Germany, May 1999.

[27] L. Massoulié and J. Roberts, “Bandwidth sharing: objectives and algorithms,” *IEEE/ACM Transactions on Networking*, pp. 320–328, vol. 10, no. 3, June 2002.

[28] S. H. Low and D. E. Lapsley, “Optimization flow control-I: basic algorithm and convergence,” *IEEE/ACM Transactions on Networking*, pp. 861–874, vol. 7, no. 6, December 1999.

[29] S. Athuraliya and S. H. Low, “Optimization flow control-II: Implementation,” in *Proceedings of IEEE INFOCOM*, 2000.

[30] K. Kar, S. Sarkar, and L. Tassiulas, “A simple rate control algorithm for maximizing total user utility,” *IEEE INFOCOM*, pp. 133–141, 2001.

[31] H. Yaïche, R. R. Mazumdar, and C. Rosenberg, “A game theoretic framework for bandwidth allocation and pricing of elastic connections in broadband networks: theory and algorithms,” *IEEE/ACM Transactions on Networking*, pp. 667–678, vol. 8, no. 5, October 2000.

[32] S. Kunniyur and R. Srikant, “End-to-end congestion control schemes: utility function, random losses and ECN marks,” *IEEE INFOCOM*, pp. 1323–1332, 2000.

[33] A. Charny and K. K. Ramakrishnan, “Time Scale Analysis of Explicit Rate Allocation in ATM Networks,” in *Proceedings of IEEE INFOCOM*, April 1996.

[34] J-Sim, “Available at www.j-sim.org.” Ohio State University.

[35] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming.* Duxbury Press/Brooks/Cole Publishing Company, 2002.

[36] K. Pawlikowski, H.-D. J. Jeong, and J.-S. R. Lee, "On credibility of Simulation Studies of Telecommunications Networks," *IEEE Communications Magazine*, pp. 132–139, January, 2002.

[37] N. Meskaoui, "A framework to model and simulate multi-agent systems within telecommunication networks: New environment, tools and behaviors." P.H.D. Thesis report, Paris 6 University, France, 2004.

# Appendix A

# Algorithm Implementation

The numerical results presented in this thesis are obtained using the simulation tool *J-Sim* ver. 1.3 [34] and the Modeling Language for Mathematical Programming (AMPL) [35].

In our work, we have proposed four algorithms for dynamic bandwidth allocation:

- Simple Dynamic Bandwidth Allocation (SDBA) algorithm;
- Iterative Dynamic Bandwidth Allocation (IDBA) algorithm;
- Optimum Bandwidth Allocation (OBA) algorithm;
- Ideal Bandwidth Allocation (IBA) algorithm.

We recall that all these algorithms proceed in two main steps:

- First, bandwidth is allocated to all active connections trying to match their near-term traffic requirements that are predicted based on statistics collected by ingress routers.
- Second, the spare bandwidth as well as the bandwidth left unused by idle and active connections is individuated on each link. Such available extra-bandwidth is allocated with guarantee during the current update interval exclusively to connections that can take advantage of it since they are already fully exploiting their subscribed rate.

To implement these two steps, it is necessary to have an accurate information regarding all connections entering the network. More specifically, at the beginning of each update interval, our algorithms need to know the state of each connection and its actual sending rate.

To this end we installed traffic monitors at each ingress router to perform online measurements on the incoming traffic flows, and we implemented a table, named *Connections' State Table* (CST), that contains the state information concerning all connections.

The J-Sim network simulation package (*drcl.inet*) offers the basic classes defined in the abstract network model. Therefore, to augment the ingress router elements with the key features mentioned above, we extended the *drcl.inet* package to implement the CST table and traffic monitors.

The structure of this Chapter is the following: Section A.1 describes in some detail AMPL, and Section A.2 provides a brief description of our extensions to J-Sim existing classes, including the CST table and dynamic bandwidth allocation algorithms implementation.

## A.1 AMPL: A Modeling Language for Mathematical Programming

AMPL is an algebraic modeling language for linear and nonlinear optimization problems, in discrete or continuous variables. Developed at Bell Laboratories, AMPL allows to formulate optimization models and examine solutions, while the computer manages communication with an appropriate solver. AMPL's flexibility and convenience make it ideal for rapid prototyping and model development, while its speed and control options make it an efficient choice for repeated production runs. For more details, the reader can refer to [35].

As described in the previous Chapters, the two algorithms OBA and IBA are based on a mathematical formulation of the network revenue maximization problem. Therefore, these latters are implemented using AMPL along with J-Sim simulator. The way in which AMPL is used is described briefly in the following.

## A.2 J-Sim Simulator Extensions

J-Sim is presented in [37] as an open, component-based, compositional simulation environment, proposed in the frame of research cooperation between the Ohio State University and the University of Illinois at Urbana-Champaign, and built in Java upon the notion of the autonomous component architecture (ACA).

The behaviors of the J-Sim components are defined in terms of contracts and can be individually designed, implemented, tested, and incrementally deployed in a software system. A system can be composed of individual components in much the same way a hardware module is composed of Integrated Circuits chips. Moreover, components can be plugged into a software system, even during execution. This makes J-Sim a platform-neutral, extensible, and reusable environment.

