

Université de Montréal

Qualité de Service dans les Services Web

Par
Ikbal Taleb

Département d'informatique
et de recherche opérationnelle

Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de
Maîtrise ès sciences (M.Sc.)
En informatique

Mars 2006

Copyright © Ikbal Taleb, 2006



QA
76
054
2006
V.026

Direction des bibliothèques

AVIS

L'auteur a autorisé l'Université de Montréal à reproduire et diffuser, en totalité ou en partie, par quelque moyen que ce soit et sur quelque support que ce soit, et exclusivement à des fins non lucratives d'enseignement et de recherche, des copies de ce mémoire ou de cette thèse.

L'auteur et les coauteurs le cas échéant conservent la propriété du droit d'auteur et des droits moraux qui protègent ce document. Ni la thèse ou le mémoire, ni des extraits substantiels de ce document, ne doivent être imprimés ou autrement reproduits sans l'autorisation de l'auteur.

Afin de se conformer à la Loi canadienne sur la protection des renseignements personnels, quelques formulaires secondaires, coordonnées ou signatures intégrées au texte ont pu être enlevés de ce document. Bien que cela ait pu affecter la pagination, il n'y a aucun contenu manquant.

NOTICE

The author of this thesis or dissertation has granted a nonexclusive license allowing Université de Montréal to reproduce and publish the document, in part or in whole, and in any format, solely for noncommercial educational and research purposes.

The author and co-authors if applicable retain copyright ownership and moral rights in this document. Neither the whole thesis or dissertation, nor substantial extracts from it, may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms, contact information or signatures may have been removed from the document. While this may affect the document page count, it does not represent any loss of content from the document.

Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé :

Qualité de service dans les services Web

présenté par :

Ikbal Taleb

a été évalué par un jury composé des personnes suivantes :

Yan-Gaël Guéhéneuc

Président-rapporteur

Abdelhakim Hafid

Directeur de recherche

Michel Gendreau

Membre du jury

Mémoire accepté le 4 mai 2006

Résumé

L'abondance des services Web sur l'Internet, la facilité de leur utilisation et d'intégration aux différents systèmes hétérogènes existants, a fait surgir un nouveau défi aux fournisseurs de ce genre de services. Comment pourront-ils assurer et garantir la qualité des services Web offerts sur le réseau Internet ?

Cette Qualité de Service (QoS) des services Web est représentée par la QoS du service Web lui-même, largement étudiée, et par la QoS du moyen de communication (par exemple : Internet) sur lequel se basent les services Web. La QoS réseau peut-être supportée par les technologies réseau telles que IntServ, DiffServ et MPLS. La QoS des services Web au niveau serveur peut être facilement supportée par une ingénierie adéquate de la ferme de serveurs. Le problème qui se pose est comment utiliser le support de QoS du côté réseau pour garantir la QoS de bout en bout sans pour autant exiger des changements de l'infrastructure (réseau et services Web) actuelle.

Dans ce projet, nous proposons un gestionnaire de ressources réseau (NRM : Network Resources Manager), qui est lui-même un service Web, qui permet de mesurer, vérifier et réserver les ressources réseaux afin de satisfaire la QoS demandée par les utilisateurs de services Web. Notre proposition n'exige aucune modification de l'infrastructure actuelle ; en effet, NRM est simplement un service Web. La performance de cette proposition a été évaluée par simulation dans un réseau DiffServ.

Pour améliorer la procédure de réservation de ressources avec NRM, nous avons développé et évalué différents algorithmes de réservation des ressources réseau à l'avance. Ces algorithmes permettent la recherche de ressources disponibles pour générer des listes de propositions dans le cas où la réservation originale est impossible. Une proposition consiste à un commencement en retard, une durée plus courte, ou une QoS dégradée pour le service demandé sans aucun retard.

Mots-clés : Qualité de service, Services Web, Gestion de ressources réseau, Réservation de ressources à l'avance

Abstract

The abundance of Web services on the Internet and their ease of use and integration to the heterogeneous existing systems pose a new challenge to the providers of this kind of services. How will they ensure and guarantee the Quality of Service of the Web services, they offer, on the Internet?

The Quality of Service (QoS) of Web services is represented by the QoS of the Web service itself, largely studied, and by the QoS of the communications network (e.g., Internet) that is used by users to access Web services. Network QoS can be supported by new emerging network technologies, such as DiffServ, IntServ, and MPLS. Web services QoS support in the server can be easily supported by careful design of server farms hosting the Web services. Even though QoS support is readily available, end-to-end QoS support for Web services is still an open issue; indeed, there is a need to support end-to-end QoS of Web services without requiring any changes to current (network and Web services) infrastructure.

In this project, we propose a Network Resource Manager (NRM: Network Resources Manager), a Web service, which reserves, measures, and checks the network resources to satisfy the QoS requested by the users of Web services. Our proposal does not require any changes to the current infrastructure; indeed, NRM is just another Web service. The evaluation of the performance of this proposal was performed, via simulations, on a DiffServ-enable network.

To support advance reservation of resources, we extended NRM with a number of algorithms. These algorithms produce, in case of requests that cannot be accommodated, a list of proposals/alternatives; each proposal consists of a delayed start, short duration, and/or a degraded QoS for the requested service.

Keywords: Quality of service, Web Services, Network Resources Management, Future Resources Reservation

Table des matières

Chapitre 1	1
Introduction.....	1
1.1 Problématique et motivation.....	2
1.2 Contributions	3
1.3 Plan du mémoire	5
Chapitre 2	6
Qualité de service	6
2.1 Introduction	6
2.2 Applications Internet et QoS	6
2.3 Définition.....	7
2.4 Les critères de la qualité de service	7
2.5 Les modèles de qualité de service.....	9
2.5.1 IntServ/RSVP	9
2.5.2 DiffServ	10
2.5.3 MPLS/ l'ingénierie de trafic.....	11
2.6 Les mesures de QoS.....	12
2.7 Conclusion.....	12
Chapitre 3	13
Services Web.....	13
3.1 Introduction	13
3.2 SOA : Architecture orientée service	14
3.2.1 SOA : caractéristiques	14
3.2.2 SOA : définition.....	14
3.2.3 Services Web : les technologies utilisées	15
3.2.4 Services Web : définition	16
3.3 Architecture services Web.....	16
3.3.1 Rôles et opérations	17
3.3.3 La Pile de protocole des services Web.....	18
3.4 Service Web et qualité de service	19

3.4.1 Introduction.....	19
3.4.2 SLA : accords de niveau de service (Service Level Agreement).....	19
3.4.3 Courtier de QoS-SW	19
3.4.4 La QoS des services Web.....	20
3.5 Services Web multimédia.....	22
3.6 Outils de développement des services Web	23
3.7 Service Web et QoS de bout en bout: revue de littérature	23
3.8 Conclusion.....	24
Chapitre 4	26
Gestionnaire de Ressources Réseau (NRM)	26
4.1. Introduction	26
4.2. Motivations.....	26
4.3 Classification des services Web.....	28
4.3.1 Les types de services Web.....	29
4.3.2 Classification orientée ressource réseau.....	29
4.4 Description de NRM.....	31
4.5 Principales fonctionnalités de NRM	31
4.6 Les tâches de NRM.....	32
4.7 Modules de NRM.....	34
4.7.1 Analyseur NRM (NRM-Analyser).....	35
4.7.2 NRM sélecteur de fonctions (NRM-Select)	35
4.7.3 Classificateur de services Web (NRM-Classifier)	36
4.7.4 Vérificateur de ressources réseau (NRM-Ckecker)	36
4.7.5 Surveillance des ressources (NRM-Monitor)	37
4.7.6 Base de données NRM	37
4.7.7 Réservation de ressources.....	37
4.7.8 Outils de mesures de réseaux.....	38
4.8 Architecture de l'environnement d'exécution de NRM.....	38
4.9 Scénario d'utilisation de NRM	39
4.10 Implémentation	42
4.10.1 Service Web vidéo-sur-demande et NRM.....	43

4.10.2 Service Web multimédia : interface et utilisation	43
4.11 Simulation.....	45
4.7 Conclusion.....	48
Chapitre 5	49
Fouille de données.....	49
5.1 Introduction	49
5.2 Définitions.....	49
5.3 Tâches de fouille de données.....	49
5.4 Étapes du processus de fouille de données.....	50
5.5 Techniques de fouille de données.....	51
5.8 Méthode KPPV (K plus proches voisins)	52
5.8.1 Description.....	52
5.8.2 Définition de la distance.....	52
5.8.3 Application de KPPV au problème de réservation	53
5.8.4 Discussion de la technique KPPV.....	55
5.9 Conclusion.....	55
Chapitre 6	56
Réservation de Ressources.....	56
6.1 Introduction	56
6.2 Objectifs et motivations	56
6.2 NAFUR	58
6.3 Structures de données et agrégation.....	59
6.3.1 Structures de données.....	60
6.3.2 Agrégation	60
6.4 L'algorithme CP-KPPV	61
6.4.1 Principe.....	61
6.4.2 Application de KPPV à la génération de propositions.....	61
6.4.3 Données de classification (disponibilités de QdS).....	62
6.4.3.1 Filtrage des données	64
6.4.4 Fonctionnement.....	66
6.5 Les autres algorithmes.....	67

6.5.1 L'algorithme CPA-NAFUR.....	67
6.5.2 L'algorithme régulier de réservation	68
6.5.3 L'algorithme hybride CP-KPPV-PCA : CPH.....	69
6.6 Paramètres de performance	70
6.6.1 MDP	70
6.6.2 Nombre de propositions	70
6.6.3 Flexibilité.....	70
6.6.4 Poids des paramètres QoS	71
6.7 Conclusion.....	71
Chapitre 7	73
NAFUR : simulation et résultats	73
7.1 Introduction	73
7.2 Données de simulations.....	73
7.3 Les paramètres de performance	75
7.4 Les algorithmes expérimentés	75
7.5 Les résultats	78
7.5.1 Le taux de rejets	78
7.5.2 Le temps de réponse.....	80
7.5.3 Flexibilité sur la bande passante	83
7.6 Discussions.....	86
7.7 Conclusion.....	87
Chapitre 8	89
Conclusion.....	89
8.1 Discussion.....	89
8.2 Perspectives	90

Liste des Tableaux

Table 4.1 : Exemple de classes de types de services Web.....	30
Table 4.2 : Services vidéo offerts.....	44
Table 4.3 : Exemple de disponibilités sur le serveur vidéo.....	44
Table 5.1 : Algorithme générique KNN.....	52
Table 6.1 : Algorithme de réservation CP-KPPV.....	65
Table 6.2 : Algorithme de réservation régulier.....	67
Table 6.3 : Algorithme de réservation CPA-NAFUR.....	68
Table 7.1 : Exemple de requêtes de réservation.....	72
Table 7.2 : Nombre de rejets des principales méthodes sur différents MDP.....	77
Table 7.3 : Temps de réponse KPPV et CPA.....	83

Liste des figures

Figure 3.1: Le Modèle d'architecture orientée services.....	16
Figure 3.2: La pile de protocoles des services Web.....	18
Figure 3.3: Exemple de document WSDL contenant les descriptions de QdS.....	21
Figure 4.1: Différents points de mesure des paramètres de QdS réseau.....	28
Figure 4.2: Modules de NRM et leurs interactions avec le courtier de QdS-SW.....	35
Figure 4.3: Exemple de scénarios d'utilisation de NRM.....	42
Figure 4.4: Client applet java du service Web VSD (Vidéo-sur-Demande).....	45
Figure 4.5 : Exemple de connexion client-serveur vidéo.....	46
Figure 4.6 : Taux de paquets perdus sur un nœud du réseau Internet.....	47
Figure 4.7 : Taux de paquets perdus sur un nœud du réseau DiffServ.....	48
Figure 5.1 : Formules de calculs de distance.....	53
Figure 5.2 : Simulation de KPPV sur des données relative à notre étude.....	54
Figure 6.1 : Représentation schématique des réservations futures.....	57
Figure 6.2 : Principe NAFUR de recherche de propositions.....	58
Figure 6.3 : Exemple d'un ASP.....	59
Figure 6.4 : Temps segmenté dans la réservation future.....	60
Figure 6.5 : Application de KNN dans le problème de réservation.....	62
Figure 6.6 : (AASP) Agrégations disponibles au sein d'un ASP.....	64
Figure 6.7 : Filtrage d'agrégations dans un AASP.....	65
Figure 6.8 : Utilisation du MDP.....	70
Figure 7.1 : Distribution normale utilisée pour les requêtes.....	73
Figure 7.2 : Liste de propositions générées par CPA-NAFUR.....	76
Figure 7.3 : Liste de propositions générées par CP-KPPV.....	77
Figure 7.4 : Taux de rejets pour différents MDP pour CP-KPPV et CPA-NAFUR.....	79
Figure 7.5 : Nombre de rejets avec les différents algorithmes sur différents MDP.....	80
Figure 7.6 : Temps de réponse moyen CP-KPPV/CPA-NAFUR sur différents MDP.....	81
Figure 7.7 : Le temps de réponse pour différents MDP et Algorithmes.....	82
Figure 7.8 : Temps de Réponse et Flexibilité sur la bande passante MDP=30.....	84
Figure 7.9 : Taux de réservation et flexibilité sur la bande passante MDP=30.....	85

Figure 7.10 : Taux de rejets avec et sans flexibilité sur la bande passante.....86

Liste des sigles et des abréviations

ASP	Available Service Projection
AVL	Arbre AVL
CBR	Case Base Reasoning
CPA-NAFUR	Calcul de Propositions Amélioré avec NAFUR
CP-KPPV	Calcul de Propositions avec KPPV
HTML	Hyper Text Markup Language
HTTP	HyperText Transport Protocol
IETF	Internet Engineering Task Force
KPPV	K Plus Proches Voisins
MDP	Maximun Delay of proposal
MWS	Multimedia Web Service
NAFUR	Negotiation Aproach for Future Reservation
NRM	Network Resources Manager
QdS	Qualité de Service
QdS-SW	Qualité de Service des Services Web
RTP	Real time Transport Protocol
SOAP	Simple Object Access Protocol
UDDI	Universal Description, Discovery, and Integration
UMTS	Universal Mobile Telecommunication System
URL	Uniform Resource Locator
VoIP	Voice Over IP (Voix sur IP)
VSD	Vidéo Sur Demande
WSDL	Web Service Definition Language
XML	eXtensible Markup Language

À ma famille, mes amis et tous ceux qui m'ont soutenu tout au long de ce travail.

Remerciements

Je tiens à remercier chaleureusement toute personne qui a contribué de loin ou de près à ce travail.

Chapitre 1

Introduction

L'avènement de l'Internet comme outil de travail, source d'informations et moyen de communication l'a propulsé au rang du média le plus convoité et le plus puissants de la planète au même titre que la télévision, la radio et la presse écrite. Cette place bien méritée n'a guère arrêté sa croissance ; la disponibilité d'ordinateurs pas chers pour la maison a créé de nouvelles habitudes de consommation de l'Internet, qui dépassent les objectifs pour lesquels il a été conçu. Cela s'explique par l'engouement de l'industrie à vouloir intégrer tous les appareils de divertissement traditionnel dans un ordinateur personnel.

Actuellement, tout est possible sur l'Internet : on peut acheter et vendre, commander des services et se distraire en ligne. Il est devenu très important et indispensable pour les entreprises, les universités, les bibliothèques et les banques. Toute entité qui est branchée à l'Internet est capable d'offrir des services d'intérêt public ou spécialisés, commerciaux ou professionnels.

Pour les utilisateurs, louer et regarder une vidéo à la maison est devenue une chose très simple, grâce aux technologies de la Vidéo-sur-Demande (VSD) utilisées par les câblodistributeurs sur leurs propres réseaux propriétaires. Pour rendre de tels services accessibles universellement, il faut que les distributeurs utilisent l'Internet comme réseau de diffusion. Cependant, en étant sur un réseau qui offre une qualité de service qui n'est pas suffisante pour la garantir pour la vidéo. Pour y remédier, il faudrait trouver un concept simple, basé sur des technologies de communications existantes, afin d'offrir ce genre de service vidéo sur l'Internet. Parmi les nouvelles architectures prometteuses, on trouve le concept "architecture orienté service" (SOA : Service Oriented Architecture), et plus précisément les Services Web accessibles à toute machine connectée à l'Internet. Cette architecture a actuellement fait son chemin tant au niveau entreprise qu'au niveau résidentiel.

1.1 Problématique et motivation

Comme nous venons de le mentionner ci-dessus, l'Internet ouvre un grand portail pour accéder à divers services : vidéoconférence, VoIP, vidéo, TV, etc. Le problème étant que pour accéder à ces services, il faut en premier les trouver et, par la suite, choisir celui qui correspond le mieux à notre demande. Bien entendu, connaître le « meilleur » service Web (p.e., Service Web de Vidéo-sur-demande) n'est pas chose facile à cause de la quantité des services Web qui existent sur le Web, sauf par expérience, par recommandation ou par publicité.

Dans le but de faciliter la découverte des services Web et leurs utilisations, Le W3C¹ a introduit un annuaire nommé UDDI (Universal Description, Discovery, and Integration), dans lequel les services Web sont publiés par leurs fournisseurs. La recherche d'un service Web qui offre de la Vidéo-sur-Demande revient par exemple à chercher des mots clés comme vidéo, VSD, etc. Le résultat renvoyé par l'annuaire UDDI est une liste contenant tous les services Web qui ont un rapport avec les mots clés, mais à aucun moment cela ne garantira au client que le service choisi offrira une bonne qualité de service. On peut garantir une certaine satisfaction du côté fournisseur de service qui dépend largement de son nom dans le marché, de sa crédibilité et de la puissance des infrastructures hébergeant ses services Web. Par contre, on ne pourra pas assurer la qualité de service des services Web au niveau du medium de communication qui est l'Internet ; car aucune garantie n'est offerte par le fournisseur du service Web aux clients quant à la continuité d'une bonne réception du service pendant la durée que prendra sa consommation, comme c'est le cas dans la Vidéo-sur-Demande où le client pourra avoir des arrêts d'images et des coupures de son, malgré une bande passante suffisante au niveau du serveur vidéo et une excellente qualité de diffusion à sa sortie. En effet, de bout en bout entre les deux points du réseau (le client et le serveur), la disponibilité de la bande passante peut varier grandement au cours du temps parce que l'Internet n'offre pas de garantie, mais seulement le meilleur effort (Best Effort).

¹ W3C: World Wide Web Consortium

Le référencement fournis par les annuaires UDDI est subjectif car, au fil du temps, certains services Web ne sont plus disponibles ou ont changé d'URL et en plus UDDI n'offre pas d'informations sur la qualité de service du service Web lui-même et celle du réseau qui le supporte.

En réalité, les clients potentiels des services Web ont principalement besoin d'un outil de recherche comme UDDIe² [5] une version étendue d'UDDI qui intègre la QoS, et d'une liste de paramètres de qualité de service à lui soumettre, afin de trouver les services Web qu'ils cherchent. Cependant, pour une simple requête, il y a souvent le risque de se retrouver face à une grande liste à explorer ou rien du tout, dépendamment des paramètres demandés. Par ailleurs, les clients ne choisiront que le(s) premier(s) service(s) Web retourné(s). En outre, certains services Web sont parfois redondants, expirés ou tout simplement hors sujet.

Il y a eu beaucoup de travail fait par la communauté des services Web sur la QoS ; cependant tout l'intérêt s'était porté sur la QoS des services Web au niveau fournisseur, au niveau de la requête de recherche [5, 13, 4, 69, 57, 51, 52], au niveau de la définition sémantique de la QoS des services Web et de leurs contraintes. Il y a eu beaucoup de travail sur la QoS dans les réseaux [2, 53, 58, 20, 37, 69] ; cependant, peu de travail [51, 52], avec beaucoup de limitations, qui traite la QoS de bout en bout. Dans notre travail, nous proposons une approche qui supporte la QoS de bout en bout entre le client et le serveur de services Web, flexible et n'exigeant aucun changement à ce qui existe. En effet, nous proposons un service Web, appelé NRM (Network Resource Manager), qui va gérer la QoS et la garantir pour les clients des services Web. Cela réside dans le choix d'un gestionnaire de ressources réseau NRM qui sera invoqué par un courtier de QoS des services Web (WS-QoS Broker) [48] pour remédier au problème du support de la qualité de service réseau des services Web.

1.2 Contributions

Le sujet de recherche que nous proposons consiste en la conception et la réalisation d'un gestionnaire de ressources réseau NRM (Network Resources Manager),

² UDDIe : Universal Description, Discovery and Integration extension

spécialement conçue comme service Web afin de supporter la qualité de service réseau pour les services Web. Ce support est traduit par des tâches de mesure, vérification et surtout de réservation de ressources réseau à l'avance au cas où il n'est pas possible de satisfaire les demandes des clients au moment souhaité.

En effet, avec notre NRM un client peut choisir un service Web en précisant dans sa requête tous les paramètres qui expriment son choix ainsi que ceux de la qualité de service souhaitée. Par exemple, ceci se traduira pour un service Vidéo-sur-Demande, par la qualité et la résolution de l'image, la qualité du son et surtout le moment souhaité pour visionner cette vidéo. Si les ressources réseau sont disponibles entre le client et le serveur, NRM enverra un accord au Courtier de QoS-SW confirmant cette disponibilité au moment voulu. Dans le cas contraire, NRM procédera à une recherche d'autres possibilités afin de générer une liste de recommandations sur les horaires de début afin que le client choisisse celle qui lui convient le mieux.

Pour atteindre ces objectifs, NRM utilise une solution mixte : il utilise à la fois des recommandations basées sur deux algorithmes de recherche de propositions. Le premier est basé sur la technique de recherche du premier horaire disponible et le deuxième utilise une technique de fouille de données : les k-Plus Proches Voisins (KPPV).

Notre étude s'est basée sur la manière d'assurer la QoS des services Web de bout en bout en proposant ce qui suit :

- Un service Web NRM offrant des fonctions de mesures, vérification et garantie de la QoS de bout en bout. NRM n'opère et n'exige aucun changement aux infrastructures existantes actuelles dans les réseaux et des services Web contrairement à certaines approches [51, 52] qui modifient la pile de protocoles afin d'assurer la QoS de bout en bout.
- Une agrégation des disponibilités de QoS qui est utilisée par les algorithmes de calcul des propositions dans le futur. Cette agrégation permet de gagner au niveau temps de recherche et rapidité dans le calcul des propositions.
- Développement d'un nouvel algorithme basé sur une technique de fouille de données : KPPV pour la génération de propositions.

Toutes nos propositions ont été évaluées via des simulations. Certaines de ces propositions ont été déjà publiées ou soumises pour publication [23, 22].

1.3 Plan du mémoire

Après cette brève description des problématiques, nos motivations et nos contributions, le reste de ce mémoire est divisé en six chapitres.

En premier lieu, dans le chapitre 2, nous abordons des généralités sur la qualité de service : définitions, critères, les classes de différenciation de services et les principaux paramètres. Nous parlons aussi des modèles de qualité de service existants.

Dans le chapitre 3, nous décrivons les services Web et l'architecture SOA, ses composantes et son fonctionnement ainsi que la qualité de service pour les services Web.

Le chapitre 4 présente le gestionnaire de ressources réseau (NRM) et les différentes interactions entre NRM et les autres modules de l'architecture. Une explication détaillée décrit l'architecture de NRM, ses composantes, les choix conceptuel et les fonctionnalités orientées qualité de service réseau.

Dans le chapitre 5, nous décrivons brièvement des généralités sur les méthodes de classification utilisées dans l'analyse et la fouille de données.

Au chapitre 6, nous nous intéressons à l'agrégation de la QoS et l'optimisation des algorithmes de réservation future des ressources réseau. Nous présentons les différents algorithmes utilisés et proposés ainsi que les éléments de performance pris en charge telle que la structure de données utilisée et l'agrégation qui y est appliquée.

Dans le chapitre 7, nous présentons les résultats de nos expérimentations et les discussions relatives aux performances.

Enfin, le chapitre 8 clôture ce mémoire par une conclusion et une discussion, en présentant les perspectives et les directions futures de notre travail.

Chapitre 2

Qualité de service

2.1 Introduction

Le plus important défi au début de l'Internet, était de transmettre les paquets de données de la source à la destination. Juste après, il y a eu des développements qui ont contribué à la gestion de la perte de paquets et la congestion. Depuis 1990, les fournisseurs de services (Fournisseurs d'Accès à l'Internet) se heurtent à des problèmes comme la progression exponentielle du nombre d'internautes qui est passé de quelques centaines de milliers en 1990 à près de 934 millions en 2003 [11]. Cette montée explosive de l'Internet et du nombre de ses utilisateurs à des conséquences non négligeables sur la stabilité et la qualité de service du réseau.

Ce nombre d'internautes qui augmente rapidement, entraîne une croissance à tous les niveaux; le nombre de sites Web et de services a suivi la demande, provoquant une augmentation très importante du trafic global et impliquant la nécessité et l'urgence d'améliorer les infrastructures du réseau pour faire face aux nouveaux besoins et services. Ce qui implique une amélioration constante des infrastructures et des protocoles afin de pouvoir contrôler et assurer la qualité des services fournis : la QoS (la Qualité de Service).

2.2 Applications Internet et QoS

Les quantités de données transitant sur l'Internet augmentent chaque jour d'une manière presque exponentielle. La création de nouvelles applications qui demandent beaucoup de bande passante (Vidéo-sur-Demande, Vidéoconférence, etc.) et l'engouement des utilisateurs pour ce type d'applications pose de nouvelles contraintes aux fournisseurs d'accès à l'Internet, tel le trafic gigantesque sur le réseau qui est la cause de beaucoup de pertes de données et surtout de la congestion.

Dans l'Internet, il y a certains trafics sensible au délai et d'autres sensible à la perte de paquets. Le trafic sensible au délai est caractérisé par un taux de perte et une durée et pourrait avoir besoin de transmission en temps réel. Les exemples incluent l'audio/vidéo, qui sont habituellement stricts côté délai, mais peuvent accepter des pertes

de paquets jusqu'à une certaine limite, sans dégrader la QoS. Par contre, le trafic sensible à la perte de paquets est caractérisé par la quantité de l'information transmise. Par exemple, il y a les pages Web, les fichiers et les courriels. Ce genre de trafic est rigoureusement exigeant en ce qui concerne la perte de données, par contre il n'y a pas de limites strictes pour la durée de transmission [20].

