

Nom Pascale PRIMET

Laboratoire : Laboratoire RESAM,ENS Lyon - UCB Lyon et Action RESO INRIA Rhône-Alpes,

Téléphone : +33 (0)4 72 72 82 28

Télécopie : +33 (0)4 78 28 74 42

Mél : [Pascale.Primet]@ens-lyon.fr

Document d'Habilitation à Diriger des Recherches

*Contribution au Support réseau des applications réparties
Qualité de Service : pour un réseau sensible aux flux*

Pascale PRIMET

20/04/2002

Remerciements

Table des matières

1	Introduction	5
1.1	Parcours	5
1.2	Evolution des applications réparties	5
1.2.1	Usage des réseaux	6
1.2.2	Applications coopératives et Grilles de calcul	6
1.3	Evolution des réseaux	7
1.3.1	Evolution des infrastructures	7
1.3.2	Evolution des protocoles	7
1.3.3	Problématique de la Qualité de Service	8
1.3.4	Réseaux programmables	8
1.4	Contexte de recherche	9
1.5	Démarche d'étude	11
1.6	Organisation du document	12
I	Applications coopératives	13
2	CSCW : CoTools et le partage de l'espace de travail virtuel	14
2.1	Introduction	14
2.2	Les systèmes coopératifs	14
2.2.1	Evolution des systèmes coopératifs	14
2.2.2	Construction et exécution des applications coopératives	15
2.3	La boîte à outils CoTools	15
2.3.1	Modélisation et architecture	15
2.3.2	Principe de transparence et flexibilité	16
2.3.3	Contrôle de concurrence flexible	18
2.4	Modèles d'architecture d'applications coopératives	21
2.4.1	Modèle AMF-C	21
2.4.2	Modeleur géométrique 3D multi-utilisateur GEO	22
2.4.3	L'application télé-pointeur TéléPTR	22
2.5	Conclusion	23
II	La Qualité de service dans les réseaux	27
3	Problématique de la Qualité de Service	28
3.1	Introduction	28
3.2	Qualité de service et travail coopératif	29
3.2.1	Evaluation de la qualité perçue dans un réseau ATM	29
3.2.2	Influence de la Qualité de Service sur le processus de coopération	31
3.3	Paramètres de performances d'un réseau	31

3.3.1	Paramètres de débit	32
3.3.2	Paramètres de délai	32
3.3.3	Paramètres de fiabilité	33
3.4	Besoins de Qualité de service des applications	33
3.4.1	Dimensions <i>utilisateur</i> de la qualité de service	33
3.4.2	Dimension temporelle	34
3.4.3	Hétérogénéité des besoins	35
3.4.4	Modèles de services et spécification des besoins	36
3.5	Conclusion	37
4	Qualité de service dans les réseaux IP	39
4.1	Introduction	39
4.1.1	Problématique IP et Qualité de Service	39
4.1.2	Considérations architecturales	40
4.2	Solutions réseaux	40
4.2.1	Mécanismes de base	40
4.2.2	Architectures de QoS-IP standard	45
4.2.3	Modèles alternatifs et Balanced Forwarding	48
4.3	Les techniques adaptatives	52
4.3.1	Classification des techniques adaptatives	53
4.3.2	Le transport TCP	55
4.3.3	Transport des applications temps-réel	55
4.3.4	Netstre@mer : vers une adaptation générique	57
4.4	Conclusion	60
5	Propositions pour un réseau sensible aux flux	62
5.1	Introduction	62
5.2	Le modèle EDS	63
5.2.1	Objectifs et hypothèses	63
5.2.2	Principes de base	64
5.2.3	Transport différencié sur EDS	67
5.3	Approche Active pour la Qualité de Service	71
5.3.1	Environnement actif	71
5.3.2	QoS active dans le plan contrôle	73
5.3.3	QoS active dans le plan données : SQoS et ADS	75
5.4	Perspectives de la QoS active	77
5.5	Conclusions	78
III	Grilles de calcul	79
6	Réseaux et grilles de calcul	80
6.1	Communication dans les grilles de calcul	80
6.1.1	Concept de grille	80
6.1.2	Problématique réseaux des grilles de calcul	81
6.1.3	Modélisation du réseau de la grille : network element	82
6.2	Mesure des performances réseau de la grille	83
6.2.1	Architecture de mesure de performance	83
6.2.2	PCP et l'ordonnement des mesures	86
6.2.3	MapCenter et la visualisation de l'état de la grille	86
6.2.4	Prédiction des performances réseau : amélioration de NWS	87
6.3	Optimisation des performances des communications	88
6.3.1	Grille et QoS réseau	88

6.3.2	Transport haute performance	90
6.4	La grille active	91
6.4.1	Convergence des problématiques grilles et réseaux actifs	91
6.4.2	Middlehardware intelligent pour la grille	92
6.5	Déploiement de plate-formes expérimentales	94
6.5.1	Support réseau du projet DataGRID	94
6.5.2	Le projet E-toile	95
6.6	Conclusion	96
7	Conclusion	98
7.1	Bilan et perspectives scientifiques	98
7.1.1	Du besoin des applications...	98
7.1.2	...aux modèles d'architecture et de services différenciés	99
7.1.3	D'une grille <i>passive</i>	100
7.1.4	... à un réseau sensible à la grille et une grille active	100
7.2	Conclusion personnelle	100

Chapitre 1

Introduction

1.1 Parcours

Ce document présente les activités de recherche que j'ai menées depuis l'obtention de mon doctorat en 1988 à l'Institut National des Sciences Appliquées de Lyon. Ces activités se sont déroulées jusqu'en 1998 au laboratoire ICTT à l'Ecole Centrale de Lyon où je suis Maître de Conférences depuis 1989 puis au sein de la jeune équipe RESAM à l'Université Claude Bernard puis à l'Ecole Normale Supérieure de Lyon.

Cette synthèse de mon travail de recherche traite de l'interdépendance du réseau et des applications réparties. Ces deux domaines sont soumis à des contraintes d'hétérogénéité et de dynamique de plus en plus importants qui requièrent des solutions flexibles, robustes et extensibles. Au cours de mes travaux, j'ai étudié cette problématique d'interdépendance au travers de l'aspect particulier de la qualité de service en l'examinant alternativement du côté des applications et du côté du réseau.

Après ma thèse de doctorat, j'ai par deux fois réorienté ma thématique de recherche. J'ai tout d'abord travaillé dans le domaine des systèmes répartis et du support informatique du travail coopératif puis dans le domaine des réseaux haut débit et des protocoles de l'Internet. Depuis un an je conduis des travaux sur les réseaux des grilles de calcul. Ce sont principalement les travaux récents, qui sont détaillés dans ce document. Cependant, je tente de montrer comment l'ensemble de mes activités antérieures alimentent et éclairent mes réflexions et mes travaux actuels.

Durant ma thèse de doctorat, j'avais travaillé sur la problématique des entrées-sorties dans les systèmes d'exploitation temps-réel et je m'étais plus particulièrement focalisée sur les aspects architecturaux et le développement des logiciels de pilotage des cartes intelligentes de robots. J'ai ainsi étudié les communications à l'intérieur d'une architecture multiprocesseurs faiblement couplés et la gestion de flux et de processus aux contraintes hétérogènes dans un système temps-réel.

1.2 Evolution des applications réparties

Dès Arpanet, la technologie réseau a été créée dans le but d'utiliser des systèmes informatiques interconnectés pour supporter la collaboration humaine [91]. L'avènement du multimédia a permis d'envisager une collaboration plus conviviale, étroite et polymorphe. Mais introduire la qualité de service et les services différenciés nécessaires à ces usages, dans une infrastructure globale telle qu'Internet, reste un défi majeur et encore largement ouvert [56].

Dans cette introduction je présente l'évolution des applications réparties et des réseaux avec les contraintes auxquelles ils doivent faire face, puis la problématique de la qualité de service et le contexte dans lequel j'ai développé mes travaux

1.2.1 Usage des réseaux

Les réseaux de transmission de données ont pour fonction initiale le transport d'informations numériques entre des ordinateurs distants. Les trois principales utilisations de l'Internet furent d'abord l'accès distant, le transport de fichiers et la messagerie électronique. L'avènement du *World Wide Web* qui permet une navigation transparente au travers de milliards de données stockées à travers le monde, a fait exploser les réseaux et particulièrement la technologie IP. Aujourd'hui, de nouvelles applications voient le jour et se répandent. Les années 2000 sont marquées par la diffusion des applications de commerce électronique (e-business), l'apparition des applications pair à pair (P2P), de calcul global [47] et le déploiement de multiples grilles de calcul, (*Grid*¹) à travers le monde [2]. Chaque jour, Internet doit faire face à l'augmentation du volume mais aussi de l'hétérogénéité des flux. Des observations sur de longues périodes ont montré une grande variabilité des usages de l'Internet [157]. Certaines applications explosent comme le Web en 1993 ou Napster en 99. D'autres arrivent au premier plan puis disparaissent rapidement. Les applications traditionnelles de messagerie et de transfert de fichiers qui totalisaient 90% du trafic sur Internet, ont été largement dépassées par le trafic http des applications WEB et le trafic multimédia (voix, vidéo, musique) au milieu des années 90. Les applications coopératives exploitant les technologies multimédia (visio-conférence, mondes virtuels partagés), ont vu le jour au cours de cette dernière décennie. Ainsi les outils du Mbone [123] ont pu atteindre jusqu'à 50% du trafic dans certains réseaux en 1995 (Bellcore, LBNL) avant de passer en second plan. Ces applications ne se sont pas diffusées réellement car la qualité des réseaux n'était pas encore suffisante pour permettre une interaction confortable et de très nombreuses barrières humaines et techniques restent encore à lever. Les deux grandes familles d'utilisation de l'Internet sont l'usage domestique et l'utilisation professionnelle. Aujourd'hui, les performances et la sécurité offertes sont insuffisantes pour les usages financièrement critiques. La qualité de service et la sécurité sont certainement les deux verrous majeurs qui freinent la diffusion, dans le monde industriel, des applications de grilles de calcul et de travail coopératif sur Internet. Dans mes différents travaux j'ai examiné en détail quelles sont les contraintes inhérentes à chacun de ces deux domaines d'application. Je tâche de mettre en lumière les problèmes spécifiques et de montrer la limite des solutions réseau que l'on peut proposer dans ces deux contextes (chap. 2 et chap. 6).

1.2.2 Applications coopératives et Grilles de calcul

Pour certains auteurs [2], les applications coopératives sont considérées comme une classe d'application particulière des grilles. Ce sera peut-être le cas à moyen terme. Ces deux domaines interconnectent et regroupent virtuellement des ressources et des humains. Ils présentent tous deux un fort caractère collectif. Cependant, leurs buts divergent : celui des applications coopératives est de faire interagir des humains, celui des grilles est d'offrir un environnement d'exécution performant à des applications à besoins de calcul et de stockage élevés. Si les applications coopératives sont caractérisées par la présence simultanée d'utilisateurs, dans les grilles de calcul, les utilisateurs n'interagissent, à priori, pas directement entre eux. Il en résulte des besoins en terme de transport assez différents. Les contraintes des flux coopératifs sont celles des flux multimédia transportés et des délais d'interaction. Dans les grilles de calcul ou de données, le temps de réponse dépend de la dynamique propre de l'application et de ses contraintes de synchronisation internes. Les échelles de délai sont liées aux modèles de calculs et de communication utilisés. Pour les grilles de données en particulier, les débits requis peuvent être très importants. C'est le cas par exemple pour transporter les fichiers de données de plusieurs Gigaoctets des expériences de physique des hautes énergies [37]. La fiabilité du transport est différente selon les flux et les applications. Les équipements terminaux du CSCW et du Grid peuvent avoir des capacités de communication et de traitement très hétérogènes. Dans le CSCW, les postes utilisés vont des stations multimédia haute performances aux terminaux sans fil et les débits d'accès peuvent varier de quelques kilo-octets pour les accès par GSM à quelques méga-octets pour les liens ADSL. Dans les grilles de calcul, les équipements d'extrémité sont des supercalculateurs ou des fermes de calcul et les liens d'accès sont supérieurs à 10Mb/s et peuvent aller jusqu'à 1 voire 10Gb/s.

¹Globalisation des ressources informatiques distribuées - Michel Cosnard

1.3 Evolution des réseaux

1.3.1 Evolution des infrastructures

Au cours des vingt dernières années, l'infrastructure des réseaux longue distance a évolué très rapidement et offre aujourd'hui des débits extrêmement importants. Par exemple, le réseau fédérateur européen pour la recherche (GEANT), offre depuis le 1 décembre 2001 un débit de 1 à 10Gbit/s pour relier les principales capitales de l'Europe alors qu'en 1995 le débit du réseau ISDN large bande n'était encore que de 34Mb/s. Aux Etats-Unis, le réseau Abilène qui fédère plus de 180 universités américaines et 70 sociétés est basé sur une épine dorsale de plusieurs Gbits par secondes. On envisage pour un futur très proche des capacités de plusieurs dizaines voire centaines de Gbits/s. Cette augmentation formidable des débits au centre de l'infrastructure est rendue possible par les avancées technologiques des télécommunications optiques : généralisation de la fibre optique, modulation optique (WDM) et commutation de longueurs d'ondes. A la périphérie, les technologies classiques (Ethernet ou RTC) changent aussi d'échelle (1 à 10 Gbit/s pour Ethernet, de 5Kb/s à 8Mb/s pour l'ADSL). Ces liaisons filaires traditionnelles sont de plus en plus concurrencées par des liaisons herziennes ou radio qui favorisent l'expansion de l'informatique mobile. On assiste donc à la périphérie des réseaux fédérateurs surdimensionnés à l'explosion de réseaux d'accès aux caractéristiques de performances extrêmement hétérogènes. Les réseaux ont tout d'abord été des réseaux privés ou propriétaires, longue distance(SNA, Arpanet), puis courte distance (Ethernet, Token Ring). Enfin, l'Internet, réseau de réseaux, a permis de fédérer l'ensemble de ces réseaux sur la globalité de la planète dès la fin des années 70. Dans les quinze dernières années nous avons assisté à une croissance exponentielle d'Internet aussi bien en nombre d'équipements connectés que dans la capacité de ses liens. La croissance est de l'ordre de 60% par an, de 16 millions de postes connectés en 1997, on est passé à 100 millions à la fin 2000. Cette taille importante amène deux problèmes : l'hétérogénéité à de nombreux niveaux (topologie, types de liens, performances, asymétrie, protocoles) et les problèmes d'extensibilité.

1.3.2 Evolution des protocoles

Sur ces infrastructures physiques, les échanges d'information sont coordonnés par des protocoles qui eux aussi ont subi de profondes évolutions. En 1981, un modèle d'architecture de protocoles, le modèle OSI (Open System Interconnexion) [226], inspiré du modèle d'architecture du réseau SNA d'IBM a été introduit afin de permettre aux systèmes hétérogènes de communiquer. Ce solide et complexe modèle a servi de base à la structuration des logiciels de communication en proposant une décomposition par niveau d'abstraction des différentes fonctions de communications à réaliser pour permettre à un émetteur de dialoguer avec un ou plusieurs récepteurs. Les niveaux d'abstraction permettent de cacher les détails d'implémentation et les déficiences des réseaux sous-jacent et fournissent une interface indépendante du réseau et des protocoles aux applications. Ainsi, dans une approche en couches, le problème du transfert de données est divisé en un ensemble de multiples sous-problèmes ce qui limite la complexité, mais peut poser des problèmes d'implémentation et de performances puisque chaque couche n'a pas de connaissance de la globalité et peut faire de mauvaises décisions sur le stockage des données par exemple. Ce fait peut dégrader fortement les performances et les rendre imprévisibles. Depuis plusieurs années, c'est le protocole IP qui s'est imposé sur l'ensemble des infrastructures car c'est une architecture qui unifie des technologies réseaux et des domaines administratifs divers. Il permet ainsi une bonne exploitation des anciennes infrastructures. Aujourd'hui, les infrastructures transportent de l'IP encapsulé dans des trames SDH, SONET parfois même Ethernet ou bien natif sur la fibre. La pile des protocoles TCP/IP est relativement légère donc a priori plus efficace que celle du modèle OSI car elle ne propose que quatre couches (physique, réseau, transport et application) au lieu de sept. Mais dès la fin des années 80 on a mis en évidence les performances décevantes de l'empilement TCP/IP [38].

Si IP tente d'offrir le maximum pour chacun, il ne donne aucune garantie d'acheminement et ne tient aucunement compte de la sémantique des flux transportés. Pour les applications à contraintes temporelles (multimédia, système, routage), à contraintes de fiabilité (transport de fichiers), ou à contraintes de débits élevés (transferts de données en masse), la fourniture de services de qualités différentes et adéquates est pourtant un besoin de plus en plus crucial. Comment faire évoluer les protocoles d'Internet pour mieux servir les besoins des applications actuelles et se préparer à accueillir les usages futurs encore inconnus ?

1.3.3 Problématique de la Qualité de Service

Les approches traditionnelles de Qualité de Service (QoS) pour les flux temps-réel sont fondées sur le "mode circuit" et la réservation stricte de ressources. Ainsi le réseau téléphonique analogique ou numérique (ISDN) offre-t-il un circuit physique dédié tout au long de la conversation ou bien le réseau de télévision utilise une bande de fréquence non partagée pour la diffusion de vidéo. La technologie ATM s'est inspirée des avantages du mode circuit en proposant un mécanisme de multiplexage temporel asynchrone basé sur la commutation rapide de cellules pour résoudre le problème de la qualité de service dans les réseaux de commutation de données. La technologie ATM a été inventée pour apporter une solution au réseau numérique à intégration de Service large bande et conçue pour offrir une couche réseau multiservices. Son utilisation à large échelle par les applications multimédia aurait pu être une solution universelle au problème de la qualité de service. Ce rêve s'est avéré impossible à réaliser parce qu'aucune "prise ATM" directe n'a pu remplacer la "prise TCP/IP" fournie en standard dans les systèmes d'exploitation des milliers d'utilisateurs d'Internet et que le mécanisme de commutation rapide de petites cellules limite les performances. Dans mes recherches j'ai eu l'occasion d'expérimenter et d'analyser les avantages et les limites d'un dispositif réseau international basé sur la technologie ATM pour le support du travail coopératif (chap. 3).

A la fin des années 90, les approches "circuit" ont été un peu délaissées en raison de leur complexité de mise en oeuvre, d'absence d'interface de programmation d'application et de manque d'extensibilité. Il subsiste cependant toujours une activité dans le domaine car les avantages en terme de performance et de sécurité du mode "circuit" sont bien là. Le mode circuit permet en effet la construction de réseaux privés virtuels (VPN) qui est un concept très attractif pour les utilisateurs industriels. Les nombreux travaux autour de MPLS ou de la commutation en longueur d'onde en témoignent. De leur côté, les applications produisent de plus en plus de volume de paquets IP. Le taux de perte de paquets et les délais de bout en bout *LossInternet* demeurent très variables dans Internet. Aussi différentes approches ont-elles été étudiées ces dernières années pour assurer ou améliorer la qualité de service (QoS) dans Internet. La réservation de ressources pour offrir une garantie stricte de service aux flux individuels dans l'approche *IntServ* [30], ou un traitement différencié des flux par une priorisation dans leur acheminement dans l'approche *DiffServ* [19] sont les deux alternatives proposées, analysées et déployées dans des plate-formes expérimentales et commencent à peine à arriver sur les réseaux de production. Parallèlement, le développement des applications multimédia s'est poursuivi même en l'absence de garanties strictes fournies au niveau du réseau. C'est ainsi que sont apparues de nombreuses techniques adaptatives qui cherchent à masquer aux utilisateurs les faiblesses du réseau. La question de la fourniture de qualité de service dans Internet demeure un problème largement ouvert. Aucune solution miracle n'a pas été trouvée ni du côté du réseau, ni du côté des applications (chap.4). C'est donc dans ce contexte que j'ai développé mes travaux sur le concept de réseau sensible aux flux (chap.5).

1.3.4 Réseaux programmables

Les modèles architecturaux traditionnels séparent la communication du traitement de données. Avec les dernières avancées technologiques, il est possible d'envisager d'étendre les fonctionnalités du réseau du simple routage et de la transmission de paquets au traitement de ces paquets. Chaque noeud du réseau peut potentiellement être un noeud de traitement et un élément de réseau. La croissance des fonctions applicatives qui infèrent sur le comportement du réseau laisse entrevoir cette évolution. Les exemples les plus significatifs sont les pare-feu, proxys en tout genre, caches web. Par ailleurs, force est de constater que le déploiement de nouveaux protocoles dans les réseaux est devenu extrêmement complexe, car il nécessite le consensus de tous les fabricants et opérateurs ainsi que la mise à jour de millions de systèmes terminaux avec des implémentations compatibles. Dès 1995, la communauté OPENSIG a avancé le concept de signalisation ouverte et de programmabilité de réseaux. Ce projet voyait le futur réseau comme un ordinateur géant, complètement programmable, et qui délivrerait des services de voix, vidéo et données de manière globale. Une solution proposée est par exemple de fournir un ensemble d'abstractions logicielles des ressources réseau qui permettent l'accès distribué aux capacités de contrôle de bas-niveau des réseaux. Ainsi l'approche IEEE P1520 [125] fournit une API de programmation de réseau (modèle de référence en 4 niveaux : éléments physiques ; périphérique réseau virtuels, services réseaux génériques, services à valeur ajoutée).

Le concept des réseaux actifs a émergé en 1996 [206] [18]. L'objectif est d'augmenter les fonctionnalités du ré-

seau pour inclure du traitement de données. Au départ, la plupart des groupes de recherche essayent de construire un cadre général permettant aux paquets actifs de transporter du code pouvant être exécuté sur les noeuds actifs du réseau. Ces approches visent un réseau programmable général. L'autre approche, plus populaire en France [48] consiste à étudier comment implémenter des services aux niveaux de noeuds actifs, qui puissent modifier les données lors de leur passage au travers du réseau et fournir aux utilisateurs des services à valeur ajoutée. Les réseaux actifs peuvent donc accélérer le processus d'évolution et augmenter la souplesse du réseau en offrant des protocoles adaptatifs, des protocoles spécialisés, déployables dynamiquement et des réseaux reconfigurables. Le concept innovant des réseaux actifs offre cependant un grand nombre de défis en particulier en termes de performances et de sécurité, défis auxquels je me suis attaquée avec l'équipe RESO. Considérant que le coeur de réseau est surdimensionné et que les routeurs traitent plusieurs dizaines de millions de paquets par secondes, l'intelligence et donc la complexité sont de plus en plus repoussées en bordure du réseau (voir figure 1.1 et figure réfig :trends. Nous avons ainsi opté pour une architecture de réseau actif dans les réseaux d'accès. J'ai choisi d'explorer en particulier l'apport de cette technologie dans le cadre de la fourniture de services différenciés avec le concept de **diffserv actif** (chap.5) et des grilles de calcul avec le concept de **grille active** (chap.6).

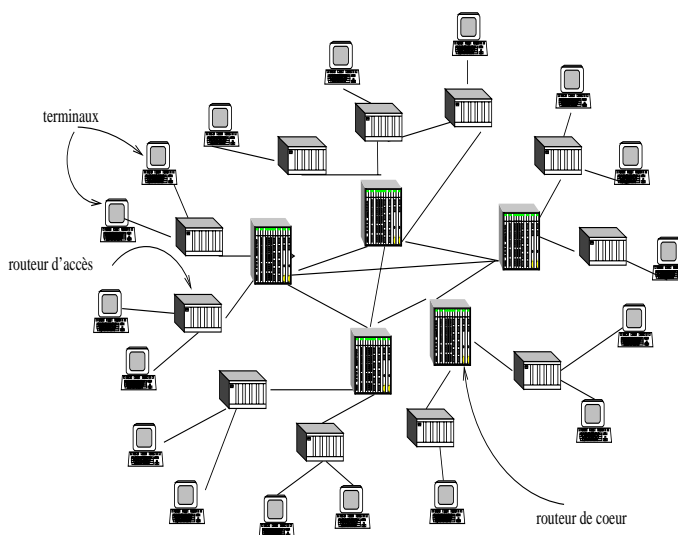


FIG. 1.1 – Modèle de réseau actuel avec des routeurs haute performance dans le coeur et des routeurs d'accès en périphérie

1.4 Contexte de recherche

Les années 90 ont vu l'explosion de l'Internet et l'avènement du multimédia. C'est dans ce contexte que j'ai développé mes recherches. Mes travaux ont été influencés d'une part par l'évolution des applications réparties et des réseaux et d'autre part par les chercheurs que j'ai côtoyés depuis ma thèse. Au cours de ma carrière de chercheur, je me suis intéressée aux aspects scientifiques et techniques des protocoles et des équipements réseau mais j'ai aussi participé à des projets *applicatifs* de grande envergure. J'ai ainsi pu mesurer les barrières techniques et humaines de l'expansion des domaines du travail coopératif supporté par ordinateur (Computer Supported Collaborative Work) ou les grilles de calcul et de données (Grid computing).

Après ma thèse, je me suis intéressée à la problématique des systèmes et applications répartis et j'ai étudié plus particulièrement le cas des applications coopératives. J'ai ainsi mis en oeuvre mes compétences en système d'une part et en réseau d'autre part pour concevoir et réaliser une boîte à outils coopératifs. Cette expérience de développement d'un *middleware* coopératif et d'applications de groupes a été menée avec les chercheurs de l'Ecole Centrale de Lyon (Samir Akkouche, Franck Tarpin Bernard, Marcello Galvao, Dany Drif et Bertrand David) et

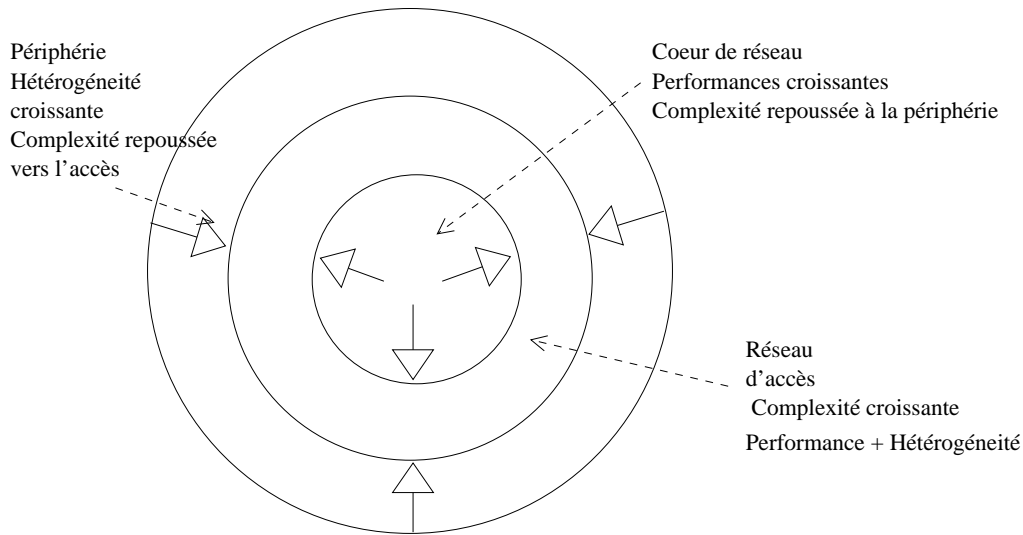


FIG. 1.2 – Evolution des performances et de la complexité dans l'interconnexion réseau

les nombreux élève-ingénieurs en fin de cursus que j'ai encadrés dans cet établissement. J'ai collaboré au projet TELECO3 avec Gérard Beuchot de l'INSA de Lyon. J'ai aussi participé au GDR-IHM et plus particulièrement au GT-SCOOP avec Joëlle Coutaz, Michel Riveill, Alain Deryke, Michel Beaudoin-Lafond et Dominique Decouchant. J'ai aussi eu l'occasion de collaborer avec des chercheurs en sciences humaines, des psychologues et des ergonomes spécialistes du télé-travail. J'ai par ailleurs mené un projet international avec le Canada pour valider l'approche ATM pour la fourniture de services multimédia, différenciés et garantis dans le cadre du travail coopératif. J'ai ainsi collaboré à distance en vidéoconférence en face à face avec des chercheurs de France Télécom R&D (CNET) tels que Bernard Grandjean ou Claude Kinzig ainsi que des chercheurs de la TéléUniversité de Montréal tels que Samuel Pierre et Lauraine André.