J-Sim also provides a script interface to allow integration with different script languages such as Perl, Tcl or Python. A fully integrated J-Sim with a Java implementation of the Tcl interpreter (with the Tcl/Java extension) already exists and is called Jacl. So, similar to NS-2, J-Sim is a dual-language simulation environment in which classes are written in Java (for NS-2, in C++) and "glued" together using Tcl/Java. However, all the public classes/methods/fields in Java can be accessed naturally in the Tcl environment without the need to export them to the Tcl environment as required in NS-2.

This component-based architecture makes it possible to compose different simulation scenarios, for different networks' architectures, from a set of basic components and classes proposed by J-Sim and/or defined by the user.

### A.2.1 Extensions to existing J-Sim classes

The basic package for network simulation in J-Sim is the *drcl.inet* package. This package contains the basic classes defined in the abstract network model, as well as a set of utility functions and classes that facilitate creation of simulation scenarios.

To implement our dynamic bandwidth allocation algorithms, we extended the J-Sim package both defining a broad set of Java classes and modifying a large number of already existing ones. In this Chapter, we will simply describe the *Connections' State Table* implementation.

Note that the Java class *Node.java* (in the *drcl.inet* package) is extended to implement our ingress routers by including a set of Java fields and methods. Among these fields, the most important is the CST table and its implementation is described in the following.

### A.2.2 Implementation of Connections' State Table (CST)

The CST table allows to maintain the state information regarding all connections established in the network.

We implemented the CST table in a vector named *CSTtable*, where each element corresponds to one connection and maintains the state information regarding such connection. Each element of the CST table is implemented in the Java class *ConnectionState.java* and it contains the following fields:

- long SourceAddress: the connection source node;
- long DestinationAddress: the connection destination node;
- Node[] ConnectionPath: the connection path;
- double SLARate: the connection subscribed rate;
- double Weight: the connection weight;

- boolean State: the connection state, in the current update interval: idle, active;
- boolean PreviousState: the connection state, in the previous update interval: idle, active;
- double RealRate: the connection transmission rate;
- double AllocatedRate: the connection allocated rate;
- double MinRate: the minimum connection rate;
- double OfferedRate: the connection offered rate;
- double FairShare: the connection fair share;
- boolean Served: when Served is true, the connection will be not taken into account in the successive iterations of the algorithm;
- Node IngressRouter: the ingress router through which the connection enters the network;

Note that the above fields are similar to those introduced in Chapter 5.

The CST table can be like the one presented in Table A.1, where each row represents one connection and contains the set of fields listed above.

Table A.1: The CST table

| | | | | | | |
|---|---|---|---|---|---|---|
| $Connection_1$ | $SourceAddress_1$ | $DestinationAddress_1$ | $Path_1$ | $SLArate_1$ | $State_1$ | ... |
| $Connection_2$ | $SourceAddress_2$ | $DestinationAddress_2$ | $Path_2$ | $SLArate_2$ | $State_2$ | ... |
| .<br>.<br>. | ...<br>...<br>... | | | | | |
| $Connection_N$ | ... | | | | | |

### A.2.3 Implementation of the Dynamic Bandwidth Allocation algorithms

In this Section, we describe in some detail the principal phases of our bandwidth allocation algorithms implementation.

First, we created the network topology and we implemented the CST tables along with a set of traffic monitors and traffic conditioners (token buckets) at ingress routers.

The basic role of traffic monitors is to perform online measurements on the incoming traffic flows. These measurements are then collected and exchanged between ingress routers to update the CST tables, accordingly, with the state information regarding each connection.

It should be clarified that the CST table, implemented at each ingress router, contains the state information concerning all connections offered to the network. In effect, each ingress router updates the CST table with the state information regarding connections that enter the network through it and then exchanges update messages with other ingress routers to maintain the state information regarding all the connections of the network.

On the other hand, token buckets are used to regulate each connection sending rate according to the allocated bandwidth computed by our bandwidth allocation algorithm.

Each link in the network, along with the correspondent parameters (the maximum capacity, the residual capacity, etc.) is implemented in the Java class *OneLink.java*. The set of links are implemented in a table named *LinksTable*, where each element corresponds to one instance of the Java class *OneLink.java*.

Every update interval (i.e. $T_u$ seconds), the proposed bandwidth allocation algorithms proceed as follows:

- CST tables implemented in ingress routers are updated with the current actual information concerning all connections entering the network. This information derives from the statistics gathered by ingress routers. Table A.2 shows how the CST table

(*CSTtable*) is updated by the ingress router (*IngressRouter*).

- At this phase, we distinguish between the four bandwidth allocation algorithms (SDBA, IDBA, OBA and IBA).

  - SDBA takes as input the updated CST table, the *LinksTable* table and the *routing matrix*, and then outputs the bandwidth allocations associated to the connections listed in the CST table. The bandwidth allocation values calculated by SDBA are communicated to token buckets installed at ingress routers to regulate the sources' sending rate.