La QoS des applications traditionnelles (HTTP, FTP, Courriel, etc.) n'est pas importante. Ces applications ne tolèrent pas les pertes de paquets, mais ils sont moins sensibles aux retards, et ne nécessitent pas de QoS au sens propre du mot. Elles sont à l'origine de l'Internet et le "Best Effort" est largement suffisant. Par contre, les applications en temps réel ont un comportement complètement opposé ; elles tolèrent la perte d'une quantité raisonnable de paquets, par contre elles deviennent critiques quant à la variation du délai, ce qui donnerait une mauvaise qualité de réception dans le cas où elle est instable.

2.3 Définition

La Qualité de Service est généralement assimilée à la discrimination de services, autrement dit, à la définition de classes différenciées de services. Mais cela signifie aussi garantir un service et pour cela lui réserver des ressources.

Le but primaire de la QoS est de fournir une priorité avec une largeur de bande dédiée, une gigue et latence contrôlées (requis par certains trafics en temps réel et interactif), et des caractéristiques améliorées de perte. Le plus important est de s'assurer que la priorité fournis pour un ou plusieurs flots ne fera pas échouer les autres dans le réseau.

2.4 Les critères de la qualité de service

La QoS peut être paramétrée de différentes manières. Généralement, les paramètres de la QoS réseau incluent le débit (bande passante), délai moyen de bout en bout, la limite du délai toléré, gigue, et le taux de perte. Les différents niveaux de QoS impliquent différents niveaux d'assurance de cette QoS et de l'utilisation des ressources.

Pour concevoir un système qui donne une satisfaction de la QoS et une utilisation optimale des ressources, il faut déterminer les conditions de la QoS de ces applications

et les convertir en paramètres de QoS réseau. Les paramètres de performance de la QoS généralement utilisés pour la qualité de service Internet sont définis dans les points suivants [37, 9].

- Le délai qui est défini comme étant le temps qui s'écoule entre l'envoi d'un paquet de données par l'émetteur et sa réception par son destinataire ;
- Variation de délai (Gigue) ;
- La bande passante qui est le taux de transfert de données (vitesse de transmission) entre l'émetteur et le receveur ;
- La disponibilité de la liaison entre deux-points : taux moyen d'erreurs d'une liaison.

Ces critères sont confrontés à certains facteurs qui perturbent leurs valeurs comme le réseau Internet et le protocole TCP lui-même.

Internet est composé d'un ensemble de routeurs et de liens de transmission. Ces liens ont des caractéristiques de délai, de débit maximum et de disponibilité et les routeurs y ont un impact significatif. En effet, les fonctions d'un routeur consistent à contrôler l'intégrité du paquet reçu, déterminer sa sortie et le stocker dans la file d'attente associée. Mais lorsque ce fonctionnement se dégrade (trop de trafic compte tenu des capacités du routeur), les files d'attente se remplissent, introduisant un délai supplémentaire dans la transmission des paquets forçant le routeur à rejeter des paquets. Ces actions influent sur le délai, la gigue, et le taux de perte. Pour tenir compte de l'évolution des réseaux, des mécanismes de contrôle du débit et de prévention de la congestion ont été ajoutés dans les implémentations du protocole TCP [61].

Dans ce contexte, offrir un service de bonne qualité consiste à fournir un service avec une gigue et un délai minimum ainsi qu'une disponibilité et un débit maximum. La Qualité de Service, quant à elle, peut-être interprétée comme une méthode permettant d'appliquer un traitement préférentiel à un certain type de trafic et d'augmenter ainsi le niveau de qualité des critères caractéristiques de cette catégorie de trafic [9].

2.5 Les modèles de qualité de service

Le Best effort n'est pas vraiment un modèle de la QoS, car c'est ce que l'Internet offre à tout le monde par défaut. Il n'y a pas de garantie pour les paquets, car ils ont tous la même priorité, la même stratégie de traitement « premier arrivé premier servi », pas de réservation de bande passante et enfin des pertes de paquets dus à la congestion. Bien que la politique Best Effort ait fait de l'Internet ce qu'il est actuellement, c'est aussi un handicap. En effet, tous les services ne nécessitent pas les mêmes ressources sur Internet, ainsi Best effort est mauvais pour les applications temps réel et de multimédia.

La QoS sert à garantir qu'une application donnée aura à sa disposition, les ressources nécessaires pour fonctionner à travers un réseau. Pour que cette garantie soit effective, la QoS se base sur plusieurs modèles afin d'assurer et garantir cette disponibilité du réseau [45, 62, 36, 68, 31].

2.5.1 IntServ/RSVP

Le modèle « Integrated Services » repose sur deux principes fondamentaux : (a) le réseau doit être contrôlé et soumis aux mécanismes de contrôle d'admission ; (b) des mécanismes de réservation de ressources sont nécessaires pour fournir des services différenciés. Sachant que le délai de transit par paquet est la caractéristique principale de la QoS, le modèle distingue deux types de services, des services qui tolèrent une variation du délai de transit, et ceux qui ne tolèrent pas de variation:

- **Le service garanti (Guaranteed Service)** qui garantit la bande passante et un délai d'acheminement limité. Ce service est destiné pour les applications audio, vidéo, et temps réel.
- **Le service contrôlé (Controlled Service)** offre un service proche de l'Internet peu chargé. Ce type de service a un intérêt si l'on considère que des outils de vidéoconférence offrent une bonne qualité quand le réseau est peu utilisé, mais la qualité se dégrade très vite si le réseau est saturé.

Le modèle IntServ spécifie des modules de décisions (Admission Control, Policy Control), de gestions de flux (Packet Classifier, Packet Scheduler) et un protocole de

signalisation RSVP (ReSerVation Protocol) pour allouer dynamiquement de la bande passante aux applications orientées réseaux. Cependant, Intserv souffre du problème de passage à l'échelle. Les ressources nécessaires pour exécuter RSVP sur les routeurs augmentent avec le nombre de réservations, résultant ainsi en de mauvaises performances de ceux-ci ; en effet, un routeur doit maintenir l'état de chaque flux (qui nécessite une réservation) qui le traverse.

2.5.2 DiffServ

DiffServ [27, 25, 18, 52] est un modèle d'architecture qui permet de fournir une discrimination de service en définissant différents types ou niveaux de service pour le trafic réseau. L'idée est qu'il est utile d'avoir des services différents à grande échelle sans avoir à maintenir des flux et une signalisation pour chaque flux dans l'Internet. DiffServ se base sur le champ TOS (Type Of Service) d'IPv4 que l'IETF3 a redéfini en champ DSCP4 pour effectuer la classification.

Les routeurs de bord (Edge Router) sont responsables de la classification des paquets entrant sur le réseau. Ils traitent les paquets en fonction de la classe codée dans le champ DSCP selon un comportement spécifique: le PHB (Per Hop Behavior). En aucun cas, ils ne traiteront différemment des paquets de sources différentes. Les routeurs du cœur (Core Router) du réseau appliquent le niveau de service approprié suivant la valeur du champ DSCP. On distingue 2 classes ou comportements de routeurs (PHB) :

- **Expedited Forwarding (EF)** ou premium service; il a pour but de garantir une bande passante avec des taux de perte, de délai et de gigue faible.
- **Assured Forwarding** regroupant plusieurs PHB garantissant un acheminement de paquets IP avec une haute probabilité; Cette famille de PHB est scindée en 4 classes garantissant de fournir une bande passante et un délai minimum, chaque classe comprenant 3 niveaux de priorité (Drop Precedence).

³ Internet Engineering Task Force

⁴ Differentiated Services Code Point

2.5.3 MPLS/ l'ingénierie de trafic

L'ingénierie de trafic peut être un autre moyen d'améliorer le support de la qualité du service, en dimensionnant plus finement les ressources dans le réseau de l'opérateur. Elle peut être liée au routage par qualité de service ou à l'utilisation de la différenciation de services. Il s'agit alors d'adapter le comportement du réseau à chaque classe de service en modifiant le routage et/ou le traitement dans les équipements intermédiaires.

MPLS (Multi Protocol Label Switching) permet de simplifier l'administration du réseau en ajoutant de nouvelles fonctionnalités particulièrement intéressantes pour la gestion de la qualité de service. Dans le même esprit que l'architecture DiffServ, MPLS permet de réduire le coût des traitements associés à l'expédition des paquets en les reportant à la périphérie du réseau et en réduisant la fréquence. Il apporte aussi un mécanisme de routage hiérarchique efficace qui consiste à établir des tunnels entre les nœuds du réseau.

Le principe de MPLS est d'attribuer un label (une étiquette) à chaque paquet lorsqu'il entre dans le réseau. Ce label est attribué en fonction de la classe d'expédition (FEC : Forwarding Equivalent Class) à laquelle appartient le paquet. La définition de ces classes dépend de l'opérateur du réseau. Généralement, une classe correspond à une entrée de la table de routage ou à un routeur de sortie du réseau, mais elle peut aussi prendre en compte la classe de service DiffServ (DiffServ avec MPLS). Une fois à l'intérieur du réseau les paquets ne sont plus traités qu'en fonction du label qui leur a été associé et l'en-tête IP n'est plus consulté. Chaque routeur utilise ce label pour déterminer le prochain routeur vers la destination.

Le protocole de distribution des labels (p.e., Label Distribution Protocol [40]) se charge de la signalisation implicite pour établir des chemins (Label Switched Paths) entre sources et destinations. Les tables de routage ne servent donc plus à relayer les paquets, mais à construire les chemins. La commutation de label est beaucoup plus efficace que le routage IP classique.

2.6 Les mesures de QoS

Ces mesures reviennent à tester les performances du réseau et son comportement avec des routeurs qui implémentent cette QoS. Il est nécessaire d'avoir des outils de mesure capable d'évaluer avec une certaine précision les performances. Les critères qui sont généralement mesurés sont la disponibilité, le délai, la gigue, et le taux de pertes de paquets. Deux types de méthodes ont été définis : la méthode active qui consiste à injecter du trafic dans le réseau de manière contrôlée et analyser les paquets retournés (Ping, traceroute...). et la méthode passive qui consiste à observer et analyser les paquets reçus sur un système terminal.

2.7 Conclusion

Une synthèse globale et finale de la Qualité de service dans l'Internet n'est pas possible avec les multiples évolutions que connaît l'Internet. Nous noterons l'explosion des réseaux sans fil et des réseaux ad hoc qui connaissent actuellement un engouement de la part des universitaires et des industriels.

Plusieurs alternatives existent aujourd'hui. Parallèlement au modèle de réservation de ressources et de services intégrés qui pose des problèmes de déploiement à grande échelle, il y a actuellement une orientation vers les services différenciés avec MPLS. Aujourd'hui, MPLS offre des techniques de « Label Switching » qui sont utilisées dans un contexte d'assurance de la qualité de service dans le réseau. Mais, en attendant son déploiement à large échelle il y a toujours la différenciation de services qui a prouvé son efficacité.

Chapitre 3

Services Web

«Aujourd'hui, l'usage principal du World Wide Web est l'accès interactif à des documents et à des applications. Dans la plupart des cas, un tel accès se fait par des utilisateurs humains, généralement via des navigateurs Web, des lecteurs audio ou d'autres systèmes frontaux interactifs. Le Web peut gagner considérablement en puissance et en portée s'il bénéficie d'une extension pour pouvoir supporter une communication inter-applications, une communication entre deux programmes. »

- La charte du groupe de travail XML Protocol du W3C.

3.1 Introduction

La puissance des services Web est essentiellement basée sur le modèle de développement des applications distribuées et l'avènement des services Web n'est qu'un résultat attendu de la transformation de l'informatique distribuée. Les services Web permettent à n'importe quelle entreprise ou particulier de publier leurs services en ligne et les mettre à la disposition d'utilisateurs de toutes catégories. L'utilisation de ces services par ces différents clients est simple et indépendante de la plateforme à partir de laquelle ils les invoquent.

On retient qu'il y a deux éléments importants dans cette architecture, le réseau et l'application. Pour le réseau, l'évolution s'est développée du mainframe centralisé vers le réseau local et enfin vers l'Internet qui représente le plus grand réseau reliant tous les ordinateurs qui y sont connectés. Pour l'application, l'évolution a commencé par l'établissement de communication inter application (par exemple CORBA [70]). Avec les réseaux locaux on a assisté au lancement de IIOP (Internet Inter ORB Protocol) d'OMG, DCOM (Distributed COM de Microsoft) et RMI (Remote Invocation Method de Sun). Avec l'émergence d'Internet et du Web, le réseau est devenu très étendu et largement distribué, ce qui a créé beaucoup de difficultés pour utiliser les protocoles déjà cités et qui ont été conçus initialement pour des réseaux locaux d'entreprise. Ces difficultés et d'autres ont amené la communauté de l'informatique distribuée et du Web à travailler sur un modèle d'architecture d'application Web orientée service.

3.2 SOA : Architecture orientée service

3.2.1 SOA : caractéristiques

L'architecture orientée service (SOA) est une forme d'architecture de systèmes distribués qui est typiquement caractérisée par les propriétés suivantes [71] :

- Vue logique : le service est une vue abstraite et logique des programmes réels, des bases de données, des processus, etc., il est défini en termes de ce qu'il fait.
- Orientation de message : le service est formellement défini en termes de messages échangés entre les agents du fournisseur et les agents du demandeur.
- Granularité : les services tendent vers l'utilisation d'un petit nombre d'opérations avec des messages relativement grands et complexes.
- Orientation réseau : les services tendent à être utilisés via un réseau, bien que ce ne soit pas une condition absolue.
- Indépendance de la plateforme : les messages sont envoyés dans un format standard via les interfaces réseau.

En général une SOA et les services Web sont plus appropriés pour (a) des applications utilisant l'Internet où la fiabilité et la vitesse ne peuvent pas être garanties, le déploiement est impossible à contrôler et dont les composants (du système distribué) doivent fonctionner sur différentes plateformes et (b) des applications nouvelles ou existantes qui doivent être publiées pour être invoquées via le réseau et peuvent être enveloppées comme service Web.

3.2.2 SOA : définition

L'architecture SOA définit un modèle d'interaction applicative mettant en œuvre des connexions en couplage lâche entre divers composants logiciels (ou agents). Ici, on entend par « service » une action exécutée par un composant « fournisseur » à l'attention d'un composant « consommateur » ; basé éventuellement sur un autre système et utilisant le réseau (Internet/Intranet) comme moyen de communication (liaison) pour transporter les données entrantes et sortantes des deux parties.

3.2.3 Services Web : les technologies utilisées

Le concept des Services Web s'articule actuellement autour des technologies suivantes [73, 60, 48] :

- **HTTP** (HyperText Transfer Protocol), littéralement « protocole de transfert hypertexte », est un protocole de communication informatique client – serveur développé pour le World Wide Web. Il est utilisé pour transférer les documents (document HTML, image, feuille de style, etc.) entre le serveur HTTP et le navigateur Web lorsqu'un visiteur consulte un site Web.
- **XML** (Extensible Markup Language ou langage de balisage extensible) est un standard du World Wide Web Consortium qui sert de base pour créer des langages balisés spécialisés ; c'est un « méta langage ». Il est suffisamment général pour que les langages basés sur XML, appelés aussi dialectes XML, puissent être utilisés pour décrire toutes sortes de données et de textes.
- **SOAP** (Simple Object Access Protocol) est un protocole d'échange inter-applications indépendant de toute plate-forme, basé sur le langage XML. Un appel de service SOAP est un flux de données ASCII respectant un format XML et transporté par le protocole HTTP.
- **WSDL** (Web Services Description Language) donne la description en XML des Services Web en précisant les méthodes pouvant être invoquées, leur signature et le point d'accès (URL, port, etc). C'est, en quelque sorte, l'équivalent du langage IDL pour la programmation distribuée CORBA.
- **UDDI** (Universal Description, Discovery and Integration) normalise une solution d'annuaire distribué de services Web, permettant à la fois la publication et l'exploration. UDDI se comporte lui-même comme un service Web dont les méthodes sont appelées via le protocole SOAP.

3.2.4 Services Web : définition

Un Service Web est n'importe quel service qui est disponible sur l'Internet utilisant les technologies précédentes, et qui est indépendant des systèmes d'exploitation (plate-forme) ou des langages de programmation. C'est un composant implémenté dans n'importe quel langage, déployé sur n'importe quelle plate-forme. Il doit pouvoir être découvert et invoqué dynamiquement par d'autres services.

3.3 Architecture services Web

On peut examiner l'architecture des services Web de deux manières. En premier, on examine les différents rôles de chaque acteur dans cette technologie ; en second, on examine les différentes couches qui composent la pile du protocole des services Web.

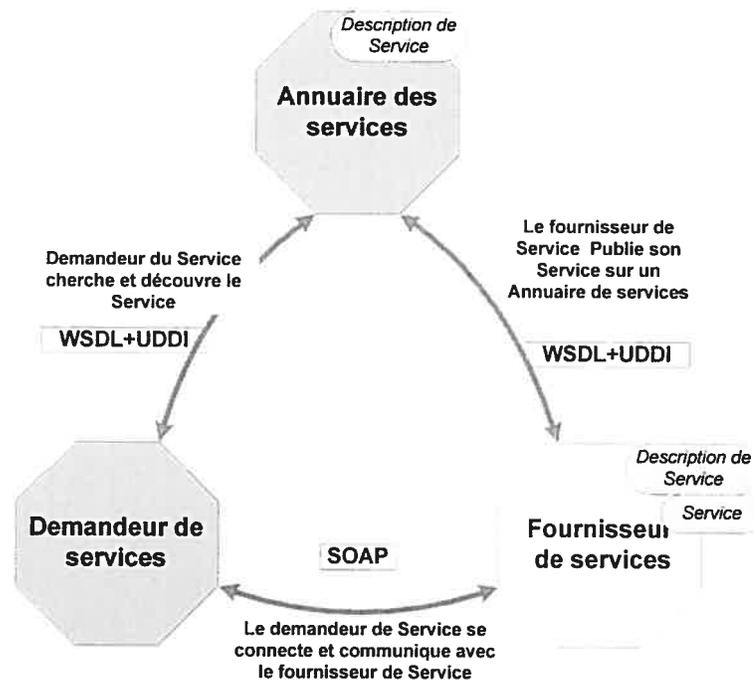


Figure 3.1 : Le Modèle d'architecture orientée services.

3.3.1 Rôles et opérations

Les rôles majeurs dans l'architecture des services Web sont (figure 3.1):

- Le fournisseur du service : il implémente le service et le publie en le rendant disponible sur l'Internet.
- Le demandeur de service: c'est n'importe quel consommateur (utilisateur, client) de service Web. Le demandeur utilise un service Web existant via une connexion réseau en envoyant une requête HTTP.
- L'annuaire de services : c'est un annuaire ou dépôt centralisé de services Web. L'annuaire fournit un serveur central où les fournisseurs peuvent publier de nouveaux services ou découvrir ceux qui sont déjà disponibles.

Les opérations spécifiques qui permettent aux trois acteurs de l'architecture SOA d'interagir entre eux sont détaillées ci-dessous [47]:

- Publier: pour être accessible, une description du service doit être publiée pour que le demandeur de service puisse la trouver. Cette publication consiste à donner le lien URI (Uniform Resource Identifier) à partir duquel le service peut être invoqué.
- Trouver: le demandeur de service recherche une description de service directement du fournisseur du service ou interroge l'annuaire.
- Relier: le demandeur du service Web entre en interaction avec le service au moment de l'exécution. Cette opération est réalisée en utilisant les informations de la description du service Web (WSDL) pour le localiser, le contacter et l'invoquer.

3.3.3 La Pile de protocole des services Web

Decouverte	UDDI
Description	WSDL
Messagerie XML	XML-RPC, SOAP, XML
Transport	HTTP, SMTP, FTP, BEEP

Figure 3.2 : La pile de protocoles des services Web

La pile de protocoles des services Web est en continuelle évolution, la figure 3.2 récapitule d'une manière générale ses couches [14]. Elle a quatre couches principales :

- **Service de transport** : cette couche est responsable de transporter les messages entre les applications. Actuellement, elle inclut le protocole de transfert hypertexte (HTTP), le Simple Mail Transfer Protocol (SMTP), le File Transfer Protocol (FTP), et aussi le Blocks Extensible Exchange Protocol (BEEP).
- **Messagerie** : Cette couche est responsable de coder les messages dans le format XML de sorte qu'ils puissent être lus à l'autre extrémité. Actuellement, elle inclut XML-RPC (Remote Procedure Call) et SOAP.
- **Service de description** : Cette couche est responsable de la description de l'interface publique du service Web. Actuellement, la description de service Web est faite en utilisant le langage de description de service Web (WSDL).
- **Service de découverte** : Cette couche est responsable de centraliser les services dans un annuaire commun, et fournir des fonctionnalités de publication et recherche simple d'utilisation. Actuellement, le service de découverte est fourni par les annuaires (UDDI).

3.4 Service Web et qualité de service

3.4.1 Introduction

La qualité de service dans les services Web (QdS-SW) concerne les aspects de qualité du service des services Web fournis aux utilisateurs. Dans une SOA, les fournisseurs de service et les utilisateurs (consommateurs) de ses services devraient pouvoir définir les paramètres et aspects de la qualité de service ; qui auront un impact assez important dans la publication, la découverte et l'utilisation de ces services.

Du côté des clients, il est important qu'ils soient informés du statut de la QdS avant son utilisation, quand le service Web est déjà ciblé. Pour les fournisseurs, c'est la base des accords de niveau de service (SLA).

3.4.2 SLA : accords de niveau de service (Service Level Agreement)

C'est un contrat définissant les engagements du fournisseur quant à la qualité de sa prestation et les pénalités engagées en cas de manquement. Cette qualité doit être mesurée selon des critères objectifs acceptés par les deux parties. Par exemple le temps de rétablissement du service en cas d'incident [29, 17].

3.4.3 Courtier de QdS-SW

Le courtier de QdS-SW (WS-QoS Broker [48]) joue le rôle d'un annuaire de services Web ; d'une certaine manière, on peut le comparer à un annuaire UDDI, mais plus orienté vers la qualité de service. Il guide les clients dans le choix des services Web, en se basant sur un ensemble de paramètres de QdS. En outre, il permet de vérifier, certifier, confirmer et contrôler la QdS dynamiquement.

Le courtier de QdS-SW est une collection de composants, exécutant des fonctionnalités dédiées à la QdS pour la sélection de services Web, conformes aux critères demandés par les clients et des paramètres de QdS publiés par les fournisseurs. Il fournit à ses clients de l'information qui les aide dans leurs décisions dans le procédé de sélection de services Web. Ce procédé d'assistance est basé sur les résultats de vérification, certification et de confirmation des services Web cible et leurs QoS.

3.4.4 La QoS des services Web

Selon le W3C, les conditions de la QoS pour les services Web incluent la performance, la fiabilité, l'extensibilité, la capacité, la robustesse, la gestion d'exception, l'exactitude, l'intégrité, l'accessibilité, la disponibilité, l'interopérabilité, la sécurité, et les exigences de la qualité de service réseau [72].

Pour réaliser la QoS désirée pour les services Web, les mécanismes de QoS fonctionnant au niveau application doivent opérer avec les mécanismes de la QoS fonctionnant au niveau du réseau; ces derniers sont indépendants de l'application (par exemple, RSVP, DiffServ, MPLS, etc.). En particulier, les paramètres de QoS des services Web doivent être convertis en paramètres correspondants à la QoS réseau. Les paramètres de base de QoS au niveau du réseau incluent le délai, la gigue, et la perte de paquet.

Un service Web est caractérisé par plusieurs paramètres de qualité de service. C'est un composant logiciel invoqué à partir d'une application cliente, puis exécuté sur son serveur hôte et ses données d'entrée ou de sortie sont transportées via le réseau Internet. Ce cheminement nous informe sur les différents paramètres impliqués dans la QoS du service Web. Si on résume ce processus par une simple formule :

$$\text{QoS-SW} = \text{QoS Serveur} + \text{QoS Service Web} + \text{QoS Client} + \text{QoS Réseau.}$$

a) Paramètres de QoS au niveau serveur (fournisseur)

La QoS au niveau du fournisseur est définie dans la description de son service ; par exemple, voir le document WSDL dans la figure 3.3 [49]. Elle peut être identifiée par le nombre d'invocations simultanées, temps de réponse, la capacité de traitement et les performances du serveur qui exécute le service Web (mémoire vive, processeur (s), disque(s), bande passante).

b) Paramètres de QoS au niveau service Web (conception et implémentation)

La QoS au niveau du service Web est définie en tant que QoS résultante de ses fonctionnalités conceptuelles et performances au niveau de son implémentation ; soit en ce qui concerne les algorithmes utilisés en fonction du domaine d'application visé, soit

dans la technique d'implémentation de telle manière à rendre son exécution (à partir d'une invocation) saine de tout « bogue » ou erreur de « exécution ». Entre autres, ça concerne aussi le temps d'exécution qui est partagé avec le serveur qui accueille l'exécution.

c) Paramètres de QoS au niveau client

Les paramètres, au niveau serveur, s'appliquent aussi au niveau client en plus d'autres qui sont spécifiques au type du service Web invoqué et requis pour le bon déroulement du processus d'exécution. Par exemple, si le client veut regarder une vidéo, il aura besoin d'un minimum de capacités graphiques supportant l'affichage de vidéo couleur avec une résolution minimale ; si le client veut télécharger un gros document, il devrait dans ce cas avoir un espace disque suffisant pour accueillir le fichier.