Dès mon arrivée dans l'équipe RESAM, j'ai réorienté mes activités sur la thématique **Réseaux Haut débit**, laissant de côté la thématique **Travail Coopératif supporté par ordinateur** et **applications réparties** que j'avais étudié au sein du laboratoire ICTT. J'ai, au côté de Bernard Tourancheau participé à la création de la jeune équipe RESAM, avec Laurent Lefevre et CongDuc Pham. Nous avons défini les axes de recherche du laboratoire et créé l'action INRIA RESO. J'ai aussi participé à la création de l'Ecole Doctorale MathIF et du DEA d'Informatique Fondamentale de Lyon. Du fait de l'absence de perspectives de la technologie ATM de bout en bout, mes travaux *réseaux* se sont focalisés sur l'évolution d'Internet pour prendre en compte le transport de flux hétérogènes. J'ai ainsi centré mes activités sur les niveaux réseau et transport et sur l'approche *réseaux actifs*. Le problème de la *Qualité de Service* dans les réseaux est resté le thème fédérateur transversal à ma reconversion thématique. Avoir pu traiter le problème du côté application puis du côté réseau, m'a permis de valider l'intérêt d'une approche verticale allant de l'étude des besoins des applications à la fourniture de modèles de réseaux et de protocoles adéquats. J'ai aussi compris tout l'intérêt du déploiement de plate-formes expérimentales pour la validation des résultats de recherche. J'en ai aussi mesuré les difficultés et les limites... J'ai plus récemment initié un certain nombre de travaux dans le domaine du support réseau aux grilles de calcul qui est un domaine d'application privilégié des technologies réseaux très haut débit. C'est l'occasion pour moi d'exploiter simultanément mes compétences en réseau, systèmes et développement d'applications réparties. J'ai ainsi participé au montage de projets nationaux (e-toile, VTHD++) et internationaux au côté de Christian Michau et de chercheurs de l'INRIA, du CNRS et du CERN. L'ensemble de mes travaux théoriques, menés avec les chercheurs de RESAM, est nourri par ces expérimentations en vraie grandeur menées dans le cadre de ces projets avec l'équipe d'ingénieurs de l'UREC de Lyon et des chercheurs français et européens que je coordonne au travers de projets IST et RNTL. J'ai ainsi l'occasion d'échanger un nombre important d'idées avec des chercheurs de renom tels que Ben Segal qui a introduit la technologie Internet au CERN, Brian Tierney de Lawrence Berkeley National Laboratory qui a écrit Netlogger, Cees de Laat de l'UvA et membre actif de l'IRTF, Tiziana Ferrari de l'INFN, de Peter Clarke et Robin Tasker à PPARC ou de

Richard Hugues Jones à l'Université de Manchester. Je suis membre actif du Grid High Performance Networking Research Group du Global Grid Forum (GGF GHPN RG). J'ai eu l'occasion de participer à une mission d'experts conduite par Michel Cosnard auprès de la NSF, de l'Internet2 et de la DOE aux Etats Unis. Je collabore avec Olivier Martin responsable des réseaux du CERN et du projet DataTAG, Howard Davis de DANTE, Dany Vandrome de RENATER et Christian Guillemot et Lionnel Thual de VTHD. J'entretiens par ailleurs de fructueuses relations scientifiques avec les membres des SUNlabs de Grenoble : Bernard Tourancheau, Gabriel Montenegro, membre actif de l'IETF.

J'ai participé à plusieurs Ecoles d'Eté Internationales sur les systèmes répartis et les réseaux haut débit multi-média (RHDM) et ai pu échanger mes idées avec les principaux acteurs de la recherche française : Serge Fdida, Michel Diaz, Christophe Diot, Laurent Toutain. Je participe aussi aux réflexions sur les réseaux actifs du groupe ASPRONET du CNRS et de loin au GRD ARP. Au travers de l'INRIA, j'ai aussi l'occasion de confronter mes idées sur les systèmes distribués, les réseaux et les grilles de calcul avec Thierry Priol, Michel Cosnard, Frédéric Desprez, Loïc Prilly, Omar Affifi.

Finalement je développe des collaborations avec les fournisseurs et utilisateurs d'applications tels que les Bio-informaticiens de l'IBCP (Christophe Blanchet, Gilbert Deleage) ou de Créatis (Johan Montagnat) ou les physiciens de l'IN2P3 (Guy Wormser, Denis Linglin, François Etienne, Vincent Breton) et du CERN (Bob Jones ou Peter Kuntz).

1.5 Démarche d'étude

Le processus de réflexion et les propositions autour du concept de réseau sensible aux flux est issu d'analyses menées au niveau des besoins réels des applications, des performances des réseaux expérimentés mais aussi du modèle architectural TCP/IP lui-même qui a, au fil des ans, exhibé une ubiquité et une puissance conceptuelle impressionnantes. Mon objectif est de me focaliser sur la réponse limite que l'on peut apporter " dans le réseau " aux problèmes de la qualité de service en conservant un réseau simple et robuste. La question que nous nous posons est la suivante : est il possible de redessiner les couches réseau et transport en apportant des services hétérogènes et flexibles, réellement nécessaires aux applications, tout en respectant la philosophie TCP/IP. Ma démarche personnelle est de marier l'approche expérimentale et l'approche théorique. La recherche sur les protocoles réseaux et l'architecture d'Internet pose en effet un certain nombre de difficultés très spécifiques qui sont analysées dans [81]. Ces défis sont aussi, je pense, ceux du Grid et du CSCW. Dans tout domaine scientifique, les outils du chercheur sont d'une part l'approche théorique avec l'analyse et la simulation et d'autre part, l'approche expérimentale avec les mesures et les expérimentations. Ces dernières fournissent un moyen d'explorer le monde réel tandis que la simulation et les analyses se restreignent à explorer un modèle abstrait et construit du monde pour en montrer les propriétés. Si dans un certain nombre de domaines, l'interaction entre les deux peut être évidente, dans la recherche Internet ces rôles ne sont pas si clairs, d'une part à cause de la très large échelle et d'autre part, à cause de l'évolution extrêmement rapide du sujet du domaine. Par exemple, un des problèmes de l'Internet, que ne connaissent pas les autres domaines est la possibilité d'un succès désastreux tel l'avènement du protocole HTTP et du WEB. La mesure est cruciale pour vérifier la réalisation et remettre en question les hypothèses. Les expérimentations sont fréquemment vitales pour lever des problèmes d'implémentations, mesurer la complexité de mise en oeuvre ou comprendre le comportement de certains systèmes. Ainsi dans l'histoire d'Internet, les plate-formes expérimentales ont joué un rôle majeur. La première plate-forme, ARPANET, a démarré en 1968 et a permis la conception des protocoles TCP/IP ainsi que le concept organisationnel des **rfc** (request for comment). Les plate-formes du passé ont montré que l'évolution est la caractéristique la plus importante pour créer des résultats révolutionnaires dans la technologie de l'information et de la communication. Ces expériences ont aussi souligné l'importance de l'interaction des chercheurs en réseau avec des utilisateurs réels pour un meilleur ancrage dans la réalité des réseaux d'aujourd'hui. Les réseaux nationaux de la recherche soutiennent aujourd'hui activement les projets de plate-forme. Nous participons à deux d'entre eux : RNRT VTHD++ (réseaux très haut débit) et RNTL E-Toile (grille expérimentale).

Il est cependant impossible de créer une instantiation du futur Internet à une échelle pertinente avec un spectre d'applications futures pertinent pour effectuer des mesures et des expérimentations valables. Pour réfléchir et inventer de nouvelles solutions, dans ces cas là, il faut utiliser des outils de simulation ou d'analyse. Pour valider nos propositions d'architecture nouvelle pour Internet et le support des applications réparties, nous nous

appuyons aussi sur la simulation notamment avec l'outil ns [143]. Au travers de mes diverses activités de recherche, j'ai donc construit une démarche qui allie l'étude de nouveaux modèles architecturaux, la création et la validation de nouveaux protocoles et logiciels en laboratoire avec la conception et le déploiement de plate-formes expérimentales.

1.6 Organisation du document

Ce mémoire développe ma démarche d'étude et les solutions conceptuelles et architecturales que je propose. Je tâche au travers des différents chapitres de montrer le cheminement qui m'a conduit vers un ensemble de propositions d'évolution des protocoles et architecture réseau. Je présente les méthodes, les modèles et les mécanismes solutions qui peuvent matérialiser le concept de réseau sensible aux flux.

Ce mémoire suit mon parcours intellectuel chronologique et est composée de trois parties qui se répondent. Les deux parties extrêmes sont orientées sur les problématiques des applications réparties tandis que la partie centrale traite, en trois chapitres, du support réseau à proprement parlé.

Dans la première partie, je présente les travaux dans le domaine du travail coopératif supporté par ordinateur que j'ai développés avec Franck Tarpin-Bernard dans le cadre de sa thèse de doctorat, Dany Drif en DEA et Marcello Galvao en DEA. Je me focalise plus particulièrement sur la flexibilité du partage de l'espace virtuel collectif. Cette partie synthétise les activités que j'ai menées au sein du laboratoire ICTT à l'Ecole Centrale de Lyon.

Dans la deuxième partie est composée de trois chapitres. Au chapitre 3, j'analyse le concept de qualité de service du point de l'application et des utilisateurs, puis dans le chapitre 4 je m'intéresse aux différents modèles et solutions de qualité de service proposées pour les réseaux IP. Mes propositions vis à vis de la gestion et du contrôle de la Qualité de Service sont développées au chapitre 5. Je décris en particulier les travaux que nous avons menés et poursuivons avec Benjamin Gaidioz dans le cadre de l'action INRIA RESO et de sa thèse de doctorat, sur le modèle DiffServ et la différenciation de service équitable ainsi que mes propositions sur une approche " réseau actif " de la gestion de la qualité de service, propositions étudiées par Julien RIO dans son DEA et complétés par différents travaux de fin d'études d'élèves ingénieurs.

La troisième partie de ce mémoire est consacrée à mes travaux sur la problématique réseau des grilles de calcul. Cette partie couvre les travaux menés depuis un an dans le cadre des projets EU DataGRID, EU DataTAG et RNTL E-Toile. Elle ouvre les principales voies de recherche que je m'attache à explorer avec Marc Herbert dans son travail de thèse, Julien Laganier dans son travail de DEA et future thèse et Franck Bonnassieux, Robert Harakaly, Geneviève Romier de l'équipe de l'UREC, Mathieu Goutelle et Fabien Chanussot ingénieurs experts de l'Inria.

Première partie

Applications coopératives

Chapitre 2

CSCW : CoTools et le partage de l'espace de travail virtuel

2.1 Introduction

L'objectif du travail coopératif supporté par ordinateur (Computer Supported Collaborative Work : CSCW) [16] est de permettre à des utilisateurs de collaborer par delà les distances en utilisant l'outil informatique. Les collaborateurs, reliés par un réseau, souhaitent avoir accès à des facilités similaires à celles de la vie réelle : interagir sur un même document, échanger des informations orales, écrites ou visuelles. Les activités de coopération sont classifiées selon une typologie temporelle et fonctionnelle. La classification temporelle distingue les activités synchrones (ou temps réel) pendant lesquelles les interacteurs opèrent en même temps, des activités asynchrones pendant lesquelles les personnes contribuent à la tâche dans des temps différents [44], [50]. Pour travailler à distance et en même temps, on peut utiliser une application mono-utilisateur classique, transparente à la coopération ou bien une application consciente de la coopération. Dans le premier cas, un support de partage multiplexe les entrées-sorties de l'application qui n'a pas à être réécrite, dans le deuxième, une application spécifique, un **collecticiel**, sera activée par les participants. Le terme collecticiel est la traduction du terme anglais groupware [66]. Il désigne le logiciel employé collectivement par les groupes de personnes engagées dans une tâche commune et qui permet le partage d'un même environnement virtuel. Dans mes travaux au sein du laboratoire ICTT, je me suis focalisée sur la problématique des activités synchrones et j'ai surtout étudié comment abstraire des collecticiels le problème du maintien d'un état cohérent. Ce problème est central dans un système coopératif temps-réel où on veut donner aux utilisateurs l'impression de vraiment travailler ensemble. Cela nous a conduit à la conception et au développement d'un système coopératif complet, d'un protocole de contrôle et de notification d'objets partagés, de plusieurs applications et outils collaboratifs et à l'adaptation d'un modèle conceptuel d'architecture d'applications interactives au mode multi-utilisateur.

2.2 Les systèmes coopératifs

Les collecticiels temps-réel, bien qu'ils soient aujourd'hui commercialisés, sont difficiles à construire. Les développeurs doivent non seulement s'occuper de la définition sémantique de leur application, mais aussi s'affronter aux difficultés techniques de distribution des données et des traitements dans le réseau. En conséquence, des **boîtes à outils** (toolkits) sont apparues pour permettre aux développeurs de construire des applications de groupe de manière plus simple.

2.2.1 Evolution des systèmes coopératifs

La première génération de systèmes coopératifs a fourni les systèmes à écran partagés (NLS, Mblink). La deuxième génération offre des systèmes à fenêtres partagées (VConf, SharedX, Timbuktu) qui permettent de trans-

former des applications mono-utilisateur en applications multi-utilisateurs transparentes à la coopération. Ces systèmes ont été importants pour l'acceptation commerciale des collecticiels. Puis les boîtes à outils d'interfaces multi-utilisateur (GroupKit v.1 [186], ArtWindow, Netmeeting), issues de la technologie des interfaces homme-machine, sont apparues et étendent les systèmes à fenêtres partagées pour supporter les applications conscientes de la coopération et des mécanismes flexibles. La quatrième génération a vu naître les architectures à partage de données avec une distinction claire de l'abstraction et de la vue graphique des données (Rendezvous [155], MMConf [45], Suite [55]). Ces architectures permettent l'accueil des retardataires, la détection d'une perte de synchronisation, le support de sessions persistantes. La nouvelle génération se focalise principalement sur le support de la flexibilité et du multimédia : GroupKit v.5 [187], Prospero [58], CoCa [130], Worlds [207], GEN [145]. C'est dans cette génération de systèmes coopératifs que nous situons nos travaux.

2.2.2 Construction et exécution des applications coopératives

Pour réduire la complexité de la conception et du développement d'une application coopérative, les boîtes à outils coopératifs fournissent des blocs de base (building blocks) génériques tels que la communication inter-processus, la distribution des événements et des données, des mécanismes qui permettent aux utilisateurs de rejoindre ou de quitter une session ainsi que des "widget" d'interface spécifiques (multi-ascenseurs...) [98], [54], [52]. L'architecture d'exécution (run-time) fournit ces blocs de base en gérant les créations et destructions de processus, les connexions pour les communications ainsi que la tolérance aux fautes. Cette architecture se scinde en deux composants fonctionnels de base : la gestion des sessions et la gestion du partage des objets. L'architecture d'exécution détermine la façon dont le système distribue les processus et les données entre les machines. Les architectures existantes se situent entre deux extrêmes : l'architecture centralisée (un seul processus d'application et une seule copie des données) et l'architecture totalement répliquée (n processus d'application et n copies des données) [124], [53]. On rencontre le plus souvent des architectures hybrides possédant des composants centralisés et d'autres composants répliqués. Les composants centralisés implémentent les fonctions délicates à assurer en mode totalement distribué telles que la gestion des utilisateurs et d'accueil des retardataires, le multicast, la sérialisation et le maintien des verrous pour la cohérence, le stockage temporaire ou persistant de l'état partagé, la notification. Ainsi dans Groupkit [188] seul le gestionnaire des sessions responsable de la connexion des participants à l'activité est centralisé. Les communications sont basées sur un mode pair à pair et l'état partagé des applications est complètement distribué. Il faut noter que, contrairement à une idée souvent répandue, les besoins en débit d'une application multi-utilisateur (jeu, tableau blanc...) sont relativement faibles au regard des exigences de délai. Chaque type d'architectures possède ses propres avantages et inconvénients, si bien que le choix résulte d'un compromis entre performances et facilité d'implémentation.

2.3 La boîte à outils CoTools

J'ai conçu et développé avec Dany Drif, Marcello Galvao [95] dans leurs stages de DEA et Franck Tarpin-Bernard dans le cadre de sa thèse de doctorat [200], l'environnement Ecoop rebaptisé CoTools [174] [162], toolkit coopératif. Cotools a ensuite été réécrit en JAVA par Julien Rio dans le cadre de son travail de fin d'études et ses fonctionnalités ont été étendues [171], [146]. Il permet le développement rapide et l'exécution d'applications coopératives sur Internet. CoTools fournit des outils plus particulièrement destinés à l'élaboration d'éditeurs coopératifs structurés. Les mécanismes sont applicables à un grand nombre de tels éditeurs (graphiques ou textuels). Les développements et les expérimentations nous ont montré que les constructions s'adaptent aussi à d'autres types d'applications coopératives (outils de conversation, jeux distribués multi-utilisateurs, environnement de réalité virtuelle

2.3.1 Modélisation et architecture

Le système coopératif CoTools est défini par un triplet $\mathcal{C} = (\mathcal{S}, \mathcal{U}, \mathcal{O})$, tel que $\mathcal{S} = \{S_1, S_2, \dots, S_n\}$ un ensemble de sessions S_i , $\mathcal{U} = \{U_1, U_2, \dots, U_m\}$, un ensemble d'utilisateurs U_j , $\mathcal{O} = \{O_1, O_2, \dots, O_l\}$, un ensemble d'outils O_k .

Les sessions S_i associent pour une durée limitée des utilisateurs U_j et des outils O_k . Elles sont caractérisées par leur durée, leur taille (nombre de participants), leur objectif. Les utilisateurs U_j sont caractérisés par leur agrégation

tion (individus ou groupe), leur rôle social, leur identité, leur autorité. Les outils O_k appartiennent à différentes classes d'applications. Ils ont en particulier un type (production, communication, coordination) et une politique de partage.

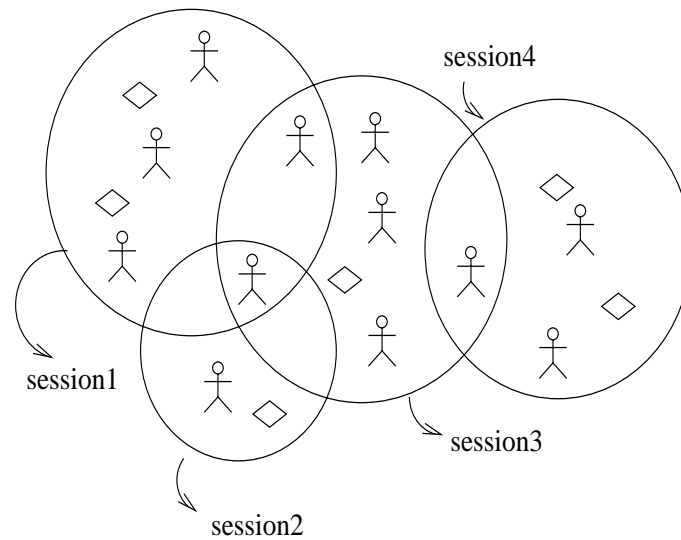


FIG. 2.1 – Modèle de l’environnement coopératif CoTools

L’environnement CoTools est responsable de la gestion des processus, et des facilités de communication, de synchronisation et de maintien de la cohérence. L’exécutif est structuré en deux modules : le gestionnaire de sessions (plan contrôle) et le module de partage de données (plan données).

Le gestionnaire de sessions gère les utilisateurs et les outils et fournit des fonctions telles que l’initialisation, les entrées et sorties dynamiques des utilisateurs dans la session. C’est lui qui permet le paramétrage des outils et l’association de politiques de partage et permet l’adaptation dynamique aux fluctuations de l’environnement. Le module de partage de données est responsable du contrôle de concurrence et de la distribution des événements coopératifs. Les mécanismes fournis permettent au développeur de spécifier quels objets sont partagés, quelles sont les opérations mono-utilisateur et les opérations multi-utilisateurs sur ces objets, et comment les modifications apportées à l’objet peuvent être rendues sur l’écran des autres participants.

L’architecture de Cotools est une architecture hybride qui cherche à trouver l’équilibre entre la centralisation de certains services critiques (support de la serialisation, du verrouillage, des retardataires..) tout en assurant des notifications à faible latence aux applications. L’état global d’un outil est stocké et maintenu par un serveur d’état centralisé tandis que l’ensemble de composants propres à l’application (présentation, contrôle, abstraction) sont répliqués sur chacun des sites utilisateurs. Un ensemble d’agent locaux, contrôlent les interactions locales et les propagent vers le serveur central lorsque c’est nécessaire. Les agents locaux jouent le rôle de représentants de la collectivité. Ce type d’architecture hybride est utilisé dans la plupart des environnements coopératifs [53], [156] et 2.2.2 car il permet d’obtenir de bonnes performances tout en étant simple à implémenter. La figure 2.2 illustre l’architecture hybride adoptée dans CoTools.

2.3.2 Principe de transparence et flexibilité

Une boîte à outils multi-utilisateurs doit fournir un ensemble de composants génériques, réutilisables et largement applicables. Le problème que pose sa conception d’une boîte est donc de trouver un bon compromis entre la régularité des caractéristiques des composants pour qu’ils puissent s’appliquer à un grand nombre de circonstances et leur flexibilité pour supporter les besoins spécifiques de différentes applications [58], [17]. Or pour développer les applications, le *toolkit* propose des entités abstraites qui encapsulent des stratégies pour la distribution, la replication, la synchronisation, la conscience de l’activité des autres (awareness). Les programmeurs

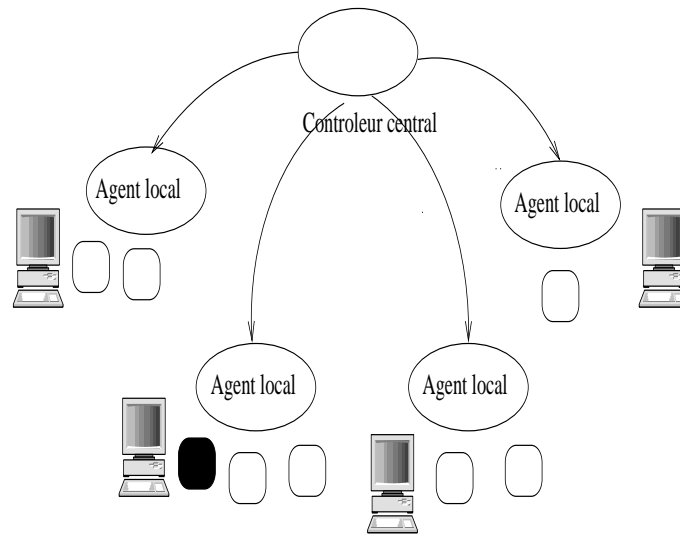


FIG. 2.2 – Architecture générale de CoTools

d'applications n'ont ainsi généralement pas de contrôle sur les mécanismes et les stratégies. Traditionnellement, pour limiter la complexité de développement, on sépare les caractéristiques *haut niveau* des applications, des caractéristiques *bas niveau* de l'infrastructure [10]. Ainsi, dans les premières générations de boîte à outils, on a utilisé une approche orientée système pour traiter les problèmes de contrôle et de distribution. On a identifié différentes formes de transparence à implémenter pour cacher aux applications et aux utilisateurs tous les problèmes causés par la distribution. Dans cette approche *système distribué*, les mécanismes de contrôle sont enfouis dans le système, et donc ne sont pas accessibles par l'application. Par exemple, si on considère le problème d'accès concurrent à une ressource partagée, une solution classique des systèmes distribués, consiste à cacher, par des techniques souvent restrictives de type verrouillage, l'existence des autres utilisateurs. Donc, le partage est fait de telle façon que les utilisateurs ignorent quasiment les activités des autres utilisateurs et ont l'illusion d'être seuls à manipuler la ressource commune [181]. Mais par essence, les collecticiels, supportent des relations interpersonnelles et sont donc des applications fondamentalement différentes des applications réparties traditionnelles [185]. Plusieurs auteurs [100], [64] ont souligné les dangers de technologies qui crispent le dynamisme inhérent à l'interaction humaine non contrainte. Le concepteur d'une application coopérative doit tenir compte du fait que :

- Plusieurs formes de coopération peuvent coexister
- Les groupes travaillent d'une façon dynamique et non prévisible
- Les groupes sont eux-mêmes dynamiques
- L'infrastructure support fournit une qualité de service variable.

De significatives interactions entre l'infrastructure d'exécution et l'application existent et doivent être prises en compte dans la construction des collecticiels et les boîtes à outils. Les éléments de l'infrastructure tels que la topologie, l'architecture logicielle, la qualité de service interagissent avec les éléments essentiellement dynamiques du travail coopératif. Il est nécessaire de considérer les solutions alternatives à la transparence de distribution, permettant de tenir compte des interactions entre les niveaux haut et bas. Tel est l'objectif central de la boîte à outils CoTools.

Pour introduire la flexibilité requise, certains auteurs remettent en cause le principe de la boîte à outil de type *boîte noire*. En effet si masquer les détails de l'implémentation permet aux développeurs de se concentrer sur l'apprentissage et l'application des blocs de base, l'interface abstraite à ces blocs de base au moyen d'une API rend l'infrastructure système et réseau sous-jacent totalement opaque aux développeurs. L'approche *open implementation* [120] vise à offrir aux programmeurs une interface beaucoup plus complète avec des primitives de haut-niveau très simples et d'autres beaucoup plus complexes et de bas niveau. Une implémentation ouverte fournit deux niveaux d'accès au programmeur : l'interface API classique et la *meta-interface*. La meta-interface décrit le comportement du toolkit et donne au développeur une possibilité *contrainte* de l'étendre. Une meta-interface peut

par exemple fournir une implémentation des sockets ainsi que des primitives de conversion d'objets en flots (sériation). C'est une API de l'implémentation du toolkit qui permet de l'adapter à ses propres besoins particuliers. Il faut déterminer quelles sont les parties de l'implémentation que le programmeur pourrait avoir à contrôler. Cette approche a été adoptée dans Prospero [57], dans la dernière version de Groukit [188] et dans Worlds [207]. L'approche *implémentation ouverte* est aussi souvent utilisée pour supporter la flexibilité dans la distribution des données. Ce principe est puissant pour l'optimisation de la programmation des applications mais peut s'avérer complexe à manipuler. La seconde approche adoptée par des auteurs tels que Grundy [101] consiste à utiliser un large ensemble de modes de collaboration au niveau conceptuel et un toolkit flexible au niveau implémentation. Le toolkit doit fournir les abstractions de programmation adéquates aptes à supporter les différents modes de collaboration.

Pour offrir la flexibilité nécessaire, nous avons choisi cette deuxième approche. Au lieu d'ouvrir le système à l'application, nous avons ouvert l'application au système. Notre objectif a été d'extraire de l'application les éléments sémantiques nécessaires à la prise de décision pour le contrôle de concurrence et à les encapsuler dans un protocole dédié. Ainsi, le système, sensibilisé aux besoins propres de l'application et de la situation prend des décisions plus fines. Nous avons en particulier défini un protocole original et flexible, le protocole NCP, pour le contrôle et la notification d'évènements relatifs aux objets partagés. La figure 2.3 donne un schéma du modèle en couches adopté dans CoTools pour le support d'application et d'outils coopératifs.

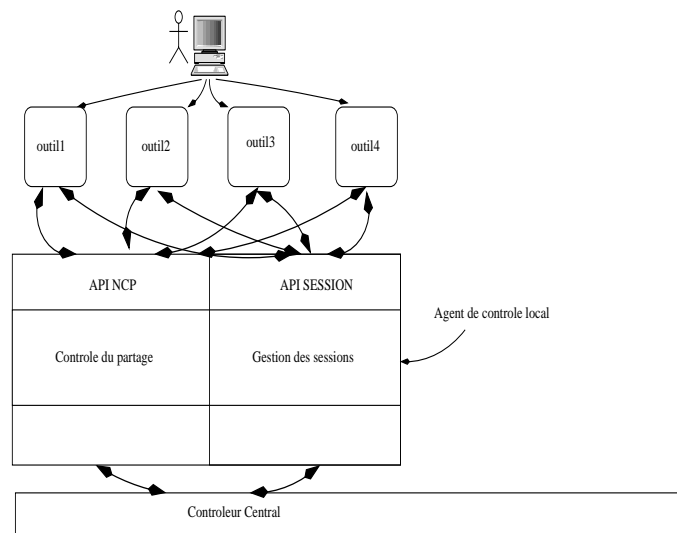


FIG. 2.3 – Modèle architectural de CoTools

2.3.3 Contrôle de concurrence flexible

Dans une application coopérative, le contrôle de concurrence gère les accès simultanés aux données partagées. Au fil des ans, la flexibilité du contrôle des données partagés dans les toolkits, aussi bien au niveau des méthodes de contrôle de concurrence (détermine comment les données sont maintenues cohérentes) que de la manière dont elles sont réparties sur le réseau est devenu un élément clé des boîtes à outils coopératifs. Ainsi, un algorithme de contrôle peut mettre en oeuvre une politique optimiste ou une politique pessimiste, selon que l'on considère que le conflit sera rare ou fréquent [65], [139]. Les stratégies optimistes sont utiles lorsque le temps de réponse du réseau est lent, mais risque de perturber l'utilisateur lorsque des retours arrière doivent être effectués (rollback). Les stratégies pessimistes vont toujours assurer une cohérence forte des données mais sont coûteuses en terme de latence [141]. S'il n'y a pas de contrôle de concurrence, des incohérences peuvent exister entre les sites, mais dans certains cas, les utilisateurs ne percevront pas les différences (whiteboard) et le gain en vitesse sera par contre substantiel. Par ailleurs, la granularité des objets peut être plus ou moins fine et la prise de décision peut être automatique ou manuelle. Considérons le cas d'une session de construction coopérative d'une scène 3D avec

un éditeur graphique multi-utilisateurs (voir figure 2.4). Le tableau 2.1 donne un exemple de découpage de la session en différentes phases avec, pour chacune, le degré de couplage et les stratégies de partage et de notification appropriées. Dans ce cas simple, on voit que les interactions requièrent un support de contrôle flexible afin de ne pas freiner la collaboration [144]. La description de la coordination pourrait être décrite formellement et s'appuyer sur les concepts de rôle. Drira [63] propose par exemple une grammaire de graphe qui permet de concevoir et d'implémenter des protocoles de coordination basés sur des règles. Dans nos travaux, nous nous sommes plutôt concentrés sur les mécanismes.