    Note that, Table A.1 has shown a general case for updating the CST table taking into account the connections' offered rate. These latters are not considered during SDBA execution, differently from IDBA presented hereafter.

  - IDBA iterates over SDBA considering also the connections' offered rates that are updated (see Table A.2), in the previous phase, based on the measurements performed by the traffic monitors. We recall that with IDBA a connection cannot be assigned a bandwidth higher than its offered rate. Thus, IDBA can exploit more efficiently network resources achieving further gains than SDBA, in terms of network revenue.

  - OBA is implemented by exploiting the important feature of Java that allows to call external programs during execution. OBA takes all the inputs considered by IDBA, as well as some setting parameters and translates all these to an environment comprehensible by AMPL.

    At this stage, AMPL is called by OBA through an external program using the Java method *getRuntime().exec(String[])*. Table A.3 explains in some detail how AMPL is called by OBA.java, which is the Java class that implements OBA.

So AMPL executes the file *NetworkModel.run* (see Table A.3), computes the connections' bandwidth allocation and writes these values in the output file *AllocatedBandwidth.out*. When OBA detects the finishing of AMPL (the Java method *p.waitFor()*), it reads the values of the bandwidth allocations from the file *AllocatedBandwidth.out* and sets these values in the CST table and then configures the token buckets at ingress routers appropriately.

– IBA, the ideal bandwidth allocation algorithm proceeds in the same manner as OBA.

Table A.2: Java code for updating the CST table

```
Node IngressRouter;
InterfaceInfo trafficStatistics;
for(int j=0; j < CSTtable.length; j++)
{
   IngressRouter = CSTtable[j].getIngressRouter();
   trafficStatistics = IngressRouter.getNodeInfo(1);
   if (trafficStatistics.getInLoad() > 0)
   {
      CSTtable[j].setActive(true);
      CSTtable[j].setRealRate((IngressRouter.getNodeInfo(0)).getOutLoad());
      CSTtable[j].setOfferedRate((IngressRouter.getNodeInfo(1)).getInLoad());
   }
   else
   {
      CSTtable[j].setActive(false);
   }
}
```

- *trafficStatistics* is a field that contains the statistics information regarding one interface of the *IngressRouter* router.
- *getNodeInfo(int if_)* is a method that allows to collect the traffic statistics at interface *if_* of the ingress router.
- *setActive(boolean)*, *setRealRate(double)* and *setOfferedRate(double)*: these three methods update the CST table with the information on the state, the transmission rate and the offered rate of each connection, respectively.
- *getInLoad()* and *getOutLoad()*: these two methods return, respectively, the incoming and outgoing traffic statistics for a particular interface of the ingress router.

Table A.3: Java code for calling AMPL from the Java file *OBA.java*

```
cmdarray[0] = "/bin/bash";
cmdarray[1] = "-c";
cmdarray[2] = "./ampl NetworkModel.run";
try
{
    process p = Runtime.getRuntime().exec(cmdarray);
    p.waitFor();
    System.out.println("—- AMPL solver has finished —-");
}
catch (Exception e_)
{
    System.out.println("Error process");
    System.exit(1); // for abnormal termination
}
```

The file *NetworkModel.run* contains the following AMPL commands:

```
model NetworkModel.mod;
data NetworkModel.dat;
option solver './snopt';
solve;
display { k in GreedyConnections } (sum { t in Intervals }
    AMPL_AllocatedBandwidth [ k,t ]) > AllocatedBandwidth.out;
display AMPL_NetworkRevenue > NetworkRevenue.out;
quit;
```

- *NetworkModel.mod* is a file that describes the network model using standard AMPL format.
- *NetworkModel.dat* is a file that provides the network model parameters.
- *AllocatedBandwidth.out* and *NetworkRevenue.out* are two output files that contain, respectively, the connections' allocated rate and the total network revenue calculated by AMPL.

# List of Acronyms

| | |
|---|---|
| ACA | *Autonomous Component Architecture* |
| AMPL | *A Modeling Language for Mathematical Programming* |
| ARM | *Active Resource Management* |
| CST | *Connections' State Table* |
| ECN | *Explicit Congestion Notification* |
| FASTRAC | *Fast Rate Computation* |
| IBA | *Ideal Bandwidth Allocation* |
| IDBA | *Iterative Dynamic Bandwidth Allocation* |
| IETF | *Internet Engineering Task Force* |
| IP | *Internet Protocol* |
| LIP6 | *Laboratoire d'Informatique de Paris 6* |
| LP | *Linear Programming* |
| OBA | *Optimum Bandwidth Allocation* |
| PEP | *Policy Enforcement Point* |
| PDP | *Policy Decision Point* |
| QoS | *Quality of Service* |
| RFC | *Request For Comments* |
| RNAP | *A Resource Negotiation and Pricing Protocol* |
| RSVP | *Resource Reservation Protocol* |
| SDBA | *Simple Dynamic Bandwidth Allocation* |

SLA *Service Level Agreement*

WFQ *Weighted Fair Queuing*

# List of Figures

# List of Tables