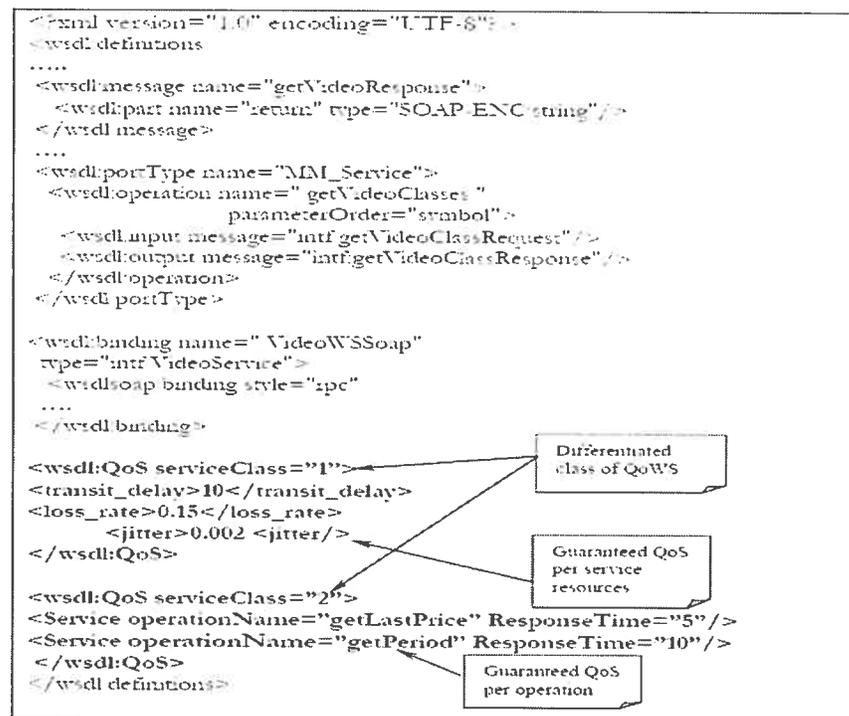


Figure 3.3 : Exemple de document WSDL contenant les descriptions de QoS.

d) Paramètres de QoS du réseau de communication.

Les clients veulent toujours définir les paramètres de QoS d'une perspective propre à eux. Ceci ne serait pas un problème si on était dans un système réparti traditionnel, où le réseau et d'autres composants d'environnement système sont sous le contrôle de la même compagnie. Mais beaucoup d'utilisateurs qui viennent de différents milieux professionnels ou résidentiels n'ont pas les garanties de QoS réseau pour leur accès aux services Web. Il faut une solution permettant de combiner/intégrer les propriétés de la QoS du service Web avec celles du réseau reliant le fournisseur et son client pour un support de QoS de bout en bout.

L'essentiel dans ce point, c'est que le réseau Internet/public actuel qui transporte la requête du client et la réponse du service Web, offre une QoS « meilleur effort » ou le meilleur qu'il peut offrir. Notre objectif est de changer le meilleur effort au meilleur disponible ; plus précisément les meilleurs services Web disponibles et ceci dépend de l'architecture et de la technologie réseau disponible entre le client et le service Web (p.e. routeurs DiffServ et contrôle d'admission [17]).

3.5 Services Web multimédia

Pour les services Web de multimédia (vidéo/audio streaming), les défis sont différents des autres types de services Web traditionnels car on peut offrir des qualités vidéo qui peuvent être médiocres en fonction des conditions de la QoS réseau disponible. La plupart des recherches sur la gestion de QoS ont été faites dans le contexte de fournir des documents multimédias. Des mécanismes de gestion de QoS ont été appliqués aux applications multimédias, dans le contexte de systèmes distribués [6]. La QoS d'une session multimédia dépend d'une combinaison de facteurs, s'étendant des caractéristiques des sources d'émission (par exemple, capacité, disponibilité du lien, débit) aux caractéristiques des chemins de réseau (par exemple, largeur de bande, taux de perte de paquets, multicast, etc.).

3.6 Outils de développement des services Web

Pour la conception et l'implémentation des services Web, plusieurs compagnies se sont mises à rude épreuve afin de fournir des environnements de développements intégrés complets. Ces environnements minimisent le temps d'apprentissage et les mènent directement à l'essentiel de leur projet. Les environnements de développement intégrés sont très complets, surtout ceux du domaine open source. Ils contiennent toute la panoplie : du serveur d'application, serveur Web, langage de programmation/compilateur et serveur de bases de données. Ceci n'est pas aussi accessible dans le cas des offres payantes comme Weblogic de BEA [7] et Websphere de IBM [24].

3.7 Service Web et QoS de bout en bout: revue de littérature

La recherche sur la QoS des services Web s'est plus concentrée sur des aspects d'ordre fonctionnel et d'interfaçage, c'est à dire, protocole SOAP, langage de description de services Web (WSDL) et d'annuaire UDDI. Récemment, les problèmes de QoS ont commencé à susciter plus d'attention dans la communauté de services Web.

La QoS n'est pas inconnue pour la communauté des systèmes distribués, mais dans les services Web il y a de nouveaux problèmes liés aux propriétés spécifiques des services Web comme leurs diversités et disponibilité sur internet.

Une large littérature existe sur la définition sémantique de la QoS des services Web et des contraintes de QoS [13, 4, 69, 5, 57, 69], aucun de ces travaux ne propose de solution pour fournir la QoS de bout en bout, et ceci, quand un service Web qui répond aux exigences de QoS de l'utilisateur est invoqué. Par exemple, Ali et al. [5] de l'université de Cardiff proposent UDDIe, comme nouvel annuaire pour les services Web en ajoutant à la norme d'UDDI certaines modifications pour supporter la qualité de service. UDDIe permet l'enregistrement de propriétés définies par le fournisseur qui publie son service Web ; ces propriétés sont propres à ce service Web. La recherche et la découverte de ce service Web seront basées sur ces propriétés. Ce travail intègre à UDDI des descriptions de QoS et des possibilités de faire des recherches multicritères. Cependant, quand un service Web, qui répond aux exigences de QoS d'utilisateur, est

choisi, il n'y a aucune garantie que le réseau soutiendra la QoS demandée. Par exemple, si la qualité audio publiée d'un service Web (par exemple, Music Player) est « qualité CD » et l'utilisateur exige une qualité audio « de qualité CD », le service Web sera choisi en réponse aux exigences de l'utilisateur ; cependant, l'utilisateur obtiendra cette qualité seulement si le réseau a assez de ressources disponibles pour fournir cette qualité.

Le travail de Tian et al. [51, 52] fait partie des rares travaux concernant le support de QoS de bout en bout pour les services Web. Ils présentent un concept d'intégration de la QoS dans les services Web. Il est basé sur un schéma XML pour la définition de QoS de services Web ; il inclut des mécanismes pour la sélection efficace des services Web en se basant sur la QoS, et il permet aussi le mappage dynamique de la QoS au moment de l'exécution et la disponibilité immédiate de l'information sur la QoS. Ils présentent un module, appelé QoS proxy, qui est située entre la couche transport et la couche de service Web. Son rôle est de marquer les paquets sortants dans le cas d'un réseau Diffserv [8]. Il peut être également employé avec d'autres technologies de gestion de réseau, tel qu'ATM et UMTS [41]. L'inconvénient principal de cette approche est qu'elle exige des changements au niveau de la pile du protocole dans toutes les entités impliquées (c.-à-d., les utilisateurs et les fournisseurs). En plus, il y a des problèmes de sécurité ; en effet, des utilisateurs malveillants peuvent marquer leurs paquets sortants pour avoir le meilleur service disponible (par exemple : Expedited Forwarding dans un réseau DiffServ [68]).

3.8 Conclusion

Nous avons parcouru dans ce chapitre les principaux éléments de bases de l'architecture SOA, ainsi que les technologies des services Web. Le modèle de programmation des services Web laisse présager une connectivité d'application à application universelle, réduisant les coûts et le temps de développement.

Nous avons défini et présenté la problématique du support de QoS, de bout en bout, dans les services Web. Après avoir parcouru les recherches majeures dans ce domaine, nous avons déterminé qu'il y a très peu de travaux qui ont été fait et les quelques solutions, proposées dans la littérature, sont préliminaires avec un certain nombre de

limitations (QdS seulement du côté serveur, changer la pile des protocoles, etc.). En effet, le problème de fournir la QdS, de bout en bout, pour les services Web est encore ouvert et beaucoup de travail reste à faire pour atteindre cet objectif.

Chapitre 4

Gestionnaire de Ressources Réseau (NRM)

Avec la croissance continue des services Web, la qualité de service (QoS) retient l'attention de la communauté des services Web. Les fournisseurs doivent employer des architectures qui supportent la QoS de bout en bout. La QoS fournie aux clients est fortement affectée par l'exécution du service Web lui-même, par le serveur hôte (p.e., serveur d'application qui exécute les services Web) et par le réseau de communication (p.e., Internet). Ainsi, même si un service Web et le serveur, qui l'exécute, fournissent une QoS acceptable, ils en demeurent toujours, que la disponibilité des ressources réseau suffisantes soit nécessaire pour assurer la QoS de bout en bout. Dans ce chapitre, nous proposons une solution pour le support de QoS de bout en bout pour les services Web. Cette solution se base sur un gestionnaire de ressources réseau que nous avons appelé NRM. Dans ce qui suit, nous présentons les détails de conception, implémentation et évaluation de NRM.

4.1. Introduction

Des solutions innovatrices sont attendues afin de pouvoir encapsuler les propriétés de la QoS d'un service Web avec les propriétés du réseau du fournisseur, du réseau du client et du réseau les reliant tous les deux [19].

Le gestionnaire de ressources réseau (NRM) propose des mécanismes, et des outils de mesure et d'informations sur le réseau de liaison entre le client et le fournisseur. L'architecture utilisée vise à assurer la QoS de bout en bout à deux niveaux (niveau serveur et niveau réseau). Dans ce but, elle utilise un courtier de QoS-SW [47, 48, 49], pour assurer les conditions de la QoS et sa surveillance au niveau serveur, et un gestionnaire de ressources réseau NRM pour garantir la QoS au niveau réseau. Les deux composants coopèrent pour assurer la QoS de bout en bout entre les fournisseurs et leurs clients.

4.2. Motivations

Les fournisseurs de contenu multimédia utilisant les technologies des services Web, peuvent offrir différents niveaux de qualité pour le même service disponible ; par

exemple, le même film peut-être disponible en 3 classes de qualité, de l'excellente à la basse résolution avec des prix différents. L'utilisateur peut choisir, généralement à travers une interface utilisateur, n'importe quelle qualité parmi les qualités disponibles ; cependant, le support de la qualité choisie va dépendre de trois éléments importants. Chaque élément jouera un rôle principal pour contribuer à avoir une qualité de service qui satisfait le choix de l'utilisateur. :

- L'état du client (utilisateur ou consommateur): Il est défini par la capacité et les caractéristiques de la machine client en plus de la capacité de la liaison qui connecte la machine au réseau. En termes de capacité/caractéristiques de la machine, nous pouvons mentionner, par exemple, le type de codecs [10] audio et vidéo et la capacité de la carte graphique de l'ordinateur (p.e., un minimum de résolution est nécessaire pour être conforme avec la classe la moins chère et la moins exigeante en termes de qualité d'image et de son). Pour la connexion réseau, le type d'accès réseau (p.e., câble, DSL, modem 56K) constitue l'élément le plus important. En effet, pour supporter une certaine qualité vidéo, l'utilisation d'un modem 56K sera inadéquate ; dans ce cas, un accès câble où DSL est nécessaire.
- L'état du serveur (fournisseur) : il est plus important de noter ici que les performances du serveur restent spécifiques à ses capacités de traitement et d'exécution. La qualité de service du fournisseur est liée à sa capacité de servir un nombre maximum d'utilisateurs simultanés pour chaque qualité vidéo correspondante. Cela se traduit par une grande mémoire vive, des processeurs puissants, etc. ; donc des serveurs vidéo assez puissants sont requis pour supporter les demandes des clients, surtout aux heures de pointe (soirées) quand le nombre de requêtes sur le serveur devient important. En plus de la capacité du serveur, la liaison réseau, qui connecte le serveur ou la ferme de serveurs au réseau, est aussi importante. Pour supporter un nombre élevé d'utilisateurs, une connexion réseau à haut débit (p.e., OC3) est requise.
- L'état du réseau (transport) : si le client possède les capacités matérielles, logicielles et de connectivité qui lui permettent de commander et regarder une vidéo et si le

fournisseur de service vidéo peut fournir la qualité demandée par ses clients, on se retrouve donc dans une situation de satisfaction mutuelle et le démarrage du service peut aller de l'avant. Cependant, pour supporter la qualité de service requise de bout en bout, le réseau qui connecte le client au serveur doit avoir les ressources nécessaires disponibles pour fournir sa part de QoS ; sinon, le client n'a pas la garantie de recevoir la vidéo avec la QoS désirée. Dans le cas où le réseau utilisé est l'Internet, qui n'offre que du "Best Effort", la garantie de la QoS d'un tel service peut facilement frôler le zéro.

Notre objectif est de proposer une solution qui permet de garantir, une QoS de bout en bout ; plus spécifiquement, nous nous intéressons à l'élément réseau qui n'a pas été largement étudié/considéré par la communauté des services Web ; l'élément serveur a été largement investigué par cette communauté.

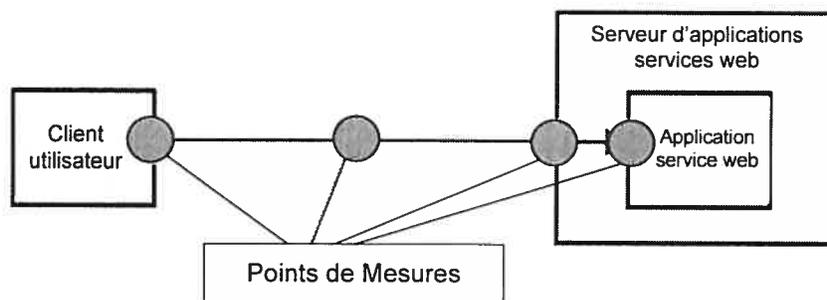


Figure 4.1 : Différents points de mesure des paramètres de QoS réseau.

Dans la figure 4.1, nous remarquons qu'en plusieurs points du chemin de bout en bout [19] qui relie le client au serveur, la QoS ne peut être la même. Soit au niveau du client, du serveur ou entre les deux.

4.3 Classification des services Web

L'annuaire UDDI [39] offre un mécanisme standard de classification, catégorisation et gestion des services Web, de sorte qu'ils puissent être découverts et consommés. Un

des buts principaux de conception de l'annuaire UDDI est la capacité de catégoriser chaque service Web. Ainsi, il est très important que les annuaires UDDI contiennent un ensemble de catégories sur lesquelles une recherche universelle peut-être faite. Comme les annuaires UDDI, NRM offre aussi une classification des services Web. Il est lui même un service Web qui incarne toutes les caractéristiques de cette architecture.

Avec NRM nous proposons une solution de classification des services Web en fonction d'informations reliées plus à la nature du service Web lui-même, au genre de données qu'il manipule et envoie sur le réseau et surtout des exigences en ressources réseau que le service Web requiert. D'où une classification orientée réseau. Cette classification est utile pour mieux servir les exigences des clients en QoS désirée. Elle permet de comparer et vérifier ces exigences avec celles qui sont publiées et découvertes par le Courtier de QoS-SW [47, 49]. Entre autres, les tâches de sélection et de vérification du Courtier de QoS-SW bénéficient de cette classification pour optimiser le processus de sélection. Dans ce qui suit, nous présentons une classification des services Web, qui est plus orientée vers les exigences des services Web en ce qui concerne le réseau.

4.3.1 Les types de services Web

Il existe différents types de services Web. Chaque service est spécialisé dans un domaine spécifique. Le type dans lequel un service Web est affecté, dépend de plusieurs facteurs y compris le domaine d'activité :

- La nature des informations que le service Web reçoit en entrée, qu'il produit en sortie vers le client et qu'il manipule pendant le traitement s'il y a lieu.
- La complexité du calcul et du traitement exécuté par le service Web.
- La quantité de données manipulées (envoyées et reçues).
- La nécessité d'une QoS garantie de bout en bout, qui peut être réservée pendant un certain temps immédiat ou dans le futur, entre le service Web et le client.

4.3.2 Classification orientée ressource réseau

Les clients et les utilisateurs des Services Web recherchent des services qui satisfont leurs besoins avec une QoS appropriée. Le calcul du temps de réponse d'un service

Web n'est pas assez informatif sur sa QoS. Il est plus spécifique à l'invocation du service Web et ne fournit qu'une information pour la QoS du serveur qui héberge ce service Web. Entre autres, la prochaine étape d'une transaction entre le service Web et ses clients dépendra plus de l'état du réseau que du temps de réponse.

Nous présentons une classification simple des services Web dans le contexte de leurs exigences envers les ressources de réseau. Le but de cette classification est de doter NRM de la capacité de classifier tous les services Web dans des catégories de types spécifiques à leurs besoins en ressources réseau.

Id	Types de Services Web	Attributs de la QoS	Importance du réseau et de la Réserve	Exemples de Services Web possibles
1	Critique	Bande passante Temps de réponse Gigue Latence Sécurité	100 %	Service Web relié à des tâches critiques
2	Traitement de données	Disponibilité Sécurité	80 %	Calcul intensif offert par des super ordinateurs
3	Streaming, Multimedia	Temps de réponse Bande passante Capacité Serveur	80 %	Services Web vidéo, audio sur demande.
4	Transfert de Données	Bande passante	50 %	
5	En temps réel	Bande passante Temps de réponse Gigue Latence	100 %	Contrôle et prise à distance
n				

Table 4.1 : Exemple de types de services Web.

Il y a des services Web qui ne sont pas exigeants en ressources réseau et ainsi n'ont pas besoin du service NRM. Par contre, il y a d'autres comme les services Web vidéo, qui exigent une bande passante assez importante, un débit constant et stable tout le long de la transaction. Cette classification est fournie par le service Web NRM.

Dans la table 4.1, nous présentons quelques groupes identifiés de services Web qui n'ont pas les mêmes besoins en termes de ressources réseau.

La table de classification est stockée dans la base de données NRM et peut être utilisée et interrogée par le courtier de QoS-SW et par les fournisseurs de services Web. Sa mise à jour doit vérifier les conditions de QoS offertes par chaque classe et service Web qui s'y ajoute afin de garder une liste fiable et disponible lors des opérations de vérification.

4.4 Description de NRM

Notre architecture n'exige aucun changement à la pile du protocole des entités impliquées. En effet, elle est basée sur un Gestionnaire de Ressources Réseau, qui doit s'assurer que le réseau (entre le service Web choisi et l'utilisateur) prend en charge sa part de la QoS exigée de bout en bout. NRM est conçu et implémenté comme un service Web, il est publié et invoqué de la même manière que les autres services Web. Il est appelé seulement quand le fournisseur de service Web fournit sa partie de la QoS exigée (par exemple, en termes de ressources de traitement du serveur hôte qui exécute le service Web). Ainsi, notre architecture permet à certaines approches (par exemple, UDDIe [5]) de supporter la QoS de bout à bout. Par exemple, quand l'Annuaire UDDIe renvoie un ou plusieurs services Web qui répondent aux exigences relatives à la QoS d'un utilisateur, NRM sera employé pour identifier le service Web qui répondra aux exigences de QoS de l'utilisateur de bout en bout.

NRM est un élément fondamental qui s'ajoute à l'architecture du Courtier de QoS-SW [47, 51, 52]. Il est impliqué dans presque toutes les tâches qu'entreprend le Courtier de QoS-SW et il jouera un rôle principal dans l'architecture SOA en fournissant des garanties de QoS réseau de bout en bout. Il effectue un certain nombre d'opérations principales qui sont nécessaires aux Courtier de QoS-SW.

4.5 Principales fonctionnalités de NRM

La fonction principale de NRM est de vérifier les disponibilités et les capacités d'assurance et de garantie de la QoS des chemins réseau, entre l'hôte du service Web et la machine du client. Pour réaliser cette fonction, NRM emploie un certain nombre de mécanismes (modules). Si le réseau sous-jacent supporte la QoS, tel que les services intégrés (IntServ [8]) ou services différenciés (DiffServ [58]), alors NRM emploie ce

support pour soutenir et offrir la QoS. Par exemple, dans le cas IntServ, NRM emploie le protocole RSVP [8] (protocole de réservation de ressource) pour faire la réservation nécessaire des ressources pour satisfaire la QoS exigée. Dans le cas d'un réseau utilisant la technologie DiffServ, NRM marque les paquets sortants selon la QoS exigée (par exemple : marquage EF (Expedited Forwarding) pour supporter la vidéo et l'audio) pour fournir une différenciation de services ou encore le cas échéant NRM peut utiliser les services d'un courtier de bande passante [18], pour fournir, si possible, la QoS exigée.

Dans les cas où aucun de ces mécanismes n'est offert par le réseau, NRM utilise des techniques de mesure pour estimer l'état du réseau de bout en bout ; il emploie par exemple des techniques de sondages pour mesurer les principaux attributs de la QoS réseau tels que le délai et le taux de perte de paquets. Ces mesures permettent d'avoir une idée sur l'état actuel du chemin réseau.

4.6 Les taches de NRM

NRM a la particularité de gérer les ressources réseau dont il a le contrôle en fonction des contrats et des accords SLA [57] qui s'y appliquent. Par exemple, il peut se réserver un canal de communication en marquant ses propres paquets, dans le cas où la technologie DiffServ existe, afin d'exécuter ses taches d'une manière fluide.

NRM regroupe plusieurs fonctions sous forme de services Web qu'il offre à ses clients. Son client principal est le courtier de QoS-SW. Il invoque NRM pour vérifier un service Web. Les fournisseurs de services Web l'invoquent quand ils publient ou mettent à jour leurs services. Toutes ces situations nécessitent une décision afin de valider les paramètres de QoS que le client exige dans sa requête.

Les tâches que NRM exécute lors de son invocation, nous recensons les taches suivantes:

- **Support des réseaux Diffserv et IntServ:** NRM utilise les capacités fournis par les technologies disponibles des réseaux actuels. Si c'est un réseau Diffserv, NRM utilise le système de marquage des paquets qui transitent du fournisseur vers le client et ainsi garantir un traitement des paquets, par le réseau, en accordance avec la QoS demandée. Si c'est un réseau IntServ, NRM utilise le

protocole RSVP afin de réserver les ressources nécessaires sur le chemin réseau pour supporter la QoS demandée.

- **Réserver des ressources** : NRM réserve les ressources réseau en réponse à une requête ; cette fonction est traitée en détails dans le chapitre 6. La réservation de ressources réseau implique une certaine disponibilité d'infrastructures capables de fournir de l'information sur la bande passante disponible entre tout point du réseau.
- **Mesurer la bande passante, le délai et la gigue de bout en bout** : obtenir des valeurs de ces attributs selon les requêtes émises par le courtier de QoS-SW. Les données en entrée sont les adresses IP (Internet Protocol) des deux bouts du lien/chemin réseau à mesurer. À noter que les mesures retournées seront spécifiques à des chemins, qui peuvent être différents, pour une série de mesures identifiées par la date, l'heure et la durée des mesures. Ceci est très important, car les chemins sur Internet ne sont pas nécessairement les mêmes pour les mêmes tests.
- **Vérifier les ressources réseau** : Vérification de bout en bout des valeurs des attributs de QoS réseau. Cette tâche est exécutée quand un fournisseur veut faire une certification de son service Web publié par l'intermédiaire du Courtier de QoS-SW. Cette vérification pourra cibler un ou tous les paramètres réseau qui existent. NRM peut fournir ou implémenter de nouvelles fonctions de vérification de paramètres réseau ou utilisera si nécessaire des services Web tiers d'autres opérateurs réseau. Par exemple, le fournisseur donnera une liste de cibles (clients) pour les tester avec son service Web. Ces cibles seront choisies en fonction de la clientèle visée par le fournisseur dans son plan de déploiement.
- **Vérifier les services Web** : la vérification concerne l'accessibilité au service Web par des clients qui sont généralement non visés en priorité par le fournisseur, mais à qui l'information s'avérera utile pour connaître les difficultés qui peuvent surgir afin de les éviter par rapport au public visé. Les informations fournies par NRM sont toujours d'ordre de performance réseau, publiées en rapports/graphes qui s'étalent sur des durées assez importantes pour donner au

fournisseur des aspects détaillés de la vérification. Par exemple la nature du trafic entre deux-points pendant une semaine.

- **Mesures et Outils de mesures fournis par des Services Web tiers :** NRM fournit de l'information entre deux points du réseau, mais peut lui-même invoquer d'autres routines ou services Web tiers afin de récupérer de l'information relative à un client, un fournisseur ou un service Web qui est nécessaire dans le processus de vérification.
- **Statistiques et informations :** NRM offre des informations utiles afin de contrôler, surveiller et faire le suivi de l'exécution d'un service Web, et aussi vérifier si sa disponibilité change dans le temps. Ces informations sont utiles pour les fournisseurs et même les clients qui voudraient s'informer sur un service Web avant même de l'utiliser (avec l'autorisation du fournisseur afin de comparer les qualités publiées à celles mesurées).

D'autres informations spécifiques peuvent être offertes aux clients et fournisseurs ; par exemple leur position géographique. Cette dernière information fournira aux clients comme aux fournisseurs des informations précieuses sur l'origine des clients et ajoutera de l'information pertinente sur les performances du service Web dans un but d'amélioration et d'élargissement de l'offre.

4.7 Modules de NRM

NRM se compose de plusieurs modules qui constituent ensemble son service Web, il est nécessaire de noter que NRM est un service Web qui peut être composé, car dans le besoin il peut faire appel à d'autres services Web externes qui sont nécessaires à l'accomplissement d'une de ses tâches.

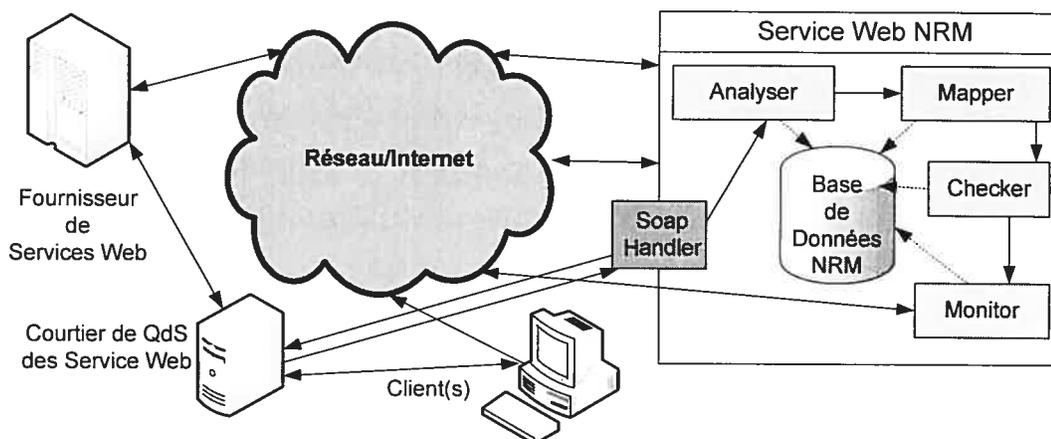


Figure 4.2 : Modules de NRM et leurs interactions avec le courtier de QoS-SW

4.7.1 Analyseur NRM (NRM-Analyser)

L'analyseur-NRM reçoit une demande/requête du Courtier de QoS-SW, identifie la catégorie du service Web et extrait les attributs de QoS fourni par le fournisseur et aussi ceux exigés par le client ; ces attributs seront comparés à ceux issus de la catégorie du service Web et ceux qui seront mesurés à travers le réseau qui relie le client au service Web. Toutes ces informations sont utiles pour fournir au Courtier de QoS-SW des éléments fiables qui vont l'aider dans la prise de décision du meilleur service Web.