Phase de production	Degré de couplage	Stratégie de partage	Stratégie de notification
creation des éléments	faible : mode parallèle	Optimiste	Notification à la fin
ajustement des dimensions	moyen : mode coordination	Optimiste	Notification gros grain
animation	fort : mode coopération	Pessimiste	Notification à grain fin
correction	très fort : mode conflictuel	Pessimiste	Mode Trace

TAB. 2.1 – Scénario de production coopérative d’une scène 3D et mode de couplage associés



FIG. 2.4 – Exemple de la scène 3D animée construite

Nous avons étudié différents algorithmes de contrôle de concurrence optimistes et pessimistes dédiés aux outils coopératifs en particulier ceux proposés par Ellis [65] et par Karsenty [5], [163] (figure 2.5). Puis nous avons défini une abstraction spécifique, l'**interaction coopérative**, et développé un modèle générique de fonction permettant d'introduire la flexibilité du contrôle de l'espace partagé [164].

Après avoir analysé les opérations critiques pour la notification mais aussi pour la gestion de la cohérence, nous avons identifié quatre phases significatives au sens de la coopération, dans une interaction typique d'un éditeur multi-utilisateur. Ces *phases* sont importantes car elles constituent des **points d'arrêt** naturels qui doivent être perçus pour assurer une bonne coopération. Lorsque le programmeur écrit son application, il doit isoler ces

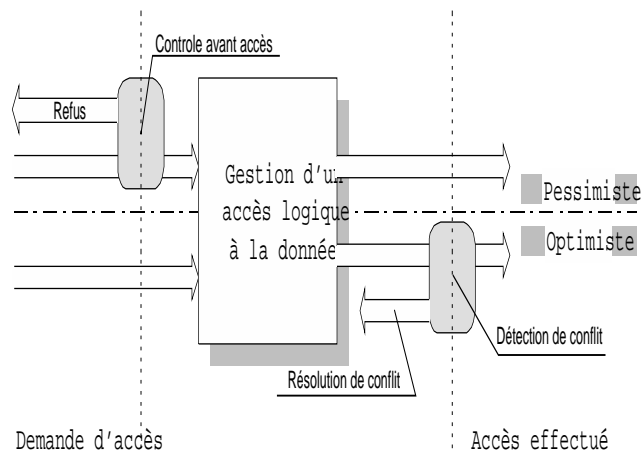


FIG. 2.5 – Principe d'un contrôle optimiste et d'un contrôle pessimiste

quatre phases dans une interaction sur un objet de l'espace partagé. La *phase* représente un événement qui doit être contrôlé et éventuellement notifié aux autres utilisateurs. Le programmeur insère à cet endroit là un appel à une primitive de l'API spécifique du noyau CoTools. Le module d'exécution est ainsi informé de l'état logique de chaque instance d'application et applique le contrôle adéquat en fonction de la politique en cours. Ce mécanisme permet de changer dynamiquement de politique pendant la session sans avoir à manipuler directement l'application.

Le protocole de Notification et de Contrôle de Concurrency, **NCP** est un protocole d'interface entre les applications et la boîte à outils. Il régule les échanges et permet d'informer systématiquement le noyau CoTools de l'occurrence d'un événement collaboratif. Une interface de programmation d'application (API) a été écrite pour encoder les primitives de service de ce protocole et est fournie avec l'environnement CoTools. La figure 2.6 représente les échanges de message NCP entre CoTools et les applications ainsi que l'automate d'état implémenté de manière répartie dans le noyau CoTools.

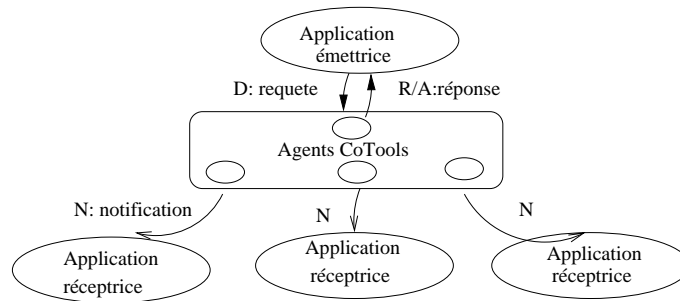


FIG. 2.6 – Le protocole de contrôle et de notification NCP

Au niveau de la boîte à outils, le module de partage de données gère les méta-données d'instances d'objets élémentaires et fournit des fonctions génériques qui encapsulent les opérations que l'on peut appliquer sur chacun des objets. Une opération est modélisée par un quintuplet

$$Op = (Obj_i, Cl_j, TypOpU_k, arg)$$

où Obj_i représente l'instance d'objet partagée, Cl_j la classe de l'objet, $TypOp$ le type d'opération (lecture, écriture),

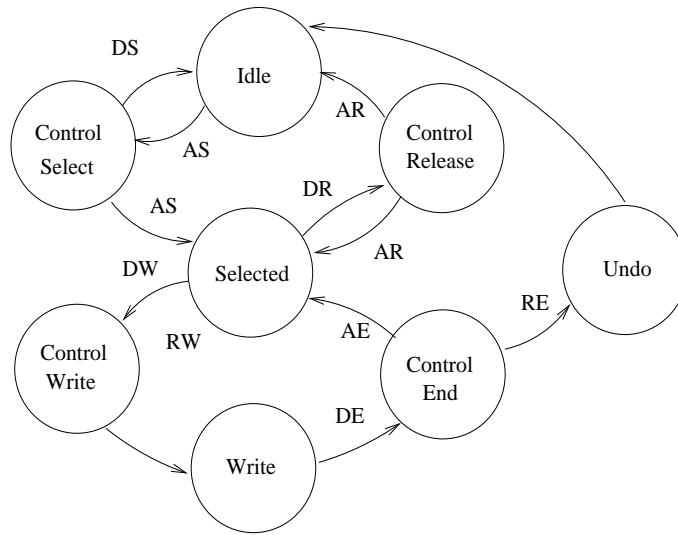


FIG. 2.7 – Automate du protocole NCP

U_k l'utilisateur qui réalise l'opération et arg est un paramètre dépendant de l'application mais transparent au noyau CoTools. CoTools distingue les méthodes nommées **opérations d'écriture**, qui altèrent les données internes à l'objet, des méthodes dites **opérations de lecture** qui ne manipulent que la présentation de l'objet. Les opérations *select* et *deselect* sont isolées car ce sont deux opérations de lecture spéciales et qui sont significatives dans un contexte collaboratif. Au delà, CoTools n'a aucune conscience de la sémantique spécifique associée à chacune de ces opérations de lecture ou d'écriture. Le serveur conserve, pour chaque instance, l'état - c'est à dire la phase - en cours et propage les évènements-opérations vers les autres utilisateurs qui sont rattachés à la session et ont explicitement demandé à être notifié de ce type d'évènement. Les instances d'application réceptrices rejouent localement l'évènement à l'aide des arguments emballé dans le message. L'abstraction d'interaction coopérative, le protocole NCP et l'API fournie se sont avérés être des outils simples et efficaces pour la construction et l'exécution des applications coopératives tests que nous avons développées ensuite. Nous avons aussi pu vérifier la pertinence de l'approche pour la mise en oeuvre d'un contrôle flexible.

2.4 Modèles d'architecture d'applications coopératives

En dépit de l'existence de boites à outils sophistiquées et flexibles, la construction d'une application coopérative demeure une tâche complexe. Une telle application est d'une part répartie et communicante, d'autre part multi-utilisateur et interactive. Depuis plusieurs années, la communauté des interface homme-machine s'est intéressée à la définition de modèles conceptuels pour les logiciels interactifs. Ces modèles organisent un système interaction en une collection d'agents qui collaborent pour supporter le dialogue entre l'utilisateur et la machine. La plupart de ces agents sont basés sur les trois composants (ou facettes) du paradigme IHM : la présentation à l'utilisateur, le noyau fonctionnel et le contrôle de l'interaction. Pour la construction des applications interactives, un certain nombre de modèles d'architectures dont les plus connus sont MVC [122] et PAC [43] ont été introduits.

2.4.1 Modèle AMF-C

AMF, modèle conçu par l'équipe de l'ICTT [201], est un autre modèle dont l'originalité est d'offrir une modélisation élégante de la partie contrôle d'interaction d'une application à l'aide d'opérateurs logiques. Dans AMF, chaque facette présente différents ports de communication, accueillant des entrées et/ou des sorties, qui peuvent être vus comme des interfaces de méthodes d'objets réels. Ces ports évitent d'avoir une association permanente

entre une fonction (un service) et son implémentation. Le composant de contrôle est défini par des entités appelées administrateurs de contrôle qui ont trois rôles :

- connecter et gérer les relations logiques entre les ports de communications (sources ou cibles).
- transformer les messages entrants en messages compréhensibles par les cibles.
- exprimer le comportement et les stratégies de contrôle en utilisant différentes règles d'activation entre un port source A et un port cible B.

Différents administrateurs ont été identifiés dont l'administrateur simple : *if A then B*, la séquence : *if A1, next A2, next An then B*, la conjonction : *if A1 and A2 then B*, etc. La figure 2.8 donne une représentation graphique du modèle AMF pour une application mono-utilisateur. On note en particulier comment les facettes présentation et abstraction sont reliées par la facette contrôle.

FIG. 2.8 – Le modèle AMF

Dans son travail de thèse, Franck Tarpin Bernard, a fait évoluer le modèle AMF pour lui intégrer la partie coopérative. Les modèles d'architecture conceptuelle des collecticiels doivent combiner les modèles d'applications mono-utilisateur et les contraintes introduites par le travail coopératif [159]. Des travaux similaires ont été réalisés parallèlement sur le modèle PAC avec PAC* [34] Nous nous sommes en particulier focalisés sur la partie contrôle et sur l'intégration de AMF-C dans CoTools pour étudier la flexibilité du contrôle de l'espace partagé [200], [202], [68]. Nous avons clairement identifié l'interaction entre la facette contrôle de l'instance locale de l'application et le module local de CoTools qui représente l'état global de l'application partagée. La figure ci-dessous 2.9 montre comment les primitives de l'API de CoTools sont activées par les opérateurs logiques de AMF-C. Les notifications asynchrones sont directement redirigée vers la facette *distant*. Chaque entrée utilisateur est capturée par l'opérateur et transmise au contrôleur de CoTools qui connaît la politique de contrôle et prend la décision ou non de propager l'évènement. Ainsi tout changement dynamique de politique globale est répercuté de manière automatique vers les applications, sans perturber leur fonctionnement. Le modèle AMF-C est détaillé dans le chapitre 3 du document d'annexes.

FIG. 2.9 – Couplage du modèle AMF-C et de CoTools

2.4.2 Modeleur géométrique 3D multi-utilisateur GEO

Dans son DEA, Marcello Galvao a développé un modeleur coopératif 3D en utilisant le modèle AMF-C et la boîte à outil CoTools. Il a ainsi pu montrer comment l'approche AMF-C/CoTools permet la réutilisation massive du code mono-utilisateur dans le passage au modèle multi-utilisateur. Nous avons mis en évidence le faible surcoût de développement d'une application multi-utilisateur complexe égal à moins de 20% du temps total dépensé pour la conception et le développement la version mono-utilisateur. Le volume de code additionnel ne représente quand à lui que 6% du volume de code total (soit 2000 lignes sur 34.000 pour le logiciel GEO) [96]. Nous avons aussi montré la flexibilité du partage à plusieurs niveaux et développé une interface spécifique pour permettre les changements dynamique de politique de partage et la notification de ces modifications aux différents participants. La connaissance du mode de coopération en cours s'est avérée être très importante pour la *conscience* (awareness) de l'activité de groupe. La figure 2.10 illustre les types de présentations différentes affichées sur les écrans de trois utilisateurs distants.

2.4.3 L'application télé-pointeur TéléPTR

Dans un contexte coopératif, l'outil télépointeur est nécessaire pour maintenir l'attention des autres participants sur un point de l'écran [140]. Avec Samir Akkouche, nous avons développé TéléPTR [147], un outil collectif de désignation, télépointeur amélioré, offrant des facilités pour exploiter et commenter des documents de nature très diverses (textes, cartes géographiques, scènes géométriques, images médicales, courbes de résultats). Les

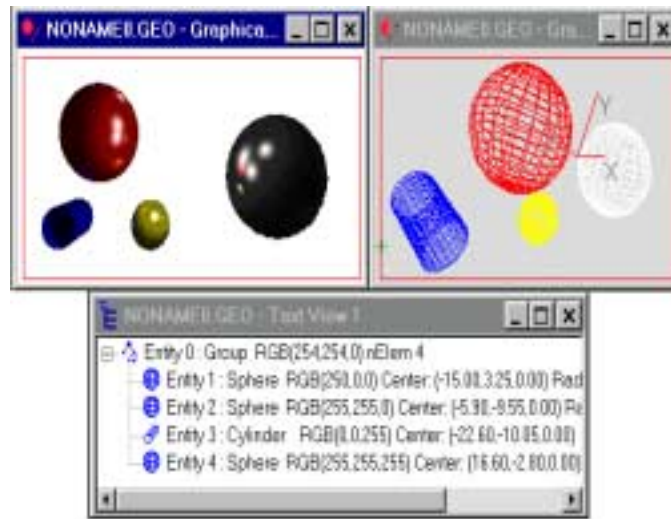


FIG. 2.10 – Vues multiples du modeler géométrique 3D coopératif GEO

fonctionnalités offertes par TéléPTR sont destinées à aider la focalisation de l'attention des participants et à permettre le dessin de croquis informels sur les vues d'une application coopérative [168], [169]. Cet outil graphique de télé-désignation partageable, souple est indépendant des applications. Il offre des fonctionnalités graphiques plus riches que l'image classique de l'index symbolisé par le télépointeur. La figure 2.12 montre le choix d'artifices de désignation mis à disposition de l'utilisateur de TéléPTR. La figure 2.13 illustre l'utilisation des masques 2D dans un modeler géométrique 3D. Ce prototype a été validé sur le plan technique et a été intégré à l'environnement CoTools. La flexibilité dans le choix d'une politique de contrôle du tour de parole et la diffusion des messages a aussi été expérimentée. Nous avons centrée notre étude sur la désignation en 2D car elle est plus générale. L'implémentation d'un télépointeur 3D dans le modeler géométrique nous a montré que pour la désignation en 3D, un raisonnement dans l'espace *objets* et non plus dans l'espace image s'imposait. La télé-désignation devient dépendante de la sémantique interne de l'application et ne peut plus être supportée simplement par un outil externe. Dans cette optique, beaucoup d'artifices de désignation peuvent être exploités : coloration des objets, points de vue, masquage sélectif selon la profondeur, spray... Le mécanisme de sauvegarde des objets logiques et de leur état par un serveur d'états tel celui de CoTools pourrait aussi apporter une réponse à cette problématique.

Le chapitre 3 du volume d'annexes de ce mémoire présente une synthèse des travaux menés autour de l'environnement CoTools. La figure reffig :coop replace nos contributions dans le domaine des systèmes et des applications coopératives dans un modèle architectural en couche.

2.5 Conclusion

La conception et le développement de la boîte à outils CoTools ainsi que du modèle d'architecture AMF-C nous a permis de mieux cerner les problématiques de production et d'exécution des applications coopératives synchrones, mais au delà, celle des intergiciels de construction et de support d'applications réparties. Nous avons proposé une nouvelle approche pour articuler de manière plus souple les différents niveaux d'abstraction et apporter de la flexibilité dans ces logiciels interactifs répartis. Des applications complexes ont pu être aisément construites à partir de notre modèle et de nos outils. La limitation principale de l'approche proposée est qu'elle nécessite des applications ouvertes dans lesquels les primitives de coopération puissent être intégrées. Pour introduire la flexibilité du partage et de la notification, l'application mono-utilisateur doit être bien structurée. L'approche utilisée pour le développement de l'application requiert la séparation nette entre les couches de présentation et d'abstraction, et l'introduction d'une couche de contrôle pour la synchronisation des vues. Cette contrainte s'avère être très utile non seulement pour faciliter le passage à un mode multi-utilisateurs, mais aussi pour améliorer la

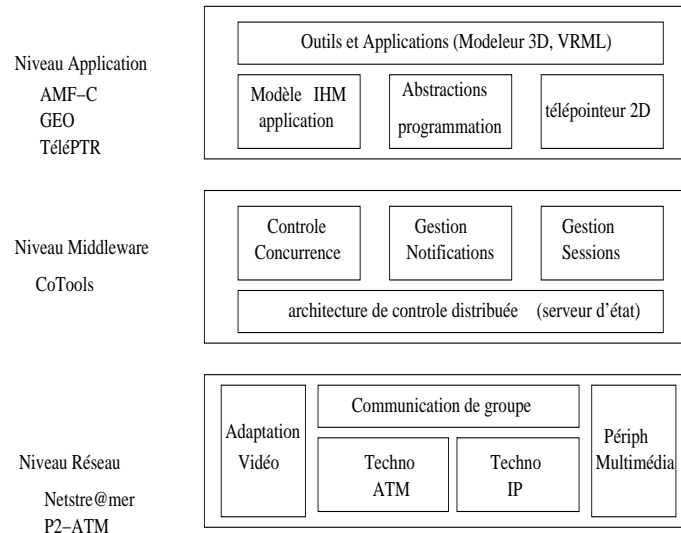


FIG. 2.11 – Localisation de nos contributions dans le domaine du CSCW

couche de présentation de l'application mono-utilisateur. Ce principe architectural commence à bien se répandre dans la communauté des développeurs d'application interactives. Cependant, nous avons eu des difficultés à valoriser nos recherches et développements sur les systèmes coopératifs. En effet, les expérimentations en local ne permettent que des évaluations fonctionnelles limitées. Les applications coopératives doivent être évaluées sur une population large et dans des environnements longue distance. Dans le milieu des années 90, il est apparu nécessaire d'expérimenter les réels apports et limites du travail coopératif supporté par ordinateur sur des plateformes expérimentales de grande envergure auprès d'une population significative. La thématique de l'évaluation est devenu un véritable axe de recherche. En 1997-1998, j'ai eu la chance de participer et même de coordonner la partie française d'un projet franco-québécois P2-ATM dont l'objectif était d'étudier l'interaction collaborative sur un réseau ATM à 2Mbps dans le cadre d'un projet de télé ingénierie de conception et de réalisation. Ce projet m'a plongée plus précisément dans la problématique de la qualité de service.

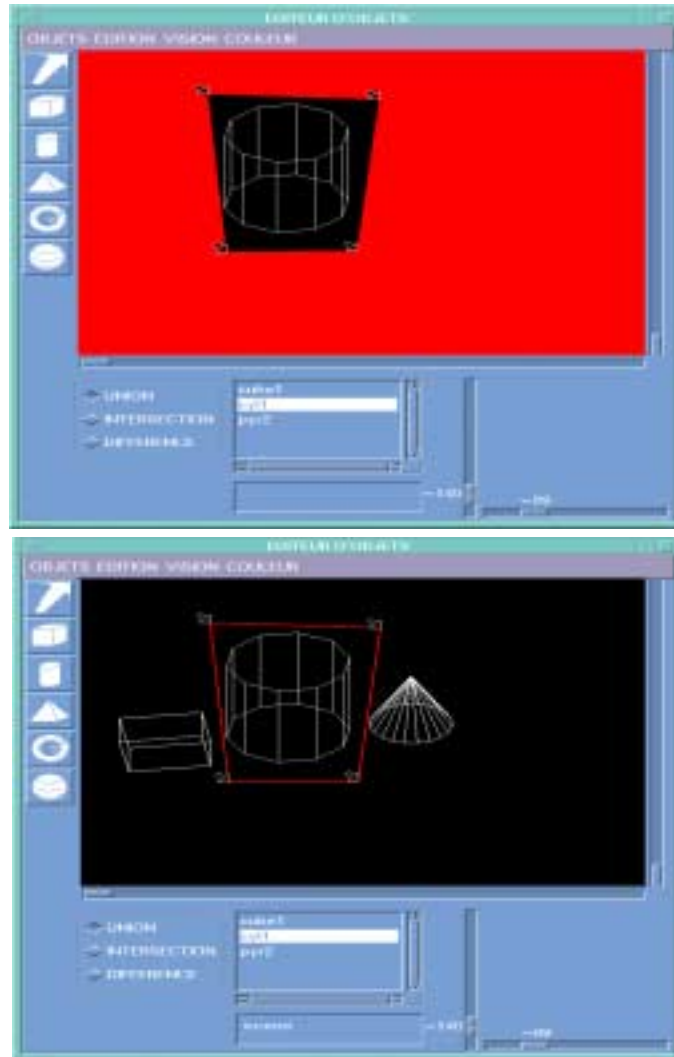


FIG. 2.13 – Utilisation de la fonction masque de téléPTR

Deuxième partie

La Qualité de service dans les réseaux

Chapitre 3

Problématique de la Qualité de Service

3.1 Introduction

Le terme **Qualité de Service** (QoS) recouvre différentes significations selon les communautés. Il est donc difficile d'en donner une définition rigoureuse et satisfaisante. Dans [73], Ferguson pose un certain nombre de questions sur l'expression Qualité de Service. Elle est composée de deux mots qui sont eux mêmes mal définis et très ambigus! Le terme **qualité** est utilisé pour décrire un processus de livraison de données d'une manière fiable ou meilleure que la normale. La notion de **service** quand à elle peut recouvrir des niveaux d'abstraction plus ou moins élevés. Il est important de distinguer la notion de Qualité de Service de celle de classes de **services différenciés** qui se réfère à la capacité de différencier les types de trafics ou de services afin que les utilisateurs puissent traiter une ou plusieurs classes de trafic de manière différente des autres.

En fait, l'origine de l'expression qualité de service est relativement ancienne dans les réseaux et est emprunté au réseau postal. On retrouve des définitions de QoS dans X25, dans le modèle OSI, dans la spécification de la couche transport. Dans ce contexte, la qualité de service définit exactement quels paramètres parmi un ensemble de paramètres sont significatifs pour un contrat de service particulier.

Un certain nombre d'auteurs s'intéressant aux systèmes distribués multimédia [197] ont cependant considéré que la définition donnée dans le cadre du modèle OSI était inacceptable car elle se limite à la partie réseau du système réparti. Par exemple, le modèle de référence pour le traitement distribué (RM-ODP) la définit comme *un ensemble de besoins qualitatifs sur le comportement collectif de un ou plusieurs objets*.

Pour traiter la qualité de service dans un système distribué multimédia, il faut :

- *évaluer les besoins en terme de vœux subjectifs* ou de satisfaction avec la qualité de l'application - performance, synchronisation, coût, etc...
- *projeter ces estimations sur des paramètres de qualité de service* pour différents composants et couches du système. Par exemple, un utilisateur va choisir la vidéo en terme de résolution et de fréquence d'images qui se traduisent en besoin de capacité.
- *négoier entre les composants systèmes et réseaux* pour s'assurer que tous les composants du systèmes peuvent honorer la requête de manière cohérente.

L'étude de [212] donne une bonne vision des différents paramètres de QoS à considérer dans les composants d'un système distribué multimédia, en particulier au niveau du système de communication, des schémas de codage, des systèmes d'exploitation, des serveurs de fichier continus et des bases de données. Dans [13] une vision globale, ainsi qu'une bibliographie détaillée de cette problématique est proposée.

J'ai abordé la problématique de la qualité de service du côté des *utilisateurs* dans le cadre du projet P2-ATM au laboratoire ICTT, puis, au sein du projet INRIA RESO, je me suis intéressée aux aspects *fournisseur du service*. Dans ce chapitre je présente l'expérience et les analyses de qualité de service que nous avons menées sur un dispositif de collaboration basé sur un réseau ATM (section 3.2), puis je présente le modèle de performance de réseau et les métriques associées (section 3.3) sur lesquelles je m'appuie dans la suite de ce mémoire avant de discuter des besoins de qualité de services des applications et des contraintes spécifiques imposées à la définition de services évolués dans Internet (section 3.4)

3.2 Qualité de service et travail coopératif

3.2.1 Evaluation de la qualité perçue dans un réseau ATM

La qualité perçue par l'utilisateur détermine le degré de satisfaction que cet usager retire du service. Dans le prolongement de mes travaux sur les systèmes coopératifs, j'ai mené un projet de télé-ingénierie collaborative de longue durée visant à évaluer la qualité perçue. Ce projet P2-ATM [178] mené en collaboration avec le CNET France Telecom et le LICEF de la TéléUniversité du Québec à Montréal, nous a permis d'appréhender réellement les caractéristiques humaines et techniques du travail coopératif synchrone et a mis en évidence les besoins de qualité de service de ce domaine d'application. Le but était d'évaluer l'écart d'une collaboration sur ATM par rapport à une collaboration sur un réseau RNIS à 2x64Kbps ou sur Internet. Un certain nombre d'obstacles avaient

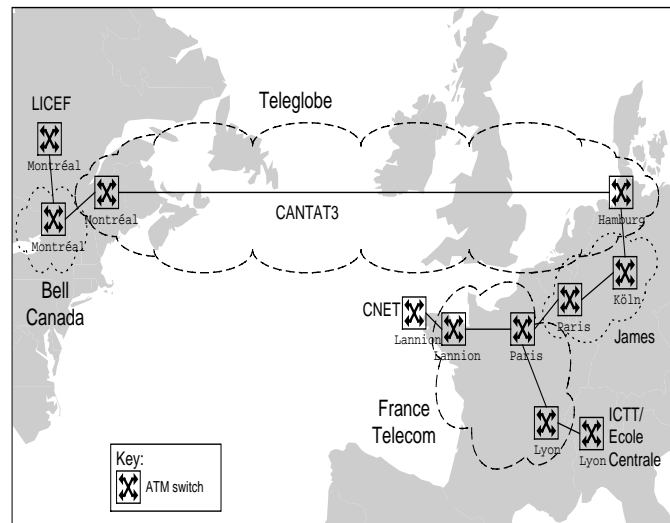


FIG. 3.1 – Infrastructure de communication du projet P2-ATM

été identifiés dans des expérimentations analogues sur Internet. Parmi les principaux :

- des problèmes d'indisponibilité ou des performances irrégulières ou insuffisantes de l'infrastructure de communication rendaient certaines fonctionnalités telles que le partage d'application complètement inutilisables.
- le travail en visioconférence sur fenêtre petit format pendant des séances régulières de plus d'une heure était difficilement supportable.
- les temps de transfert de fichiers volumineux au cours des séances de travail étaient prohibitifs et entraînaient des ruptures d'interaction importantes.

Pour immerger l'équipe d'ingénierie dans un réel contexte de production, l'expérimentation était soumise à un certain nombre de pressions réalistes et exigeantes : l'objet de l'expérimentation - un module multimédia de télé-formation à la technologie ATM devait être finalisé, et il le fut [11].

J'ai eu en charge la conduite de l'équipe française d'une quinzaine de personnes de ce projet coopératif d'une durée de six mois à raison de trois séances de 2 heure trente par semaine. Avec René Chalon, nous nous sommes plus particulièrement focalisés sur la mise en oeuvre et l'étude du dispositif de coopération, avec le test et l'évaluation des outils logiciels, des matériels et des possibilités offertes par les réseaux ATM [170].

La technologie ATM est une technique de commutation intelligente qui permet de commuter très rapidement toute forme de trafics (voix, données, images), encapsulés dans des cellules de 53 octets, tout en maximisant l'utilisation des débits offerts par les liaisons optiques. La figure 3.1 illustre les canaux virtuels ATM mis en place pour le projet. La description détaillée du dispositif mis en oeuvre ainsi que les résultats obtenus sont donnés dans le chapitre 2 du document d'annexe à ce mémoire. La figure 3.2 ci-dessus en donne un aperçu. L'expérimentation a montré que l'utilisation d'un réseau haut débit et à qualité de service garantie permettait de réaliser une tâche coopérative complexe. La bande passante de 2Mb/s permet d'obtenir une qualité d'image très confortable et de

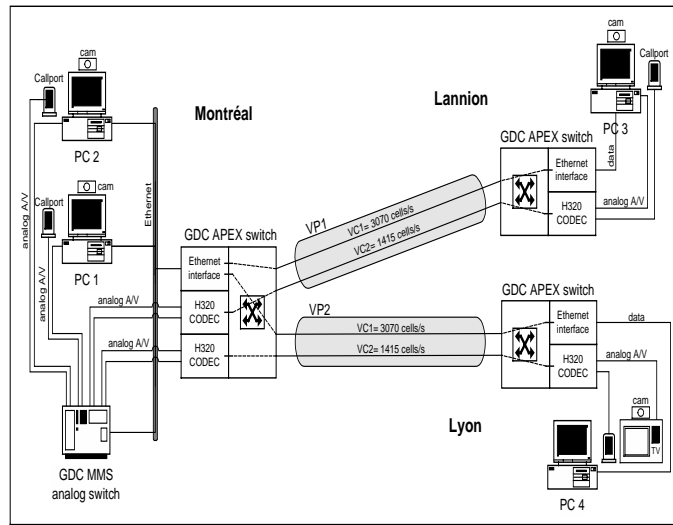


FIG. 3.2 – Dispositif de collaboration

visualiser simultanément les quatre vidéo des sites distants. Elle offre la possibilité de partager des applications 3D et des animations, fonctionnalités inaccessibles par exemple sur d'autres supports réseau. Les transferts de fichiers volumineux (supérieurs à 1Mo) se sont avérés réalisables avec des débits de l'ordre de 500kb/s, ce qui demeure cependant faible sur une liaison offrant un débit constant de 1,3Kb/s (canal virtuel CBR à 3072 cellules/s). Plusieurs facteurs expliquent ce faible rendement :

- le surcoût en terme de temps de calcul et d'information d'entête de la pile protocolaire :

$$ftp \rightarrow tcp \rightarrow ip \rightarrow ethernet \rightarrow AAL5 \rightarrow ATM$$

- des rafales de pertes de cellules, inexpliquées par les opérateurs, rencontrées sur le réseau expérimental international.
- la dynamique interne du protocole tcp

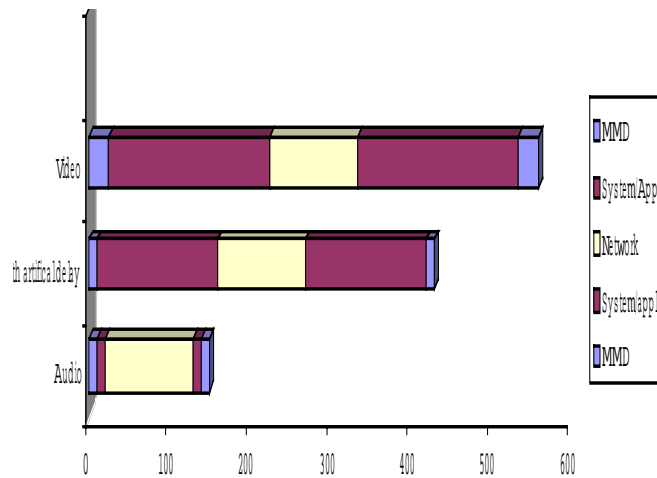


FIG. 3.3 – Synchronisation intermédia dans P2-ATM

De nombreuses ruptures et dégradations technologiques (sons, images et données) ont été observées et mises en perspective avec les ruptures d'interaction. D'autre part, les équipements audio utilisés reproduisaient un son d'une qualité insatisfaisante (pas de contrôle automatique de gain, difficulté d'équilibrage des niveaux sonores et absence de dispositif de suppression d'écho). La qualité résultante était inférieure à la qualité d'un système téléphonique classique RTC64 ou RNIS. Ce problème est un problème connu et récurrent dans les environnements distribués multimédia expérimentaux. Il n'est pas associé au réseau de transport lui-même mais est dû aux lacunes des périphériques d'extrémité qui sont parfois complexes à paramétrer. Nous avons aussi relevé une difficulté d'utilisation et d'initialisation du dispositif expérimental ATM plus importante que dans un dispositif classique, tel Pictoretel, basé sur une technologie Numéris.

3.2.2 Influence de la Qualité de Service sur le processus de coopération

Dans le cadre du projet P2-ATM, nous avons présenté une étude du processus de coopération et de la corrélation des dimensions humaines et des dimensions technologiques [?]. Nous avons analysé l'importance et l'utilisation des différents canaux (audio, video, data) dans chacune des phases identifiées. Nous avons aussi étudié de l'influence de la variation des performances du réseau sur le processus de collaboration. Nous avons cherché en particulier à définir une fonction f telle que : $QoI = f(QoS)$, où QoS est la qualité de service de bout en bout observée et QoI (Quality of Interaction) *qualité* ou confort de l'interaction, concept que nous avons cherché à objectiver et à mesurer en collaboration avec les chercheurs en Sciences Humaines. Les expérimentations de P2-ATM ont mis en évidence le fait que les facteurs de QoS de niveau utilisateur, traduits subjectivement par *bien se voir*, *bien s'entendre*, contribuent à une *bonne interaction*. De manière générale, nous avons observé qu'une dégradation de la qualité de service entraîne un changement de registre. L'acteur passe à un niveau de contrôle (il vérifie, il répète...) [209]. Toutes ces actions le coupent dans sa pensée et le déconcentrent de sa tâche, ce qui se répercute sur la productivité globale. Ainsi les durées d'attente, le nombre et la fréquence de changements de registres sont les métriques que l'on associe à la notion de qualité d'interaction (QoI). Les paramètres subjectifs peuvent ainsi être objectivés. Le *bien/mal se voir* s'exprime d'un côté sous forme de paramètres de QoS de niveau performance, format ou coût qui agissent sur la précision et la fluidité de l'image et de l'autre sous forme de paramètres de QoI. Finalement nous avons constaté que lorsque les utilisateurs sont automatiquement informés des changements de qualité de service du réseau et peuvent se faire une idée de ce qui se passe sur la ligne, nombre de ruptures d'interaction peuvent être évitées et le travail coopératif peut se poursuivre de manière quasi fluide. L'influence de la qualité de service sur le processus d'interaction reste cependant relativement complexe à objectiver car de nombreux paramètres sont difficiles à quantifier. A la suite de ce projet, les chercheurs en ergonomie de FT R&D ont poursuivi ce travail d'analyse. De mon côté, j'ai orienté mes recherches sur les problématiques purement réseau.

3.3 Paramètres de performances d'un réseau

Le fournisseur de service réseau requiert un ensemble d'outils et de services pour contrôler les paramètres de performances du réseau afin de fournir un service meilleur et plus prévisible. Un modèle de performance de réseau [73] s'appuie sur cinq métriques principales.

- **le débit**, noté r ,
- **le délai**, noté d ,
- **la variation de délai** ou gigue, notée j ,
- **le taux de pertes**, noté l ,
- **le taux d'erreurs**, noté e .

Dans le cadre des réseaux ATM, le débit est exprimé en nombre de cellules à la seconde, les délais ou les taux de pertes en délais de transfert de cellules et de taux de pertes de cellules. Pour les réseaux IP, l'unité de base est généralement le bit. Le groupe IPPM (IP Performance Supervision) de l'IETF a décrit un cadre pour les métriques de performances IP dans la rfc 2330 [158]. Ces métriques peuvent être définies dans une direction (aller simple) ou dans les deux directions (aller-retour). Dans la suite de cette section, l'unité par défaut sera le bit et la technologie sera IP. Chacune de ces métriques peut être définies de la manière suivante :

3.3.1 Paramètres de débit

Le débit binaire, r , ou, par abus de langage, la bande passante, entre deux systèmes communicants est le nombre de bits que le réseau est capable d'accepter ou de délivrer par unité de temps. C'est le taux de transfert maximum pouvant être maintenu entre deux points. Le débit utile dépend du niveau auquel on se place dans la hiérarchie protocolaire. Par exemple, la bande passante d'un lien réseau, métrique analytique définie par le groupe IPPM, représente la capacité de transport d'un lien réseau, mesurée en bits par seconde, dans laquelle les données n'incluent pas les bits nécessaires pour les entêtes de niveau 2. Lorsque l'on se place à niveau supérieur à la couche réseau, on considère la capacité du lien (throughput) qui correspond au volume effectif de données application transmis. La capacité utile du lien (goodput) est égale au nombre total de bits issus de l'application et correctement transmis par unité de temps. Par exemple pour un flux TCP, on ne comptabilise que les bons paquets reçus et on retire des statistiques les paquets retransmis. C'est cette capacité utile qui est le paramètre pertinent pour l'application. Comme il existe différentes implémentations du protocole TCP pour acheminer l'information, cette métrique est empirique et délicate à manipuler. L'IPPM a défini un cadre pour la mesure de cette capacité de transport (Bulk Transport Capacity (BTC) [195], définie par la formule : $BTC = V/T$ où V est le volume de données reçues et T le temps écoulé.

3.3.2 Paramètres de délai

Délai de transit

Le délai de transit de bout en bout, d , est le temps écoulé entre l'envoi d'un paquet par un émetteur et sa réception par le destinataire. C'est une des caractéristiques principales de la QoS. Le délai de transit d est composé d'une partie fixe d_f et d'une partie variable d_v .

On a $d = d_f + d_v$ avec $d_f = d_p + d_s + d_t$ et $d_v = d_q + d_j$ où

- d_p est le délai de propagation (5 à 6 micros seconde par km sur une liaison filaire)
- d_s est le temps de transmission qui est fonction du débit binaire et de la taille des paquets émis.
- d_t est le temps de traitement qui correspond par exemple à la traversée d'un codeur-décodeur.
- d_q est le délai cumulé dans les files d'attente des routeurs
- d_j est le délai introduit par le buffer de compensation de la gigue pour assurer la synchronisation

Le temps de traitement est d_t composé de plusieurs facteurs : le délai système, le délai de traversée des interfaces matérielles au niveau de l'émetteur et du récepteur. Typiquement pour une information de type voix numérisée sur un PC on a $d_t = 20ms$. Ce délai est causé par la paquetsation. Il faut 10ms pour constituer une trame G.729. Pour un codec qui met en oeuvre un algorithme de compression, l'étape d'anticipation (look ahead) peut prendre 5ms.

Le délai d_q induit par la mise en file d'attente des paquets dans les systèmes intermédiaires dépend de la taille des tampons mémoire dans les routeurs et de leur encombrement. Un délai de file d_q de 50ms sur un lien à 25Mbps indique qu'il y a environ 15koctets en attente dans les routeurs. Le délai dépend donc de la distance entre deux points, du nombre d'équipements intermédiaires et du délai de file dans ces équipements.

Dans les réseaux IP, on considère aussi souvent le délai d'aller-retour ou Round Trip Time, RTT , car il est facile à calculer en utilisant la méthode du ping-pong qui consiste à émettre des paquets de contrôle de type *echo* du protocole ICMP et de calculer le délai entre l'émission et le retour de l'écho. Un certain nombre de travaux récents [203] ont montré les limites de l'approche *aller-retour* par une mise en évidence de l'asymétrie des performances sur les liens Internet [158]. Ainsi de nouvelles techniques de mesures du délai sont apparues [113]. Elles nécessitent la synchronisation des horloges de l'émetteur et du récepteur par un système GPS (Global Positioning System) ou un protocole de type NTP [175].

Gigue

La gigue est la variation de délai de bout en bout notée j . La gigue de délai pour un flux de paquets est défini par la différence maximum de délai expérimentée par n'importe quels paquets pris deux à deux dans le flux. si d_0 est le délai de référence et d_k le délai du *k*ème paquet alors $j_k = d_k - d_0$

A cause de la gigue introduite par le réseau, les flux audio/vidéo capturés au même instant peuvent ne pas arriver simultanément chez le récepteur. Au niveau du récepteur, les applications temps-réel doivent bufferiser les données pour enlever la gigue ajoutée par le réseau et retrouver les relations temporelles originales. Dans [?], les auteurs proposent des mécanismes pour borner la gigue des canaux temps-réel dans les réseaux à commutation de paquets. Les liens ATM de type CBR, sont caractérisés par une gigue très faible. Par contre, dans Internet, le problème de la gigue est délicate et n'est pas réellement étudiée [73] comme nous le verrons au chapitre suivant.

L'IETF a aussi défini formellement la notion de gigue instantanée : variation de délai instantanée (instantaneous packet delay variation : IPDV) . C'est la différence de délai de transmission entre deux paquets k et $k + 1$ consécutifs. Cette gigue instantanée reflète l'évolution de l'état de congestion du lien. Si elle est stable, la charge du lien est constante. Si par contre elle augmente, elle indique la dérivation vers un état de congestion. Elle est utilisée dans certaines implémentations de TCP (TCP vegas) par exemple pour anticiper les pertes de paquets, donc les congestions et mettre en oeuvre les mécanismes d'évitement plus rapidement. Les travaux de Busse [33] ont montré l'exploitation de cette gigue instantanée dans un protocole adaptatif (voir section 4.3). La gigue ne permet d'anticiper un état de congestion que si le délai de bout en bout est relativement important. Cette contrainte en limite donc son usage dérivé.

3.3.3 Paramètres de fiabilité

Taux de pertes

Le taux de pertes l correspond au rapport du nombre de paquets non arrivés sur le nombre total de paquets transmis. Les pertes dans Internet sont causées par la congestion, l'instabilité du routage, les défaillances de liens et la non fiabilité des liaisons téléphoniques ou sans fil. La congestion est la cause la plus importante de pertes. La perte de paquet peut se produire soit par dépassement de capacité des buffers dans les routeurs ou dans les systèmes d'extrémité soit par violation de délai borné. La distribution des pertes est aussi une métrique très importante pour les protocoles adaptatifs tels que TCP. Un lien réseau peut être caractérisé par son taux d'erreur e qui est calculé par intervalle de temps relativement long. Il correspond au nombre de bits reçus erronés sur le nombre total de bit reçus ou le nombre de paquet erronés sur le nombre total de paquets reçus. Dans les liaisons filaires actuelles, ce taux d'erreur résiduel est très faible. Cependant, dans les réseaux sans fil, le taux d'erreur n'est plus négligeable.

3.4 Besoins de Qualité de service des applications

3.4.1 Dimensions *utilisateur* de la qualité de service

Les besoins de qualité de service des applications sont exprimés en terme de voeux subjectifs et ces estimations sont projetées sur les paramètres de performance du réseau. Par exemple, un utilisateur va choisir la vidéo en terme de résolution et de fréquence d'images qui se traduisent en besoin de débit réseau. L'acheminement des données entre les entités distribuées par un réseau de communication recouvre plusieurs dimensions qui reflètent les besoins opérationnels des applications [103] :

- la dimension volumique concerne la quantité de données pouvant être transportées par unité de temps (garantie de débit).
- la dimension temporelle est relative aux aspects temporels de la livraison (délai et gigue).
- la dimension sémantique reflète la qualité de la transmission (en terme de taux de perte ou d'erreur).

Une qualité de service réseau implique une sorte d'engagement sur une ou plusieurs de ces dimensions. Le modèle best effort de l'Internet ne différencie pas ces dimensions et le réseau ne prend aucun engagement quand à la livraison des données. A l'opposé, le modèle ATM offre une variété de types de services et des garanties strictes sur l'acheminement des données. Ainsi un service CBR, à débit constant, offre-t-il un service avec une gigue quasi nulle et un délai borné. Ce type de service convient particulièrement bien aux trafics pour lesquels il existe une relation temporelle forte entre l'émetteur et le récepteur. La table 3.1 ci-dessous donne des exemples d'utilisation des services ATM qui optimisent plus ou moins les métriques réseaux. Les services ATM correspondants sont :

- **CBR** le service à débit constant,
- **VBR** le service à débit variable temps-réel *rt* ou non temps-réel *nrt* ,
- **ABR** le service disponible
- **UBR** le service non spécifié. Enditemize

Application	CBR	rt-VBR	nrt-VBR	ABR	UBR
Données critiques	++	+	+++	+	NA
Interconnexion de réseaux locaux	+	+	++	+++	++
Emulation de circuits	+++	++	NA	NA	NA
Vidéo-conférence	+++	+++	NA	NA	NA
Audio compressée	+	+++	++	++	+
Distribution vidéo	+++	++	+	NA	NA
Multimédia interactif	+++	+++	++	++	+

TAB. 3.1 – Domaines d’application des catégories de services ATM (source ATM forum)

3.4.2 Dimension temporelle

Pour déterminer les besoins des applications en termes de qualité de service, on peut s’appuyer sur une classification qui permet de cerner, pour chaque type d’application, les dimensions critiques. La classification des applications de l’Internet définie dans le cadre du modèle IntServ [30] distingue les applications élastiques des applications rigides. Les applications rigides génèrent des flots de données dont la livraison à débit constant (CBR) est prévisible. Les applications élastiques, génèrent des rafales avec une livraison imprévisible de blocs de données à débit variable (VBR). Les applications telles que le transfert de fichiers envoient des données en masse ce qui augmente le débit de la source et peut utiliser toute la bande passante disponible (il n’y a pas de borne supérieure ni en délai, ni en débit).

Une autre typologie [221] distingue les applications personne-personne des applications personne-serveur. Il existe aussi des applications serveur-serveur. Lorsque deux personnes sont en communication, comme dans le cas des applications coopératives, les contraintes temporelles sont très fortes, les délais doivent être courts et ne pas varier. En effet, le contenu numérisé des applications multimédia temps-réel échantillonné et émis à intervalle de temps régulier, doit être transmis de manière isochrone afin d’être restitué correctement. Pour ces applications personne-personne, et en particulier pour la voix numérisée, le cas idéal est celui dans lequel le réseau introduit un délai faible et constant, c’est dire une gigue nulle. Un délai trop important a un impact néfaste dans une communication interactive telle qu’une conversation téléphonique. Une faible gigue est requise pour éviter des pertes tardives par dépassement de délai. Le fait d’avoir un service à gigue bornée, permet au site destinataire de calculer la quantité d’espace de buffer requise pour éliminer la gigue. Plus la gigue est faible, plus cet espace est limité. Ainsi, il est plus important d’avoir un délai et une gigue bornée qu’un délai moyen faible. En effet, plus un paquet arrive tôt, plus son temps passé dans le buffer de compensation de gigue sera long. Si chaque paquet est reçu, la qualité perçue est parfaite. Cependant, si un paquet est perdu, la qualité sera dégradée. Ainsi le délai, la gigue et dans une moindre mesure le taux de perte sont des facteurs déterminants de la qualité de service pour une application multimédia temps-réel. L’impact des conditions du réseau sur la qualité perçue des conversations voix ont été largement étudiées. Une série de publication de l’ITU-T propose le *e-model* qui analyse de façon subjective les conversations comme une fonction de nombreux paramètres telles que le délai de bout en bout, le taux de perte et le type d’encodage. Le *e-model* calcule un facteur taux de transmission (rating) R qui représente la qualité perçue par un nombre compris entre 1 et 100. $R = 100$ correspond à la qualité parfaite, au dessus de 60 la qualité est acceptable. R peut être exprimé de manière simplifiée par l’expression suivante : $R = R_0 - I_d - I_l$ où R_0 est la qualité de base (sans perte et dans délai), I_d est la pénalité due au délai et I_l est la pénalité due aux pertes de paquets. I_d a une expression analytique relative complexe et I_l est donnée par un ensemble de valeurs fini. Pour la plupart des applications de téléphonie, l’ITU-T G114 [116] recommande une valeur de 150ms comme borne supérieure de délai unidirectionnel. 150ms et 400ms sont potentiellement intolérables et au dessus de 400ms le délai est inacceptable. Au niveau de l’encodage, la norme G723.1 [?] est préférée pour la

voix sur IP (VoIP) car elle requiert un débit relativement faible (6.4 Kb/s contre 64Kb/s de la norme G711). Par exemple, pour une qualité subjective d'environ 60, G723.1 peut supporter jusqu'à 4% de pertes et jusqu'à 100ms de délai. Dans ces 100ms, 20ms sont pris pour la mise en paquets et le délai de transmission, laissant 80ms pour le délai variable dans les files. La tolérance pour les applications vidéoconférence est-elle de 200 à 300ms [116] [103]. Pour les applications interactives un délai d'aller-retour supérieur à 600ms dégrade l'interactivité d'une application [31].

La table 3.2 ci-dessous donne une classification des flux selon la dimension temporelle.

Type	Spécificité	Exemple
Asynchrones	Pas de contrainte sur la date de remise (élastique)	e-mail
Synchrones	Données sont sensibles au temps mais certaine flexibilité	telnet
Interactifs	Délais potentiellement perçus mais n'affectent utilisabilité ou fonctionnalité	jeu multi-utilisateurs
Isochrones	Sensibilité au délai peut affecter utilisabilité	audio-conférence
Mission-Critical	Délais de livraison peuvent affecter fonctionnalités	télé-chirurgie

TAB. 3.2 – Classification des flux selon la dimension temporelle

3.4.3 Hétérogénéité des besoins

Dans le domaine de la QoS, l'usage est de considérer les besoins des applications en terme de délai car ce sont les besoins les plus exigeants et difficiles à honorer dans un réseau partagé. C'est ainsi que l'on assimile souvent la notion de qualité de service à cette dimension temporelle. Cependant, les réseaux informatiques, et Internet en particulier, sont aujourd'hui utilisés pour des applications de plus en plus variées. Les applications communicantes couvrent un large spectre allant de la messagerie textuelle, à la distribution de logiciel, à la vidéo et l'audio de loisir, à la vidéo conférence, l'e-commerce, les jeux temps-réel multi-utilisateur, la visualisation scientifique et l'imagerie médicale. Il en résulte une très grande diversité des caractéristiques de trafic et des besoins en performances des applications existantes, ainsi qu'une profonde incertitude face aux applications futures. L'hétérogénéité et la fluctuation des besoins est vaste et recouvre plusieurs formes :

- Hétérogénéité des besoins pour un même type de flux
- Fluctuation des besoins dans le temps
- Hétérogénéité des besoins et des capacités des récepteurs

Hétérogénéité des besoins pour un même type de flux

Des études ergonomiques ont montré que les personnes peuvent utiliser l'audio et la vidéo aussi longtemps que le contenu informatif est au dessus d'un niveau minimum [216]. Ce niveau dépend du contenu du média et de la tâche à faire. Par exemple, l'utilisation d'une langue étrangère est plus délicate que la langue maternelle dans une audio-conférence. On comprend aisément que la qualité audio doit être meilleure dans le premier cas que dans le deuxième. Dans le cas de la vidéo, un utilisateur peut souhaiter accéder rapidement à une information incomplète pour jeter un coup d'oeil furtif sur une visioconférence ou une présentation. Un participant réel préférera une qualité image et audio supérieure. Selon la situation et l'exploitation de l'information que l'on veut faire, la qualité requise est donc différente. Un même média peut être considéré comme un flux isochrone, interactif ou synchrone selon les cas.

Fluctuation des besoins dans le temps

Symétriquement à la variation des performances du réseau, on constate aussi des fluctuations dans les besoins des usagers au cours d'une même session. Les expériences que nous avons présentées dans la section 3.2 précédente, nous ont montré que les sessions coopératives sont longues et que les besoins de communication peuvent évoluer en fonction de la phase de travail. Le gestionnaire de session devrait pouvoir renégocier dynamiquement des paramètres pour augmenter ou diminuer les performances selon telle ou telle dimension pendant des intervalles de temps précis. Parfois, lors des phases de congestion du réseau où les garanties ne

peuvent être maintenues ou offertes, il faudrait aussi pouvoir relâcher les contraintes sans rompre la session ou l'empêcher de démarrer.

Hétérogénéité des besoins pour de multiples récepteurs

Lors de sessions multicast en large groupe dans un environnement très hétérogène, les besoins des utilisateurs et leurs conditions de réception peuvent être très variés. Ainsi, pour un même flux, le service requis ne sera pas identique pour tous les récepteurs. Il est évidemment inutile de transporter une information vers un récepteur incapable de l'exploiter. Ce problème a été étudié dans le cadre des applications adaptatives du Mbone et nous le développons dans le chapitre suivant à la section 4.3.3.

3.4.4 Modèles de services et spécification des besoins

Pour honorer les besoins de qualité de service des applications, des modèles de services différenciés sont proposés dans les réseaux. Lorsqu'un réseau offre des services garantis, la clause la plus importante dans un contrat est la spécifications du profil de trafic injecté par l'utilisateur et dont l'acheminement est assuré par le fournisseur. Les spécifications du trafic et du service désiré peuvent être spécifiées sur la base d'un canal virtuel dans ATM, sur une base par flux dans IntServ ou dans un accord de service, **service level agreement (SLA)** dans l'architecture DiffServ [182].

Réseaux ATM

Quatre classes de services ont été identifiés dans le cadre de la technologie ATM. C'est la couche d'adaptation de bout en bout, **AAL, Application Adaptation Layer**, qui donne à ATM la flexibilité de transporter des types de services différents avec le même format de cellules.

AAL1	Services CBR	relation temporelle forte entre émetteur et récepteur	Audio-Vidéo à débit fixe
AAL2	Services VBR	cf AAL1 + débit variable	Audio-Vidéo à débit variable
AAL3/4	Services CO et CL	trafic en rafale	Transfert de fichiers
AAL5	Services CL	Simplification de AAL3/4	Réseaux locaux haut débit

TAB. 3.3 – Classes AAL et exemples d'utilisation (source ATM forum)

Un certain nombre de paramètres doivent être définis par l'utilisateur lorsqu'il négocie un contrat de trafic avec le fournisseur. Les principaux paramètres sont le débit crête *Peak Cell Rate (PCR)*, le délai de transfert de cellule *Cell Transfert Delay (CTD)*, la variation de délai *Cell Delay Variation (CDV)*, le débit de cellules *Sustainable Cell Rate (SCR)*, la taille maximale des rafales *Maximum Burst Size (MBS)*, le taux de pertes de cellules *Cell Loss Ratio (CLR)*, et le débit minimal de cellule *Mean Cell Rate*. La table 3.4 suivante, explicite les paramètres qui doivent être spécifiés pour chaque service.

Service	Fréquentiel	Temporel	Sémantique
CBR	PCR	CTD + CDV	-
rt-VBR	PCR + SCR + MBS	-	-
nrt-VBR	PCR+SCR+MBS	CTD	CLR
ABR	PCR+MCR	-	-
UBR	-	-	-

TAB. 3.4 – Paramètres de spécification des services ATM selon les dimensions (source ATM forum)

Services définis dans IntServ

Le groupe IETF Intserv [30], a défini plusieurs modèles de services. Les trois principaux sont :

- le service garanti (GS),
- le service à charge contrôlée (CL)
- le service best effort (BE)

Le service GS définit une relation contractuelle entre le client et le fournisseur de réseau. Chaque partie devant honorer son contrat. Le réseau peut rejeter une requête s'il n'est pas capable de l'honorer. Dans le service GS, l'application n'a pas à changer son comportement.

Dans le service prédit CL, le contrôle d'admission est basé sur une évaluation de la charge actuelle du réseau mesurée, alors que dans le service garanti, la charge est basée sur la caractérisation pré-spécifiée des connexions existantes. Ainsi comme la charge mesurée est variable, l'engagement est moins fiable. L'application peut s'adapter.

Dans IntServ, la spécification d'un trafic se fait avec un **Tspec** [208] sous forme d'un token bucket (seau à jetons) qui exprime le débit moyen du flux r , ainsi que la durée et l'amplitude des rafales b . Pour un service garanti, l'utilisateur doit de plus préciser le débit maximal p (peak rate). L'enveloppe de trafic $A(t)$ définit la quantité maximale de trafic que l'utilisateur peut injecter dans le réseau. Elle s'exprime ainsi :

$$A(t) \geq \min(M + pt, b + rt)$$

où M la taille maximale d'un paquet, p , b , r les paramètres précédents.