4.7.2 NRM sélecteur de fonctions (NRM-Select)

Les attributs de QoS réseau qui vont être vérifiés par NRM sont ceux qu'on récupère de la table de classification. Pour chaque attribut, il existe une fonction de mesure, de test et de vérification que NRM invoquera afin de récupérer les valeurs sur lesquelles se basera la décision du meilleur service Web dans une liste de sélection par exemple. Sélectionner la fonction qui exprime le mieux la QoS du service Web choisi revient à prendre le paramètre de qualité de service réseau qui a un poids important ; alors, le sélecteur choisira les fonctions adéquates qui seront utilisées par le NRM-Checker.

Des poids sur les attributs de QoS réseau sont donnés au moment de la publication par le fournisseur du service Web qui jugera du paramètre le plus important selon lui en vue d'offrir le meilleur service. Ces poids et attributs seront ajoutés chaque fois qu'une

nouvelle classe ou un type de service Web est publié ou créé pendant le processus de la publication. L'intervention du NRM dans le processus de publication n'a pas été détaillée dans ce travail. Elle est prise en charge par le Courtier de QdS-SW. Entre autres, le service Web NRM contribue d'une manière importante dans ce processus.

4.7.3 Classificateur de services Web (NRM-Classifier)

Le classificateur fournit différentes fonctions usuelles de création de nouvelles classes, types, catégories de services Web et attributs de QdS réseau. Toutes ces fonctions et procédures sont destinées à contrôler et mettre à jour la base de données NRM. Cette partie de classification a été déjà présentée dans la section 4.3.2.

4.7.4 Vérificateur de ressources réseau (NRM-Checker)

Le NRM-Checker emploie ses propres fonctions pour mesurer les paramètres simples de la QdS de réseau, le délai, la gigue, temps de réponse et la bande passante. Pour d'autres mesures plus détaillées ou plus pertinentes de ces mêmes paramètres ou d'autres comme la latence et la perte de paquets, il est nécessaire d'invoquer d'autres services Web spécialisés dans la mesure de réseau chez les professionnels en communications, chez les fournisseurs de services Internet et les principaux opérateurs de réseaux dans le Monde, etc. Par exemple, Observer 10.1 de Network Instruments [54] qui n'est pas encore dispensé en service Web comme d'autres produits existants, mais la tendance service Web donnera naissance à ce genre de services.

L'information utilisée en entrée par NRM-Checker (fournie par le Courtier de QdS-SW) est :

- Adresses IP de la machine du client qui peut être un utilisateur (humain) ou un autre service Web dans le cas de services Web composés.
- Adresses IP du fournisseur du service Web ou du serveur qui l'héberge.
- La fonction sélectionnée pour effectuer la mesure de l'attribut à mesurer ; si elle n'est pas disponible dans NRM, celui-ci procédera à l'invocation d'un service Web tiers qui fournit une telle mesure. Ce service Web est déjà identifié à l'avance par NRM vu qu'il subit les mêmes tests avant d'être ajouté dans la base de données NRM.

4.7.5 Surveillance des ressources (NRM-Monitor)

Une autre tâche, qui est la surveillance d'un service Web, pour assurer qu'il n'y a aucune violation dans le contrat de certification du fournisseur. Cette surveillance est importante pour le courtier de QoS-SW et les fournisseurs ; elle est indépendante du processus de base de NRM. Il faut spécifier en entrée des paramètres tels que le temps de début, de fin et la durée du monitoring et les attributs de QoS réseau à surveiller (délai, gigue, débit, etc.).

4.7.6 Base de données NRM

Elle conserve toutes les informations nécessaires au fonctionnement de NRM. Elle garde des traces d'exécutions et des résultats dans un but analytique, statistique et prédictif ou dans des cas de litiges sur la violation de contrat d'un service Web. Ces informations sont spécifiques au processus de sélection du Courtier de QoS-SW. On y trouve aussi les tables relatives à la classification des services Web.

4.7.7 Réserve de ressources

Pour garantir la QoS des services Web, la QoS réseau doit être assuré en utilisant un Courtier de bande passante (Bandwidth Broker) qui implémente un contrôle d'admission. Une requête est acceptée seulement dans le cas où il existe assez de ressources disponibles pour satisfaire ses exigences. L'implémentation de NRM réalise la réserve de ressources en utilisant des algorithmes de réserve et de propositions basée sur l'approche NAFUR [8]. Cette approche est présentée dans le chapitre 6. Le but de cette approche est de proposer au client une alternative quand sa requête ne peut pas être satisfaite ; une alternative correspond à une réserve inférieure à celle demandée (QoS dégradée), une réserve retardée par rapport au temps de début demandé et/ou une réserve sur une courte durée plus courte que celle demandée.

4.7.8 Outils de mesures de réseaux

NRM peut invoquer des services Web autres que ceux qu'il implémente ; ces services appartiennent à des entreprises qui offrent des outils de mesures réseau ou des entreprises qui gèrent plusieurs domaines de réseaux (dans le monde). Cette diversité mondiale permettra à NRM d'ouvrir un potentiel de clientèle mondiale aux fournisseurs des services Web en offrant des assurances de QoS sur différents réseaux avec la collaboration d'autres NRM.

4.8 Architecture de l'environnement d'exécution de NRM

La figure 4.3 montre tous les acteurs, éléments et modules qui interagissent entre eux afin de supporter la QoS pour les services Web. Cet environnement comporte quatre acteurs principaux :

Fournisseur du service Web : il héberge le service Web (par exemple Service Web vidéo). Il peut être un serveur d'Applications Weblogic, Websphere ou autres. Le client récupère le fichier WSDL (le fichier de description du service Web qui contient les informations pour la connexion) qui est publié sur les annuaires, puisque le fournisseur offre du contenu multimédia, il dédie un serveur de diffusion vidéo (video streaming server) qui héberge tous les modules de diffusion et les fichiers MultiMedia. Le serveur offre les capacités nécessaires pour subvenir aux demandes des clients.

NRM : il est responsable d'assurer, garantir ou estimer la QoS entre le client et le serveur de bout en bout. Il est hébergé sur le serveur de celui qui le fournit (par exemple : un fournisseur de service IP). Il est préférable de le faire cohabiter avec le Courtier de QoS-SW qui l'invoque pour améliorer la performance des échanges entre les deux entités ; cependant, ceci n'est pas requis, spécialement, dans le cas où plusieurs NRMs sont disponibles (par exemple : un NRM par domaine réseau) ; les différents NRM peuvent communiquer entre eux afin que chacun assure sa partie de QoS du chemin réseau de bout en bout.

Client : Il télécharge la partie application (par exemple : applet Java) du service Web qui lui permet de choisir et visualiser du contenu multimédia. Certaines conditions (par exemple : codec vidéo) peuvent être exigées au moment de l'exécution de l'applet Java.

Annuaire : ils contiennent les descriptions des services Web qu'ils indexent suivant la catégorie et le domaine. Il existe plusieurs types d'annuaire qui se basent sur le modèle standard UDDI : **(1) Courtier de QdS-SW** : un annuaire orienté QdS des services Web ; **(2) UDDI** : l'annuaire de base ; **(3) UDDIe** : un annuaire standard étendu avec des capacités d'intégration de la QdS dans la publication et la recherche.

4.9 Scénario d'utilisation de NRM

Il y a un certain nombre de fournisseurs de services Web qui ont publié leurs service Web multimédia (SWM) sur un annuaire standard UDDI, ou étendu UDDIe [5] ou encore un courtier de QdS-SW pour intégrer des données de QdS dans le fichier WSDL publié.

Un portail Web permet aux utilisateurs de chercher des services Web (par exemple, Vidéo-sur-Demande). Plus précisément, l'utilisateur définit une description du service qu'il cherche et qui satisfait les conditions de QdS désirée. Les paramètres de QdS saisis par le client sont souvent d'ordre littéraire sélectionnés dans une liste prédéfinie, par exemple « excellente, moyenne ou basse qualité d'image ». Cette description sera traduite par le courtier de QdS-SW vers des valeurs numériques qui faciliteront la recherche en fonction de ce qui est publié. Dans le cas où aucune information n'existe au niveau du courtier de QdS-SW, NRM analysera la requête, et prendra en charge cette information. Il convertira la description (WSDL) en attributs de QdS réseau spécifique aux services Web vidéo. La conversion se basera sur les valeurs issues de la table de classification et stockées dans sa base de données. NRM les utilisera comme repère pour vérifier et soutenir la QdS demandée par le client.

La figure 4.3 montre les interactions entre les différents composants. Les flèches numérotées représentent l'ordre des interactions dans le temps. Chaque flèche inclut une description des interactions correspondantes ; par exemple, la flèche (1) exprime

l'invocation du Courtier de QdS-SW par le client tandis que la flèche (9) montre la diffusion de la vidéo en utilisant RTP. Les étapes suivantes sont exécutées lors de la réception d'une requête client :

- Le client (Web portal/application Java) recherche et se connecte au service Web du courtier de QdS-SW; puis, l'invoque en générant une requête utilisateur (flèche 1 sur la figure 4.3).
- Le courtier de QdS-SW lance un processus de recherche en se basant sur les éléments fournis, dans la requête, pour récupérer une liste triée, L, de SWMs qui satisfont aux exigences de l'utilisateur, à partir des annuaires UDDI. (flèches 2 et 3).
- Le courtier de QdS-SW considère le premier service Web de la liste (le meilleur) L et invoque NRM pour vérifier s'il y a assez de ressources réseau disponibles et suffisantes pour supporter le MWS entre son fournisseur et son utilisateur (flèche 4). Nous notons qu'avant d'invoquer le NRM, le courtier de QdS-SW recherche et se connecte au NRM, car c'est un autre service Web comme les autres.
- NRM analyse la demande du courtier de QdS-SW et identifie l'information principale qu'il utilisera pour traiter la demande (flèche 5) ; les informations principales sont : adresses IP de l'utilisateur et du serveur qui exécute le MWS sous considération, et les attributs QdS demandés.
- Si le réseau qui relie l'utilisateur et le MWS a des fonctionnalités de support de QdS, alors, NRM utilise les protocoles et mécanismes fournis par le réseau
 - Si le réseau utilise IntServ, NRM utilisera le protocole RSVP pour faire la réservation de ressources entre l'utilisateur et le MWS.
 - Si le réseau utilise DiffServ, alors NRM marquera les paquets sortants (par exemple, le trafic de paquets vidéo diffusé par MWS) selon la QdS demandée. Dans la simulation du prototype de NRM (Section 4.11),

NRM marque les paquets sortants en supposant que le réseau supporte la technologie DiffServ.

- Si le réseau n'offre pas de QoS, NRM utilisera des outils de sondage pour vérifier le statut du réseau (ou plutôt le chemin réseau) entre l'utilisateur et le SWM dans un temps qui ne dépassera les délais acceptables dans un réseau.
- Si NRM réussit à faire de la réservation des ressources requises ou estime (dans le cas où le réseau ne fournit pas de QoS) qu'il y a assez de ressources pour accommoder la demande de l'utilisateur, il renvoie une réponse positive à Courtier de QoS-SW ; sinon, il renvoie rejet (flèche 6).
- Si le courtier de QoS-SW reçoit un rejet, il considère le prochain service Web dans la liste L et relance NRM ; le même processus est répété (flèches 4 et 5). Ce processus se termine quand une réponse positive est reçue ou quand tous les services Web dans la liste L ont été vérifiés sans succès. Dans le cas d'une acceptation, le fichier WSDL du SWM sélectionné est retourné à l'utilisateur pour procéder à l'invocation ; dans l'autre cas, un rejet est envoyé à l'utilisateur ou encore une proposition dans le cas où le NRM a les fonctionnalités nécessaires (voir Chapitre 6) pour proposer des alternatives.
- Si l'utilisateur reçoit un rejet, il renonce ou change sa demande en ce qui concerne ses préférences de QoS (c'est à dire, il initie une renégociation) ; autrement, il utilise le document WSDL fourni par le courtier de QoS-SW (flèche 7) pour se connecter au SWM et lancer le service (flèche 8). Après ces étapes, le serveur vidéo associé au SWM sélectionné commence la diffusion du média choisi vers le client en utilisant le protocole RTP (Realtime Transport Protocol) (flèche 9).

If faut noter que NRM peut être plus complexe ou très sophistiqué en fournissant des fonctionnalités de réservation à l'avance (voir Chapitre 6) ; dans ce cas, en réponse à l'invocation du service Web Courtier de QoS-SW, NRM vérifie la disponibilité de

ressources réseau, calcule, et retourne la QoS qui peut être fournie au moment de la requête au BWS (c.-à-d., immédiatement), ou dans un certain temps futur. Par exemple, si la QoS demandée ne peut pas être assurée au moment de la requête, NRM détermine le temps le plus proche, dans le futur, où l'utilisateur peut commencer le service avec la QoS désirée.

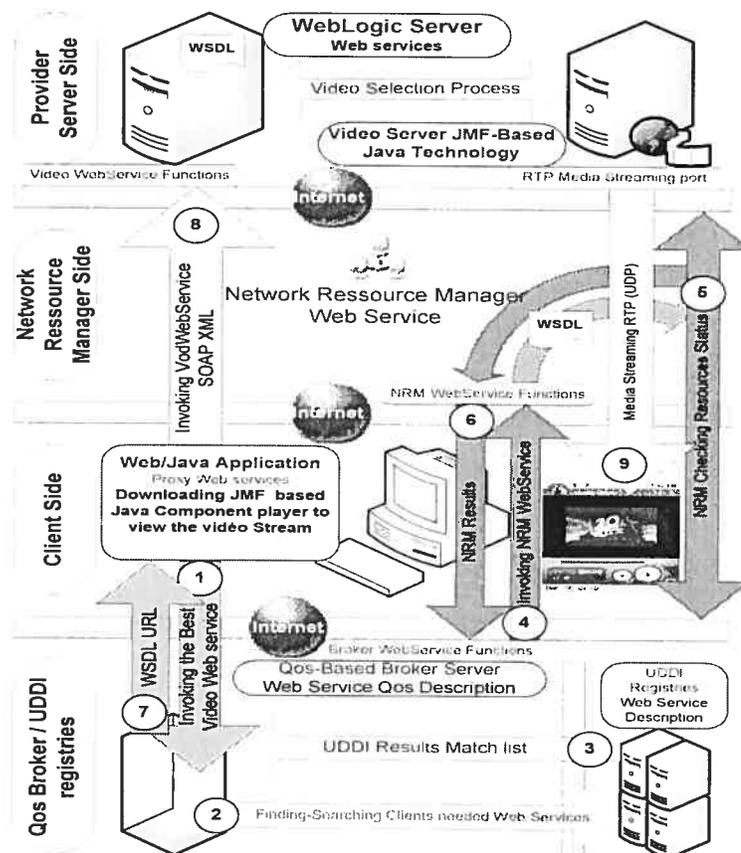


Figure 4.3 : Exemple de scénarios d'utilisation de NRM.

4.10 Implémentation

Nous avons implémenté un service Web vidéo qui permet à l'utilisateur de se connecter et choisir une vidéo en fonction de la QoS offerte et celle disponible sur le réseau, ce qui met en valeur l'architecture NRM proposée.

4.10.1 Service Web vidéo-sur-demande et NRM

Le service Web multimédia (vidéo-sur-demande) fournit à ses utilisateurs des fonctionnalités pour choisir et jouer un film. Les films et leurs métadonnées sont stockés dans une base de données locale MySQL. Nous avons doté SWM de plusieurs fonctions de diffusion et de réception de contenus multimédias. Nous avons également implémenté NRM comme un service Web qui utilise le marquage DiffServ [8] pour fournir une QoS différenciée ; le service Web multimédia et les utilisateurs qui s'y connectent sont sur un réseau qui utilise la technologie DiffServ.

Pour implémenter notre prototype, nous avons utilisé Bea Weblogic workshop [7] afin de concevoir des services Web dans un environnement conforme à J2EE [65]. Le serveur Weblogic est une plateforme de déploiement et de publication des services Web. L'API JMF (Java Media Framework) [66] est utilisé pour diffuser du contenu vidéo et/ou audio via le réseau (Internet). JMF implémente le protocole RTP [28] (basé sur UDP : User Datagram Protocol [33]) pour transmettre et recevoir des données multimédias.

4.10.2 Service Web multimédia : interface et utilisation

L'utilisateur peut choisir entre plusieurs classes de QoS. Chaque classe de QoS est caractérisée par le coût équivalent à la bande passante (Tables 4.2, 4.3). Par exemple ; une vidéo d'excellente qualité exige 5 Mb/s ; donc, pour garantir cette qualité, le serveur vidéo devrait avoir les ressources nécessaires pour la supporter et un chemin réseau devrait exister entre le serveur et le client avec une largeur de bande libre de 5 Mb/s.

QoS Classes	Qualité vidéo	Coût	Bande passante nécessaire
Classe 1	Haute	5 \$	5mbps
Classe 2	bonne	2 \$	2mbps
Classe 3	basse	1 \$	100 kb/s

Table 4.2 : Services vidéo offerts

Pour vérifier si le fournisseur du service Web peut supporter la demande de l'utilisateur, nous supposons qu'il existe une table qui indique le nombre maximum d'utilisateurs qui peuvent être servis simultanément par le serveur vidéo avec le détail de chaque classe de QoS offerte pour le même service (la capacité d'un service à offrir un nombre précis de clients peut être facilement mesurée ; ainsi, nous pouvons omettre cette table, mais il existe des cas où cette table est utile ; par exemple : à un moment donné il n'existe plus de disponibilité dans la classe 2 qui est demandée par le client, par contre la classe 1 ne sert aucun utilisateur, d'où la possibilité de la convertir en totalité ou en parti vers plusieurs ports de disponibilité de la classe 2).

La table 4.3 maintient également en permanence le nombre d'utilisateurs courants ; cette information est utilisée pour déterminer si une nouvelle demande peut être servie ou non par le serveur vidéo.

QoS Classes	Qualité vidéo	Ports disponibles	Maximum ports	Ports utilisés
Classe 1	Haute	2	5	3
Classe 2	bonne	14	15	1
Classe 3	Basse	10	25	15
Total		26	45	19

Table 4.3 : Exemple de disponibilités sur le serveur vidéo

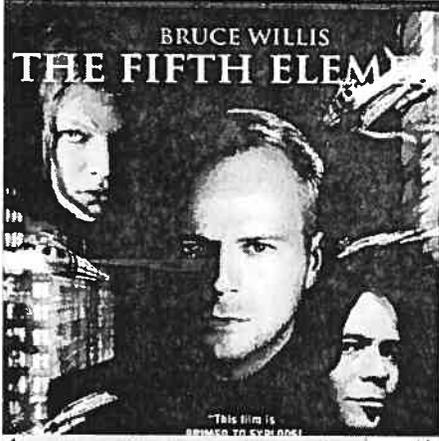
La figure 4.4 présente l'interface client du service Web multimédia ; en effet, quand le client se connecte au service Web choisi (dans ce cas-ci vidéo-sur-demande), il télécharge une applet Java, qui le dote d'une interface utilisateur (GUI) lui permettant de choisir le film qu'il veut et de vérifier si son ordinateur est prêt pour accueillir le flux vidéo. (Carte graphique capable d'afficher la vidéo, codecs nécessaires, JMF installé, mémoire, etc.).

Video Streaming Web Service

Liste des video disponible:

Name	Type	Description	Duration	Data
StarTrek Insurrection	SciFi	The Battle For Parades	103	startrekinsurrection.jpg
The dead zone	Mystere	The most entertaining	40	The dead zone.jpg
The Fifth Element	SciFi	Two hundred andfifty	125	Selement.jpg
Event Horizon	SciFi	The year is2047 Year	125	Eventhorizon.jpg
Kill Bill Volume 2	Action	The New Film By G	125	killbill2.jpg
Police Academy 1	Comedy	The Namean Law Enfa	125	policeacademy.jpg
Exorcist	Horror	The horror isLegion A	125	Exorcist3.jpg
StarTrek Insurrection	SciFi	The Battle For Parades	103	startrekinsurrection.jpg
The dead zone	Mystere	The most entertaining	40	The dead zone.jpg
The Fifth Element	SciFi	Two hundred andfifty	125	Selement.jpg
Event Horizon	SciFi	The year is2047 Year	125	Eventhorizon.jpg

Two hundred andfifty yearsin thefuture, lifeas we know it isithreatened bythearrival of Evil. Only thefifth element (played byMilla Jovovich) can stop theiEvil from extinguishing life,as it tries to do every five thousand years. She is helped byex-soldier, current-cab-driver, Corben Dallas (played byBruce Willis), who is in turn, helped byPrince/Arsenio clone, Ruby Rhod. Unfortunately, Evil isbeing assisted byMr. Zorg (Gary Oldman), who seeks to profit from thechaos that Evil will bring, andhis alien Mercenaries.



Quality CheckOos Stop Pa... St...

Figure 4.4 : Client applet Java du service Web VSD (Vidéo-sur-Demande).

4.11 Simulation

Nous présentons les simulations que nous avons réalisées pour évaluer NRM de base sans le module de réservation qui sera détaillé dans les chapitres 6 et 7. Pour NRM de base nous avons utilisé «J-Sim» la plateforme de simulation Java [54] pour simuler le comportement d'un réseau IP supportant DiffServ.

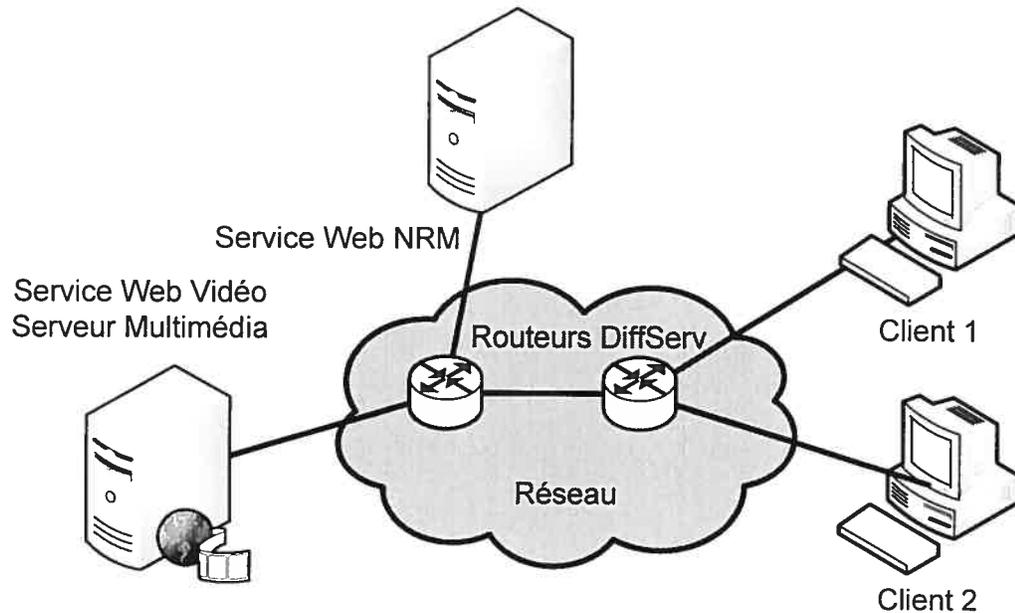


Figure 4.5 : Exemple de connexion client-serveur vidéo.

Nous avons mis en application un Service Web Multimédia (SWM) qui offre le service de vidéo-sur-demande ; SWM fournit à des utilisateurs la fonctionnalité de choisir et de jouer un film. Il implémente les fonctionnalités nécessaires pour accéder aux données sur le serveur et pour diffuser le contenu multimédia. Un client se connecte à un serveur vidéo (figure 4.5) ; chaque réseau peut avoir son NRM qui est responsable de la gestion des ressources. Les expérimentations que nous avons exécutées supposent un seul NRM et un seul réseau.

Les paquets entrant dans un tel réseau sont marqués dépendamment de la QoS désirée. Dans nos expérimentations, nous marquons les paquets EF (client 1) ou bien BE (client 2). Les paquets marqués EF reçoivent le meilleur service disponible (un routeur expédie les paquets EF en priorité avant tout les autres paquets) ; les paquets BE (Best Effort) sont les derniers à être expédiés. Nous avons considéré deux scénarios pour les simulations : (1) deux utilisateurs qui demandent en même temps de visualiser un film avec un service BE ; et (2) deux utilisateurs qui demandent en même temps de visualiser un film : un avec un service BE et un avec un service EF.

La capacité du lien entre les deux routeurs dans la figure 4.5 est de 0.75 Mbps. Chaque vidéo génère un débit de 0.4 Mbps entre le client et le serveur vidéo, d'où les deux vidéos simultanément sur le même lien vont créer une congestion et perte de données. Dans le premier scénario (1), il n'y a aucune garantie de la QoS. Les paquets traverseront les routeurs sans avoir de priorité. La figure 4.6 montre qu'il y a des pertes de paquets pour les deux sources ce qui justifie la faible garantie de QoS par NRM quand il n'y a pas de garantie de la QoS (Best effort). NRM fera des simples mesures sur le temps de réponse et la capacité du lien de bout en bout. Ces mesures sont faites quand il n'y a pas de technologie qui garantie la QoS de bout en bout pour aider les consommateurs dans leurs décisions du choix du service Web. Ceci ne veut pas dire que les clients vont recevoir de la mauvaise qualité ; il y a toujours un seuil acceptable de perte de paquets pour la vidéo.