Pour un simple utilisateur exprimer les besoins de qualité de service des différents flux de son application à l'aide de token bucket est un processus relativement lourd. Se pose aussi le problème du type de services et du type de garanties réellement requis par les applications ainsi que celui de la flexibilité de l'usage du réseau.

Services définis dans DiffServ

Le groupe IETF DiffServ a défini deux standards de services pour les agrégats de trafic **Premium Service** qui assure un débit et un délai borné et **Assured Service** qui offre des garanties de débit et de taux de pertes. Nous développons les caractéristiques de ces standards dans la section 4.2.2 du chapitre suivant. Ces services ne sont pas offerts directement aux applications.

Un accord de service, SLA, est un contrat de service entre un client et un fournisseur de service. Le client peut être un utilisateur final ou bien un domaine adjacent amont. Le SLA spécifie le trafic mais aussi tous les aspects concernant l'acheminement des paquets (forwarding) que le client doit recevoir du fournisseur. Les prix ainsi que les procédures de facturation, les services de cryptage, les mécanismes d'authentification utilisées pour vérifier l'utilisateur peuvent être inclus dans le contrat. Bien qu'un SLA est relativement stable, il peut être mis à jour pour refléter les changements de profil de trafic et le service requis. Ceci nécessite une renégociation. Une définition de SLA contient les paramètres suivants :

- Identificateur de source : adresse IP, N° port, protocole
- Identificateur de destination : adresse IP, N° port, protocole
- Débit
- Taux de perte
- Date de début du flux
- Date de fin du flux

L'expression d'un SLA est beaucoup moins précise et plus simple que celle d'un token bucket. Au fil des années et de l'évolution des technologies les garanties de qualité de service offertes par les réseaux sont de moins en moins strictes et fines. Les mécanismes requis pour spécifier les trafics et les services en deviennent de moins en moins complexes. Dans le chapitre suivant, j'essaie d'analyser cette évolution.

3.5 Conclusion

Dans la projet P2-ATM, nous avons pu observer que la technologie ATM, conçue pour les réseaux numériques à intégration de service large bande [196], est effectivement un bien meilleur support pour la collaboration et la vidéoconférence que l'Internet ou le réseau RNIS à bande étroite. Aux yeux de l'opérateur participant au projet, le gain obtenu est cependant apparu insuffisant au regard des coûts de mise en oeuvre. Par ailleurs

nous avons montré que la qualité du réseau n'était pas le seul facteur déterminant la qualité d'un dispositif de collaboration. La chaîne de la qualité de service de bout en bout s'est avérée très complexe à configurer et à ajuster. Les équipements d'extrémité (ordinateur, codec, cartes d'interface, dispositifs d'entrée-sortie) jouent tous un rôle fondamental. Une des principale difficulté de l'obtention de bonnes performances est liée à l'interdépendance de multiple éléments dans une chaîne de qualité de service à la fois horizontale d'un bout à l'autre du réseau et verticale de haut en bas du modèle architectural. L'absence d'API et d'applications informatiques exploitant pleinement les possibilités de qualité de service d'ATM a été par ailleurs un grave handicap à son déploiement massif jusqu'à l'utilisateur final. Rappelons que l'API socket a joué un rôle majeur dans le succès de la technologie TCP/IP et en freine peut être aussi son évolution aujourd'hui.

L'expérience P2-ATM a mis en évidence l'hétérogénéité des besoins pour un même flux dans le temps et selon les usages. Même dans le cadre d'un contrat de trafic, les caractéristiques de cette interconnexion internationale complexe n'ont pas été stables au cours des longues sessions de travail. Face à la très grande hétérogénéité des applications et des besoins, des questions fondamentales et récurrentes se posent :

- quelles performances sont réellement nécessaires ?
- quand et pourquoi faire ?
- quels types de garanties veut-on offrir aux usagers.
- comment capturer les besoins des applications ?
- comment permettre aux applications de les exprimer ?
- quels mécanismes faut-il inventer pour permettre une adaptation à des facteurs aussi variés que la tâche à effectuer, le support technique disponible mais aussi l'expertise technique des acteurs, la maturité du groupe, l'aspect financier ?

La multitude de besoins et d'usages requiert la fourniture de modèles services et de disciplines de services flexibles pour allouer dynamiquement différents profils de performances aux différentes connexions. Suite à mon changement d'environnement de recherche correspondant à mon intégration dans l'équipe RESAM, je me suis concentrée sur l'aspect performance réseau et sur les mécanismes de gestion des ressources réseaux que je développe au chapitre 4. Dans le chapitre 5, je développe mes idées sur les moyens d'introduire la flexibilité de qualité de service nécessaire. Dans le cadre de mon activité sur les grilles de calcul, je m'intéresse à nouveau à la problématique utilisateur de la capture et de la spécification des besoins des applications aux extrémités (voir 6.3.1 qui demeure une question ouverte et fondamentale aussi bien pour les chercheurs que les équipementiers ou les fournisseurs de capacités et de services réseau. Mon expérience concrète de la technologie ATM et mes analyses des besoins des utilisateurs ont servi de points d'appuis à tous ces travaux ultérieurs.

Chapitre 4

Qualité de service dans les réseaux IP

4.1 Introduction

Vers la fin des années 90, la technologie ATM a commencé à s'effacer au profit de l'expansion toujours plus importante de l'Internet. Les efforts de recherche sur la QoS se sont nettement focalisés sur le protocole IP. Les travaux sur la Qualité de service dans Internet, poussés par les évolutions des applications vers le multi-média, ont commencé dès le début des années 90 au sein de l'IETF. Avec Benjamin Gaidioz, nous avons mené un étude approfondie des différentes approches proposées au niveau réseau, c'est à dire du protocole IP ainsi qu'au niveau des extrémités, dans l'approche adaptative. Dans ce chapitre, je développe les travaux menés au plan réseau avec la proposition d'un modèle de service équitable **Balanced Forwarding** dans le cadre du modèle DiffServ et nos investigations au plan des techniques adaptatives avec la conception et la réalisation d'un module d'adaptation vidéo générique **Netstre@mer**. Ces études, nourries par des implémentations et des évaluations expérimentales, nous ont conduits à développer de nouvelles solutions, en rupture avec l'existant et que je présente au chapitre suivant.

4.1.1 Problématique IP et Qualité de Service

Les réseaux IP classiques offrent un simple service : le service best effort. Un tel modèle de service permet aux routeurs d'être sans état et de ne garder aucune information de grain fin à propos du trafic. L'architecture Internet est donc basée sur le concept que tous les états relatifs à un flux doivent être dans le système d'extrémité. Cette propriété confère au système global un grande robustesse. En fournissant un modèle de service minimaliste, l'Internet est extensible en taille et en hétérogénéité des applications et des technologies. Ensemble, elles sont les deux raisons techniques majeures de son succès. Par ailleurs, l'utilisation de la couche réseau est libre de toute tarification puisque les mêmes ressources sont disponibles pour tous les utilisateurs. L'inconvénient est que, puisqu'il n'y a pas de contrôle d'admission, le réseau peut être perturbé par des utilisateurs trop gourmands. Comme IP est un protocole sans connexion, le concept de contrat de trafic n'existe pas. Si le débit avec lequel le trafic est dirigé sur les interfaces dépasse la vitesse avec laquelle ces mêmes interface sont capables d'acheminer le trafic vers l'aval, des congestions peuvent se produire. Le trafic en excès est placé dans les files d'attente des dispositifs physiques jusqu'à débordement de ces files. Ainsi, les applications peuvent faire l'expérience de délai variables ou de pertes de paquets. Les congestions peuvent entraîner des pertes transitoires ou bien des pertes de longue durée. Le protocole d'extrémité TCP a été conçu pour assurer la fiabilité et la retransmission si nécessaire. Par ailleurs, comme le réseau n'effectue pas contrôle de congestion, cette fonction doit impérativement être assurée par les extrémités [118]. Le service réseau minimaliste fournit par IP a bien fonctionné et ce de manière surprenante, pendant plus de trente ans sans que les alternatives telles que le relais de trame ou ATM, fournissant des capacités de qualité de service, ne connaissent un tel succès et ne freinent la croissance exponentielle d'IP. Cependant, l'émergence d'applications sensibles au délai telles que la téléphonie IP, la vidéoconférence et les jeux vidéo interactifs, commencent à menacer l'ubiquité d'IP. Même si un ensemble de mesures prises en 1998 sur le réseau vBNS

(Very High Performance Backbone Service) suggère que 95% du trafic IP est de type TCP avec 70% de trafic HTTP, 5% de trafic FTP et 5% de trafic SMTP, cela ne veut pas nécessairement dire qu'il n'y a pas de besoin de transport temps-réel sur RTP/UDP (voir ??). Nous pensons que les usagers boudent les applications temps-réel parce que la qualité offerte actuellement est médiocre. Le fort développement des applications de streaming audio et vidéo qui utilisent TCP a été un moyen de contourner le problème des délais dans Internet en réalisant du téléchargement. Cela ne résout pas le problème des applications interactives et coopératives. Il y a aussi beaucoup d'incitation à utiliser le protocole TCP dans Internet pour son rôle fondamental dans le contrôle de congestion [83].

Nous avons montré dans le chapitre précédent que les flux qui voudraient circuler dans Internet ne sont pas tous également sensibles aux pertes et aux délais d'acheminement. Si, par sa simplicité, IP a pu apporter beaucoup, il est aujourd'hui trop plat, pauvre et limité et le paradigme Best effort doit être réexaminé.

4.1.2 Considérations architecturales

Généralement, on oppose deux approches architecturales pour résoudre le problème de la qualité de service réseau : l'approche **dans le réseau** et l'approche **aux extrémités**. Dans l'approche **aux extrémités**, la qualité de service peut être traitée au niveau utilisateur (a) [33], au niveau application (b) au niveau système (c) ou au niveau de la couche transport (d) [197]. Les applications adaptatives traitent la QoS au niveau (b) en mettant en oeuvre des algorithmes de compensation d'erreurs ou des mécanismes d'adaptation pour pallier le manque de qualité du réseau. Une activité importante est menée au niveau (c) avec les architectures de QoS et des interfaces de programmation, API de QoS [13]. La couche transport est la première couche susceptible de réaliser l'adaptation de bout en bout de niveau (d). La couche AAL, *adaptation application layer*, apporte la flexibilité du modèle ATM. TCP dans le modèle IP est un algorithme adaptatif.

Dans l'approche **dans le réseau** il existe des solutions à tous les étages du modèle en couches. Les architectures IntServ et DiffServ proposent des solutions de QoS pour la couche réseau IP. Pour la couche 2, l'IETF a défini une signalisation à la RSVP pour la réservation de bande passante dans Ethernet et d'autres réseaux de la famille 802. IEEE a proposé aussi d'étendre Ethernet pour ajouter 8 niveaux de priorité dans la norme 802.1p qui est exploitée aussi dans le cas des réseaux mobiles (norme 802.11). La commutation en longueur d'onde (*lambda switching*), proposera des longueurs d'onde dédiées offrant des capacités distinctes et garanties aux utilisateurs [?].

L'opposition d'approche *dans ou hors réseau* est l'objet de débats récurrents, preuve qu'aucune solution unique n'existe pour répondre à l'ensemble des questions. Selon la philosophie IP, le coeur de réseau devrait rester le plus simple possible et les services être assurés par les extrémités [39]. Au contraire, dans la technologie ATM, la qualité de service est gérée dans le réseau, mais sa mise en oeuvre de bout en bout s'est avérée très complexe.

Dans la suite nous examinons les avantages et les limites d'une gestion de la QoS au niveau réseau dans IP (section 4.2) puis ceux amenés par un traitement aux extrémités (section 4.3). Dans quel cas est-il plus pertinent d'avoir un réseau élastique avec des applications élastiques, dans quel cas le modèle de performance doit-il être plus rigide ?

4.2 Solutions réseaux

4.2.1 Mécanismes de base

Le surdimensionnement

Les âpres défenseurs de l'IP Best effort proposent la solution du surdimensionnement. Le principe est d'augmenter toujours la bande passante disponible pour éviter la congestion, les délais de file d'attente, les pertes de paquets. Ainsi, s'il y a toujours suffisamment de capacité réseau, les garanties de service peuvent être offertes. L'évolution des technologies optiques, telle que le *DWDM* délivrant une capacité croissante en augmentant le nombre de longueurs d'onde, permet la construction de réseaux multiterabit. Il suffit de faire confiance aux progrès des technologies optiques des supports de transmission et de ne rien changer au ni-

veau de la couche IP. Cependant le surdimensionnement ne peut être qu'une solution partielle à la fourniture de qualité de service et ce pour plusieurs raisons :

- ce n'est pas une solution de bout en bout. S'il est possible de sur-provisionner une épine dorsale et offrir de la QoS à ce niveau du réseau, les transactions Internet, traversant deux ou plusieurs domaines administratifs n'ont aucune garantie. On ne peut en rien assurer que deux réseaux surdimensionnés individuellement auront des capacités de couplage, *peering*, suffisantes pour écouler le trafic entre eux. Ainsi, des campagnes de mesures Internet ont montré qu'une large part des problèmes de performances (délais et pertes) se produisent aux points d'accès réseaux publics qui interconnectent plusieurs réseaux.
- pour réduire la congestion et maintenir des délais de bout en bout faibles, un réseau peut avoir à faire du contrôle de trafic en écartant des paquets à ces points d'entrée du réseau. Puisque ces pertes se produisent aux points d'interconnexion, un opérateur peut toujours blâmer les autres réseaux tout en affichant un faible taux de pertes à l'intérieur de son propre réseau.
- on donne la même qualité de service aux paquets de tout type d'application. Et, en cas de surcharge occasionnelle, les paquets sensibles au délai seront plus pénalisés que d'autres par une mise en file d'attente tandis que ceux sensibles aux pertes seront plus pénalisés par les *drops*.
- le dernier kilomètre est traditionnellement le moins bien dimensionné. D'aucune manière un surdimensionnement du coeur de l'Internet ne pourra permettre d'obtenir une qualité vidéo suffisante à partir d'une connexion modem. Il est donc tout aussi important d'approvisionner correctement ce dernier kilomètre. Mais l'émergence des communications sans fil avec leur faibles capacités de communication et de traitement, laisse présager que l'hétérogénéité des connexions d'accès sera toujours considérable.
- finalement, le sur-provisionnement est une solution inefficace en termes économiques et de gestion de ressources. Tout le trafic reçoit la même qualité très élevée, même si toutes les applications n'en ont pas besoin.

Prévenir et contrôler les congestions

Puisque le principal problème du Best Effort et le responsable de la faible qualité de service d'Internet est la congestion, une autre solution est de prévenir les congestions avant qu'elles ne se produisent et de les contrôler si elles arrivent. Une étude détaillée des mécanismes de contrôle de congestion pour les flux unicast et multicast est présentée dans [62]. Ces mécanismes de gestion active des files d'attente tels que RED [84] tendent à améliorer le service Best Effort en ayant plus de contrôle sur les congestions. Mais ils ne permettent pas d'offrir des garanties ni des services différenciés. Des mécanismes de plus en plus sophistiqués pour le contrôle du trafic et des ressources ont été progressivement ajoutés à ces mécanismes de gestion de files dans les routeurs IP. Ces mécanismes sont les briques de base de la conception des architectures de Qualité de Service de l'Internet. On distingue généralement ceux qui opèrent dans le plan contrôle de ceux du plan données. La table 4.1 ci-dessous répertorie les mécanismes proposés dans les différents plans. La figure 4.1 donne un schéma général de l'intégration de ces différents modules fonctionnels au sein d'un routeur IP actuel. Je les détaille ci-dessous.

Plan données

Les mécanismes du plan données implémentent les actions que les routeurs doivent entreprendre sur chaque paquet pour servir différents niveaux de service. On distingue :

- les mécanismes de conditionnement : classification, marquage, mesure, contrôle et lissage
- les mécanismes de gestion de files
- les mécanismes d'ordonnement

Mécanismes de conditionnement

Lorsqu'un paquet est reçu, le classifieur détermine à quel flux ou à quelle classe il appartient en fonction du contenu d'une certaine portion de l'entête et selon certaines règles. Il existe une classification générale qui associe une signature avec des informations de niveau transport (port) et des champs de niveau IP (adresses). Cette opération est coûteuse. Elle est effectuée dans tout routeur IntServ et dans les routeurs de frontière

	Plan	Architecture cible
Classification MB	données	Intserv - DiffServ (bordure)
Classification BA	données	DiffServ (coeur)
Mesure	données	Intserv - DiffServ (bordure)
Lissage	données	Intserv - DiffServ (bordure)
Ecartement	données	Intserv - DiffServ
Marquage	données	DiffServ(bordure)
Gestion de file	données	Intserv - DiffServ (coeur)
Ordonnancement	données	Intserv - DiffServ (coeur)
Contrôle d'admission	contrôle	Intserv - DiffServ (bordure)
Contrôle de politique	contrôle	DiffServ (bordure)
Gestion de trafic	contrôle	DiffServ (bordure)
Reservation de ressources	contrôle	Intserv

TAB. 4.1 – Récapitulatif des mécanismes de QoS

FIG. 4.1 – Architecture actuelle d'un routeur Internet type

de DiffServ (multifield classification : MF). Une classification basée sur un schéma bit ordonne les paquets selon un seul champ de l'entête IP. C'est une opération simple et très rapide. Dans DiffServ c'est la *behavior aggregate* classification (BA), elle est employée uniquement dans les routeurs de coeur. Après la classification, le paquet est passé à une instance logique de conditionnement qui peut marquer un champ, mesurer les propriétés temporelles d'un flux et les comparer à un certain profil pour décider si le paquet est " dans " ou " hors " profil. Un *shaper*, basé sur un seau percé, *leaky bucket* ou sur un seau à jetons *token bucket* [73], peut retarder certains ou tous les paquets pour le rendre le flux conforme au profil. C'est ce que l'on appelle le lissage de flux. Certains paquets peuvent être éliminés par un *dropper*.

Mécanismes de gestion de file

Un des buts principaux de la QoS IP est de contrôler la perte de paquets. L'espace mémoire alloué aux files d'attente dans les routeurs est conçu pour absorber les rafales de données de courte durée. Limiter la taille de ces files peut aider à réduire les bornes de délai. Traditionnellement, les paquets sont jetés lorsque les files débordent. Ce sont soit les derniers arrivés (tail drop) soit les plus anciens dans la file (front drop), soit des paquets choisis au hasard qui sont éliminés. Il y a deux inconvénients à cette technique : le lock-out

ou monopolisation de la file par certains flux et le problème des files pleines (full queue) qui allongent le délai. Pour éviter ces deux problèmes, les mécanismes sophistiqués de gestion active de file d'attente ont été préconisés et introduits dans les routeurs IP. Ainsi le mécanisme RED (Random Early Detection) [84] est un algorithme de gestion active de file d'attente, de plus en plus populaire et recommandé par l'IETF [?]. Le principe de RED est de surveiller l'évolution de la file d'attente et d'éliminer des paquets aléatoirement pour prévenir les congestions. Ce mécanisme écarte les paquets de la file de transmission d'un lien avec une probabilité de p calculée comme une fonction croissante de la moyenne courante de la taille q de cette file, $p = H(q)$. RED offre plus de flexibilité que le mécanisme classique de Tail Drop dans le choix du comportement du routeur en cas de congestion. Un certain nombre de travaux ont montré les inconvénients et les limites du mécanisme RED simple [134]. Dans certains routeurs le mécanisme WRED [152] (weighted random early detection) est proposé comme alternative à RED. L'écartement se fait non plus de manière purement aléatoire mais est basé sur la priorité du paquet représentant le poids du flux ; cette priorité est marquée dans le sous-champ IP precedence du champ type de service de l'entête IP. RIO [40] raffine le concept RED en introduisant les bits " in " et " out " selon que les paquets sont dans ou hors profil. Ainsi les paquets " out " sont éliminés préférentiellement. Le mécanisme ECN, Explicit Congestion Notification quand à lui permet d'envoyer un signal à la couche de transport pour prévenir de la congestion en positionnant un bit spécifique de l'entête du paquet [180]. C'est un mécanisme inspiré de la notification FECN (forward explicit congestion notification) défini dans le relais de trame. Son apport au niveau des performances de bout en bout observée dans TCP par exemple est encore en cours d'évaluation [80].

Mécanismes d'ordonnement

La fonction d'un ordonnanceur est de sélectionner le paquet à transmettre au cycle suivant, pour chaque interface de sortie d'un routeur et parmi les paquets disponibles et appartenant aux flux partageant la même file de sortie. L'ordonnement joue une part importante dans le contrôle du délai. Les disciplines de services doivent être simples pour permettre une analyse aisée ainsi qu'une implémentation efficace très haut débit. Les ordonnanceurs font l'allocation effective des ressources telles que la bande passante (quels paquets doivent être transmis), de la rapidité (quand ces paquets doivent-ils être transmis) et de l'espace mémoire (quels paquets sont écartés). Ces disciplines de service affectent donc trois paramètres de performance qui sont : le débit, le délai et le taux de perte.

Dans [224] Zhang présente une étude très détaillée des disciplines de services disponibles pour garantir les performances dans un réseau à commutation de paquets . Les disciplines de service peuvent être classifiées selon qu'elles sont *work conserving*, un serveur n'est jamais inactif lorsqu'il y a un paquet à transmettre ou *non work conserving*, à chaque paquet une date d'éligibilité est affectée. Même lorsque le serveur est inoccupé, si aucun paquet n'est éligible, aucun paquet ne sera transmis. Zhang a montré, que le type discipline de service (work conserving ou non) affecte le délai de bout en bout, les besoins en espace mémoire (bufferisation), les caractéristiques de la gigue.

Il existe une grande variété d'algorithmes d'ordonnement qui sont analysés dans [102], [199]. Les plus courants sont :

- FIFO : first in, first out qui est la politique d'ordonnement la plus simple. Pas de différenciation de flux ou de classe, pas de garantie de délai ou de débit. Cet ordonnanceur a été optimisé pour le Best effort.
- PQ : strict priority queuing insère les paquets prioritaires en tête de file. Ce type de discipline permet de fournir le service premium [75] de l'architecture DiffServ.
- CBQ : class based queuing : fournit une file séparée pour chaque classe. En général, chaque file est gérée en FIFO. Aucune garantie de délai et de débit ne peut être fournie aux flux individuels.
- WRR : Weighted Round Robin est un algorithme classique d'ordonnement de type tourniquet qui permet de retirer les paquets dans différentes files selon une certaine pondération par file. Pour la mise en oeuvre du service scavenger sous Linux nous utilisons CBQ associée à WRR (voir section 4.2.3).
- WFQ : weighted fair queuing est une variante de WRR dans laquelle les poids sont couplés aux débits réservés. Cet ordonnanceur peut fournir une garantie de délai aux flux individuels. Mais il ne peut pas séparer la garantie de débit de la garantie de délai. Le problème résultant est qu'un flux qui a une bande

passante allouée faible peut obtenir un délai de bout en bout important. Un nombre important de variantes ont été proposées.

- GPS : general processor sharing est défini comme le modèle fluide de trafic et sert de modèle théorique de référence.
- EDF : Earliest deadline first est une forme dynamique d'ordonnancement par priorité. A chaque paquet on assigne un délai limite qui est la somme de la date d'arrivée et de la garantie de délai. Couplé avec un trafic shaper, EDF peut séparer la garantie de délai et de débit.

Un certain nombre de critères peuvent être identifiés pour classer ces mécanismes d'ordonnancement et de gestion de file. On distingue les propriétés :

- isolation : protection vis à vis du trafic en excès des autres utilisateurs,
- équité : accès à la capacité en excès, efficacité, nombre de flux pouvant être servi par un certain niveau de service,
- complexité : en terme de surcoût d'implémentation et de contrôle.

Un ordonnanceur doit optimiser un certain nombre de critères, délai, débit, etc, mais doit rester très simple. En effet, il est censé opérer à la vitesse du lien, ce qui l'oblige, pour un débit OC-48, à prendre une décision en 100ns pour chaque paquet. Dans un ordonnanceur à classement par priorité, une variable globale, le temps virtuel, est associée à chaque lien de sortie de l'équipement. Une estampille, calculée comme fonction de cette variable, est associée à chaque paquet du système. Les paquets sont classés selon leur estampille et sont transmis dans cet ordre. La complexité d'une implémentation de tout algorithme basé sur la priorité dépend de la complexité de calcul de l'estampille associée à chaque paquet. Par exemple, pour maintenir le temps virtuel dans un algorithme WFQ, il faut traiter au moins V évènements, correspondant au nombre V de connexions actives, pendant la transmission d'un seul paquet. La complexité est en $O(V)$. Ainsi l'algorithme Weighted Fair queueuing est l'algorithme d'ordonnancement idéal en terme de propriétés de délai et d'équité, mais son implémentation est difficile et complexe.

Mécanisme du Plan contrôle

Un certain nombre de mécanismes concernent la configuration des routeurs pour assurer que certains paquets reçoivent un traitement spécial et qu'un certain nombre règles concernant l'utilisation des ressources sont bien appliquées. Trois type de mécanismes sont identifiés :

- le contrôle d'admission
- le contrôle de la politique
- la gestion des ressources

Le contrôle d'admission

Le contrôle d'admission doit décider s'il y a assez de ressources dans le réseau pour accepter une nouvelle connexion. Il s'agit de calculer correctement une région d'admission pour ne pas refuser inutilement l'accès à certains flux tout en évitant les violations de QoS. Il existe trois approches pour contrôler l'admission d'un nouveau trafic : l'approche déterministe, l'approche statistique et l'approche basée sur la mesure. Les deux premières utilisent une estimation *à priori* tandis que la dernière est basée sur des mesures courantes de certains paramètres critères. L'approche déterministe effectue un calcul du pire des cas pour éviter toute violation de QoS. Cette technique, si elle est efficace pour les trafics fluides, l'est beaucoup moins pour les trafics en rafale et conduit à une sous-utilisation des ressources. Les deux autres approches autorisent une faible probabilité de violation de la QoS pour optimiser l'utilisation des ressources.

Le contrôle de la politique

La politique de QoS spécifie les règles d'accès aux ressources et aux services selon des critères administratifs. Elle définit, dans un domaine administratif donné, si tel utilisateur, site ou application peut accéder aux ressources et dans quelles conditions. Depuis quelques années, des architectures basées sur les politiques ont été définies [179], avec des modèles à trois tiers ou à deux tiers, un point de mise en oeuvre de la politique, *Policy Enforcement Point PEP*, un point de prise de décision, *Policy Decision Point PDP*, qui peuvent

être combinés en une seule entité, et un dépôt des politiques. Pour échanger les informations entre ces entités, des protocoles standards sont nécessaires. *COPS, Common Open Policy Service*, [29] permet les échanges entre PDP et PEP. *LDAP, light weight access protocol* [223] permet l'accès aux politiques stockées dans le dépôt dans des *PIB : policy information base*. Un nombre important de travaux autour des réseaux basés sur les politiques, *policy based network*, ont vu le jour ces dernières années. On commence à voir apparaître aussi des propositions de fusion des aspects politiques de sécurité et politiques de QoS.

La gestion des ressources

La gestion des ressources entre domaines administratifs est un problème très important et a une grande influence sur la qualité de service de bout en bout. Dans l'architecture DiffServ, un courtier de bande passante, le *Bandwidth Broker BB*, a, dans chaque domaine administratif, la mission de contrôler les opérations au niveau des routeurs de bordure. Il inclut des fonctions de *PDP* tandis que le routeur de bordure sert de *PEP*. L'architecture d'un *BB* a quelques similarités avec le protocole de routage *BGP4* qui sert de protocole standard pour le routage inter-domaine. Le routage et la QoS sont souvent présentés de manière indépendante, mais certains proposent une fusion du routage et de la QoS pour résoudre le problème de la QoS [184].

4.2.2 Architectures de QoS-IP standard

Depuis la fin des années 90, ces mécanismes sophistiqués du plan contrôle et du plan données se développent dans les équipements. Ainsi, avec un tel soutien des fabricants et des organismes de standardisation, on peut penser que des services évolués vont se déployer à large échelle dans les années futures. Mais le choix de l'architecture et des modèles de services appropriés à Internet reste encore une question ouverte. Le problème de la qualité de service revient à un problème d'optimisation du partage des ressources. Plusieurs approches sont donc possibles au niveau *réseau* :

- fournir un service déterministe avec des **garanties de performances strictes**. Cette solution requiert une réservation préalable des ressources.
- fournir un service statistique avec des **garanties statistiques absolues**. Pour cela une priorisation des flux est nécessaire. Les garanties peuvent être absolues ou relatives, c'est à dire dépendant ou non de la charge courante. Pour obtenir des garanties statistiques absolues il faut mixer une réservation stricte aux agrégats de flux et traiter les paquets selon des classes de priorité.
- fournir un service statistique avec des **garanties statistiques relatives**. Les garanties statistiques relatives sont offertes lorsque l'on ne fait que de la priorisation de paquets par classes.
- fournir un **service sans garantie** ; Aucun mécanisme à l'intérieur du réseau n'est requis mais une surveillance des performances et une adaptation aux extrémités est indispensable. C'est le cas TCP et des applications adaptatives sur IP Best effort.