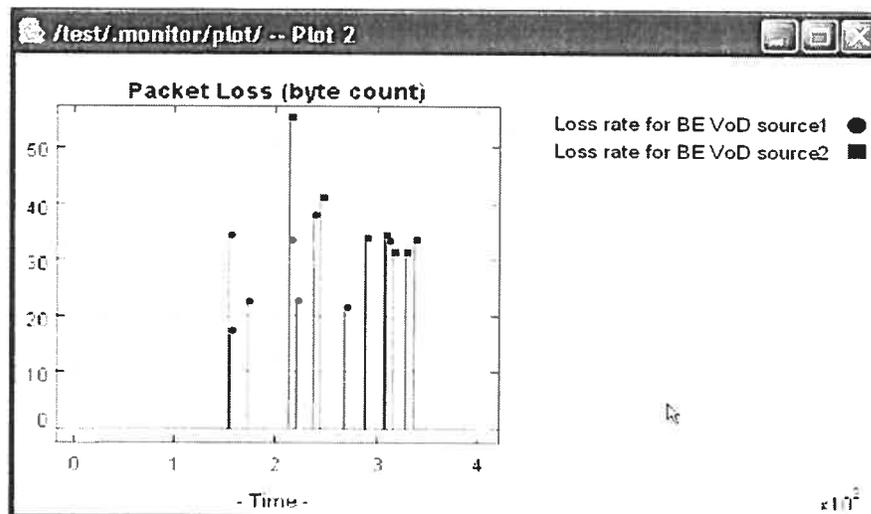


Figure 4.6 : Taux de paquets perdus sur un nœud du réseau Internet.

Dans le second scénario (figure 4.7), on remarque que la source 2 qui a été marquée EF par NRM n'a pas subi de perte de paquets. Cette garantie est due aux routeurs DiffServ (figure 4.5) qui traitent en priorité les paquets marqués EF.

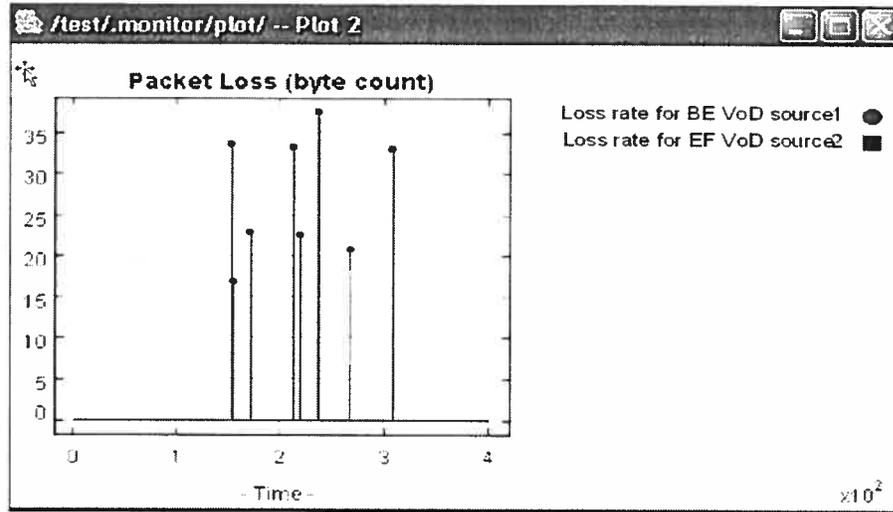


Figure 4.7 : Taux de paquets perdus sur un nœud du réseau DiffServ.

4.7 Conclusion

Dans ce chapitre, nous avons présenté une solution au problème du soutien de la QoS de bout en bout pour les services Web (Multimédia) afin de la garantir. Notre solution n'exige aucun changement à l'infrastructure réseau actuelle soit des utilisateurs ou des fournisseurs de services Web. Plus spécifiquement, nous avons présenté la conception et l'implémentation du service Web gestionnaire de ressources réseau (NRM). C'est simplement un autre service Web qui est publié dans les annuaires de services Web. Il est recherché, trouvé, et invoqué par le Courtier de QoS-SW. Son rôle principal est l'appui, si possible, de la QoS dans le chemin réseau qui connecte le service Web à l'utilisateur. L'assurance de la QoS dépend des fonctionnalités de réseau en termes de QoS (par exemple : DiffServ, IntServ, MPLS). Si le réseau n'offre pas de QoS, NRM utilise des approches basées sur des mesures réseau afin d'estimer la QoS entre les deux points du chemin réseau. Il est à noter que les détails de conception et d'implémentation de NRM ont fait l'objet d'une publication [23].

Chapitre 5

Fouille de données

Le présent chapitre est consacré à la fouille de données (Data Mining) ; il décrit ses tâches, ses objectifs, ses techniques ainsi que les étapes de son fonctionnement. Il détaillera entre autres la technique des k plus proches voisins que nous avons adaptée pour notre approche de réservation des ressources réseau (voir Chapitre 6).

5.1 Introduction

Aujourd'hui, presque toutes les activités dans le monde entier utilisent des ordinateurs et des applications qui génèrent beaucoup de données qui sont importantes à différents degrés ou niveaux. Toutefois, ces montagnes de données ne seront pas bien exploitées si nous n'avons pas des outils pour répondre à des interrogations de ces données. Il est intéressant de développer des stratégies claires et efficaces, pour pouvoir analyser et extraire des informations qu'on interprétera plus tard. Pour répondre à tous ces besoins, la mise en place de fouille de données devient alors incontournable.

5.2 Définitions

La fouille de données, aussi connue sous les noms d'exploration de données et « Data Mining », a pour objectif l'extraction du savoir implicite contenu dans de grandes quantités de données, par des méthodes automatiques ou semi-automatiques [15]. Elle présente l'avantage de trouver des corrélations entre les données et permet de mieux comprendre les liens entre des phénomènes en apparence distincts et d'anticiper des tendances encore peu discernables [50].

5.3 Tâches de fouille de données

Quelle que soit la nature du problème posé par la fouille de données, la solution réside habituellement dans l'une des tâches suivantes [50] :

La classification consiste à examiner les caractéristiques d'un élément (objet) nouvellement présenté et l'affecter à une classe d'un ensemble déjà prédéfini. Cette tâche repose sur des techniques comme les arbres de décision, le CBR (Case-Based

Reasoning), etc. Cette tâche est utilisée dans notre travail afin de classer une requête de réservation comme nouvel élément dans un ensemble de disponibilités.

La description (description and Profiling) permet de décrire des données trop complexes en vue d'en fournir des explications.

L'estimation consiste à estimer une variable continue en fonction des variables dites explicatives, qualitatives ou quantitatives.

La prédiction est similaire à la classification et l'estimation, sauf qu'elle a pour objectif de prédire les comportements des objets ou leurs valeurs futures (par exemple : prédire la perte de paquets, ou aussi prédire le moment de la congestion dans le réseau)

La segmentation (clustering) consiste à créer des groupes homogènes, qui se ressemblent le plus à partir de données hétérogènes.

Le groupement par affinité (affinity grouping) consiste à identifier les dépendances qui existent entre des caractéristiques observées sur un ensemble de données.

5.4 Étapes du processus de fouille de données

Dans cette section, nous décrivons les différentes étapes du processus de fouille de données :

Préparation des données (data processing) : les données proviennent généralement d'entrepôt de données, des fichiers binaires, des textes, etc. Certaines transformations sont faites sur ces données afin d'éliminer ce qui est inutile. Dans notre projet, les données sont des agrégats de disponibilités de QoS en temps réel issues de la projection de service disponible (ASP : Available Service Projection) qui seront utilisées en entrée à la technique appropriée à notre recherche (voir Chapitre 6).

Nettoyage des données (data cleaning) : une fois toutes les variables nécessaires identifiées, il est recommandé de corriger et éliminer les données erronées/extrêmes.

Intégration des données (data integration) : il est parfois utile d'avoir des données externes si nécessaire. L'objectif est donc de consolider et d'enrichir les données sur lesquelles le processus de fouille de données va s'exécuter.

Normalisation et transformation des données : consistent à faire des changements dans la base de données. Par exemple, transformer excellent, très bien, bien, moyen et faible en 5, 4, 3, 2 et 1 respectivement.

Recherche des modèles (search for patterns) : le choix des techniques est important pour élaborer des modèles adéquats ; il se fera selon les objectifs visés par l'étude. Les techniques les plus populaires sont citées dans la section 5.5.

Évaluation et interprétation : Cette étape représente la phase finale du processus de fouille de données. Elle consiste à interpréter et évaluer [32] les modèles.

5.5 Techniques de fouille de données

Les techniques de fouille de données constituent le coeur de ce processus. En effet, ces techniques permettent d'élaborer des modèles pour l'aide à la décision à partir de données brutes. Elles sont basées sur des aspects mathématiques et statistiques complexes. Les techniques les plus utilisées sont [50] : les arbres de décision (Decision Tree), les réseaux bayésiens (Bayesian Network), les réseaux de neurones (Neural Network), l'analyse des regroupements (Cluster Analysis), les règles d'association (Association Rules) et le raisonnement à base de cas (Case Base Reasoning). Nous décrirons seulement la technique de raisonnement à base de cas utilisée dans notre approche de réservation de ressources avec proposition.

C'est une technique dans laquelle l'expérience antérieure peut s'appliquer à de nouvelles situations. Autrement dit, nous nous basons sur des cas résolus dans le passé pour trouver des solutions à des problèmes similaires actuellement affrontés. Si l'un des nouveaux problèmes n'a aucune référence dans la base de cas comprenant les expériences passées, il sera interprété, indexé et mis à jour dans cette base. Le raisonnement à base de cas est vital dans les activités où le rôle de l'expérience est décisif, c'est-à-dire, les activités où nous ne disposons ni de théories, ni de modèles formels pour leurs résolutions. En fouille de données, le raisonnement à base de cas est très pratique pour les tâches de la prédiction, l'estimation et notamment la classification, où il s'agit de positionner des objets nouvellement présentés par rapport à ux plus proches voisins déjà résolus [67]. Ces algorithmes reposent sur certaines formules de calcul de distance (figure 5.1)

5.8 Méthode KPPV (K plus proches voisins)

5.8.1 Description

La méthode des k plus proches voisins (k Nearest Neighbor en anglais) est une méthode dédiée à la classification qui peut être étendue à des tâches d'estimation. La méthode KPPV est une méthode de raisonnement à base de cas. Elle se base pour la prise des décisions sur l'utilisation des cas similaires déjà résolus en mémoire. Contrairement aux autres méthodes de classification déjà citées ci-dessus (arbres de décision, réseaux de neurones, etc.), il n'y a pas d'étape d'apprentissage consistant en la construction d'un modèle à partir d'un échantillon d'apprentissage. C'est l'échantillon d'apprentissage, associé à une fonction de distance et une fonction de choix de la classe en fonction des classes des voisins les plus proches, qui constitue le modèle. L'algorithme générique de classification d'un nouvel exemple par la méthode KPPV est comme suit :

Algorithme KPPV

- 1 **Entrée** : le nombre k de voisins.
 - 2 **Données** : soit un échantillon de N enregistrements $X_i (x_1, x_2, \dots, x_n)$
 - 3 **Entrée** : un nouvel enregistrement $Y(y_1, y_2, \dots, y_n)$
Déterminer les k plus proches enregistrements de Y en utilisant une fonction de distance
 - 4 appropriée. $D(X_i, Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$ (Distance Minkowski)
 - 5 **Sortie** : les k exemples les plus proches qui ont la distance $D(X_i, Y)$ la plus faible.
-

Table 5.1 : Algorithme générique KNN.

5.8.2 Définition de la distance

Le choix de la distance est primordial au bon fonctionnement de la méthode ; les distances les plus simples permettent d'obtenir des résultats satisfaisants.

Dans notre travail, les points X_i représentent les enregistrements d'une table contenant les alternatives possibles à une requête représentée par Y dans l'algorithme. Nous désignons par plus proche voisin un point qui est le plus proche de l'enregistrement autre que lui-même.

1. **La distance Euclidienne** : $d_{Eucl}(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$
2. **La distance Manhattan** : $d_{Manh}(x, y) = \sum_{i=1}^n |x_i - y_i|$
3. **La distance de Minkowski** : $d_{Mink}(x, y) = \sqrt[q]{\sum_{i=1}^n (x_i - y_i)^q}$ où $q \in \mathbb{N}$

Figure 5.1 : Formules de calcul de Distance.

Pour définir la fonction de distance, nous définissons d'abord une distance sur chacun des champs x_i et y_i , puis nous combinons ces distances pour définir la distance globale entre enregistrements. La figure 5.1 présente les différentes distances utilisées entre deux valeurs numériques x et y . La combinaison des distances entre champs définit une distance entre enregistrements. Soit $X = (x_1, \dots, x_n)$ et $Y = (y_1, \dots, y_n)$ deux enregistrements, d_1, \dots, d_n sont les distances définies sur les différents champs, la distance entre deux enregistrements peut être définie par la distance euclidienne.

5.8.3 Application de KPPV au problème de réservation

Considérons la simulation suivante afin d'évaluer la méthode KPPV dans le contexte de notre travail (figure 5.2). Les huit enregistrements de données existantes proviennent des disponibilités d'un attribut de QoS, où le premier champ est le temps de début « StartTime », le second indique la durée de cette disponibilité, et le troisième la bande passante disponible (Available bandwidth). Ceci constitue l'ensemble de données qu'on va utiliser afin de classer le nouvel enregistrement Y qui représente une requête de ressources constituée des mêmes champs.

Le Nouveau enregistrement. requête Y			Les Poids de chaque Champs		
Requested resources			Request Attributes Weight		
<i>S_{Tr}</i>	<i>D_r</i>	<i>A_{br}</i>	<i>StartTime</i>	<i>Duration</i>	<i>Available Band</i>
139	35	400	0,849	0,15	0,05
Agregated Available bandwidth			KNN Results		
<i>Id</i>	<i>StartTime</i>	<i>Duration</i>	<i>AB</i>		<i>KNN</i>
1	152	18	310	3	24,33
2	147	5	384	2	14,22
3	149	35	418	1	
4	150	12	37	6	82,28
5	151	45	29	7	83,78
6	220	5	310	5	78,17
7	217	23	384	4	72,11
8	139	35	4	8	88,55

Les enregistrements de données existants

Figure 5.2 : Simulation de KPPV sur un échantillon de données.

Nous constatons à première vue que le plus proche voisin de notre requête Y est l'enregistrement 3 dont la distance euclidienne est la plus basse (10,05). Nous remarquons aussi, que le 2^e plus proche voisin est l'enregistrement 2, ce qui ne reflète pas réellement les résultats attendus qui normalement doivent donner en 2^eme position le 8^e enregistrement. Cela amène la possibilité de pondérer l'importance des différents champs, chose que nous avons faite en donnant un poids de mesure à chaque champ afin de le favoriser par rapport à l'autre. L'enregistrement 2 est sorti comme le 2^e plus proche voisin à cause du fort poids donné au 1^{er} champ et aussi sa valeur qui est la plus proche de la requête. Même si le temps de début et la durée sont assez proches de la requête, on remarque que la bande passante est largement inférieure à celle demandée (une différence de 396).

Une étape intermédiaire déjà citée dans les processus de fouille de données est nécessaire afin de filtrer les données qui peuvent biaiser notre approche avec KPPV. Ce filtrage permettra d'éliminer ces erreurs comme la durée qui doit être supérieure ou égale à celle recherchée ; même chose pour la bande passante. Pour ce qui est du temps de début, il est normal que nous n'allions pas considérer des données antérieures par

rapport au temps actuel. Mais, il serait possible de les prendre en compte pour proposer des temps, durées et QdS inférieurs à ceux demandés.

5.8.4 Discussion de la technique KPPV

La technique KPPV permet la recherche de cas similaires au cas à résoudre, et d'utiliser non pas les décisions des cas proches déjà résolus pour choisir une décision, mais prendre les k plus proches voisins comme alternatives à la requête soumise à NRM. Ça revient à considérer les cas les plus proches comme propositions.

Elle peut s'appliquer dès qu'il est possible de définir une distance sur les champs. Or, il est possible de définir des distances sur des champs complexes tels que des informations géographiques, des textes, des images, du son, etc. C'est parfois un critère de choix de la méthode KPPV, car les autres méthodes traitent difficilement les données complexes. On peut noter, également, que la méthode est robuste au bruit. Elle permet de traiter des problèmes avec un grand nombre d'attributs.

Les performances de la méthode dépendent du choix de la distance, du nombre de voisins et du mode de combinaison des réponses des voisins. En règle générale, les distances simples fonctionnent bien. Sinon, il faut envisager le changement de la distance ou de la méthode.

5.9 Conclusion

Notre objectif dans ce chapitre était d'introduire le domaine de fouille de données afin de mieux comprendre les outils et algorithmes utilisés à des fins de décisions, prédictions et estimations. Nous avons présenté la méthode KPPV que nous allons utiliser dans notre approche de recherche de propositions. Nous avons appliqué KNN sur un échantillon de données afin de simuler son comportement et vérifier son efficacité de produire des solutions alternatives. Le résultat était satisfaisant ce qui nous a encouragés à l'implémenter dans notre système.

Dans le chapitre suivant, nous allons décrire en détail comment la technique KPPV a été utilisée pour calculer des propositions alternatives.

Chapitre 6

Réservation de Ressources

6.1 Introduction

Quand un client veut commander un film en utilisant un service Web multimédia MWS (Vidéo-sur-Demande) sur l'Internet, il doit sélectionner le film, la qualité vidéo et audio qu'il veut, etc. Le support de la qualité désirée dépendra de la bande passante, du moment souhaité pour commencer à visualiser le film, et de sa durée. Ces paramètres sont requis par NRM pour faire la réservation de ressources.

6.2 Objectifs et motivations

L'efficacité de la réservation future (voir figure 6.1) dépendra des paramètres de QoS réseau entre le client et le serveur. Quand le système de réservation traite les demandes des utilisateurs, il devrait pouvoir répondre à ces demandes avec un ticket de réservation affirmant que la « réservation est faite ». Cependant, dans les systèmes courants les utilisateurs reçoivent uniquement deux réponses : acceptée ou rejetée s'il n'y a aucune ressource disponible, qui en soi est une réponse logique, mais qui ne laisse aucune possibilité pour des solutions de rechange. Pour éviter les rejets, nous utilisons dans notre travail une approche appelée NAFUR [2] (Negotiation Approach with Future Reservation). Dans le cas d'une impossibilité de réservation de ressources demandées par l'utilisateur, NAFUR calculera et enverra une liste de possibilités évitant de ce fait cet état de rejet.

L'application de NAFUR signifie que le système doit trouver les propositions appropriées qui accommodent au maximum la demande initiale de l'utilisateur. Pour obtenir les meilleures propositions, NRM exécute un processus de recherche qui doit être efficace et dont le temps de réponse doit être le plus court possible. Les auteurs de NAFUR [2] proposent un certain nombre d'algorithmes pour calculer les propositions alternatives ; cependant, ils n'abordent pas le problème d'implémentation qui pourra avoir un impact considérable sur le temps de réponse de calcul de propositions. En réponse à une requête, NAFUR propose de parcourir l'état du réseau sur une période de

temps qui couvre toutes les requêtes acceptées (en cours et prévus dans le futur) à partir du temps de début de la requête (voir figure 6.1) ; ce parcours est nécessaire pour le calcul de propositions alternatives. Ceci pourrait avoir un impact négatif sur le temps de réponse de NAFUR, particulièrement si l'on est en présence d'un réseau chargé sur lequel transitent des dizaines ou des centaines de milliers de flux (requêtes) avec des temps de départ et de fin différents.

Une première contribution de ce chapitre est l'utilisation d'agrégat hiérarchique de l'état du réseau (au lieu de l'état du réseau 'plat') pour calculer les propositions et la définition de structures de données qui permettront d'accueillir ces agrégats. Nous croyons que l'utilisation d'agrégat pour calculer les propositions va réduire le temps de réponse (voir Chapitre 7).

Une deuxième contribution est le développement et l'évaluation d'un certain nombre d'algorithmes qui permettent de calculer les propositions en utilisant les résultats de la première contribution. Plus précisément, nous avons implémenté et évalué quatre algorithmes : (a) CP-KPPV : calcul de proposition avec KPPV ; (b) Une version améliorée de l'algorithme PC [2] que nous appelons CPA-NAFUR calcul de proposition amélioré avec NAFUR; (c) un algorithme qui combine les deux, CP-KPPV et CPA-NAFUR que nous appelons CPH : calcul de proposition hybride ; et (d) l'algorithme de réservation régulière/classique qui est introduit afin de comparer ses résultats avec les autres algorithmes proposés.

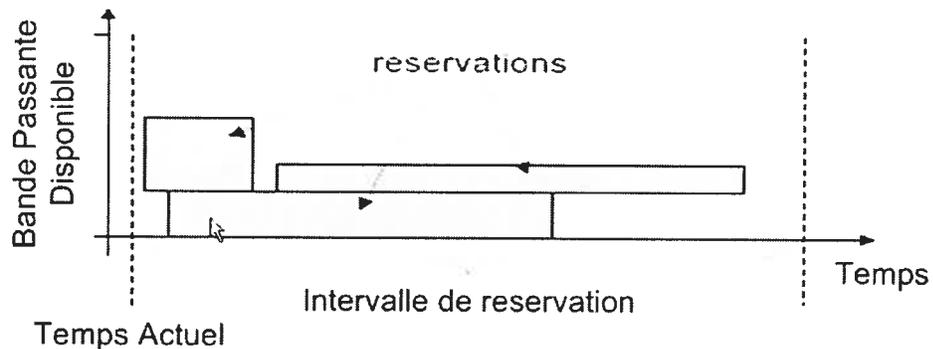


Figure 6.1 : Représentation schématique des réservations futures.

6.2 NAFUR

En réponse à une demande de service avec une QdS désirée, la plupart des approches de négociation de QdS renvoient une acceptation ou un rejet simple de la demande [53, 46, 43, 1, 44]. NAFUR [2] est une approche de négociation pour des réservations futures ou à l'avance. Elle est basée sur un principe de proposition d'alternatives à défaut de pouvoir satisfaire la requête de réservation initiale. Le résultat que NAFUR produit est un ensemble de propositions où chacune indique la QdS disponible pendant une durée « Duration » à partir d'un temps « Start Time » dans le futur. Formellement, une proposition est définie comme un tuple (Time, Duration, QdS) où QdS représente la Qualité de Service qui peut être offerte par le système pendant l'intervalle [Time, Time+Duration].

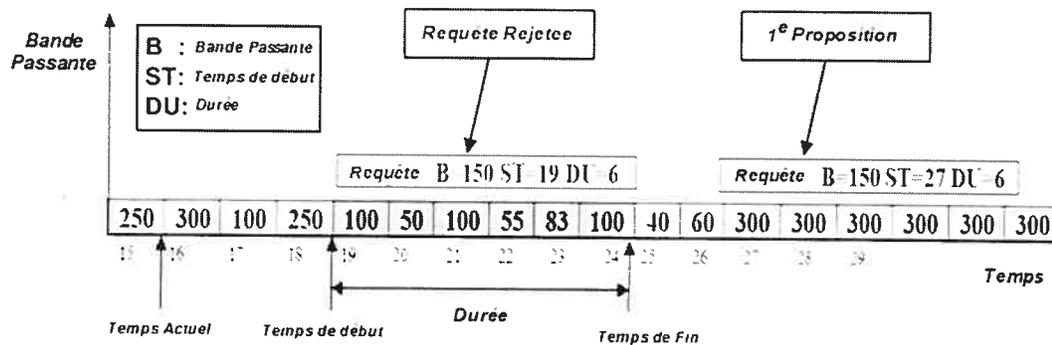


Figure 6.2 : Principe NAFUR de recherche de propositions.

Une requête de réservation est définie par le temps de début de la réservation *TimeStart*, sa durée *Duration* et la QdS réservée (par exemple : Bande passante : BP) ; donc, une requête R est définie par (TimeStart, Duration, BP).

NAFUR se base sur ASP (projection de service disponible) pour calculer les propositions en réponse à une requête. Formellement, ASP se compose d'une liste de tuples (*Time*, *QdS*), où *QdS* correspond à la QdS qui peut être soutenue à partir du temps *Time*.

L'ASP est définie comme une projection des disponibilités de service (Paramètres de Qualité de service) dans le temps entre deux points du réseau; La figure 6.2 montre, par exemple, qu'à l'instant 16, 300 unités de bande passante sont disponibles alors qu'à

l'instant 25, 40 unités de bande passante sont disponibles. La requête (19, 6, 150) ne peut pas être accommodée puisque la bande passante disponible entre 19 (temps de début) et 25 (temps de la fin) est égale à 50 unités (minimum (100, 50, 100, 55, 83, 100)) ; dans ce cas, NAFUR retourne une proposition qui consiste à commencer la réservation en retard de 8 unités de temps soit à l'instant 27.

ASP est un élément fondamental pour NAFUR. Il est utilisé pour le calcul des propositions en réponse à une requête de réservation. Les détails de calcul d'ASP peuvent être trouvés dans [2] ; plus particulièrement, il faut mentionner l'algorithme BC (Basic Combination) qui est fondamental pour le calcul d'ASP de bout en bout ; il combine les ASPs des nœuds qui constituent le chemin entre la source et destination pour calculer l'ASP de bout en bout.

Dans ce projet, nous supposons qu'un ASP, de bout en bout, est disponible ; les algorithmes proposés l'utilisent, entre la source et la destination de la requête, pour produire les propositions à retourner à l'utilisateur. La figure 6.3 montre une présentation graphique d'ASP ; par exemple : à l'instant $time = 0$, il y a 5 Mbps disponible.

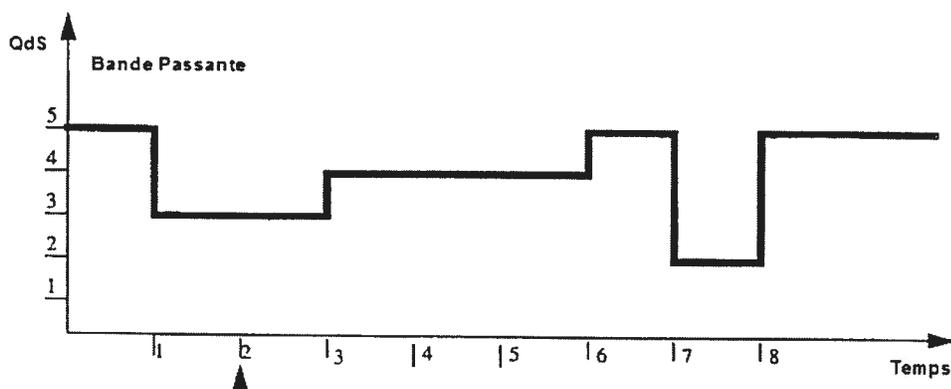


Figure 6.3 : Exemple d'un ASP.

6.3 Structures de données et agrégation

Dans cette section nous présentons l'agrégation d'ASP afin d'optimiser le temps de recherche de disponibilités ainsi que les structures de données utilisées.

6.3.1 Structures de données

L'ASP représente la QdS disponible entre le client et son fournisseur de service. L'information sur la QdS doit être disponible à tout moment pour l'agent de réservation responsable du contrôle d'admission. Toute interrogation et vérification ou mise à jour des disponibilités est directement appliquée par l'agent sur l'ASP qu'il contrôle afin de garder un état réel des ressources de bout en bout.

Nous considérons que l'ASP est représenté par un tableau qui est la plus simple des structures de données. Dans [43, 44] les comparaisons entre les tableaux et les arbres segmentés montrent que les tableaux sont les mieux adaptés pour le problème de réservation. La structure utilisée est définie comme suit : chaque entrée du tableau représente une unité de temps segmenté (un créneau) qui contient la valeur de la bande passante réservée durant cet intervalle de temps.

À chaque instant, elle contient la QdS disponible et la modifie dans le cas de réservation de la ressource. La figure 6.2 montre les détails de cette structure de données, qui offre une possibilité simple et rapide d'agrèger les QdS disponible entre deux points du temps. Pour représenter la QdS en fonction du temps, beaucoup de travaux de recherche [53, 46, 43, 1] utilisent des arbres binaires et des tableaux comme structure de données.

6.3.2 Agrégation

Nous avons utilisé la forme de structure arborescente pour agréger la QdS. En réalité le plus simple était de construire un arbre dans un tableau simple à une dimension (voir figure 6.6). On a aussi utilisé la possibilité de segmenter le temps en unités usuelles appelées créneaux. Chaque agrégation représente une fusion de deux créneaux de niveau inférieur fusionné en un seul créneau dans le niveau supérieur d'une durée double et affecté de la QdS la plus faible. L'opération est répétée jusqu'à ce qu'il n'y ait qu'un seul agrégat représentant la durée totale. Les créneaux sont employés pour réduire le traitement sur la durée et l'espace mémoire. Par exemple, si nous devons examiner 10 créneaux de 1 minute pour vérifier la bande passante pour 10 minutes, ce sera 2 créneaux de 5 minutes. Dans la figure 6.4, nous remarquons qu'un créneau est

représenté par une disponibilité de bande passante qui est par défaut une agrégation des disponibilités de chaque unité de temps qui représente le créneau.

6.4 L'algorithme CP-KPPV

6.4.1 Principe

La méthode des k Plus Proches Voisins est basée sur une des techniques de classification et de fouille de données (voir Chapitre 5) ; nous considérons les agrégats de disponibilités de bande passante comme des données déjà classées (par similarité dans le concept de fouille de données) ou des cas qu'on va utiliser pour classer une nouvelle donnée parmi eux qui est celle issue de la requête du client.(voir figure 6.5)

Quand une demande de réservation d'une quantité de bande passante (QoS) pour une durée spécifique (et aussi à partir d'un temps de début spécifique), est traitée par le système de réservation, elle est ou bien acceptée ou rejetée. Dans le cas de rejet, l'utilisateur doit reformuler et renvoyer sa nouvelle demande sans aucune garantie de succès. Pour éviter cette opération, nous utilisons l'approche NAFUR pour calculer des propositions à la place d'un rejet.

6.4.2 Application de KPPV à la génération de propositions

Pour décrire CP-KPPV, nous considérons la tâche de classifier un nouvel objet (une nouvelle requête dans notre cas) parmi un certain nombre d'exemples connus (AASP : éléments agrégés d'ASP comme ensemble de données d'entraînement (figure 6.5)). Cette tâche consiste à classifier (estimer) le point requête en se basant sur un nombre « K » choisi de ses voisins les plus proches. Pour une bonne compréhension, nous considérons l'exemple de la figure 6.5 ; les deux lignes qui se croisent au niveau du point de requête divisent le graphique en quatre zones, délimitant de cette manière l'intervalle des requêtes acceptables, car nous n'acceptons pas les points (propositions) avec une heure de départ inférieure à celle demandée et de même pour la bande passante. Pour cet exemple nous remarquons qu'on élimine la zone (1) et (4) de la liste avant appliquer l'algorithme KPPV en raison de l'importance du temps de début dans notre travail. Les autres zones (2) et (3) sont éligibles pour être traitées par l'algorithme KPPV et les résultats sont triés suivant le point le plus proche et qui est située dans la

zone (3). Les propositions qui sont dans la zone (2) ne sont pas complètement éliminées si on considère que NRM connaît les classes de services Web offertes par le fournisseur pour le même service. Nous pourrions dire que ces disponibilités seront prises en compte si la zone (3) ne contient pas assez de propositions ou si les temps de début « Start Time » des propositions existantes sont très éloignés du temps de la requête initiale. Dans ce cas, les propositions seront soumises avec une QoS moindre que celle demandée si le client spécifie dans sa requête qu'il acceptera d'autres alternatives de QoS inférieure.

Nous pourrions imaginer que le coût de ces alternatives sera plus bas, chose qui encouragera l'utilisateur à les accepter. Dans tous les cas, nous pourrions donner au client deux classes de listes de propositions, une de la zone (3) comme première classe et l'autre zone (2) comme deuxième classe de QoS si le temps de début est plus important que la QoS.

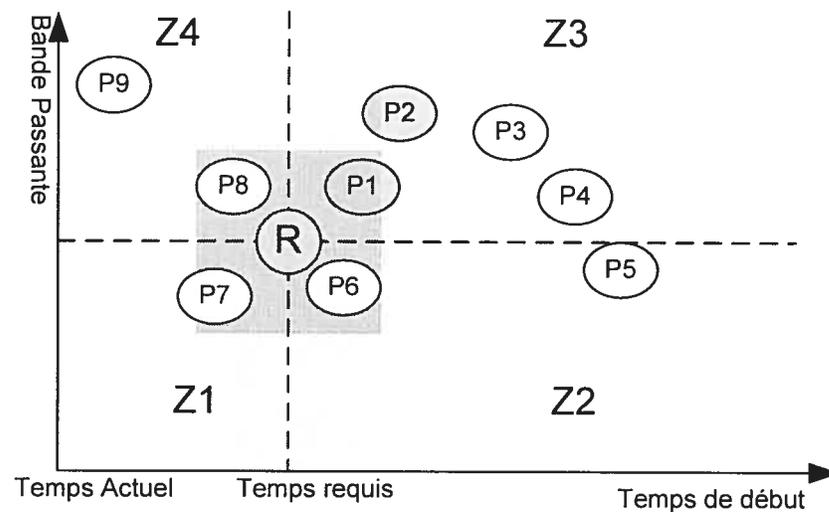


Figure 6.5 : Application de KPPV dans le problème de réservation.

6.4.3 Données de classification (disponibilités de QoS)

L'algorithme CP-KPPV doit être appliqué sur une structure de données (AAB : Aggregated Available Bandwidth) qui contiendra comme données des enregistrements contenant des champs tels que l'heure de début « StartTime », la durée « Duration » et la bande passante « Bandwidth » disponible d'une possible proposition. Ces possibilités de propositions sont obtenues en créant des agrégations issues de l'ASP.

Avant de présenter le processus de création et mise à jour de cette structure de données nommée AASP (agrégation ASP) (figure 6.6), nous définissons toutes les structures de données utilisées dans notre approche et leurs contenus :

ASP : tableau à une dimension, chaque élément $ASP[i]$ du tableau représente la bande passante disponible à l'instant i . La durée que représente un élément du tableau est un créneau de temps (principe de l'arbre segmenté). L'intervalle $[i, i+1]$ a une durée de $2*i$ minutes ; par exemple : si un créneau i est égal à 5 minutes la durée est égale à 10 minutes. La taille de l'ASP est définie par $MaxTime$ (on utilise 1440 minutes qui représentent une journée).

À chaque réservation, une mise à jour touchera les éléments contenus dans l'intervalle $StartTime$ et $StartTime + Duration$; cette mise à jour consiste à une diminution (égale à la nouvelle réservation) de la bande passante disponible de chaque créneau.

AASP : est un arbre binaire segmenté équilibré (conforme aux conditions des arbres AVL) qui est une représentation et agrégation de l'ASP. Nous utilisons un tableau de même taille que l'ASP dans lequel l'AASP est créé (figure 6.6). Le premier élément $AASP[1] = Minimum[ASP[1], ASP[2]]$ où $AASP [1]$ correspond à AB21 dans la figure 6.6 et $ASP [1]$ (resp. $ASP [2]$) correspond à AB1 (resp. AB2) dans la figure 6.6.

Le premier niveau d'agrégation correspond au tableau ASP qui contient des agrégats d'une durée d'un créneau, le 2ème niveau contient des agrégats de deux créneaux et ainsi de suite jusqu'à arriver au dernier niveau qui est un agrégat de durée $MaxTime$ et qui contient la bande passante disponible pendant cette durée. La mise à jour de cette représentation AASP est faite seulement s'il y a rejet. Donc, une autre structure AAB est nécessaire afin de lui appliquer l'algorithme KPPV.

Level 4	$AB41 = \text{MIN}(AB31, AB32)$							
Level 3	$AB31 = \text{MIN}(AB21, AB22)$				$AB32 = \text{MIN}(AB23, AB24)$			
Level 2	$AB21 = \text{MIN}(AB1, AB2)$		$AB22 = \text{MIN}(AB3, AB4)$		$AB23 = \text{MIN}(AB5, AB6)$		$AB24 = \text{MIN}(AB7, AB8)$	
Level 1	AB1	AB2	AB3	AB4	AB5	AB6	AB7	AB8

ASP(Mbps)	ASP[1]	ASP[2]	ASP[3]	ASP[4]	ASP[5]	ASP[6]	ASP[7]	ASP[8]
Time(Minutes)	5	10	15	20	25	30	35	40
	5 minutes Slot							

Figure 6.6 : (AASP) Agrégations disponibles au sein d'un ASP.

AAB : cette structure est créée au moment de l'invocation de l'algorithme basé sur KPPV ; en fait, AAB représente l'espace de recherche pour l'algorithme. La source des données pour AAB n'est autre que l'AASP. Toutes les agrégations de tous les niveaux (c.-à-d. de toutes les durées) seront prises en compte pour remplir la table AAB. Mais puisque ce processus est un processus de la fouille de données, un filtrage est nécessaire pour réduire le temps de recherche de KPPV (voir section 6.4.3.1).

Une entrée de la table AAB est définie comme suit :

- ST (StartTime) : Heure de début ;
- D (Duration) : La durée (en minutes) qui est un multiple entier de créneaux ; elle est toujours supérieure ou égale à la durée demandée.
- AB (Available Bandwidth) : bande passante disponible durant [ST, ST+D].

La requête de l'utilisateur : est définie comme suit $R = (ST_R, D_R, B_R)$ où ST_R représente le temps de début, D_R représente la durée, et B_R représente la bande passante.

6.4.3.1 Filtrage des données

Dans notre travail, nous avons appliqué un certain nombre de filtres. Pour ce qui est de la bande passante, nous considérons seulement les agrégats dans lesquels la bande passante est supérieure ou égale à celle requise ; il peut y avoir des préférences des utilisateurs dans lesquelles ils peuvent accepter une bande passante moindre à celle requise ; par exemple 5 % moins de celle demandée. Pour ce qui est de la durée, nous considérons seulement les agrégats sur une durée supérieure ou égale à celle dans la demande.

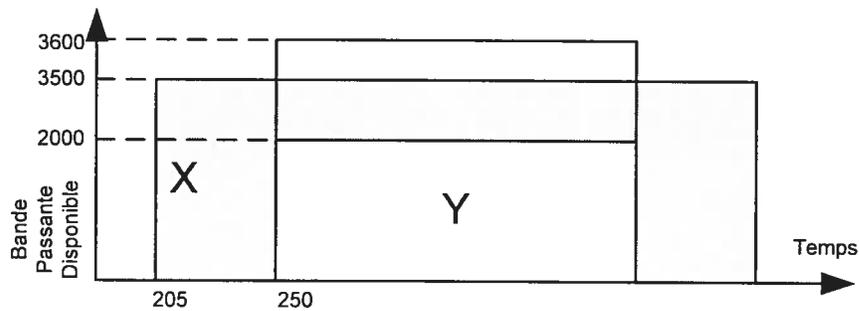


Figure 6.7 : Filtrage d'agrégations dans un AASP.

Le filtrage ciblera et éliminera aussi les blocs qui ont une durée qui est incluse dans d'autres blocs plus grands en durée et en bande passante. Par exemple, dans la figure 6.7 on a un agrégat X qui a comme temps de début 205, d'une durée de 128 minutes et de bande passante disponible 3500 ko ; $X = (205, 128, 3500)$. S'il existe dans notre échantillon un autre agrégat $Y = (250, 50, 2000)$, alors le fait que ses valeurs sont contenues dans X, il n'y a aucune utilité de le garder, car il va consommer du temps de calcul et de la mémoire. D'un autre côté si $Y = (250, 50, 3600)$, il restera dans l'échantillon, car sa bande passante est supérieure à celle de X. Aussi nous ne considérons pas les enregistrements/agrégats qui ont une heure de début supérieure à l'heure de début demandé plus une valeur prédéfinie appelé MDP. Elle représente le temps maximum, toléré par les utilisateurs, par lequel une demande peut être retardée (voir Section 6.7.1).

Algorithme CP-KPPV

- 1 : **Début**
- 2 : **Entrée** : Requête $R = (ST_R, D_R, B_R)$; ASPs ;
 KPPV[N] : Table de distances Dist et propositions AAB (ST_p, D_p, B_p)
 /* un élément de KPPV est une paire (distance, élément de AAB) où distance représente la distance entre l'élément AAB et la requête à traiter ;
 initialement, tous les éléments de KPPV sont initialisés à nil*/
- 3 : **Sortie** : Liste de propositions LP où $LP[i] = (ST_{pi}, D_{pi}, B_{pi})$
- 4 : $MaxB = val$; (valeur maximale de Bande passante)
- 5 : convertir ST_R en minutes à l'indice i correspondant dans le tableau ASP
 /* si $ST_R = 17$ minutes donc $i = 5$ dans la figure 6.6*/

```

6 : Convertir  $D_R$  en minutes à  $D_R$  en créneaux
    /* si  $D_R = 28$  minutes donc  $D_R = 6$  créneaux dans la figure 6.6*/
7 :  $AB = \text{Max}B$  ( $AB$  : Bande passante disponible)
8 : Pour  $x = ST_R + 1$  à  $ST_R + MDP$  Faire
    /*  $MDP$  est un intervalle de temps défini en créneaux */
9 :  $AASP = \text{Agrégation} (ASP)$  ; (Voir Section 6.3.2)
10 :  $AAB = \text{Filtrage} (AASP)$  ;
    /* il en résulte un nombre  $N$  d'éléments de  $AAB$  produit par le processus de
        filtrage. (Section 6.4.3.1)*/
11 : Fin Pour
12 : Pour  $y = 1$  à  $N$ 
    Faire
     $KPPV[y].Dist = \text{Distance} (AAB[y], R)$  ;
    /* Calcul de la distance entre chaque enregistrement  $AAB$  et la Requête  $R$  */
    Fin Pour
13 : Tri ( $Dist$ , croissant) ;
    /* ordonne  $KPPV$  de la paire avec la plus petite distance jusqu'à la paire avec
        la plus grande distance*/
14 : Pour  $y = 1$  à  $N$ 
    Faire
     $LP[y] = KPPV[y].AAB$  ;
    /* affecter les propositions de la meilleure en premier à la mauvaise en
        dernier
    Fin Pour
15 : Fin

```

Table 6.1 : Algorithme de réservation basée sur K Plus Proche Voisins.

6.4.4 Fonctionnement

Quand le système reçoit une requête d'un utilisateur pour réserver de la bande passante, le processus démarre automatiquement, par une vérification de la bande passante disponible à partir de l'heure de début pendant la durée requise. Si la bande passante est disponible pendant la durée demandée, la réservation est faite et une mise à jour de l'ASP est faite, et il n'y a aucun besoin de créer ou mettre à jour les tables AASP et AAB.

Pour la table AAB il n'y a pas de mise à jour, car elle est créée à chaque fois que le processus de propositions est enclenché, ainsi nous construisons les données de la table AAB à partir de AASP lui-même issue de ASPs.

Actuellement, à la création de la table AAB, l'algorithme KPPV calculera la distance de similarité entre les éléments $AAB_i = (ST_i, D_i, AB_i)$ et la demande $R = (ST_R, D_R, B_R)$. La distance $D = (R, AAB_i)$ est calculée en utilisant la formule euclidienne (voir Section 5.4). La meilleure proposition est celle avec la distance minimale. Les k premières propositions sont envoyées à l'utilisateur pour en choisir une. Le nombre k peut être choisi par le client au moment de la formulation de sa requête.

L'algorithme KPPV n'est pas utilisé pour classer notre requête parmi un ensemble de données (voir chapitre 5), mais le but est de pouvoir obtenir les données les plus proches d'elle.

6.5 Les autres algorithmes

6.5.1 L'algorithme CPA-NAFUR

Dans le cas où la demande ne peut pas être accommodée, CPA-NAFUR examine la largeur de bande globale en incrémentant l'heure de départ par une constante pour chaque étape jusqu'à trouver la largeur de bande voulue ou atteindre la valeur maximale de l'heure de départ qui est inférieure ou égale à l'heure de départ demandé + MDP (figure 6.8). CPA peut être paramétré pour s'arrêter quand la première proposition est déterminée ou pour continuer afin de produire une liste de propositions.

La différence entre CPA-NAFUR et l'algorithme de calcul de propositions de NAFUR [2] est que CPA-NAFUR est appliqué sur une structure de données qui offre l'agrégation des données donc une exécution plus rapide sur un ASP diminué de la moitié ou plus, dépendamment des performances voulues. Si on augmente les agrégations, le temps de réponse et le nombre de propositions diminue aussi. Cela est dû à l'agrégation, car quand on agrège on perd de la précision.

Algorithme CPA -NAFUR

```

1   : Début
2   : Entrée : Requête R = (STR, DR, BR) ; Table ASP [1..MaxTime]
3   : Sortie : Liste de propositions LP où LP[I]= ( STpi, Dpi, Bpi)
      I=0 /* aucune proposition dans la liste LP
4   : MaxB= val ; (valeur maximale de Bande passante)
5   : Convertir STR en minutes vers indice i correspondant dans le tableau ASP
6   : Convertir DR en minutes à DR en créneaux
7   : AB= MaxB (AB : Bande passante disponible)
8   : Pour x= STR +1 à STR + MDP
      /* on commence avec STR +1, car il n'a pas été possible de réserver avec
      STR par l'algorithme classique (régulier)
      Faire
9   : Pour y = (x à x + DR)
      /* pour la même durée DR on calcule la Bande Passante maximale disponible
      Faire
10  : Si ASP [y] < AB
      alors AB=ASP [x] ;
11  : Fin Pour
12  : Si AB >= BR
      alors I=I+1 ; LP[ I ]=( y, DR, AB) ;
      /* Si cette Bande Passante est satisfaisante par rapport à celle demandée, on
      l'ajoute à la liste des propositions.
13  : Fin Pour
14  : LP contient les I propositions trouvées.
15  : Fin

```

Table 6.3 : Algorithme de réservation NAFUR-PC.

6.5.2 L'algorithme régulier de réservation

Quand une demande arrive, la méthode de réservation régulière est très simple. D'abord, une vérification dans ASP de la bande passante qui est disponible pendant la durée demandée et à partir du temps de début souhaité dans la requête. Si elle est plus grande ou égale à celle demandée, nous réservons et mettons à jour la table ASP, autrement, une réponse de rejet est envoyée à l'utilisateur (voir table 6.2)

Algorithme régulier de réservation

```

1  : Début
2  : Entrée : Requête  $R = (ST_R, D_R, B_R)$  ; Table  $ASP[1..MaxTime]$ 
3  :  $MaxB = val$  ; (valeur maximale de bande passante)
4  : Convertir  $ST_R$  en minutes à l'indice  $i$  correspondant dans le tableau  $ASP$ 
5  : Convertir  $D_R$  en minutes à  $D_R$  en créneaux
6  :  $AB = MaxB$  ( $AB$  : Bande passante disponible)
7  : Pour  $x = ST_R$  à  $ST_R + D_R$ (en créneaux) Faire
      Si  $ASP[x] < AB$  alors  $AB = ASP[x]$  ; Fin Si
      /*  $AB$  contiendra la fin de la boucle, la Bande passante maximale disponible
8  : pour la durée  $D_R$ 
9  : Fin Pour
      Si  $AB \geq B_R$  alors
10 : /* Cas où la Bande Passante demandée est disponible
      Pour  $x = l$  à  $l + D_R$ (en créneaux)
      Faire
           $ASP[x] = ASP[x] - B_R$  ;
      Fin Pour
11 : /* On procède à la Réservation en mettant à jour l'ASP
      Sinon Rejet ;
12 : FinSi
13 : Fin

```

Table 6.2 : Algorithme de réservation régulier.

6.5.3 L'algorithme hybride CP-KPPV-PCA : CPH

L'algorithme hybride est une combinaison des deux algorithmes CP-KPPV et CPA-NAFUR ; en premier, l'algorithme de réservation régulière démarre, puis si un état de rejet est détecté, un appel à l'algorithme CP-KPPV est émis pour produire des propositions. Si CP-KPPV ne peut pas produire des propositions alternatives, la demande est réorientée vers l'algorithme PCA-NAFUR et ensuite s'il n'y a aucun succès à trouver une proposition, l'état de rejet en résulte.

6.6 Paramètres de performance

6.6.1 MDP

Le MDP (Maximum Delay of Proposal) est l'intervalle dans lequel les algorithmes de recherches de propositions vont puiser les alternatives possibles afin d'en tirer les meilleurs. Il permet de limiter la recherche sur un intervalle entre le temps de début voulu par la requête à ce temps de début + MDP.

Temps Actuel	StartTime Demandé	MDP Intervalle de Recherche
TN	ST	ST+ MDP

Figure 6.8 : Utilisation du MDP

6.6.2 Nombre de propositions

Le nombre de propositions est relatif à K pour l'algorithme CP-KPPV et il peut être spécifié par le demandeur de service au moment de la formulation de sa requête. Cependant, pour l'algorithme CPA-NAFUR le nombre de propositions dépendra de l'intervalle MDP parcouru à la recherche d'alternatives. Dans le cas où le nombre de propositions est fourni à l'algorithme, le calcul de proposition s'arrêtera au moment où il est atteint.

Le nombre de propositions retournées peut aussi être limité par le temps de réponse. En effet, on peut limiter le temps de réponse à une valeur x , et on va retourner les propositions produites avant d'atteindre le temps de réponse égale à la valeur x .

6.6.3 Flexibilité

La flexibilité est un paramètre qui permet d'obtenir les meilleures propositions pour le temps de départ et la QdS si l'utilisateur spécifie dans sa requête l'intervalle d'acceptation de chacun des paramètres tels que si la requête est $R = (ST_R, D_R, B_R)$, le client donnera les valeurs qu'il tolérera dans le cas où sa requête n'est pas disponible. Mais cette flexibilité n'intervient pas directement dans les propositions, car on peut trouver des disponibilités en utilisant ces valeurs au moment de l'exécution de l'algorithme régulier de réservation. La réservation se fera dans un deuxième temps en

utilisant le même algorithme régulier avec des vérifications sur des intervalles donnés par les différentes flexibilités tolérées. Les paramètres de flexibilité issus de la requête concernant le temps de début, la durée et la bande passante sont les valeurs que l'utilisateur acceptera si sa requête n'est pas respectée. Si la liste issue de cette opération contient plusieurs propositions, on appliquera CP-KPPV pour prendre la meilleure.

6.6.4 Poids des paramètres QoS

Pour de meilleures propositions, on donne des poids aux attributs de QoS de notre requête. Ces poids donneront des priorités dans l'application de l'Algorithme KPPV au moment du calcul de la distance euclidienne, afin de ressortir les propositions dont le paramètre a le plus grand poids. Par exemple, si on donne un poids important à la bande passante l'algorithme priorisera les enregistrements qui ont une bande passante qui se rapproche le plus de celle de la requête et ignorera d'une manière générale les autres paramètres (durée et temps de départ) qui ont un poids faible. Ces priorités sont reliées à la nature de l'application qui fait la requête et/ou l'importance des attributs pour l'utilisateur.

En général, il y a des profils adaptés à chaque genre d'utilisateur et type d'application utilisant le réseau. Par exemple, dans un système de Vidéo-sur-Demande, l'utilisateur donnera un poids élevé à l'heure de début ce qui la priorisera au dépend des autres paramètres tel que la bande passante, ce qui va nuire à la qualité du vidéo. Par contre dans un service Web de téléchargement, la bande passante est plus importante que l'heure de départ.

6.7 Conclusion

Dans ce chapitre nous avons présenté des algorithmes de recherche de propositions qui peuvent être utilisés pour réaliser NAFUR. Plus spécifiquement, nous avons présenté quatre algorithmes qu'on a implémentés ; CP-KPPV, CPA-NAFUR, CPH et l'algorithme de réservation classique. Nous avons décrit les structures de données utilisées pour la réalisation des différents algorithmes. L'objectif était de choisir les structures appropriées pour un accès rapide, une recherche et une mise à jour facile et simple. Une évaluation de la performance des quatre algorithmes est présentée dans le chapitre 7 en plus d'une analyse comparative.

En implémentant un des algorithmes décrits dans ce chapitre, NRM est capable de fournir un service sophistiqué à ses utilisateurs ; en effet, NRM offre des alternatives/propositions aux utilisateurs des services Web qui l'invoquent pour une garantie de QoS.

Chapitre 7

NAFUR : simulation et résultats

7.1 Introduction

Nous présentons dans ce chapitre les simulations faites et résultats obtenus pour la partie NRM étendu avec le module de réservation de ressources en avance. Nous discutons ces résultats pour les différents algorithmes de réservation de ressources. Ces expérimentations ont été réalisées sur une machine basée sur Intel Pentium 4 avec 1 Go de Ram et une fréquence de 3 Ghz. Nous avons développé notre propre environnement de simulations en C++ pour modéliser les requêtes de réservation et exécuter les différents algorithmes.