Différentes architectures ont été proposées dans Internet. Nous les présentons selon le niveau de garantie offert. Cette présentation correspond aussi à leur chronologie d'apparition.

IntServ : Garanties strictes et réservations de ressources

Plusieurs modèles de gestion de qualité de service adoptent la solution de réserver des ressources afin de garantir des services aux utilisateurs. Les ressources réseau sont réservées selon une requête de QoS émanant de l'application. Une telle approche nécessite un protocole de réservation qui se fait généralement par une signalisation explicite dans le plan contrôle. Les seuils minimum de qualité sont déterminés par les programmeurs d'application ou par les utilisateurs puis transmis au gestionnaire de QoS. Une phase de négociation est entamée entre le gestionnaire et l'application pour déterminer si ces seuils peuvent être garantis. On détermine alors les besoins en ressources système et réseau, ces ressources sont ensuite réservées pour garantir le service.

Ainsi dans un réseau basé sur la technologie ATM, on réserve au préalable un canal virtuel avec une bande passante et des caractéristiques de débit (CBR, VBR) fixes. C'est au gestionnaire du réseau (fournisseur) d'assurer que le service offert l'est bien avec les caractéristiques spécifiées et négociées.

L'architecture **Intserv** est la première à avoir été élaborée pour fournir à Internet un paradigme prenant en considération les services temps-réels. Elle est basé sur ce modèle. Le but est de fournir un lien de communication à **qualité constante** en terme de débit, délai, taux d'erreur dans Internet. En fait, la plus grande force de cette architecture réside dans l'approche systématique et rigoureuse qu'elle propose au niveau de :

- l'identification des services intéressants
- l'analyse des spécifications réseaux nécessaires pour les supporter
- la proposition d'un cadre de référence pour l'implémentation

L'architecture Intserv est donc un cadre de services . Les principaux standards ont été développés entre 1993 et 1995 [219] [194], [208]. Les services qui ont été identifiés sans ambiguïté sont le service garanti et le service à charge contrôlée.

Le **service garanti, GS**, est défini pour émuler au mieux un circuit virtuel dédié. Il fournit des bornes, que l'on peut prouver mathématiquement, sur les délais de file d'attente de bout en bout en combinant les paramètres des différents éléments du réseau le long du chemin tout en assurant la disponibilité de la bande passante selon les paramètres d'une spécification. Le **service à charge contrôlée, CLS**, est équivalent à un service Best Effort sous des conditions de faible charge. Ainsi, il est mieux que le Best Effort mais ne fournit pas un service garanti borné comme le promet le service garanti.

Les composants de l'architecture IntServ sont le protocole de réservation de ressource, le contrôle d'admission et les mécanismes d'ordonnancement des paquets dans les routeurs. Le protocole RSVP [220] à états dynamiques, *soft state*, permet l'acheminement d'une requête de réservation ainsi que la réservation effective des ressources le long d'un chemin. Ce protocole de réservation, retenu par le groupe IntServ, présente cependant plusieurs faiblesses qui en rendent l'implémentation complexe. Certaines évaluations d'implémentation du protocole RSVP [151] ont montré que le surcoût induit par le traitement par flux augmente linéairement avec le nombre de flux. Cela pose un problème de passage à l'échelle d'une part dans le plan donnée pour le traitement des paquets, et dans le plan contrôle pour la signalisation. Plusieurs protocoles de réservation plus légers ont été proposés tels que SRP, scalable reservation protocol [7] ou YESSIR [151]. Par exemple, contrairement à RSVP, ce dernier utilise une seule passe et fait de la réservation partielle tout en mettant en oeuvre un mécanisme d'évitement de la fragmentation des ressources. Les principaux problèmes d'IntServ sont donc l'extensibilité, le maintien des états de contrôle et d'ordonnancement pour chaque flux dans tous les routeurs, les fonctions de gestion et de comptabilité ainsi que l'interface application-réseau requis. Ces inconvénients majeurs ont entravé le déploiement d'Intserv dans l'Internet.

Garanties statistiques : approche DiffServ

Dans l'approche par priorisation, le trafic est classifié et les ressources sont partagées selon des critères de gestion de la bande passante. Les classifications attribuent un traitement préférentiel aux applications qui ont des demandes plus exigeantes. On a une signalisation implicite qui s'opère dans le plan donnée (identifiant de classe dans un champ d'entête). Pour garantir un niveau de service requis, généralement on réserve plus de ressources car il est difficile de caractériser précisément le trafic à l'avance. Les mécanismes de priorisation, dans lesquels les paquets sont labellisés suivant leur priorité et sont traités différemment au niveau des routeurs offrent des solutions à priori plus légères et des garanties de qualité de service non plus absolues au niveau flux mais statistiques. Un réseau qui n'offre pas de garanties de service strictes, est à priori moins coûteux à mettre en oeuvre.

Ainsi, pour contourner les faiblesses des propositions faites par le groupe IntServ, en 1998, un nouveau groupe de l'IETF nommé Differentiated Services Group, **DiffServ** a suggéré de mener des investigations dans cette direction : au lieu se concentrer sur les flux individuels pourquoi ne pas gérer des agrégats de trafic (large ensemble de flux ayant les mêmes besoins de service) et discriminer les paquets en fonction de leur priorité (**precedence**). Cette idée a conduit au concept de services différenciés - à l'opposé des services intégrés - qui ont l'avantage de pouvoir être plus facilement implémentés même dans les réseaux existants [142] et [19]. Diffserv se présente donc comme architecture plus extensible et plus facile à gérer. Le concept de services différenciés est basé principalement sur un schéma orienté **paquet** et que, dans le réseau, la notion de flux d'utilisateur final n'existe pas. Un autre objectif majeur et initial de la différenciation de services est de permettre de facturer différemment les services d'Internet. Plusieurs travaux ont été précurseurs de l'architecture DiffServ et des modèles de services standards. Une revue de l'état de l'art chronologique est

donnée dans [225].

Un domaine DiffServ doit avoir des frontières bien définies. Chaque domaine effectue ses propres marquages et ses vérifications. Diffserv repousse les fonctions complexes de conditionnement et de classification du trafic aux frontières du domaine. La figure 4.2ci-dessous illustre l'architecture d'un domaine DiffServ qui distingue clairement les routeurs à l'intérieur du domaine des routeurs de sa bordure.

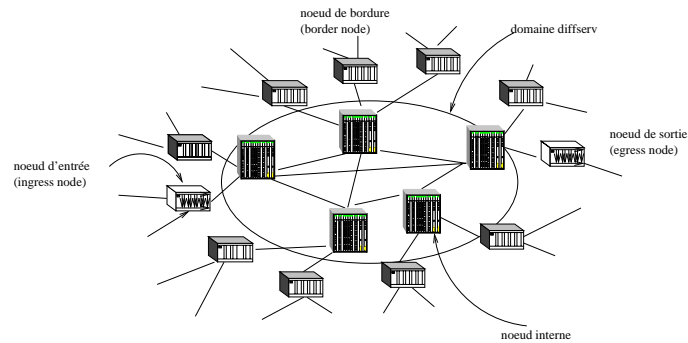


FIG. 4.2 – Architecture DiffServ

Le modèle DiffServ repose sur la classification et le marquage des paquets IP. La définition du champ type de service concerne la sémantique des comportements plutôt que les mécanismes spécifiques d'implémentation. Dans la rfc [142], le champ TOS et le champ Traffic class de Ipv6 [49] sont redéfinis et intitulé **DSFIELD**. Le champ TOS avait été initialement spécifié dans la rfc 791, mais son utilisation n'avait jamais été très importante, exceptée l'exploitation du champ IP precedence dans les routeurs CISCO. La figure 4.3 donne la structure de ce champ TOS et sa réutilisation pour la marquage DiffServ (DiffServ Code Point : DSCP). Nous donnons dans la figure 4.3 et tables 4.2 et ??, l'évolution de la signification de ce champ dans IP puis dans DiffServ.

Precedence(3)	Type of Service(4)	MBZ
DSCP(6)		CU(2)

FIG. 4.3 – Définition du champ TOS (A) et du Définition du DSCP (B)

Valeur du TOS	Interprétation
1000	Minimiser le délai
0100	Maximiser le débit
0010	Maximiser la fiabilité
0001	Minimiser le coût
0000	Normal

TAB. 4.2 – Utilisations du champ TOS

On note que la rfc2474 autorise le remarquage du DSCP d'un domaine à l'autre, ce qui n'est pas sans poser de problèmes pour l'obtention d'un service de bout en bout.

Le comportement sur le chemin PHB (per hop behavior) inclut le traitement différentiel d'un paquet individuel implémenté à l'aide d'une discipline de gestion de file d'attente et/ou d'un discipline de service de file (mécanisme d'ordonnancement) L'architecture DiffServ tout comme l'architecture Intserv définit un cadre

Modèle	IntServ	DiffServ absolu
Type de différenciation	garanties absolues	garanties statistiques absolues
Granularité de différenciation	microflux	agrégats
Etats	par flux	par agrégats
Base de classification	Plusieurs champs d'entête	DS field
Signalisation	explicite	explicite en bordure/implicite dans le coeur
Réservation	Requise(RSVP)	Semi-statique ou dynamique (BB)
Types Routeurs	identiques	bordure/coeur
Contrôle d'admission	requis	requis
Facturation	requis	requis
Coordination	de bout en bout	locale (per hop)
Extensibilité	limitée par le nombre de flux	limitée par le nombre de classes
Gestion réseau	Similaire réseaux de circuits	Similaire réseaux IP

TAB. 4.3 – Tableau comparatif des approches de QoS IP traditionnelles

de services. Plusieurs services ont été étudiés. Deux ont été standardisés : le service express (EF) et le service assuré (AF).

Expedited Forwarding Le concept d'Expedited Forwarding (EF) [119] est de créer un sous-réseau à faible latence où les paquets ne sont jamais (ou très peu) détruits. Les conditions d'acheminement des paquets qu'offre ce sous-réseau virtuel sont *a priori* favorables aux flux temps-réel puisque le temps de traversée du réseau est diminué. Cependant, les flux élastiques y obtiennent évidemment aussi de meilleures performances puisque le chemin est sans perte ni délai de file d'attente.

Assured Forwarding Le PHB Assured Forwarding (AF) [106], sépare les paquets dans n files qui ont chacune une part de débit déterminé. Au sein d'une file, il existe m sous-classes dont le taux de perte est plus ou moins élevé. Bien que les valeurs de m et n ne soient pas standardisées, on les trouve souvent instanciées à $n = 4$ et $m = 3$. Certains distinguent les services or, argent ou bronze. En effectuant une différenciation sur le taux de perte, ce service n'est pas vraiment destiné à améliorer le temps de traversée du réseau et profite essentiellement aux flux élastiques [222]. Plusieurs études s'intéressent au comportement de TCP sur AF [72] (voir section 5.2.3).

Le service premium, implémenté à l'aide du PHB "expedited forwarded" a été le premier déployé dans les réseaux expérimentaux. La première implémentation à large échelle a été le Qbone [205]. Une première démonstration du service Premium sur Abilène a été réalisée en février 1999. Un constat établi à la fin de l'année 2001 [14] fait état de l'ampleur des difficultés de déploiement de Premium Service dans le Qbone. Au niveau européen, le projet SEQUIN ainsi qu'un certain nombre de travaux au niveau de TF-NGN ont mené au test et au déploiement de Premium Service sur des réseaux expérimentaux. Dans VTHD, la différenciation de service est aussi en cours d'évaluation. Nous sommes en train de mettre au point un plan d'expérimentations au sein du réseau européen GEANT (voir section 6.5.1) en collaboration avec les réseaux nationaux (NRNs), le projet SEQUIN et le projet DataGRID. Aucun test en Europe n'est aujourd'hui réellement probant et le déploiement de Premium Service n'est pas encore très large. Ceci devrait cependant évoluer rapidement.

4.2.3 Modèles alternatifs et Balanced Forwarding

DiffServ a été proposé pour résoudre le problème d'extensibilité de RSVP/IntServ mais en a introduit de nouveaux.

- le besoin d'une gestion de la bande passante au niveau global de l'interconnexion réseau (Bandwidth Broker)
- la difficulté d'obtenir une QoS par flux dans un contexte d'agrégation [102]

- un coût supplémentaire est induit par le contrôle d'admission et l'authentification requis aussi bien par RSVP/IntServ au niveau micro-flux que par DiffServ au niveau agrégat.

Dans le plan contrôle, DiffServ requiert du contrôle d'admission, réalisé par les bandwidth broker centralisés ou répartis. Dans le premier cas, une signalisation explicite est nécessaire, dans le deuxième, les BB répartis sont capables de prendre des décisions à partir de bases de données locales qui doivent être maintenues cohérentes par ailleurs. Les traitements à effectuer au plan données sont par contre très allégés et donc beaucoup plus extensibles. En fait le modèle DiffServ n'exprime pas explicitement ces besoins en terme de fonctionnalité car DS ne définit pas de services, mais tout service définit selon le modèle DS, et en particulier ceux qui ont été standardisés par l'IETF nécessitent un contrôle d'accès.

Modèle	DiffServ relatif	Internet
Type de différenciation	Garanties relatives	Pas de garanties
Granularité de différenciation	agrégats	tous flux
Etats	par agrégats	sans état
Base de classification	DS field	aucune
Signalisation	implicite dans le coeur	aucune
Réservation	non	non
Routeurs	identiques	identiques
Contrôle d'admission	non requis	non
Facturation	requis ou non	non
Coordination	locale (perhop)	pas de coordination
Extensibilité	limitée par le nombre de classes	non limitée
Gestion réseau	Similaire à réseaux IP	Normale

TAB. 4.4 – Tableau comparatif des approches de QoS IP relatives et de IP Best Effort

Pour faire face à ces handicaps, certains auteurs ont proposé une différenciation de service sans garanties absolues. Les solutions de Différenciation de Service relative ou de Best Effort amélioré proposent des services dits *non évolués*. L'Internet 2 suggère aujourd'hui de concentrer les efforts sur ce type de modèles et d'architectures au dépend des modèles DiffServ traditionnels [14] qui n'ont pas réellement été des succès. Les travaux théoriques de Martin May [135] laissent déjà percevoir dès 1999 les limites du modèle DiffServ absolu. Depuis 2 ans, nous travaillons avec Benjamin Gaidioz sur les approches alternatives qui offrent, selon nous, de grands intérêts dans le contexte Internet. Nous avons ainsi proposé le modèle **BF**, inspiré du modèle, puis nous nous sommes intéressés au modèle de différenciation proportionnelle (voir section 3.4.1. Les tables ?? et ?? donnent respectivement les propriétés des architectures de QoS IP traditionnelles et celles des approches relatives comparées au modèle best effort classique. La table 4.5 liste les approches alternatives proposées aux modèles DiffServ absolu standard.

	nom	laboratoire	nombre de classes	mécanismes
Alternative Best Effort	ABE	EPFL	2	dédié
Best Effort Differentiated Services	BEDS	Nortel	2	WFQ+RED
Scavenger Service	QBSS	Internet2	1	CBQ
Balanced Forwarding	BF	RESAM	2	dédié
Proportional Differentiated Services	PDS		N	BPR ou WTP
Equivalent Differentiated Services	EDS	RESAM	N	BPR+WTP

TAB. 4.5 – Propositions de services Best Effort Amélioré

Asymmetric Best Effort

Les premiers travaux dans le domaine du Best Effort Amélioré sont issus de l'équipe de Jean-Yves Le Boudec à l'EPFL. Le modèle Asymmetric Best Effort, ABE, initialement baptisé Alternative Best Effort [115], [114]

propose deux classes de services, l'une avec un haut débit et l'autre avec un faible délai. L'idée est de proposer un service faible latence dans un réseau best-effort. Contrairement à EF dont c'est aussi un objectif, la classe faible latence (la classe « verte ») n'est pas strictement plus intéressante que l'autre (la classe « bleue »). Dans un routeur, chaque classe gagne sur un terrain et perd sur un autre : les paquets verts (faible latence) admis dans un routeur sont assurés d'y rester moins longtemps que les paquets bleus. En revanche, les critères d'admission pour les verts sont plus sévères et ils subissent un taux de perte plus élevé que les bleus. Le paramétrage du système (différenciation délai/taux de perte) est tel qu'il garantit que les flux bleus ne voient pas leurs performances diminuer par rapport au best-effort classique malgré l'existence du service vert. ABE est basé sur l'hypothèse que tous les trafics sont TCP friendly, c'est à dire qu'ils utilisent le mécanisme de congestion de TCP.

BF : Balanced forwarding

Dans [94] nous avons présenté un nouveau modèle de service, inspiré de ABE. Ce modèle propose deux classes de services mais ne fait pas d'hypothèse sur le comportement des flux de bout en bout et en particulier ne suppose pas qu'ils sont tous TCP-coopérants. La classe 1 proposée est la classe *plus faible délai* et la classe 2 est la classe *plus faible taux de perte* dans lequel en cas de congestion, le routeur dégrade différemment les performances des deux classes selon le type de trafic qu'elles transportent. Dans le routeur, les statistiques de chaque classe sont telles qu'on conserve l'égalité entre l_1/l_2 et d_2/d_1 où l_i et d_i sont respectivement le taux de perte et le délai de file de la classe i . Lorsque le réseau est peu chargé, la valeur de ces rapports est proche de 1 et les classes ont des performances comparables. Lorsque la charge augmente, la valeur des rapports augmente, c'est-à-dire que le taux de perte de la classe 1 augmente plus que celui de la classe 2 alors que c'est l'inverse pour les délais de file d'attente (à cause de l'asymétrie de l'égalité).

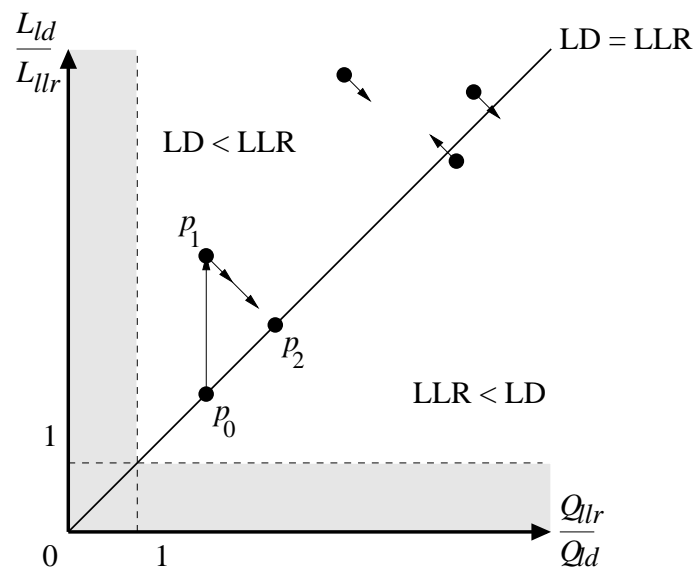


FIG. 4.4 – Evolution du délai et du taux de perte pour maintenir l'équité

BF est inspiré de ABE mais contrairement à ABE qui est basée sur une propriété globale des flux (TCP-friendliness), BF ne s'appuie que sur des critères locaux, ce qui en facilite le déploiement. Le caractère dynamique de la différenciation dans BF donne au réseau une certaine réactivité puisqu'il prend lui même la décision de faire diverger les performances relatives des classes selon la charge. Nous avons proposé un ordonnanceur pour ce modèle (figure 4.5). Cet ordonnanceur maintient l'historique des pertes et des délais dans deux files circulaires indépendantes.

L'implémentation sous Linux du modèle BF nous a permis de mener des expérimentations sur un réseau local (figure 4.7). Nous avons validé les propriétés locales et établi la différenciation des performances (figure

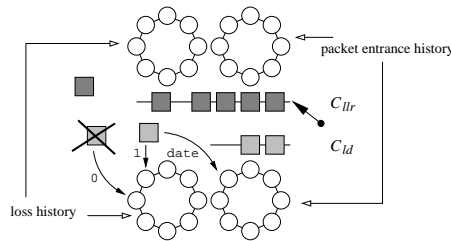


FIG. 4.5 – Ordonnanceur de BF

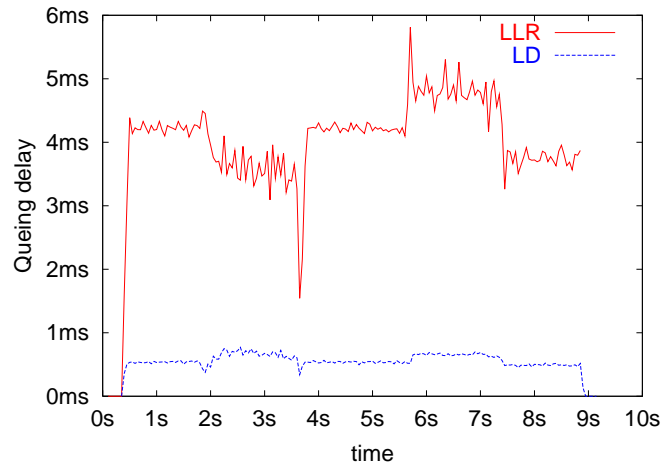


FIG. 4.6 – Mise en évidence de la différenciation en délai

4.6). Nous n'avons pas réussi à démontrer la pertinence du modèle asymétrique $l_1/l_2 = d_2/d_1$ pour les flux de bout en bout. Ce modèle signifie intuitivement : le taux de perte est au flux sensible aux pertes, ce que le délai est aux flux sensibles au délai. C'est une assertion qui, si elle permet de faire un premier pas vers le concept d'équité différenciée, n'est pas tout à fait satisfaisante. Certains auteurs ont montré par exemple que les courbes d'utilité du taux de pertes et du délai divergeaient [193]. Même si elles sont toutes deux croissantes, celle du taux de perte est linéairement croissante, tandis que celle du délai croît de manière non monotone. D'autre part, le modèle analytique de calcul du débit d'une connexion TCP [148] avance que le taux de pertes a une influence d'ordre 2 par rapport au délai. Nous avons donc initié une étude approfondie de la problématique de bout en bout de la différenciation de services. Et nous avons fait évoluer le modèle BF vers le modèle EDS, proposant plusieurs classes de services différenciés équivalents. Nous présentons ces travaux au chapitre suivant.

BEDS : Best Effort Differentiated Services

Plus récemment, le modèle BEDS, Best Effort Differentiated Services a été proposé. Il vise aussi les mêmes objectifs que ABE : offrir un service *loss conservative* avec une probabilité de perte plus faible mais un délai plus grand que le service *delay conservative*. Contrairement à l'approche ABE qui requiert l'implémentation d'un nouvel algorithme de gestion de file et d'ordonnancement, pour réaliser l'architecture BEDS, les auteurs cherchent à utiliser les mécanismes déjà existants dans les routeurs. BEDS peut être mis en oeuvre à partir d'une combinaison de l'ordonnanceur Weight Fair Queuing (WFQ) et du mécanisme de gestion active de file Random Early Detection (RED). Ainsi le déploiement du modèle dans Internet est simplifié. Les auteurs montrent aussi qu'il peut être incrémental. La complexité des mécanismes est analogue à celle des implémentations connues de RED et de WFQ.

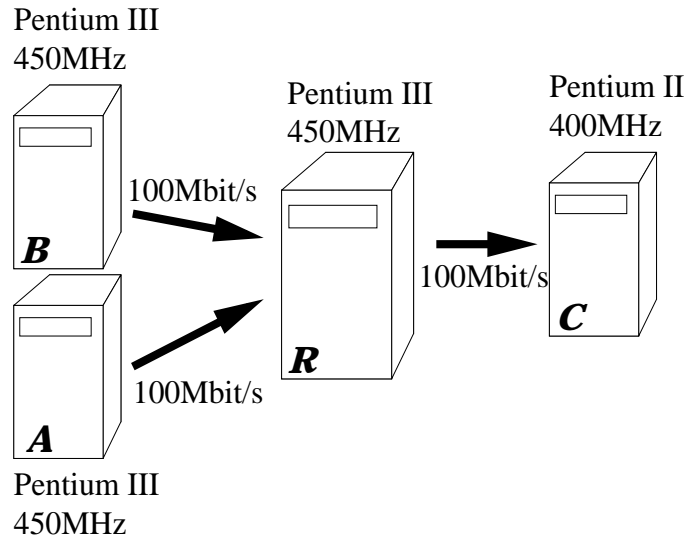


FIG. 4.7 – Réseau mis en place pour les expérimentations

QBSS : Qbone Scavenger Service

A l'extrémité du spectre, le service **scavenger** se veut *moins bien* quequ'un service Best Effort. Qbone Scavenger Service [110] a été proposée en 2001 comme une classe additionnelle au service Best Effort. Une faible part de la capacité du réseau est allouée d'une manière non rigide à ce service lorsque la capacité Best Effort classique est sous-utilisée voire non utilisée. Ainsi, pendant ces phases, les flux scavenger peuvent consommer toute la capacité non utilisée. Comme le service Best Effort est difficile à définir en termes positifs, scavenger est encore plus délicat à spécifier. Au lieu de décrire le service offert, les auteurs préfèrent en donner une description opérationnelle en spécifiant la configuration des routeurs. Les applications qui sont relativement tolérantes aux pertes, aux délais et à la gigue peuvent marquer leur trafic et recevoir un service potentiellement dégradé par rapport au service Best Effort par défaut. Marquer le trafic ainsi peut être comparé à attribuer une valeur *nice* positive à un processus UNIX. Les routeurs participants au domaine QBSS peuvent traiter le trafic scavenger (DSCP 001000) de manière indépendante du trafic Best effort et l'acheminer de manière moins rapide que le trafic BE (DSCP 000000) ou bien le traiter comme du trafic Best Effort. Ainsi, tout routeur qui ne s'occupe pas du DSCP et ne le change pas satisfait les condition de QBSS, ce qui permet un déploiement incrémental.

Nous avons implémenté le service QBSS sous Linux en configurant dans nos routeurs les mécanismes CBQ et WRR. Nous avons évalué les performances obtenues par des flux TCP et UDP marqués en scavenger sur un réseau expérimental local. La figure 4.8 ci-dessous met en évidence les résultats que nous avons obtenus et qui correspondent bien aux spécifications du service. Dans la première partie de la courbe, le flux QBSS est seul et consomme l'intégralité de la capacité. A l'instant 37 :10 un flux TCP BE arrive. Le flux QBSS tombe à 1. Un des intérêts majeurs de QBSS est selon nous, qu'un flux scavenger étant par essence **amical** avec les flux best-effort, le sera automatiquement avec tous les flux TCP et n'entravera pas, par conséquent, le mécanisme de contrôle de congestion d'Internet. Ceci, nous le verrons dans le dernier chapitre de ce document, permet d'envisager de faire évoluer la couche transport de manière beaucoup plus libre.

4.3 Les techniques adaptatives

Pendant que s'élaboraient ces architectures de qualité de service au niveau réseau, du côté des applications multimédia temps-réel, des recherches ont été menées pour trouver des mécanismes capables d'absorber les fluctuations de performances de l'interconnexion. En l'absence de garanties de service strictes en terme de délai ou de taux de perte à l'intérieur du réseau, des algorithmes adaptatifs ont donc été développés aux

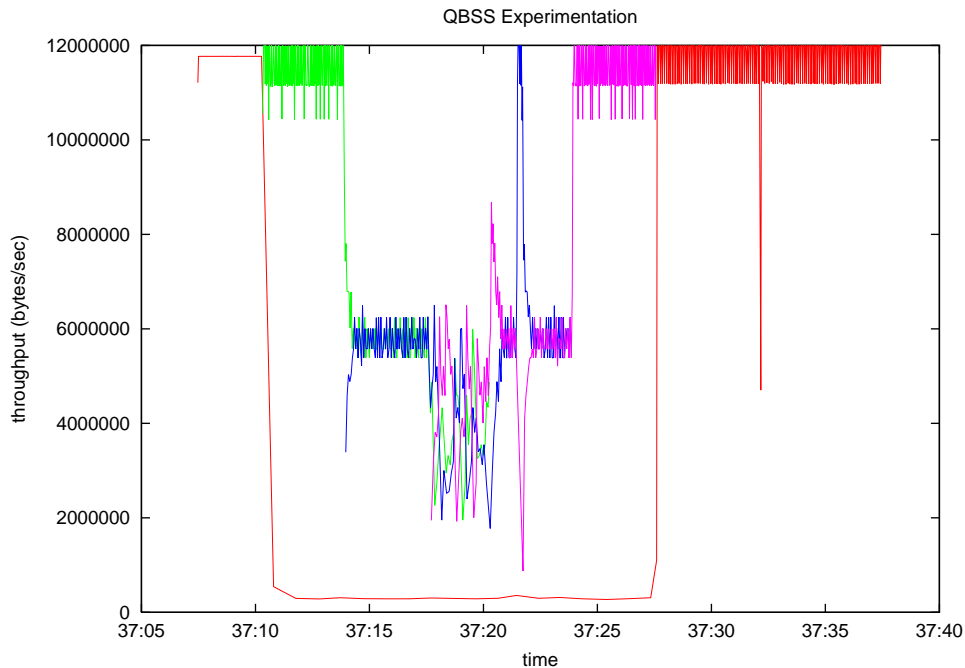


FIG. 4.8 – Comportement des flux TCP-QBSS en présence de trafic Best Effort (A), en l’absence de trafic Best Effort (B)

extrémités.