7.2 Données de simulations

- Les requêtes de réservation

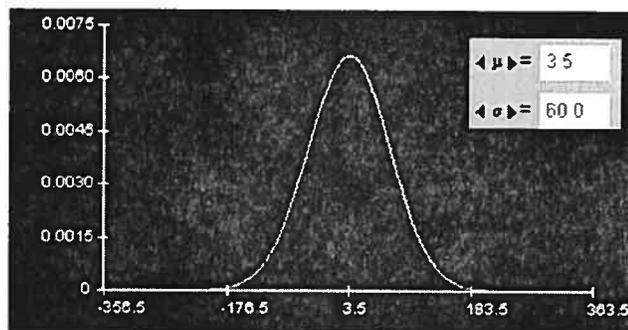


Figure 7.1 Distribution normale utilisée pour les requêtes.

Pour simuler les algorithmes de réservation et de propositions, nous avons utilisé une distribution normale de moyenne = 3.5 et de sigma = 60 afin de générer un trafic de 1500 requêtes sur une durée de temps de quatre-heures (figure 7.6) et qui représente un pic de requêtes entre la 2ème et 3ème heure. Ce pic représente 20h00 de la soirée où les

demandes sont au maximum. Le nombre de requêtes choisi est spécifique à la bande passante disponible de 60 Mo utilisée.

Les valeurs sur les axes du graphe sont générées par une distribution normale. Ces valeurs représentent seulement la distribution qui les génère. On établit un changement de repères pour ajuster ces valeurs à ceux qu'on utilise dans nos expérimentations.

- **Contenu des requêtes**

Chaque requête $R = (ST, Du, B)$ est constituée d'un temps de début « ST », d'une durée « Du » et de la QdS voulue (ici la QdS est la bande passante B). ST varie de 30 à 60 du temps actuel, ($ST = \text{temps actuel} + X$; $30 \text{ min} \leq X \leq 60 \text{ min}$), Du varie de 60 min à 90 min, et B prend comme valeur soit 128, 256, ou 512 ko. Nous supposons que la valeur maximale de la bande passante est égale à 60 000 ko (60 mb). L'intervalle de temps utilisé est d'une journée (1440 minutes). Il est représenté par une structure de données de type tableau ASP [1..1440]. Le tableau ASP est initialisé avec la valeur maximale de la bande passante (60mo) ; dès que le système commence à accepter des requêtes, la valeur de la bande passante disponible va diminuer.

Les requêtes sont générées aléatoirement suivant une loi normale comme cité précédemment. Elles se présentent comme l'exemple qui suit dans la table 7.1 : à l'instant 98 minutes de l'origine, on reçoit 2 requêtes identifiées chacune par son numéro de requêtes qui indique l'ordre de traitement.

Temps Actuel	Nombre de Requêtes	Numéro Requête	ST	Du	B
.....
95	1	1	112	76	128
96	1	1	138	78	512
97	1	1	124	71	512
98	2	1	114	79	128
98	2	2	152	71	512
99	2	1	155	75	128
99	2	2	157	78	256

Table 7.1 : Exemple de requêtes de réservation générées.

7.3 Les paramètres de performance

- **MDP (Délai maximum de propositions)**

Pour les expérimentations, la valeur de MDP prend les valeurs 30, 60, 120, 240 minutes. L'utilisation de ces valeurs nous permet d'évaluer l'impact de MDP sur les réservations ou plutôt sur les propositions.

- **Flexibilité dans les propositions**

En plus du MDP, les propositions peuvent être contraintes par des paramètres ou intervalles de flexibilité que le client peut définir pour limiter le domaine des propositions qu'il préfère dans le cas où sa requête est rejetée.

- **Valeur de la durée de réservation**

La durée de la réservation de chaque requête est distribuée d'une façon aléatoire sur l'intervalle [60, 90] pour simuler un système de Vidéo-sur-Demande...

- **Granularité du temps (Créneaux)**

Nous avons utilisé un créneau d'une valeur d'une minute qui est le plus bas possible. Chaque agrégation est multiple de 1. Par exemple, nous aurons des agrégats de durées paires (2, 4, 8, 16, 32, 64, 128, 256, 512, etc. minutes).

7.4 Les algorithmes expérimentés

- **Algorithme Régulier**

Cet algorithme a deux états : soit il réserve, soit il rejette. Il est le plus rapide vu qu'il cherche des alternatives. Il n'y a aucun effet du MDP sur cet Algorithme.

- **Algorithme CPA-NAFUR**

CPA-NAFUR démarre sa recherche à partir du temps demandé + un créneau (1 minute) en vérifiant sur la durée demandée si la bande passante requise est disponible, et ainsi de suite jusqu'à ce qu'il arrive à la limite de temps de début de la requête + MDP. CPA-NAFUR récupère de cette manière les disponibilités qu'il va soumettre au client. Un exemple de liste propositions de CPA-NAFUR (figure 7.2) se traduit par des

temps de début qui se suivent, la même durée pour toutes les propositions et des bandes passantes supérieures ou égales à celle de la requête.

La première colonne représente le temps de début, la seconde la durée et la dernière la bande passante disponible. Le temps actuel qui correspond au moment où la requête $R = (\text{Temps de début} = 181 \text{ minutes}, \text{Durée} = 81 \text{ minutes}, \text{Bande Passante} = 512 \text{ Ko})$ a été générée est égal à 135 minutes à partir de l'origine. D'où la première proposition est $P = (183, 81, 848)$ qui est largement adéquate, car le retard est de seulement de 2 minutes sachant que le MDP est de 30 minutes.

```

--PC Algorithm with MDP = 30
Time now: 135 Request-> 4
Request : 181 81 512
-----
 183    81    848
 184    81    976
 185    81   1104
 186    81   1104
 187    81   1104
 188    81   1104
 189    81   1360
 190    81   1672
 191    81   2256
 192    81   3024
 193    81   3152
 194    81   3664
 195    81   3920
 196    81   4944
 197    81   4944
 198    81   4944
 199    81   5200
 200    81   6992
 201    81   8016
 202    81   8528
 203    81   8528
 204    81   8528
 205    81  10320
 206    81  10320
 207    81  11088
 208    81  11472
 209    81  12368
 210    81  12880
 211    81  14544

```

Figure 7.2 : Liste de propositions générées par CPA-NAFUR.

- **Algorithme CP-KPPV**

```

1 -----Knn with MDP = 30 ---
2 Time now: 135 Request Numero-> 0
3 Request : 143 67 256
4 -----
5 48.101 144 64 208
6 67.956 176 32 208
7 77.421 176 16 208
8 77.627 144 128 208
9 82.91 176 8 208
10 87.419 180 4 208
11 89.722 182 2 208
12 195 144 256 208
13 344.22 180 2 592
14 447.58 144 512 208
15 723.58 184 8 976
16 723.91 184 4 976
17 724.09 184 2 976
18 958.2 144 1024 208
19 1106.7 188 4 1360
20 1106.8 188 2 1360
21 1361.9 176 4 1616
22 1362 178 2 1616
23 1362.2 186 2 1616

```

Figure 7.3 : Liste de propositions générées par KPPV.

L'algorithme CP-KPPV puise ses échantillons de données ou éléments dans les disponibilités de bande passante. Nous remarquons dans la (figure 7.3) que les disponibilités commencent juste après le temps de début de la requête + un créneau. La première colonne n'est affichée que pour identifier les lignes. La deuxième montre la distance calculée dont la plus petite valeur identifie la meilleure ou première proposition disponible. Les autres colonnes qui restent représentent respectivement le temps de début, la durée et la bande passante disponible.

Dans cet exemple les poids sur les attributs dans la formule de distance sont égaux à 1. Nous n'avons pas discriminé un paramètre par rapport à l'autre. Nous trouvons des durées plus longues avec la même bande passante et un même temps de début (par exemple les lignes 5, 8, 12, 14..). Normalement un filtrage doit être fait sur cet

échantillon afin d'éliminer les durées les plus petites qui représentent les premiers niveaux d'agrégations (nous avons laissé les petites durées pour monter l'utilité du filtrage dans cet algorithme). Les durées des agrégations sont toutes des multiples de 2, ce qui se traduit par une agrégation faite par couple de 2 créneaux. D'où des durées de 2, 4, 8, 16, 32, 64, 128, 256, 512. Les durées inférieures à la durée de la requête sont éliminées du calcul de la distance.

7.5 Les résultats

Les expérimentations sont ciblées afin de mesurer certains paramètres importants. Parmi ces paramètres, on trouve : le taux de rejet, taux de réservation et le temps de réponse. On suppose que le temps 0 minute est l'origine et à l'instant 86 la première requête est traitée par le système, ceci est dû à un intervalle de 26 dans lequel aucune requête n'est générée par la distribution normale, ce n'est qu'à la 26^{ème} minutes que commence l'arrivée des demandes. L'arrivée des requêtes est comprise entre les minutes 86 et 273. Le maximum de requêtes est situé entre 166 et 193 inclusivement. Dans cet intervalle les requêtes atteignent le maximum de 15 requêtes par minute (pendant environ 28 minutes, ce qui donne un nombre de requêtes totales de 375 requêtes). Le nombre total de requêtes traitées entre 86 et 273 est de 1473 requêtes.

Les graphes suivants représentent les résultats de l'évaluation des trois algorithmes CP-KPPV, CPA-NAFUR, et l'algorithme de réservation classique.

7.5.1 Le taux de rejets

Pour les trois algorithmes on remarque que le taux de rejets (requêtes non acceptées) finit par croître à partir de la 138^{ème} minute. Pour les quatre valeurs de MDP choisies, on remarque que le temps de début de rejet des requêtes est retardé en fonction de la valeur de MDP. Pour les valeurs de MDP = 60, 120 et 240 minutes, les rejets sont retardés de 5 minutes, 23 minutes et 50 minutes respectivement, en les comparant avec les rejets de l'algorithme régulier. Les algorithmes CPA-NAFUR et CP-KPPV donnent à 99.5% les mêmes résultats en ce qui concerne le taux de rejets. (figure 7.4).

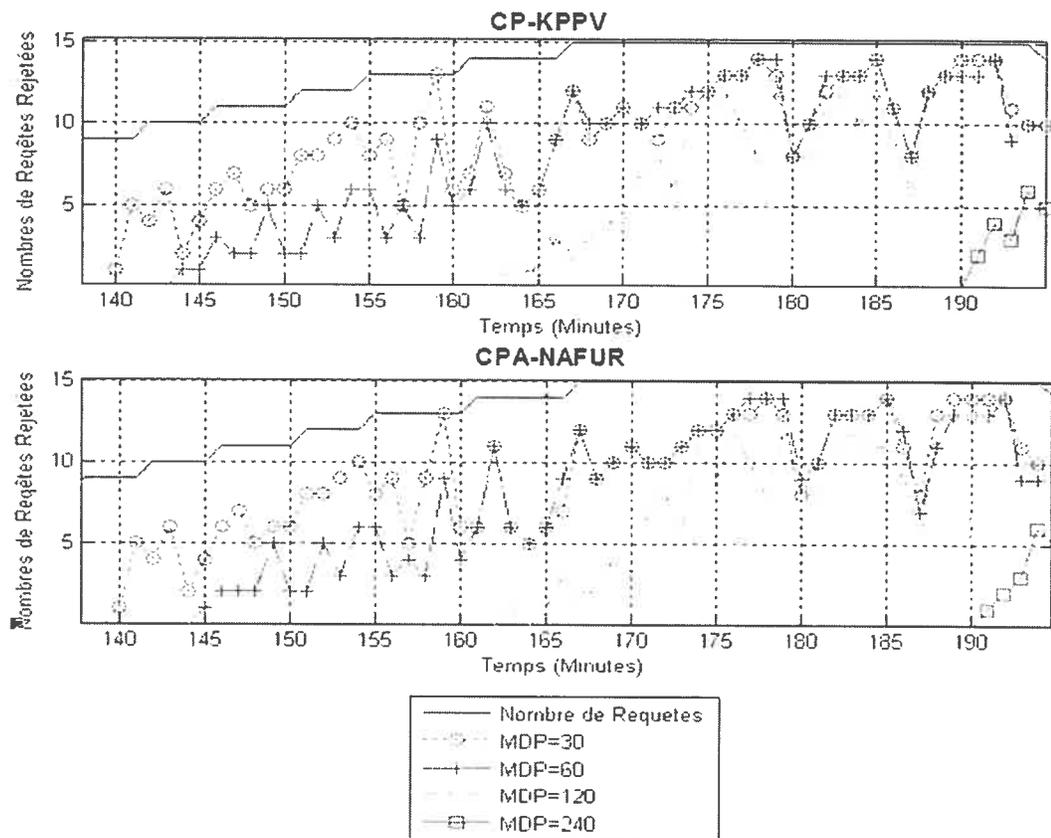


Figure 7.4 : Taux de rejets pour différents MDP pour CP-KPPV et CPA-NAFUR.

MDP (Minutes)	Algorithme Régulier	Algorithme CPA-NAFUR	Algorithme CP-KPPV
30	982	910	911
60	982	825	823
120	982	662	659
240	982	345	341
480	982	0	0

Table 7.2 : Nombre de rejets des principales méthodes sur différents MDP. Sur 1476 requêtes.

Nous remarquons dans la table 7.2 que les résultats sont très étroits entre CPA-NAFUR et CP-KPPV ; nous concluons simplement que les deux algorithmes ont les

mêmes résultats pour les mêmes MDP. Ces résultats sont largement supérieurs à ceux de l'algorithme régulier (Classique).

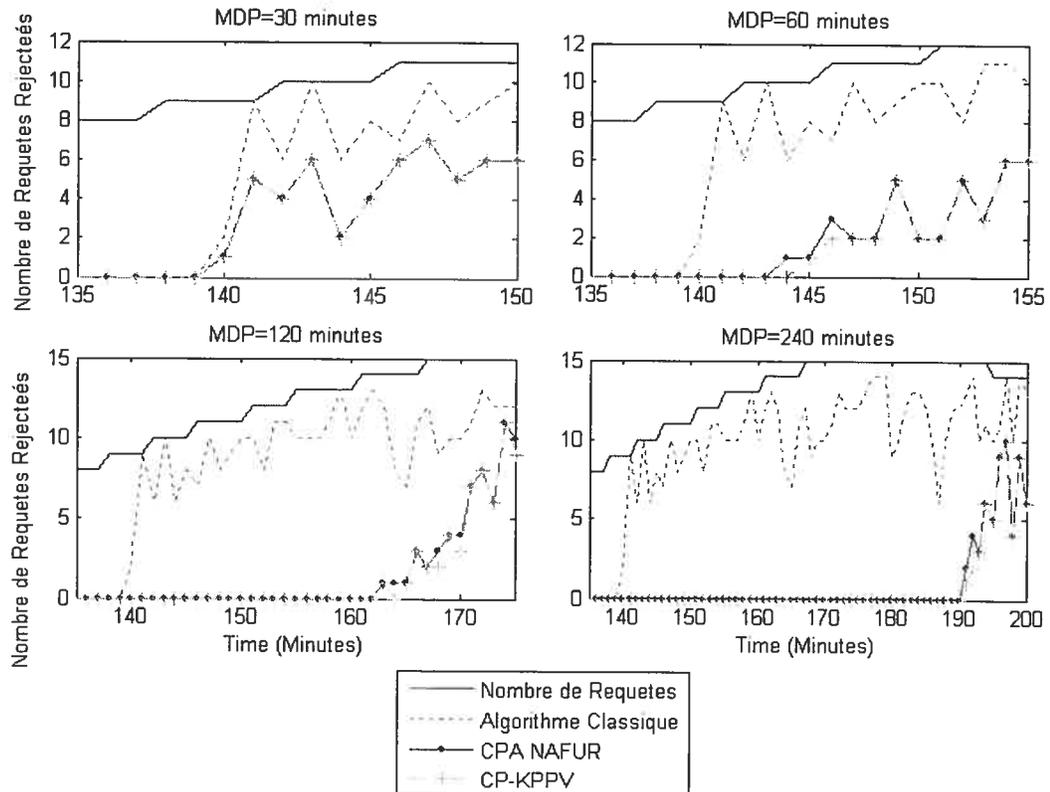


Figure 7.5 : Nombre de rejets avec les différents algorithmes sur différents MDP

7.5.2 Le temps de réponse

La figure 7.7 montre clairement la supériorité de CP-KPPV en termes de temps de réponse par rapport à CPA-NAFUR pour toutes les valeurs de MDP. Évidemment, le temps de réponse de l'algorithme de réservation classique est le plus court ; ceci s'explique par le fait que cet algorithme ne calcule aucune proposition alternative quand la requête ne peut pas être accommodée. La figure 7.5 montre aussi que quand le MDP augmente, CPA-NAFUR prend beaucoup de temps afin de balayer l'espace des disponibilités pour trouver des propositions.

MDP (Minutes)	Moyenne Temps de Réponse (Micro Seconds)		KPPV -	CPA
	CP-KPPV	CPA-NAFUR	CPA (%)	KPPV (%)
30	98,61	71,90	137%	73%
60	127,10	125,22	101%	99%
120	173,36	274,59	63%	158%
240	281,04	755,16	37%	269%
480	379,47	2459,99	15%	648%

Table 7.3 Temps de réponse CP-KPPV et CPA-NAFUR

Moyenne de temps de réponse CPA-NAFUR et CP KPPV

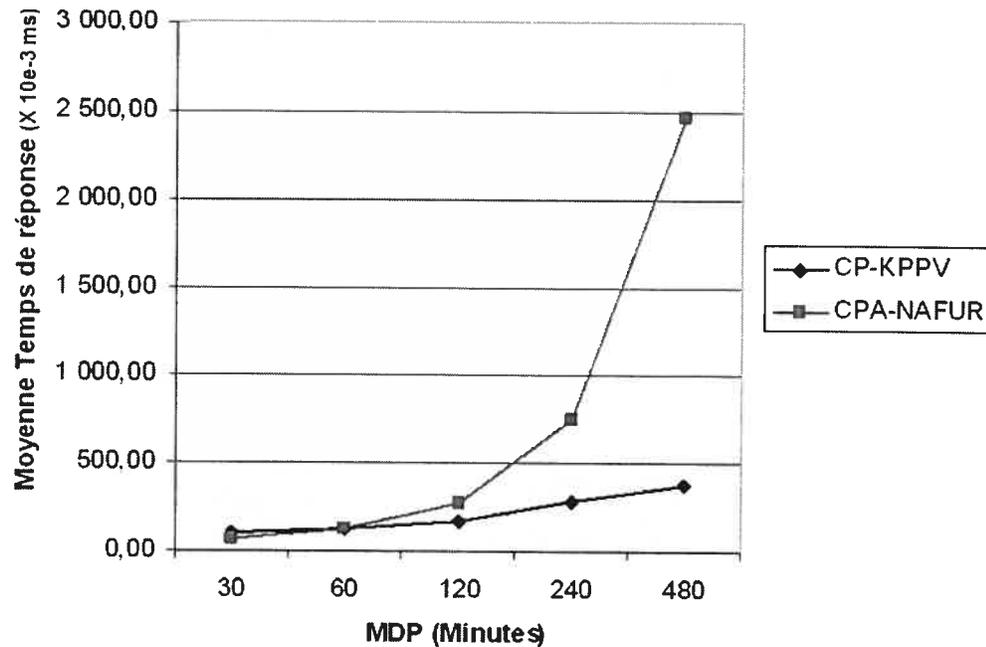


Figure 7.6 : Temps de réponse moyen CP-KPPV et CPA-NAFUR sur différents MDP

Le meilleur temps de réponse est toujours associé à l'algorithme régulier, mais au dépend d'un taux de rejet élevé. Selon la table 7.3, pour différents MDP, nous avons remarqué que l'algorithme CP-KPPV réduit son temps de réponse chaque fois que les MDP grandissent (voir aussi la figure 7.6). Par contre, l'algorithme CPA-NAFUR perd sa performance de temps de réponse pendant la recherche de propositions dès que MDP

dépasse les 60 minutes parce qu'il parcourt un intervalle plus grand à la recherche de propositions ce qui cause des temps de réponse plus long générés par le nombre de comparaisons faites. En conclusion, nous pouvons affirmer que CP-KPPV est très supérieur, en termes de temps de réponse, à CPA-NAFUR pour de grandes valeurs de MDP. La différence varie de 100% et peut atteindre 650% pour des valeurs de MDP entre 60 et 480. Cependant, pour de petites valeurs (par exemple : entre 30 et 60) de MDP, CPA-NAFUR produit des temps de réponses légèrement inférieures à celles de CP-KPPV.

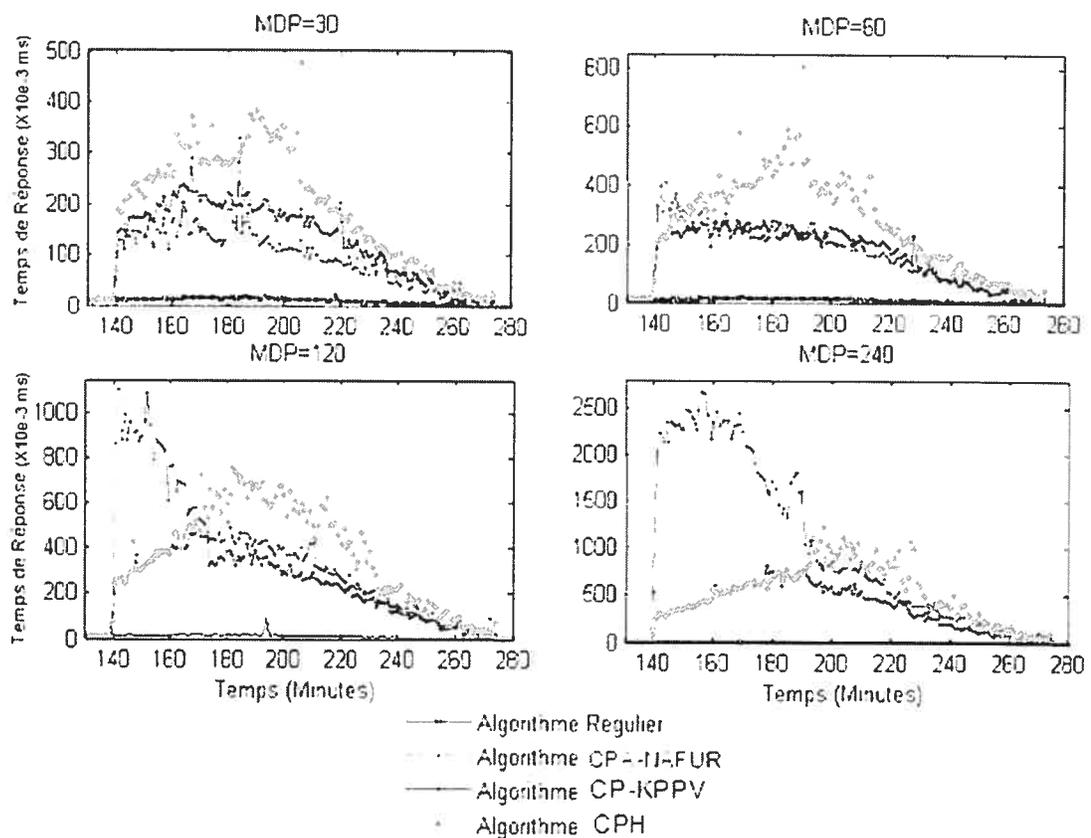


Figure 7.7 : Le temps de réponse pour différents MDP et algorithmes.

Dans la figure 7.7 on remarque que l'algorithme hybride CPH prend, dans la majorité des cas, beaucoup de temps de calcul d'où un temps de réponse assez grand par rapport aux autres algorithmes, mais cela s'explique facilement par le fait qu'il exécute dans le cas d'un rejet les algorithmes CP-KPPV en premier lieu et CPA-NAFUR dans le

cas où CP-KPPV ne trouve pas de propositions. Nous avons testé cet algorithme dans le but de voir les performances des deux algorithmes ensemble.

7.5.3 Flexibilité sur la bande passante

Dans le cas de rejet, CP-KPPV et CPA peuvent proposer une valeur de bande passante inférieure à celle demandée ; par exemple, le client accepte qu'il soit servi de seulement 80 % de ce qu'il a demandé. Pour comparer ces deux algorithmes, nous avons expérimenté avec trois paramètres de performance avec un MDP de 30 minutes ; ces paramètres sont : taux de rejet, taux de réservation, et le temps de réponse.

Temps de réponse

La figure 7.8 montre que l'algorithme CP-KPPV a un temps de réponse plus grand que celui de l'algorithme CPA-NAFUR en ce qui concerne un MDP de 30 minutes. Quand il ya flexibilité sur bande passante, le temps de réponse de CPA devient beaucoup plus important que CP-KPPV (voir figure 7.8). Ceci s'explique par le fait que CPA doit parcourir plus de plages de données à la recherche d'une proposition avec ceux définis par la flexibilité, sans omettre le cas où il peut trouver mieux. En ce qui concerne CP-KPPV, ce n'est qu'un simple calcul qui va inclure les valeurs qui sont comprises dans l'intervalle défini par la flexibilité. La différence en termes de temps de réponse entre les deux algorithmes est considérable (environ 100% voir figure 7.8 avec flexibilité) dans le cas de flexibilité sur la bande passante.