Traditionnellement c’est la couche transport, première couche de bout en bout qui réalise l’adaptation (définition de la couche transport OSI). Sa complexité dépend de la qualité de service offerte par le réseau sous-jacent. Ainsi dans OSI, 5 classes de couches transport ont été définies. TP4 étant la plus complète et la plus complexe. Le modèle ATM, offre aussi plusieurs services d’adaptation à l’application. Dans la couche d’extrémité AAL, application adaptation layer, les paquets d’information sont mis en forme et les informations utiles pour la recombinaison de l’information émise chez le récepteur sont insérées. ATM propose 5 couches d’adaptation, de AAL1 à AAL5, correspondant au différentes classes de services offertes au niveau ATM. Mais ces couches n’implémentent pas d’algorithme spécifique d’adaptation aux variations de performances du réseau puisque les canaux offerts par la couche réseau ATM ont une qualité prévisible.

Dans le modèle Internet, deux principaux protocoles de transport sont disponibles : un protocole rudimentaire UDP qui ne fait que du multiplexage de flux et un protocole fiable très sophistiqué, TCP, qui réalise l’adaptation aux pertes de paquets et le contrôle de congestion. Comme les applications multimédia temps-réel ont plus d’exigence de délai que de fiabilité, TCP ne convient pas et ce sont les applications, qui, pour des raisons de performance, sont elles-mêmes en charge de l’adaptation. Dans cette section je présente les principes de ces techniques d’adaptatives et ainsi que les travaux que j’ai mené au niveau de l’extraction de l’algorithme d’adaptation d’une application, afin de réaliser un module d’adaptation indépendant et générique.

4.3.1 Classification des techniques adaptatives

Il existe différents mécanismes d’adaptation et de contrôle de la QoS. Les approches peuvent être classifiées selon leur type. En effet, comme le réseau peut introduire des variations de délai, de pertes de paquets et de débit, trois types d’adaptation sont possibles : l’adaptation au délai, l’adaptation au débit, l’adaptation aux pertes. Par ailleurs, l’adaptation peut se faire par la source ou par le récepteur. Les mécanismes basés sur l’adaptation par la source ont été les premiers développés, nous les synthétisons ci-dessous. Nous étudierons,

dans une section suivante, les problèmes relatifs au multipoint et les techniques d'adaptation par le récepteur.

Adaptation au délai

L'objectif d'un tel algorithme est d'effectuer un contrôle efficace de la variation de délai. Il s'agit d'ajuster dynamiquement le délai de buffering artificiel afin de compenser la variation de gigue. En effet, la taille du buffer de compensation de délai au niveau du récepteur a des conséquences sur la qualité de réception : si le buffer est trop petit, il peut déborder, s'il est trop important, il n'y aura jamais de perte, ce qui accroît la qualité mais augmente par là même le délai de réception. La taille optimale dépend de la gigue induite par le réseau. Or cette gigue est inconnue a priori. Dans [136] McCanne détaille l'algorithme utilisé dans *vat*. Des travaux plus récents [?] ont raffiné cet algorithme de détermination de la borne maximale du délai de rejeu, encore appelé *modèle du point de play-back*. Cette technique peut être employée dans les applications de streaming ou pour faire de la synchronisation intermédia. On peut considérer que l'adaptation au délai est aujourd'hui un problème bien compris et maîtrisé.

Adaptation au débit

Cette adaptation consiste à contrôler le débit d'émission de façon à rendre les besoins en bande passante compatibles avec la capacité disponible du réseau. L'objectif est de minimiser la congestion du réseau et le nombre de paquets perdus. Cette adaptation n'évite pas les pertes. Dans [25] on trouve une description claire des mécanismes d'adaptation du taux de transmission possibles. Différents problèmes doivent être résolus : comment adapter la demande en débit en fonction des besoins de l'application, comment ajuster le débit d'émission, comment caractériser l'état du réseau, comment sélectionner et acheminer l'information de retour (feedback). L'application peut par exemple définir un critère de performance critique qui devra être privilégié dans toute décision d'adaptation. Par exemple, pour une transmission vidéo, la précision du rendu peut être plus importante que la perception du mouvement. Une source peut jouer sur plusieurs paramètres afin de modifier son débit d'émission. Pour la vidéo on peut augmenter le seuil de détection du mouvement, diminuer le taux d'images ou la résolution. Selon le type d'application, différents paramètres sont contrôlables. Pour la vidéo, on peut réguler la résolution spatiale, la quantification ou le taux d'images. Pour un flux audio, on agit sur l'encodage. On peut aussi choisir d'assurer une garantie minimum : pour une vidéo de loisir un taux d'images à 15 images/s sera une borne inférieure, pour l'audio, un débit de 32kb/s sera un minimum. Si une telle garantie ne peut être obtenue, l'utilisateur peut décider de différer son appel. La plupart des algorithmes d'adaptation sont basés sur un feedback relatif aux paquets perdus. Il est aussi possible d'utiliser la variation de délai (gigue), de calculer un délai de bout en bout à partir d'estampilles. Cette technique n'est cependant valable que si le délai intersite est suffisamment grand. C'est au niveau de cette adaptation au débit que l'on peut escompter le plus de flexibilité.

Adaptation aux pertes de paquets

Cette adaptation vise à contrôler les pertes de paquets afin de minimiser leur impact au niveau de l'application réceptrice. La perte de paquet peut engendrer une dégradation de la qualité du signal et peut être contrôlée de plusieurs manières du côté des applications :

- Requête de répétition automatique lorsqu'un paquet a été perdu (ARQ). Cette technique optimise la fiabilité de la transmission mais augmente la variabilité des délais et est incompatible avec des performances interactives. C'est la technique utilisée dans TCP.
- Correction automatique d'erreur (FEC). On améliore la transmission de l'information en ajoutant de la redondance qui permet de recomposer le signal de départ à partir d'un sous-ensemble de paquets reçus. Cette technique est plus efficace si les pertes sont dispersées.
- Codage robuste. Ce codage modifie la représentation du signal de telle sorte que le flux est moins sensible aux pertes de paquets, certainement au prix d'une baisse du taux de compression, c'est à dire une augmentation du débit de transmission.

Ces différentes techniques d'adaptation ont été implémentées dans les protocoles et applications existantes. Nous étudions en particulier l'adaptation dans TCP

4.3.2 Le transport TCP

TCP est un protocole de bout de bout dont la vocation est de compenser les faiblesses du protocole IP et de fournir aux applications un transport fiable, c'est à dire avec un taux d'erreur et de perte résiduel nul. TCP s'adapte à la fois au débit et au taux de perte, mais ne fait aucun contrôle sur le délai. Il utilise une technique de type ARQ. TCP effectue le contrôle d'erreur, de perte, de séquençement par détection et signalisation et la compensation de perte par retransmission. Par ailleurs, TCP effectue des fonctions de multiplexage (par port) et de contrôle de flux (par fenêtre glissante à anticipation) ainsi que du contrôle de congestion qui est en principe une fonction " réseau ", mais que IP n'assure pas pour les raisons de simplicité que nous avons explicité auparavant. TCP s'avère être un protocole extrêmement sophistiqué et dont la dynamique est complexe. TCP a été donc conçu pour s'adapter à un type de couche réseau (IP), pour des réseaux locaux et des réseaux longue distance à faible débit et pour des classes d'application bien spécifiques (transfert fiable de fichiers, messagerie textuelle, accès distant). Dans [39] Clark explique la genèse de TCP par séparation du protocole NCP en deux sous-couche IP et TCP, puis l'apparition du protocole UDP pour servir d'autres besoins de transport.

Avec l'évolution des applications et des infrastructures, TCP, malgré un grand nombre d'optimisations que nous détaillerons au chapitre 5, a atteint aujourd'hui ses limites. La couche transport est donc en plein ébranlement pour suivre l'évolution des protocoles d'application, des protocoles réseau et de la technologie des supports. Nous pouvons considérer trois grands axes d'évolution :

- le transport des applications temps-réel
- le transport sur un support très haut débit
- le transport sur support aérien (non filaire)
- le transport sur couche IP à services différenciés
- le transport multicast

Dans le chapitre 4 je traite des aspects de TCP sur une couche réseau offrant des services différenciés . Dans le chapitre 5 nous étudierons TCP pour le haut débit. Nous développons ci-dessous la problématique du transport et de l'adaptation pour les applications temps-réel. Dans ce document nous ne traiterons pas directement de la problématique de transport multicast dont l'étude recouvre cependant souvent des aspects similaires à ceux que nous étudions en unicast, ni de la problématique du transport sur les réseaux non filaires.

4.3.3 Transport des applications temps-réel

Pour le transport des flux multimédia, TCP ne convient pas car il fait de la retransmission pour corriger les erreurs et les pertes. Les retransmission entraînent des délais qui peuvent être prohibitifs pour la régénération du signal à l'arrivée. C'est donc le protocole UDP qui a été choisi comme protocole de transport des paquets multimédia temps-réel sur Internet. Mais le protocole UDP n'offre aucun mécanisme de contrôle de délai, de débit ou de perte. UDP ne coopère pas non plus avec les autres flux pour la gestion des congestions dans le réseau. Le protocole RTP, real-time transport protocol [12], protocole de bout en bout défini par l'IETF et porté par UDP a été défini pour le transport des flux continus et temps-réel. C'est un protocole non orienté connexion, et qui n'offre aucune garantie de fiabilité. Il apporte uniquement la capacité de distinguer les différents types de média (sous-module) et de conserver les traces de différentes statistiques décrivant la qualité de la transmission vue par les autres usagers. Ce protocole est composé de deux parties : une partie dédiée au transfert de données (RTP), l'autre au contrôle (RTCP). Le protocole de contrôle RTCP (real time control protocol) transporte des paquets contenant l'information nécessaire au supervision de QoS et au contrôle de congestion. Les applications émettrices envoient un rapport d'émission aux récepteurs qui peuvent ainsi estimer le taux de données actuel, des informations pour la synchronisation intermédia. A l'inverse, les récepteurs envoient des rapports de réception aux membres de la session, ce qui permet à l'émetteur de récupérer des informations sur les numéros de séquence reçus, le taux de perte, une mesure de la variance du délai d'arrivée, une estimation du délai d'aller-retour, etc.

Ni UDP, ni RTP n'implémentent de mécanisme d'adaptation comme TCP. Il est donc largement admis que ce sont les applications qui doivent s'adapter à un certain intervalle $[QoS_{min}, QoS_{max}]$ pour fournir une réaction gracieuse à la disponibilité dynamique des ressources . Un nombre importants de travaux ont été menés dans le cadre du Mbone [123]. Les trois types d'adaptation, au délai, aux pertes et au débit sont mises

en oeuvre, et ce souvent de manière combinée [136]. Conçues pour Internet, ces applications ne font aucune hypothèse sur la qualité de service offerte par le réseau sous-jacent. Il est à noter qu'une telle approche n'offre pas de garantie de QoS, mais vise seulement à améliorer la qualité perçue.

Différents objectifs doivent être considérés lorsque l'on conçoit un schéma de contrôle adaptatif. Le schéma doit

- résulter en un taux de perte de faible
- atteindre une utilisation globale de la bande passante élevée
- conserver la distribution équitable de la bande passante entre les connexions.
- s'étendre aux larges groupes multicast.

Réduire le taux d'erreur résultant

Pour la compensation des pertes de paquets, la technique la plus utilisée dans les applications audio-vidéo adaptative est le codage robuste. C'est le cas des outils de diffusion vidéo *nv*, *vic* [138] ou *IVS*. Comme les algorithmes de compression vidéo MPEG ou H261 sont basés sur un codage intra et inter image, s'il y a des pertes de paquets la dégradation est importante jusqu'à la réception d'une image intracodée (intervalle périodique). Le codage robuste est issu du concept de la théorie formelle de la communication le " Joint source/channel coding " (JSCC). JSCC établit que l'on peut souvent obtenir de meilleures performances du système en combinant la conception du codage de contrôle d'erreur et de la compression plutôt que de traiter ces deux problèmes de manière indépendante [?]. Pour faire un contrôle d'erreur adaptatif, trois méthodes permettent de contourner le problème des erreurs :

- réduire l'intervalle entre blocs intracodés. On tend vers une seule image intracodée et élimination des trames intercodées. C'est la technique utilisée dans le codage M-JPEG. Cette méthode est gourmande en bande passante.
- intra-coder les images et ne transmettre que les blocs d'une image dont les changements sont au dessus d'un certain seuil. C'est ce qui est mis en oeuvre dans *vic* et *nv*. Les besoins en bande passante augmentent de 30% par rapport à un codage inter-frame.
- utiliser un mécanisme de Forward Erreur Correction comme dans *IVS*.

Bolot et Turlitti [23] ont proposé un algorithme basé sur un mécanisme de contrôle de débit et un mécanisme de compensation d'erreur (FEC : forward Error Correction) pour minimiser l'impact des erreurs sur les applications audio/vidéo. Le mécanisme de base de contrôle des pertes par Forward Erreur Correction est basé sur une opération XOR. A chaque séquence de k paquets, on émet un paquet $k + 1$ calculé par une opération XOR sur les k paquets précédents. On peut ainsi réparer une erreur dans un message de k paquets. Le débit requis augmente de $1/k$ paquets et la latence s'accroît. L'originalité de l'algorithme de Bolot et Turlitti est d'introduire une FEC variable et fonction du comportement du réseau en terme de perte de paquets.

Conserver l'équité de distribution de bande passante

Le déploiement massif d'applications temps-réel émettant des flux RTP/UDP non contrôlés, peut être très préjudiciable aux flux TCP qui diminuent leur débit d'émission en présence de congestion. Les pertes engendrées par des émissions importantes de flux UDP peuvent entraîner la famine des flux TCP et une forte iniquité du partage des ressources globales entre les flux TCP et les flux UDP. Plusieurs approches ont été suggérées pour ajuster le comportement des flux UDP et le rendre similaire à celui des flux TCP [132]. Dans [118] Jacobson présente un schéma basé sur le mécanisme de contrôle de congestion de TCP sans retransmission des paquets perdus. Ainsi, l'émetteur maintient une fenêtre qui avance en fonction d'acquittements émis par le récepteur. Cette approche est limitée à l'émission unicast.

Floyd a proposé un modèle qui estime le débit utile (throughput) d'une connexion TCP sous des conditions de délai et de taux d'erreur connus. Madhavi [131] and Turlitti [24] ont proposé un schéma de contrôle de bout en bout dans lequel les systèmes d'extrémités mesurent les pertes et les délais dans le réseau et restreignent leur taux de transmission selon la valeur estimée de l'équation de débit ci-dessus. Floyd définit aussi des mécanismes de routeur qui identifient les flux utilisant plus de bande passante que les flux de l'équation. L'objectif est d'éliminer en premier les flux n'étant pas coopératifs, c'est à dire non *TCP-friendly*.

Ainsi plusieurs propositions se sont basées sur le modèle de TCP dépendant de la taille maximale des paquets, du RTT et du taux de perte moyens. Baser l'adaptation sur ce modèle a des limites car l'algorithme ne peut se baser que sur des valeurs courantes et non des valeurs moyennes et on peut obtenir ainsi un comportement très oscillatoire et une qualité perçue très désagréable.

Problématique du multipoint et adaptation par le récepteur

Les algorithmes présentés s'appuient sur un contrôle par la source. Cette solution est effective dans le cas des transmissions unicast ou en faible groupe multicast. Pour les groupes plus larges, dans un environnement hétérogène, elles engendrent un nivellement par le bas dans lequel, le récepteur le moins bien connecté détermine la qualité d'un nombre beaucoup plus important de récepteurs bien reliés. Ainsi un algorithme basé sur le récepteur *ayant le plus fort taux de perte* évite toute congestion mais pénalise tous les récepteurs. L'algorithme peut être plus sophistiqué et se baser sur un rapport proportionnel de récepteurs **non chargés, chargés et congestionnés**. Dans l'approche adaptative proposée dans IVS avec adaptation à la source, lorsque le nombre de paquets perdus atteint un certain seuil, le récepteur informe l'émetteur. L'émetteur choisit de décroître le débit si une certaine proportion de récepteurs souffre d'un taux de perte excessif.

Pour résoudre ce problème de variabilité des conditions de réception, plusieurs auteurs [211], [137] ont proposé une distribution hiérarchique des données en une couche de base formée par les informations offrant la qualité minimum et des couches additionnelles pour l'amélioration de la qualité. Dans ce mode de transmission, ce sont les récepteurs qui s'adaptent en choisissant dynamiquement de s'abonner ou de quitter une couche, c'est à dire un groupe multicast. Si la transmission hiérarchique permet d'améliorer la réception, elle a un impact sur le délai. En effet, les paquets des différentes couches, n'empruntant pas nécessairement le même chemin doivent être resynchronisés à l'arrivée, ce qui peut considérablement complexifier le système global. Par ailleurs, les join/leave dynamiques peuvent entraîner un problème d'oscillation.

Certains auteurs se sont intéressés à des solutions hybrides alliant l'adaptation par la source et par les récepteurs. Dans Vosaic [78] par exemple, l'adaptation est effectuée au niveau du récepteur, si un trame arrive en retard, on ne l'affiche pas. La régulation se fait sur le taux réception par suppression images. Les images en avance sont stockées afin de compenser la gigue de présentation.

4.3.4 Netstre@mer : vers une adaptation générique

Pour des raisons de performance, les applications adaptatives intègrent des fonctionnalités qui relèvent de la couche transport. Ainsi les mécanismes proposés ne sont pas facilement réutilisables. Dans les travaux que j'ai conduit avec Julien Laganier et Jean-François Fleury pour la conception du logiciel Netstre@mer [79], nous avons étudié comment extraire un algorithme d'adaptation d'une application de diffusion vidéo afin d'en faire un mécanisme générique et réutilisable .

Le problème du traitement intégré

Au début des années 90, les protocoles en couches constituaient un goulet d'étranglement pour les applications temps-réel à fortes contraintes temporelles. Les deux problèmes majeurs identifiés étaient les accès mémoire (copie multiples) ainsi que les allers-retour de paquets dans le réseau. Pour pallier cette inadéquation des protocoles en couches, un nouveau modèle architectural a été proposé [85]. Ce modèle est basé sur le principe de mise en trame par l'application (Application Level Framing : ALF) et de traitement intégré des couches (Integrated Layer processing : ILP). Les opérations sont découpées en 2 phases : une phase de traitement des données et une phase de contrôle du transfert. La première phase regroupe les opérations de lecture et copie d'octets, de détection d'erreurs, de stockage et de présentation des données à application. La deuxième phase s'occupe du contrôle de flux et de congestion, du traitement des acquittements, de la détection de pertes et du reséquencement. La phase 1 est critique pour les applications multimédia temps-réel, car les volumes à traiter sont très importants. Il faut donc minimiser le nombre de fois où ces données traversent le bus système. Pour la phase 2, il est nécessaire d'effectuer toutes les opérations le plus souvent possible en une seule boucle de traitement pour minimiser les accès mémoire. Le traitement simultané de plusieurs fonctions réseaux indépendantes est donc en contradiction avec traitement en couches. Ce concept

ALF a été mis en oeuvre dans les logiciels multimédia du Mbone, et en particulier dans le milieu des années 90 dans les logiciels vic [138], vat (visual audio tool) et les codec logiciels : H261 [117], nv, JPEG... En réalité le traitement intégré est bénéfique lorsque les ressources sont critiques. Au fur et à mesure des évolutions technologiques le facteur critique des ressources système (mémoire, bus, processeur) est à rééquilibrer avec les performances offertes par les réseaux et les interfaces réseau. Avec les évolutions des codeurs matériels, les principes ALF et ILP semblent apporter aujourd'hui plus de limites au plan de la modularité et de la généralité qu'ils n'amènent de bénéfices. De même le principe du codage robuste basé sur JSCC remet en question les principes de la modularité et de séparation des fonctions de Shannon. Avec les progrès technologiques des supports de transmission et des services, le taux de pertes ira diminuant et le contrôle d'erreur pourra être indépendant de la compression. Les algorithmes de transport n'auront plus nécessairement à être conçus comme des algorithmes de codage de canal.

Objectifs de recherche

L'objectif des développements autour de netstre@mer et de l'adaptation était d'explorer plus en détail les techniques adaptatives dans le but de les réutiliser soit au niveau application, dans un outil flexible tel que CoTools afin d'apporter à l'utilisateur des mécanismes de contrôle dynamique de la qualité de service, soit au niveau réseau afin de déporter l'adaptation de type *la source* au plus près des récepteurs. Plusieurs auteurs travaillent à la problématique de l'adaptation indépendante des applications dans le but de la construction de *middleware* adaptatifs. Avec l'avènement de l'informatique multimédia mobile, le besoin devient en effet de plus en plus pressant. Ces recherches sont très orientées sur les aspects mobilité. Nos travaux se sont basés sur le modèle proposé dans la passerelle de transcodage videogateway [9] qui réutilise des codeurs logiciels écrits en C.

Comme ces travaux se situaient dans la suite du portage de CoTools en Java et le cadre du développement de la plate-forme active Tamanoir écrite en JAVA, nous n'avons étudié que les solutions Java. Notre objectif ici était donc aussi d'examiner les fonctionnalités et les performances de la librairie Java Media Framework (JMF), qui permet de traiter une grande variété de formats multimédias de façon quasiment identique. La majeure partie de cette librairie est basée sur un processeur JMF qui se charge de chaîner les composants nécessaires à la lecture et l'adaptation du flux multimédia. Le schéma ci-dessous présente un exemple du chaînage qui est effectué pour la lecture d'un fichier MPEG dont la piste vidéo est au format MPEG-1 et la piste audio au format MPEG-1 Layer 3.

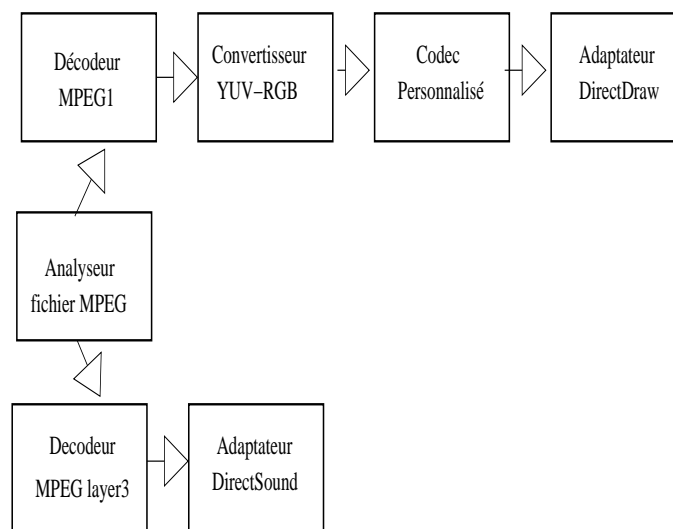


FIG. 4.9 – Principe de traitements des flux vidéo dans JMF

Ne souhaitant pas rivaliser avec les performances des composants matériels, nous avons focalisé nos tra-

voux, non pas sur les aspects performances mais plutôt sur les aspects fonctionnels et architecturaux. Nous nous sommes intéressés au problème du déport de l'adaptation au débit dans le réseau sur le chemin entre une source et plusieurs destinataires. Nous avons limité notre étude au cas des flux vidéo et nous sommes concentrés sur l'adaptation au débit. On se proposait d'adapter le flux multimédia au point de passage (passerelle) entre deux réseaux, en fonction des ressources disponibles sur le réseau situé en aval de la passerelle. Cette adaptation du flux est en fait une dégradation : elle consiste en une réduction du nombre d'images par secondes, une réduction de la taille de l'image, du nombre de couleurs, ainsi qu'une modification de la qualité de compression des images.

Le format d'encodage

Lorsque l'on cherche à extraire l'adaptation de l'application, il faut envisager un format qui permettent un transcodage à la volée aisé. Les opérations de transcodage sont d'autant plus gourmandes en mémoire que le format de compression est inadapté au traitement individuel des images. Nous avons choisi d'utiliser M-JPEG comme format de compression. Le format M-JPEG compresse chaque image d'une séquence vidéo comme une image individuelle, sans tenir compte des autres images, c'est à dire sans profiter de la redondance temporelle du flux, comme les codages MPEG ou H32x. Si le format JPEG s'avère être un mode efficace de compression des images fixes, il n'est pas optimisé pour le codage d'une séquence. M-JPEG est donc plus exigeant en débit que son concurrent MPEG. Cependant ce dernier souffre de deux défauts majeurs : il nécessite de forts calculs pour la détection des similarités visuelles et il rend la manipulation image par image difficile puisque chaque image est liée à ses voisines. MPEG est un donc format approprié pour le stockage et la diffusion mais pas pour la manipulation des images (stockage sur CD-ROM (MPEG1) ou sur DVD (MPEG2)) et il est très sensible aux pertes. La puissance des processeurs actuels ne permet pas de décompresser puis re-compresser du MPEG à la volée en temps réel. Par contre, le format JPEG se prête mieux à une telle manipulation. M-JPEG est par ailleurs insensible à la perte de trames. Cet avantage doublé de la rapidité de codage/décodage compense de loin la bande passante supérieure que nécessite le M-JPEG par rapport au MPEG pour une qualité d'image sensiblement égale.

La transmission de M-JPEG sur RTP a fait l'objet de plusieurs rfc (rfc 2035 puis rfc 2435). En général, la transmission de vidéo sur RTP a fait l'objet de nombreux efforts de standardisation (rfc 2250, rfc 2343 et rfc 3016). Ces efforts de standardisation visent à réduire la bande passante nécessaire tout en conservant plus ou moins bien l'invulnérabilité face aux pertes de paquets en effectuant de la compression des entêtes RTP par exemple.

Régulation des rapports de réception

Dans un contexte multi-utilisateur, les rapports de réception servant au contrôle du flux peuvent engendrer un important trafic car ils sont émis en unicast par chacun des récepteurs en aval de la source. Une régulation est nécessaire pour éviter les implosions de rapports de récepteurs et la saturation des canaux. Il est communément admis qu'une limite supérieure de 10% de rapports (trames RTCP) est un bon compromis. Nous avons écrit un algorithme de régulation générique original basé sur l'émission contrôlée par la source d'une trame SDES (description de la source) qui déclenche l'envoi des rapports RR par les récepteurs. Ainsi, le nombre de rapports reçus entre 2 émissions de trames SDES donne une valeur approchée du nombre de récepteurs. On peut aussi estimer la bande passante relative consommée. Ce système a un avantage supplémentaire : si un noeud de l'arborescence décide de ne plus faire d'adaptation, et se contente de faire relais pour les paquets RTP et SDES, les rapports seront automatiquement remontés à la plus proche source en amont. Ces rapports contiennent les informations classiques (gigue, taux de perte, nombre cumulé de paquets perdus, numéro de séquence le plus élevé reçu).

Algorithme d'adaptation

L'algorithme adaptatif est basé sur le taux de pertes des paquets. Le module de contrôle fournit des statistiques sur les taux de réception de paquets RTP observé par chaque récepteur ; on calcule la moyenne, le minimum, le maximum ainsi que l'écart type des taux de réception sur les N derniers paquets RR reçus. La grandeur caractéristique est la moyenne du taux de pertes. Pour éviter le problème du nivellement par le bas, qu'une simple utilisation de la moyenne peut engendrer, l'algorithme utilise aussi le minimum, le maximum, ainsi que l'écart type afin d'adapter l'émission de paquets RTP aussi aux cas singuliers.