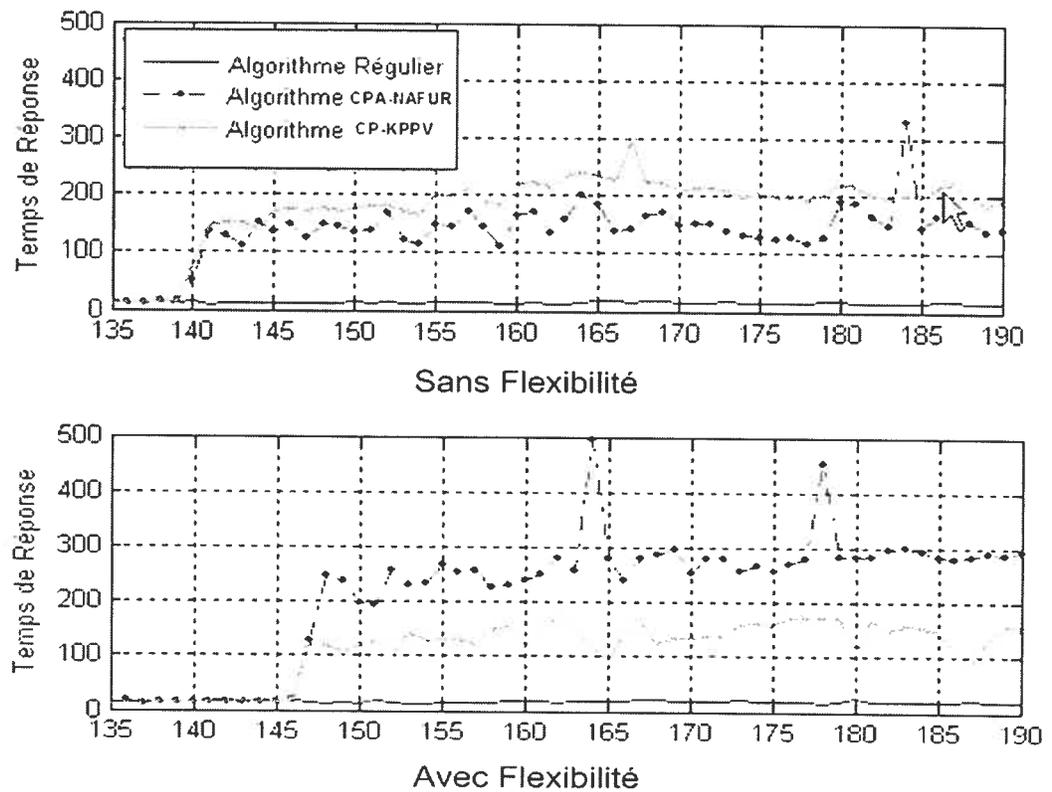
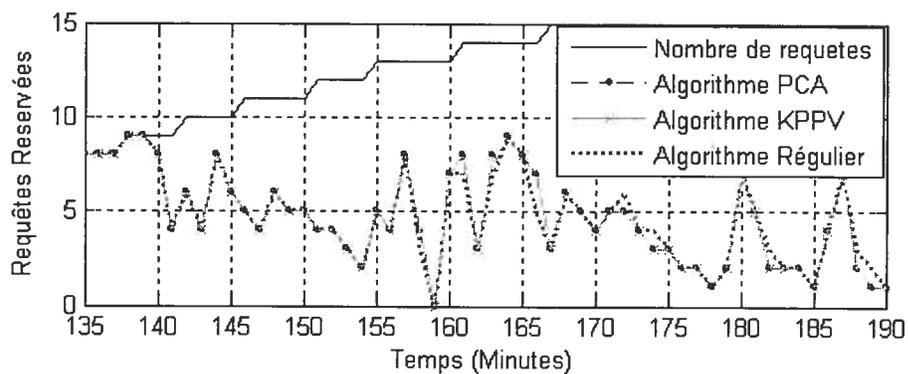


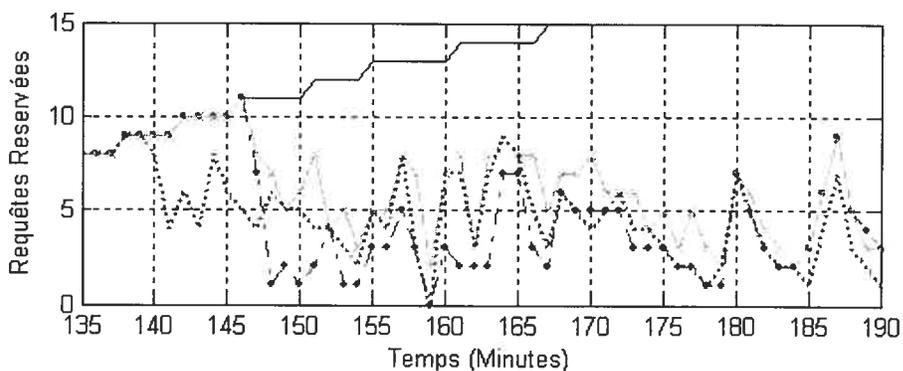
Figure 7.8: Temps de Réponse. flexibilité sur la bande passante. MDP=30.

Taux de réservations

Dans la figure 7.9 nous remarquons que les 2 algorithmes donnent du 100 % de réservation jusqu'à la 145^e minute. Pour le cas de sans flexibilité, c'est à partir de la 139^e minute que les rejets commencent. D'où le gain de 6 minutes de plus de réservations 100 % pour un MDP de 30 minutes. Il n'y a pas de grandes différences avec l'algorithme classique, car le MDP est de 30 minutes, ce qui diminue les disponibilités sur cet intervalle.



Sans flexibilité



Avec flexibilité

Figure 7.9 : Taux de réservation et flexibilité sur la bande passante. MDP=30

La figure 7.10 confirme que les rejets sont différés jusqu'à la 146^e minute par rapport au cas sans flexibilité pour les algorithmes CP-KPPV et CPA-NAFUR. Un avantage pour CPA-NAFUR par rapport à CP-KPPV est mis en évidence dans la figure 7.10; en effet, CPA-NAFUR rejette en moyenne 25% de requêtes moins que CP-KPPV.

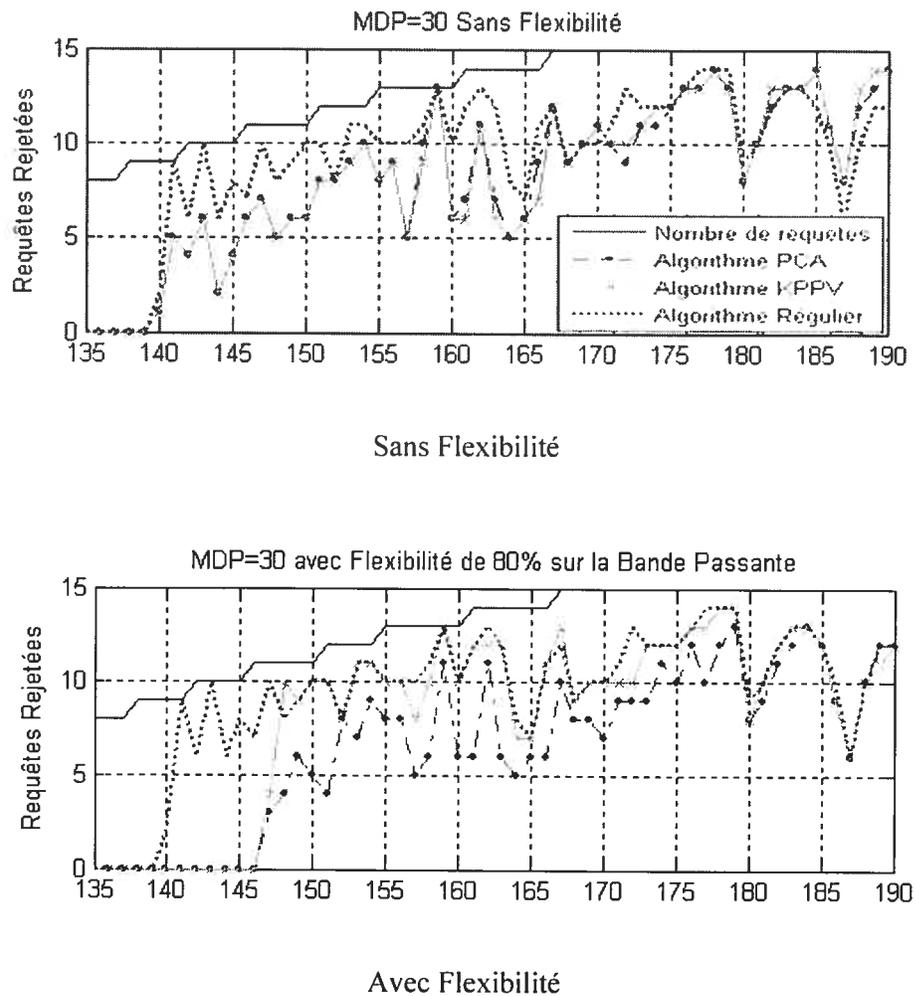


Figure 7. 10 : Taux de rejets MDP=30 avec et sans flexibilité sur la bande passante.

7.6 Discussions

Ces expérimentations nous ont montré que l'algorithme CP-KPPV donne des résultats similaires à celles de CPA-NAFUR, mais il est encore beaucoup mieux au niveau du temps de réponse quand il s'agit de valeurs de MDP supérieures à 60 minutes (et aussi dans le cas de flexibilité de la bande passante pour des valeurs arbitraires de MDP ; cependant, dans ce cas, CPA-NAFUR rejette moins de requêtes que CP-KPPV). Ce gain de performance ne profite pas aux applications où les clients ne voudraient pas que leurs demandes soient différées de plus d'une heure de la requête initiale. Mais pour

d'autres applications, il est préférable de planifier les demandes avec des valeurs de MDP beaucoup plus grandes. Ces demandes seront servies dans les intervalles définis dans le futur, au cas où le système ne pourrait faire une réservation régulière. Citons par exemple, des réunions qui utilisent des services de visioconférence qui sont planifiés à l'avance ou les applications de calcul distribué (grid computing).

Les expérimentations ont aussi montré que la variation du MDP et la flexibilité, en termes de la bande passante à réserver, permettent à CP-KPPV de performer mieux par rapport à CPA-NAFUR et à l'algorithme de réservation classique. Cette performance est surtout concentrée au niveau du temps de réponse qui est largement inférieur grâce à une structure de données qui lui permet d'agrèger les disponibilités pour une performance de recherche de propositions et des mises à jour.

La variation des autres paramètres comme la durée peut donner lieu à des applications intéressantes de CP-KPPV. Par exemple, pour des services de téléchargements de grandes quantités de données qui nécessitent de la bande passante et une durée qui dépendra de la bande passante (une large bande passante avec une petite durée ou une petite bande passante avec une grande durée). En réponse à une requête pour supporter un téléchargement qui nécessite 30 minutes avec une bande passante de 6 Mo/s, CP-KPPV produira des propositions, en cas de rejet, comme (Durée 64 minutes, bande passante 3 Mo/s), (Durée 128 minutes, bande passante 1.5 Mo/s), etc. Nous remarquons que les durées dans ces propositions sont des multiples de 2 parce que la valeur du créneau est égale à une minute.

Finalement, il faut noter que CP-KPPV ne se limite pas seulement aux trois paramètres de la requête $R = (ST, Du, B)$; il peut fonctionner avec un nombre plus grand de paramètres au besoin.

7.7 Conclusion

Nous avons présenté dans ce chapitre les simulations que nous avons faites pour évaluer la performance des algorithmes de réservation proposés. Les performances au niveau des réservations montrent une supériorité de CPA-NAFUR et CP-KPPV par rapport à la méthode régulière. En termes de temps de réponse, CP-KPPV est supérieure, dans la majorité des cas, à CPA-NAFUR. En effet, la seule instance où

CPA-NAFUR performe légèrement mieux en termes de temps de réponse est le cas où les valeurs de MPD sont petites et il n'y pas de flexibilité dans la réservation de la bande passante demandée. Cependant, dans le cas de cette flexibilité, CPA-NAFUR rejette moins de requêtes que CP-KPPV.

Finalement, nous pouvons confirmer que CP-KPPV performe mieux que les CPA-NAFUR dans le cas de flexibilité dans le choix des autres paramètres de la requête (par exemple : durée). Finalement, nous notons que CP-KPPV peut être utilisé dans le cas où la requête contient plus des trois paramètres (temps de début, durée, bande passante) utilisés dans ce projet.

Chapitre 8

Conclusion

8.1 Discussion

Nous avons présenté dans ce mémoire une solution au problème du soutien de la QoS réseau de bout en bout pour les services Web. Notre approche ne propose aucun changement dans le réseau ni dans les services Web actuellement déployés. Elle présente un service Web (NRM) gestionnaire de ressources réseau qui garantit, si c'est possible, la QoS du chemin réseau entre le service Web et son demandeur. Cette garantie dépend des fonctionnalités et des technologies de réseau supportant la QoS, si le réseau ne les offre pas, NRM utilisera des mesures réseau afin d'estimer la QoS entre les deux points du chemin réseau.

Nous avons aussi présenté une version étendue de NRM basé sur NAFUR [2] et expérimenté une approche basée sur KPPV qui est une technique, de raisonnement à base de cas, utilisée dans la fouille de données. Cette approche permet de produire des propositions et des alternatives à des requêtes qui sont normalement rejetées par les algorithmes existants. Pour déterminer ces propositions, nous recherchons les disponibilités futures qui peuvent être utilisées comme données d'entrées pour l'algorithme CP-KPPV. Nous exécutons CP-KPPV sur cet échantillon de disponibilités afin de ressortir celles qui sont les plus proches de la requête de réservation. Toutefois, nous pensons que l'étude menée ici est un premier pas dans l'utilisation de techniques de fouille de données afin de résoudre un problème de réservation de ressources réseau à l'avance.

L'avantage de CP-KPPV réside non seulement dans l'utilisation d'une structure de données simple mais aussi dans l'agrégation de l'ASP qui produit les disponibilités futures en forme d'agrégats de bande passante. L'évaluation de ces disponibilités par une distance de similarité simple permet de produire des propositions très proches de la requête, comparativement à CPA-NAFUR, en plus d'avoir un meilleur temps de

réponse. En effet, avec CP-KPPV les utilisateurs pourraient économiser le temps consacré à la recherche de services Web suivant leur disponibilité en spécifiant leurs préférences temps et qualité. Dans le cas de non-disponibilités de ressources, ces préférences seront utilisées pour offrir les meilleures alternatives. Ils peuvent également élaborer leurs requêtes en spécifiant les attributs de qualité de service qu'ils sélectionnent. En conséquence, CP-KPPV peut être déployé pour les utilisateurs afin qu'ils puissent eux même réserver des ressources pour des besoins plus spécifiques et précis moyennant un certain coût.

8.2 Perspectives

Comme travaux futurs pour ce projet, nous pensons améliorer les performances de CP-KPPV afin qu'il donne plus de propositions proches des demandes en la dotant de modules supplémentaires. Ces modules permettront d'utiliser les profils des applications qui utilisent des ressources réseau et de les associer avec CP-KPPV pour réaliser un système de recommandations de propositions basé sur ces profils (par exemple la classification des services Web selon leur besoin en ressources réseau). L'idée est de permettre à ce système de réservation/propositions de faire des prédictions pour mieux proposer. Ce module fera parti des fonctionnalités de NRM qui pourra l'utiliser pour mieux aider le Courtier de QdS-SW à prendre des décisions en ce qui concerne les services Web avec la meilleure QdS.

D'un autre côté, l'utilisation des techniques de fouille de données a permis de voir le potentiel d'outils existant pouvant être utilisés sur les données de mesures et de statistiques disponibles au niveau de la base de données NRM, afin de prédire et d'estimer la QdS dans le futur et surtout de produire des propositions qui vont avec les requêtes. On peut aussi dire que les fournisseurs des services Web peuvent eux aussi réserver des ressources dans le futur s'ils savent à l'avance la QdS réseau qu'il doivent fournir pour attirer le maximum de requêtes sur leurs services Web. La base de données NRM stocke toutes les informations nécessaires en prévision de cette tâche future.

Bibliographie

- [1] A. Brodnik, A. Nilsson, Data Structure for a Time-Based Bandwidth Reservations Problem, July 10, 2004.
- [2] A. Hafid, G. v. Bochmann, R. Dssouli, A quality of service negotiation approach with future reservations NAFUR: a detailed study. Computers Networks and ISDN Systems. 1998.
- [3] A. Hafid, M. Maach, J. Drissi, DARSION: A Distributed Advance Reservation System for Interconnected Optical Networks, In Proceedings of the 9th IEEE/IFIP Optical Network Design and Modeling (ONDM'05). 2005.
- [4] A. Keller and H. Ludwig, The WSLA framework: Specifying and Monitoring Service Level Agreements for Web Services. IBM Research Report. 2002.
- [5] A. s. Ali, O. f. Rana, R. Al-Ali, D. w. Walker, UDDIe: An Extended Registry for Web Services International Journal of Simulations Systems, Science & Technology, 5 (5), December 2004.
- [6] A.Hafid, G.v. Bochmann, An Approach to Quality of Service Management in Distributed Multimedia Application: Design and an Implementation.
- [7] BEA, WebLogic platform, <http://www.bea.com>, 2005.
- [8] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and Weiss, W., An Architecture for Differentiated Services. December 1998. RFC 2475.
- [9] C. Chassagne., Qualité de Service dans l'Internet, CNRS-UREC, Août 1998, <http://www.urec.fr/metrologie/article-qos.html>.
- [10] Codec, définition, <http://fr.wikipedia.org/wiki/Codec>
- [11] Computer Industry Almanac, <http://www.c-i-a.com/pr0904.htm>, September 3, 2004.
- [12] D. Clark, W. Fang, Explicit Allocation of Best Effort Packet Delivery Service, IEEE/ACM, Transactions on Networking, August 1998, Vol. 6, No. 4, pp. 362-373.

- [13] DAML-S Coalition, DAML-S: Web Service Description for the Semantic Web. In Proceeding of the International Semantic Web Conference. 2002.
- [14] E. Cerami, Web Services Essentials : Distributed Applications with XML-RPC, SOAP, UDDI & WSDL , First Edition, February 2002.
- [15] Encyclopédie WIKIPEDIA, <http://fr.wikipedia.org>, septembre, 2005.
- [16] F. Denis, R. Gilleron, Apprentissage à partir d'exemples. Technical report, Grappa - Université de Lille 3, 1999.
- [17] G. Mardente, M. Mellia, C. Casetti, Admission control and path allocation for SLAs in DiffServ networks, High Performance Switching and Routing, 2004. HPSR. 2004 Workshop on 2004 Page(s):38 – 42.
- [18] G. Stattenberger and T. Braun, Performance of a Bandwidth Broker for DiffServ Networks, TR, Institute of Computer Science and Applied Mathematics, University of Bern, Switzerland. 2003.
- [19] H Ludwig, Web services QoS: external SLAs and internal policies or: how do we deliver what we promise?, 4th International Conference on Web Information Systems Engineering Workshops, 2003. Proceedings, Pages: 115 – 120.
- [20] H. j. Chao, X. Guo, Quality of Service Control in High-Speed Networks, John Wiley & Sons, Inc., November 2001, 448 pages, page
- [21] H. y. Tyan, J-Sim: A Java Network Simulation platform. <http://www.j-sim.org/>
- [22] I. Taleb, A. Hafid, J. Driss, A Quality of Service Negotiation Approach with Future Reservations (NAFUR): An Evaluation of Proposal's Generation Algorithms.
- [23] I. Taleb, A. Hafid, M. a. Serhani, QoS-aware Multimedia Web Services Architecture. WEBIST 2005: 133-139.
- [24] IBM, Websphere, <http://www-306.ibm.com/software/Websphere/>
- [25] IETF, Differentiated Services working group. <http://www.ietf.org/html-charters/diffserv-charter.html>

- [26] IETF, Integrated Services working group. <http://www.ietf.org/html-charters/intserv-charter.html>
- [27] IETF, IntServ & DiffServ WGs., Internet Engineering Task Force Society, (<http://www.ietf.org>).
- [28] IETF, RTP: Real-time Transport Protocol, <http://www.ietf.org/html.charters/avt-charter.html>
- [29] J. Guillot, JDNNet, SLA/SLM : comment garantir la qualité de service, Avril, 2003, http://solutions.journaldunet.com/0304/030417_faqlaslm.shtml.
- [30] J. Han, M. Kamber, Data Mining: Concepts and Techniques. Morgan Kaufmann, 2002.
- [31] J. Heinanen, R. Guerin, RFC 2698: A Two Rate Three Color Marker, Juin 1999.
- [32] J. m. Adamo, Data Mining for Association Rules and Sequential Patterns. Springer, 2001.
- [33] J. Postel, RFC 768: User Datagram Protocol (UDP), <http://www.ietf.org/rfc/rfc0768.txt?number=768>, August 1980.
- [34] J. Rachlin, S. Kasif, S. Salzberg, and D.W. Aha, Towards a better understanding of memory-based reasoning systems. In Proceedings of the Eleventh International Conference on Machine Learning, p.242--250. Morgan Kaufmann, 1994.
- [35] J. Souza, S. Matwin, N. Japkowicz, Evaluating Data Mining Models: A Pattern Language. In Proceedings of the 9th Conference on Pattern Language of Programs (PLOP'2002), 2002.
- [36] J. Wroclawski, RFC 2211: Specification of the Controlled-Load Network Element Service. Septembre 1997.
- [37] J. Shin, D. c. Lee, C. c. Jay Kuo, Quality of Service for Internet Multimedia, Prentice Hall PTR, July 24, 2003, 205 pages, page 14.
- [38] K. Bakour, BLED : Système d'aide à la recherche d'informations sur Internet , Université de Montréal, Avril 2005.

- [39] K. Januszewski, E. Mooney, UDDI Version 3 Features List, http://uddi.org/pubs/uddi_v3_features.htm, Juillet 2005.
- [40] L. Andersson, et al., MPLS: Label Distribution Protocol, <http://www.ietf.org/internet-drafts/draft-ietf-mpls-rfc3036bis-03.txt>, IETF, October 2005.
- [41] L. Bos, S. Leroy, Toward an all-IP-based UMTS system architecture, Network, IEEE Publication Date: Jan/Feb 2001, Volume: 15, Issue: 1, page(s): 36-45.
- [42] L. Breiman, J. h. Friedman, R. a. Olshen, and C. j. Stone, Classification and regression trees. Technical report, Wadsworth International, Monterey, CA, 1984.
- [43] L. o. Burchard, Analysis of Data Structures for Admission Control of Advance Reservation Requests. In IEEE Transactions On Knowledge And Data Engineering, Vol. 17, No. 3, March 2005.
- [44] L. o. Burchard, Advance Reservations of Bandwidth in Computer Networks, School of Electrical Engineering and Computer Sciences, 2004-07-14.
- [45] L. Toutain, J.-M. Bonnin , O. Medina, La qualité de service dans l'Internet, Algotel'2000, 2000
- [46] L. Yuan, C. k. Tham, A. I. Ananda, A Probing Approach for Effective Distributed Resource Reservation. In Proceedings of 2nd International Workshop on QoS in Multi-Service IP Networks (QoSIP 2003), Milan, Italy, 24-26 Feb 2003.
- [47] M. a. Serhani, A. Hafid, H. Sahraoui, A. Benharref, QoS broker-based architecture for Web Services, NOTERE 2004.
- [48] M. a. Serhani, R. Dssouli, A. Hafid, H. Sahraoui, A QoS Broker Based Architecture for Efficient Web Services Selection Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on 11-15 July 2005 Page(s):113– 120.

- [49] M. A. Serhani, R. Dssouli, H. Sahraoui, A. Benharref, M. E. Badidi, QoS integration in value added Web services, the Second International Conference on Innovations in Information Technology (IIT'05), 26-28 Septembre 2005, UAE.
- [50] M. Kanatardzic, Data Mining Concepts, Models, Methods, and Algorithms. Wiley - Interscience, 2003.
- [51] M. Tian, A. Gramm, H. Ritter, and J. Schiller. Efficient Selection and Monitoring of QoS-aware Web services with the WS-QoS Framework. 2004 IEEE/WIC/ACM International Conference on Web Intelligence (WI'04), September 2004, Beijing, China
- [52] M. Tian, A. Gramm, T. Naumowicz, H. Ritter, J. Schiller, A Concept for QoS Integration in Web Services 1st Web Services Quality Workshop (WQW 2003), in conjunction with 4th WISE, Rome, Italy, December 2003.
- [53] O. Schelén, A. Nilsson, J. Norrgård, S. Pink, Performance of QoS Agents for Provisioning Network Resources. In Proceedings of IFIP Seventh International Workshop on Quality of Service (IWQoS'99), London, UK, June 1999.
- [54] Observer, Network Instruments, <http://www.networkinstruments.fr>.
- [55] P. Adriaans, D. Zantinge, Data Mining, Addison-Wesley, 1996.
- [56] P. Ferguson, G. Huston, Quality of Service: Delivering QoS on the Internet and in Corporate Networks, John Wiley & Sons, Inc., 1998.
- [57] R. Al-Ali, S. Sohail, O. f. Rana, A. Hafid, G. von Laszewski, K. Amin, S. Jha and D. Walker. Network QoS Provision for Distributed Grid Applications.
- [58] R. Braden, D. Clark, S. Shenker, Integrated Services in the Internet Architecture: an Overview, 1994. RFC1633.
- [59] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification, RFC 2205, September 1997, Proposed Standard.
- [60] R. Fielding, et al., HTTP: Hypertext Transfer, <http://www.w3.org/Protocols/rfc2616/rfc2616.html>, June 1999.

- [61] R. Jain, Digital Equipments Corp., Congestion Control in Computer Networks: Issues and Trends, IEEE Network Magazine, May 1990, pp.24-30.
- [62] S. Shenker, C. Partridge, R. Guerin, RFC 2212:Specification of Guaranteed Quality of Service. Septembre 1997.
- [63] Softeam, Web Services,
http://www.softeam.fr/technologies_Web_services.php, juillet 2005.
- [64] Statsoft, k-Nearest Neighbors, <http://www.statsoft.com/textbook/stknn.html>
- [65] Sun Microsystems, J2EE, <http://java.sun.com/j2ee/>, 2005.
- [66] Sun Microsystems, JMF: Java Media Framework API,
<http://java.sun.com/products/java-media/jmf/>, 2005.
- [67] T. m. Cover and P.e. Hart, Nearest Neighbor Pattern Classification, IEEE Transactions on Information Theory, Volume IT-13(1), pp.21-27, 1967.
- [68] V. Jacobson, K. Nichols, K. Poduri, RFC 2598: An Expedited Forwarding PHB. Juin 1999.
- [69] V. Tasic, B. Pagurek, K. Patel, WSOL- A Language for the Formal Specification of Classes of Service for Web Services. Proc. of ICWS'03, June 23-26 2003: LV, Nevada, USA. 2003.
- [70] W. Sadiq, F. a. Cummins, Developing Business Systems With Corba: The Key to Enterprise Integration, Cambridge University Press, 1998 - 287 pages.
- [71] W3C, The World Wide Web Consortium, HTTP - Hypertext Transfer Protocol, <http://www.w3.org/Protocols/http>.
- [72] W3C, The World Wide Web Consortium, QoS for Web Services: Requirements and Possible Approaches, W3C Working Group Note 25 November 2003. <http://www.w3c.or.kr/kr-office/TR/2003/NOTE-WS-QoS-20031125/>
- [73] W3C, The World Wide Web Consortium, Web Services Architecture, W3C Working Group Note 11 February 2004. <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211>

- [74] Weiss and Kulikowski, K-Nearest Neighbor, 1990. [1] A. Brodnik, A. Nilsson, Data Structure for a Time-Based Bandwidth Reservations Problem, July 10, 2004.