Le temps de décodage des images JPEG au niveau du client varie entre 15 et 50 ms en fonction de la qualité des images et de leurs dimensions. On note que le nombre des images dont le temps de décompression est supérieur à 40 ms, ce qui correspond à la limite permettant un affichage fluide à 25 images/secondes, est suffisamment faible pour permettre un affichage correct.

La conception et le développement de l'outil netstre@mer nous a montré la faisabilité mais aussi la difficulté de l'extraction des algorithmes d'adaptation ou de transcodage de l'application pour le cas particulier de la transmission vidéo. La figure ?? donne l'architecture générale de Netstre@mer. Dans ce schéma un seul noeud d'adaptation est représenté avec ses cinq modules : C contrôle, I interpréteur, A adaptation, R récepteur, S émetteur. Parmi les paramètres sur lesquels on peut agir, celui qui semble le plus efficace par rapport à la réduction du débit est le facteur de qualité de la compression JPEG, qui a de plus l'avantage de conserver à l'image sa lisibilité ; on perd quelques détails et des aplats apparaissent, mais la topologie de l'image n'est pas ou peu altérée. Le principal défaut de la proposition est que le processus implémenté remonte systématiquement au niveau de la couche applicative. Lorsqu'il n'y a pas d'adaptation à faire, ce détour est inutile et pénalisant. Il serait préférable de rester au niveau de la couche réseau (couche IP) quand aucune adaptation n'est nécessaire, ce n'est malheureusement pas possible en Java car aucun accès natif à la couche IP n'est fourni.

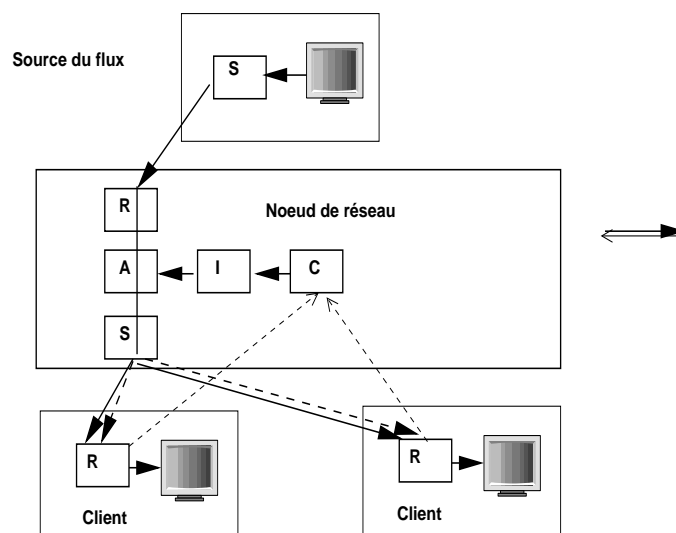


FIG. 4.10 – Architecture générale de netstre@mer

4.4 Conclusion

Au niveau réseau IP, des mécanismes de gestion active de file d'attente, tels que RED, ont été développés depuis plus d'une dizaine d'années pour limiter les congestions dans Internet. Ils sont conformes à la philosophie IP et se déploient progressivement. Mais ces mécanismes ne fournissent pas explicitement de services évolués aux applications. Comme le sur-dimensionnement, ils sont une des réponses partielles à l'amélioration des performances globales de l'Internet et au problème de la qualité de service. Les propositions Int-

Serv/RSVP et DiffServ ont cherché à apporter des solutions pour la fourniture de services différenciés dans IP avec des garanties de QoS tout en conservant un service Best Effort bien éprouvé. Dans le plan contrôle, IntServ requiert une signalisation explicite et le stockage et le traitement d'états par flux. Dans le plan données, IntServ requiert de la classification, de l'ordonnement et de la gestion de buffer par flux. Cet ensemble de contraintes s'est avéré être un lourd handicap pour le déploiement à large échelle de IntServ. Le modèle DiffServ paru donc dès le début attractif pour sa simplicité. L'IETF n'a pas explicitement recommandé des services de bout en bout comme dans IntServ, mais s'est donné pour objectif de fournir un nombre suffisamment riche de PHB à partir desquels, des services de bout en bout pourraient être construits. Ainsi le groupe DS a standardisé un petit nombre de PHB. Le service Premium est apparu comme celui pouvant le mieux répondre aux besoins des applications temps-réel qui requièrent des bornes précises pour les pertes et la gigue. Un service Premium visant à offrir un service de type *liaison spécialisée* s'appuie sur trois mécanismes de QoS différents : la différenciation par marquage des paquets au sens DiffServ, le sur-dimensionnement du service au niveau des routeurs d'entrée, le contrôle d'admission de type Intserv en bordure. Mais le déploiement de Premium Service s'avère lui aussi très complexe et peu concluant. Le modèle d'architecture DiffServ, tel qu'il a été défini au départ, ne semble pas être la solution de QoS de l'Internet du futur. J'avais dirigé nos efforts vers des solutions de QoS plus simples telles que les services *différents mais égaux* proposés dans ABE ou QBSS et la proposition du modèle **Balanced Forwarding**. La réorientation actuelle des travaux du Qbone nous montre la pertinence de ce choix. Dans le chapitre suivant, j'explique pourquoi et comment nous avons poursuivi nos explorations dans cette direction.

D'un autre côté, les techniques adaptatives se sont beaucoup développées dans les applications audio-vidéo pour Internet. Une bonne connaissance de la problématique des applications multimédia est aujourd'hui acquise. L'inconvénient majeur des techniques adaptatives est qu'elles ajoutent de la complexité au développement de l'application et qu'elles sont peu réutilisables et extensibles. Le principe de nos travaux sur le module Netstre@mer a été de dissocier les mécanismes d'encodage et d'émission des mécanismes de mesure de performances et d'adaptation dynamique. Ainsi, en fonction de l'information de retour émise par le ou les récepteurs, une source est capable d'augmenter le débit d'émission ou de modifier le codage pendant les phases où le réseau est sous-utilisé et au contraire, le réduire dans le cas inverse. Ces techniques sont particulièrement attractives pour les sessions longues de réunion en vidéoconférence pendant lesquelles les variations tant des performances réseau que des besoins usagers peuvent être importantes. L'objectif visé était de construire des mécanismes génériques qui s'adaptent d'une part aux variations de performances du réseau mais aussi aux variations des besoins des applications comme nous en avons identifié le besoin dans nos précédentes études.

A l'issue de ces travaux à différents niveaux, application, techniques adaptatives et réseau, j'ai acquis l'intime conviction que c'est la réponse architecturale à la QoS que l'on proposera qui offrira le compromis performance - flexibilité adéquat. Il est clair que plus on sophistique les services du niveau réseau avec des solutions à états, nécessaires pour obtenir un meilleur niveau de garantie, plus la complexité des routeurs augmente au détriment de la performance. Plus la complexité est repoussée aux extrémités, plus les routeurs sont simples et performants, au détriment de la flexibilité et de la prise en compte de l'hétérogénéité et de la dynamique. Ainsi je pense qu'il est nécessaire de combiner les approches proposées à différents niveaux du modèle architectural pour obtenir un compromis final intéressant. Les solutions *dans le réseau et aux extrémités* ne doivent pas s'exclure mutuellement. Par la suite, j'ai cherché à explorer les limites du modèle DiffServ et à étudier le domaine des réseaux actifs pour la qualité de service. L'expérience acquise avec Balanced Forwarding et Netstre@mer servent de base à nos recherches sur les services différenciés équivalents et actifs et les protocoles de transport programmables.

Chapitre 5

Propositions pour un réseau sensible aux flux

5.1 Introduction

A l'issu de nos travaux sur les solutions classiques de QoS-IP, nous étions convaincus que la prise en compte des besoins hétérogènes des flux et des variations dans les scénarii d'utilisation requérait une flexibilité au niveau du réseau coordonnée à une adaptabilité des protocoles d'extrémités. Le concept de **conscience** traduit du terme anglais **awareness** s'oppose au terme de transparence, un concept clé des réseaux et des systèmes répartis. Ce concept est apparu ces dernières années dans le domaine des réseaux et des applications communicantes. Une première approche est de rendre les applications conscientes du réseau (*network aware application*) [21], [214]. L'*implémentation ouverte* (open implementation) et la programmation réflexive [120] vont dans cette direction. Une autre approche tente de rendre le réseau sensible aux applications ou aux flux (*application aware network*). C'est cette voie que nous avons choisi d'explorer, motivés par la volonté de conserver au maximum la simplicité de programmation des applications réparties, tout en tirant au maximum le potentiel du réseau. Notons que ces deux approches ne sont pas antagonistes mais complémentaires comme je le schématise sur la figure 5.1. Je préfère utiliser le terme de sensibilité est la **propriété qu'à un être vivant de sentir son milieu extérieur, et qui par extension, le rend intelligent**.

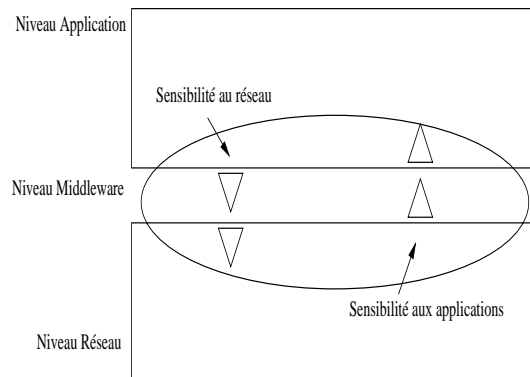


FIG. 5.1 – Complémentarité des approches *sensibles*

A la lumière des études précédentes, nous avons développé deux propositions qui sont le modèle **EDS (Equivalent Differentiated Services)** de services différenciés équivalents, étudié dans le cadre de la thèse de Benjamin Gaidioz (section 4.1) et le modèle **ADS (Active Differentiated Services)** de services différenciés actifs, issu de la technologie des réseaux actifs. Julien Rio a présenté ce modèle dans son travail de DEA et je

continue à l'approfondir avec différents étudiants du département Telecom de l'INSA de Lyon et de l'Ecole Centrale de Lyon (section 5.3.3). Dans ce chapitre, je décris les principes de ces deux types de solutions et j'en étudie les limites et les perspectives. Il s'agit de deux solutions originales qui, comme les approches SCED+ [46] ou Dynamic packet State [?], se démarquent profondément des solutions classiques et qui ouvrent des perspectives intéressantes pour l'évolution des infrastructures IP et leur utilisation. Ces deux approches ont en commun un caractère **hybride** et supportent un **déploiement incrémental**. La nature hybride est due au fait qu'elles utilisent toutes deux à la fois des mécanismes de QoS dans les routeurs du réseau et des mécanismes d'adaptation situés soit dans des équipements de bordure soit au niveau des extrémités. Un déploiement incrémental est possible car tous les routeurs n'ont pas besoin d'être *xDS-capable* et les applications ne sont pas nécessairement adaptatives. Dans les deux cas, nos solutions s'adressent à des flux dont les contraintes, temporelles en particulier, ne sont ni trop rigides ni trop fortes. Les latences ne sont ni très faibles ni strictement bornées. Nous faisons l'hypothèse que la majorité des flux des applications de l'Internet actuelles et futures entrent dans cette catégorie de flux élastiques. Les flux critiques tels ceux des applications de chirurgie à distance devront, à notre avis, emprunter des réseaux dédiés et extrêmement bien dimensionnés ou s'appuyer sur des mécanismes de réservation stricte de ressources proposés par le modèle Expedited Forwarding de DiffServ ou de réservation de longueur d'ondes. Par contre, la téléphonie IP et la vidéoconférence peuvent bénéficier de tels modèles souples.

5.2 Le modèle EDS

5.2.1 Objectifs et hypothèses

L'objectif de nos travaux sur le modèle **Equivalent Differentiated Services** a été de rechercher la limite que peut apporter **le réseau** au problème de la qualité de service dans le cadre de l'Internet. Le modèle TCP/IP a prouvé sa formidable robustesse et ubiquité. Nous pensons que s'il est nécessaire de le faire évoluer, il n'est pas souhaitable de le remettre en question fondamentalement. L'objectif premier de la philosophie de conception d'Internet [39] était de développer une technique efficace pour l'utilisation de réseaux interconnectés. Examinons les objectifs secondaires ordonnés comme suit :

- 1) la communication Internet doit se poursuivre en dépit des pannes de réseaux ou de passerelles.
- 2) Internet doit supporter de multiples types de services de communication
- 3) l'architecture d'Internet doit s'accommoder à une grande variété de réseaux
- 4) la gestion des ressources ou la comptabilité sont d'autres objectifs, mais reconnus au départ comme étant de moindre priorité.

Pour le concepteur, les multiples types de services (2) doivent être localisés au niveau transport. Le type de service traditionnel est la remise de données fiable et bidirectionnelle. Ce fut le premier et quasiment seul service fournit. Selon ce principe initial, il était clair que l'adaptation relève de la couche transport et non de l'application elle-même qui doit demeurer simple à programmer. Pour faire évoluer Internet, un certain nombre de conseils ont été formulés [82]. En particulier, la robustesse est considérée comme plus importante que l'efficacité. Comme Internet est caractérisé par une hétérogénéité à de nombreux niveaux et que les changements peuvent être non anticipés et non contrôlés, le développement et le déploiement de l'infrastructure sont nécessairement incrémentaux et les solutions extensibles.

La question que nous nous sommes posée alors fut : est il possible de redessiner la couche réseau en lui apportant de la QoS tout en respectant la philosophie IP, pour laisser au réseau la simplicité et la robustesse dont il fait preuve ?

Le modèle de Service Différentiés opérant sur des classes, c'est à dire des agrégats de flux, même s'il présente de nombreuses limites, nous semble demeurer un bon concept pour introduire des services nouveaux dans Internet. Pour étendre son domaine d'application d'un bout à l'autre du réseau, un certain nombre d'hypothèses, sous-jacentes au modèle DiffServ doivent être relâchées et une approche incrémentale privilégiée. Il est en effet impossible de changer tous les routeurs de l'Internet simultanément. Or les solutions proposées dans IntServ et DiffServ requièrent une mise à niveau de tous les routeurs d'un même domaine ainsi qu'une collaboration des domaines adjacents. C'est ce qui en rend le déploiement de bout en bout très ardu. Le modèle BEDS [76] et ou notre modèle BF [93] présentés dans le chapitre précédent sont des exemples

d'approches incrémentales qui permettent la cohabitation de routeurs classiques avec des routeurs QoS tout en apportant un gain de performance et une différenciation de services. Un traitement différencié des flux effectué là où l'hétérogénéité de performances et les congestions sont fréquentes peut apporter un certain bénéfice aux flux de bout en bout. Les mécanismes exacts pour positionner les niveaux de priorité et le gain effectif apporté par de telles approches sont encore des problèmes ouverts comme nous l'étudions dans la section suivante de ce chapitre.

5.2.2 Principes de base

Nous avons introduit le concept de services différenciés équivalents dans le but de faire évoluer le paradigme IP non pas vers une couche réseau offrant de la qualité de service mais vers une couche IP à services différenciés dont aucun n'est intrinsèquement meilleur qu'un autre. Le principe d'équivalence est requis pour assurer la gratuité des services, gage de l'extensibilité du modèle. L'objectif du modèle EDS est d'optimiser l'acheminement de classes de paquets au sein d'un réseau IP en fonction de leur besoins propres en termes de performances, tout en optimisant l'utilisation des ressources des routeurs. La couche EDS de niveau 3 est conçue comme un bloc de base de l'architecture et doit être associée à des protocoles de transport en charge d'une compensation de performances spécifique à l'application utilisatrice. Dans le cadre de sa thèse, Benjamin Gaidioz a défini au niveau 3 la couche EDS comme évolution de IP et il étudie à présent une couche de transport différencié comme évolution de TCP/UDP pour l'architecture EDS.

Caractéristiques de EDS

Au niveau 3, EDS propose une alternative simplificatrice de l'architecture DiffServ afin de relâcher les contraintes relatives

- au contrôle d'admission (indispensable dans EF),
- au concept de domaine qui pose des problèmes d'interopérabilité entre domaines et ne garantit pas aisément la fourniture de service de bout en bout.
- au principe de tarification implicite de l'architecture DiffServ qui impose des mécanismes de facturation et de comptabilité.

Les trois caractéristiques fondamentales de EDS sont donc une **approche incrémentale, pas de contrôle d'admission, pas de tarification différenciée**.

Pour permettre un déploiement incrémental, la définition du comportement de chaque noeud s'appuie uniquement sur des paramètres locaux. Aucun critère global du réseau tel que le comportement des flux, le nombre de routeurs traversés n'est prise en considération dans les algorithmes de mise en oeuvre. Un routeur prend des décisions à partir de statistiques qu'il mesure et stocke localement et d'informations contenues dans chaque paquet traité. Afin de pouvoir assurer les propriétés du service sans pratiquer de contrôle d'admission, les garanties offertes sont *relatives* plutôt qu'absolues. Proposer une garantie absolue impliquerait un dimensionnement des ressources allouées à une classe et le contrôle que le trafic ne les sature pas. Si les garanties sont relatives, il est toujours possible de les assurer quelle que soit la charge du réseau. Pour s'affranchir de la tarification, les classes proposées ne doivent pas pouvoir être ordonnées strictement selon la qualité d'acheminement des paquets qu'elles offrent. Ainsi, on assure une sorte d'équité entre les classes, bien que les paquets reçoivent des performances statistiques différentes.

Spécification du modèle

Le modèle EDS propose un nombre quelconque N ($N \geq 2$) de classes. La différenciation est pratiquée sur les métriques délai et le taux de perte de chaque classe. On peut envisager une extension du modèle à un nombre quelconque de métriques. Dans le modèle délai-perte initial, on attribue à une classe i un coefficient de délai d_i et un coefficient de taux de perte p_i . Ces deux coefficients sont constants. Pour deux classes i et j , le routeur ordonnance les paquets et choisit ceux qu'il détruit de telle sorte qu'il y ait un rapport d_i/d_j entre leurs délais moyens *locaux* et un rapport de p_i/p_j entre leurs taux de perte *locaux*. Afin de ne pas définir de classe aux performances strictement meilleures que les autres par rapport aux autres, l'attribution des coefficients est telle que si pour deux classes i et j on a $d_i > d_j$, alors on a aussi $p_i < p_j$ et vice-versa.

Ainsi les extrémités disposent d'une gamme de services d'acheminement de paquets qui peuvent être utilisées librement. Lorsque la charge du réseau augmente, les performances de bout en bout obtenues par les flux varient et peuvent ne plus correspondre à leurs besoins. L'extrémité a alors la possibilité d'emprunter un autre service plus adéquat. EDS propose d'accélérer l'acheminement de paquets ou (dans le sens exclusif du terme) de le rendre plus fiable, contrairement à un réseau Best Effort qui n'offre qu'un acheminement plat. Dans le modèle Balanced Forwarding, BF que nous avons précédemment proposé (voir chapitre 3) la valeur des rapports augmentait avec la charge pour accentuer la différenciation. Ainsi, l'adaptation dynamique à la charge variable du réseau se faisait dans les routeurs. On obtenait une réactivité très grande qui avait pour conséquence néfaste d'engendrer une instabilité importante du système. Dans EDS, les rapports sont constants, l'adaptation doit donc se faire plutôt en périphérie au regard des performances effectivement observées. Un flux peut utiliser une classe au rapport plus élevé ou plus bas selon les performances mesurées. Nous détaillons ce point dans la section 5.2.3 suivante.

Dans EDS, le marquage se fait à la source, il doit être transporté de manière non altérée de bout en bout puisqu'il sert à spécifier de manière abstraite les contraintes du flux. Dans le draft que nous avons soumis à l'IETF [92], nous avons soulevé un certain nombre de contraintes relatives à la mise en oeuvre de EDS. En particulier le champ d'identification de classe, le *Class identifier*, doit avoir un traitement différent du DSCP de DiffServ [142]. Dans DiffServ, en effet, le codepoint (DSCP) peut être modifié à l'entrée de chaque domaine et il n'y a aucune garantie de bout en bout. Par ailleurs, la taille du champ fixe la valeur N du nombre de classes. Nous avons défini une classe par défaut, de valeur 0 et qui approche le comportement d'une classe Best Effort.

La différenciation de service proportionnelle

La différenciation proportionnelle, permet de s'affranchir du problème de contrôle d'admission. Cette différenciation est telle que la performance d'une classe est proportionnelle à un coefficient qui lui est associé. Le modèle **Proportional Differentiated Service** [59] fournit par exemple un ensemble de classes avec une différenciation de délai proportionnelle à un ensemble de paramètres. L'intérêt d'offrir de telles garanties relatives est qu'il n'est plus nécessaire de contrôler le trafic de chaque classe pour assurer les propriétés comme c'est le cas lorsque les garanties sont absolues. Dovrolis a défini deux ordonnanceurs permettant d'effectuer la différenciation selon le délai [61] et le taux de perte [60]. Une des limites du modèle est de nécessiter une facturation car il est possible d'ordonner strictement les performances offertes par les différentes classes de service. Si aucun mécanisme spécifique de régulation n'est additionné au modèle proportionnel, tous les trafics pourraient se concentrer dans la même classe à plus forte priorité et rendrait la différenciation caduque. Tous les flux obtiendraient le même service relatif qui deviendrait un simple service Best Effort. La symétrie des performances, telle qu'elle est spécifiée dans le modèle EDS permet de lever cette contrainte de facturation.

Disciplines de service

Pour réaliser le service EDS des mécanismes spécifiques doivent être implémentés dans les routeurs. Benjamin Gaidioz a développé des mécanismes de différenciation proportionnelle sur le délai et sur le taux de perte basés sur les algorithmes **WTP, Waiting Time Priority** et **BPR (backlog-proportional rate)** proposés par Dovrolis [59]. Les algorithmes de Dovrolis approximent le modèle DS proportionnel pour une différenciation de délai et sous des conditions de charge élevée. L'ordonneur BPR (backlog-proportional rate) est basé sur l'algorithme GPS [153] modifié. Les taux de service d'une classe sont dynamiquement ajustés afin d'être rationnés proportionnellement au ratio correspondant à la charge de la classe mesurée. Cet algorithme montre des variations de délai en dent de scie sur des échelles de temps courtes. Le second algorithme, Waiting Time Priority (WTP), est basé sur l'algorithme Time Dependant Priority (TDP) de Kleinrock [121]. La priorité d'un paquet du flux i au temps t est proportionnelle au temps d'attente du paquet au temps t , où la constante de proportionnalité, s_i , est le paramètre de service pour i . Les auteurs ont montré par simulation (et nous l'avons vérifié expérimentalement) que le délai moyen relatif observé par deux flux i et j dans un serveur WTP a une valeur proche de s_i/s_j , pour des durées de mesure de l'ordre de quelques dizaines de temps de transmission paquets. Ainsi, WTP donne une bonne approximation du modèle de différenciation proportionnelle sous des conditions de trafic élevées. Les paramètres

de WTP doivent être adaptés en fonction de la distribution de la charge du trafic. Certains auteurs [129], ont reproché le manque de caractérisation de ce modèle proportionnel et l'absence d'indication sur les conditions dans lesquelles cette approximation reste valable. Par exemple, la notion d'échelle de temps *courtereste* imprécise. Ainsi Leung a montré de manière analytique les conditions dans lesquelles l'ordonnanceur est correct et produit une large différenciation des temps d'attente.

- **condition 1** : il faut qu'il y ait suffisamment de paquets de classes élevées pour occuper le serveur et faire en sorte que les paquets des classes basses soient retardés de manière adéquate.
- **condition 2** : il faut que le serveur retarde les paquets de basse classe. Si leur charge est élevée, un grand nombre d'entre eux seront mis en file d'attente et leur temps d'attente va augmenter.
- finalement, il existe des distributions du trafic pour lesquelles il n'y a pas de solution analytique, ce qui signifie que le système ne peut obtenir les ratios de temps d'attente.

Ces auteurs ont montré qu'en utilisant les valeurs des paramètres de contrôle proposée par Dovrolis, on obtenait des valeurs effectives de 1.3 au lieu de 2. Les courbes que nous avons obtenues montrent des résultats similaires. Benjamin Gaidioz étudie comment améliorer l'ordonnanceur proportionnel. Dans un modèle proportionnel pur, qui offre des classes de services strictement meilleures que d'autres, et doit donc être associé à des mécanismes de facturation différenciées, il est important que les mécanismes de mise en oeuvre garantissent des rapports de proportionnalité exacts. Cependant, dans le cas du modèle EDS, nous cherchons à améliorer le traitement des paquets selon une métrique et à le dégrader selon une autre. Ainsi les contraintes sur l'exactitude et le déterminisme des conditions d'applicabilité du système ne sont pas aussi strictes. Nous avons donc implémenté cet ordonnanceur et ce mécanisme de gestion de file pour en étudier le comportement et analyser les propriétés du modèle.

Selon la spécification EDS, la valeur des coefficients de proportionnalité peut différer d'un routeur à l'autre. La seule contrainte sur le positionnement des coefficients est de conserver deux ordres asymétriques. La répartition des coefficients peut être simplement linéaire comme dans la colonne 1 de la table suivante ou bien quadratique (colonne 2).

	d1	l1	d2	l2
1	8.0	1.0	4.0	0.5
2	7.0	2.0	3.9	1.0
3	6.0	3.0	3.8	2.0
4	5.0	4.0	3.7	4.0
5	4.0	5.0	2.5	8.0
6	3.0	6.0	2.0	8.1
7	2.0	7.0	1.5	8.2
8=N	1.0	8.0	1.0	8.3

TAB. 5.1 – Deux type d'initialisation des coefficients de proportionnalité

Le paramétrage de la colonne 2, consiste à regrouper les classes 1 à 4 et les classes 5 à 8. Dans le premier groupe il y a un plus fort écartement des classes selon la métrique perte de paquets. La différenciation en délai est linéaire, celle en perte est quadratique. Ainsi la définition des coefficients de proportionnalité est ouverte et flexible. Elle peut être adaptée à chaque type de routeur selon sa localisation dans un réseau. Nous poursuivons nos travaux de simulations pour dégager des propriétés intéressantes et des règles d'initialisation pertinentes. Ces règles dépendent des propriétés de bout en bout attendues, comme nous le développons dans la sous-section suivante.

Validation du modèle

Pour valider le modèle EDS, nous avons implémenté les mécanismes dans le simulateur de réseau ns [143] et effectué un certain nombre de simulations. Nous menons d'autres travaux actuellement sur une plate-forme locale expérimentale avec l'émulateur de réseau longue distance nistnet [149]. L'étude consiste à montrer :

- La validité de l'approche incrémentale
- La pertinence des coefficients de proportionnalité

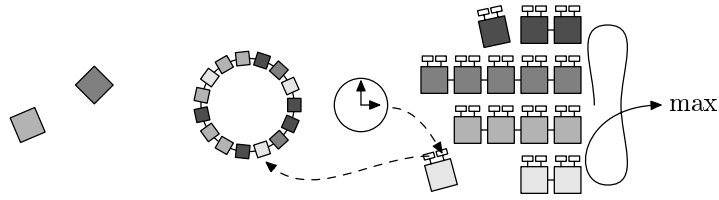


FIG. 5.2 – L'ordonnanceur de EDS

- La capacité du modèle à différencier correctement des flux selon différents critères
- Les performances de bout en bout obtenues par les flux.

5.2.3 Transport différencié sur EDS

Couche transport et services différenciés

Les protocoles de transport traditionnels ont été spécifiquement développés au-dessus d'un modèle Best-Effort plat. Nous posons donc l'hypothèse suivante : 1 : il n'y a à priori aucune raison pour que les protocoles TCP/UDP exploitent au mieux les modèles de services réseaux différenciés qui améliorent le taux de perte, le délai ou le débit.

Nous prétendons donc que si l'on modifie la couche IP, il faut conjointement faire évoluer la couche de transport. Cette évolution doit se faire dans plusieurs directions :

- Le principe de contrôle de congestion par la couche transport (en l'occurrence TCP) doit être réexaminé, en fonction des mécanismes réseau mis en oeuvre qui ont, par construction, une action sur les phénomènes de congestion.
- Plusieurs types de services transport doivent être proposés pour servir au mieux les besoins hétérogènes des applications. La couche TCP unique optimise la fiabilité au détriment du délai. Elle est donc contournée par les applications à contraintes temporelles. La couche UDP, elle ne fournit pas de service transport spécifique. Les protocoles RTP et RTCP agissent en dehors du noyau du système d'exploitation et au dessus la couche transport alors que leurs fonctionnalités relèvent de cette couche.
- Les fonctions d'adaptation doivent être intégrées autant que possible aux services de transport.
- La couche transport doit évoluer vers une couche modulaire et adaptable

Pour aborder cette problématique vaste et ouverte, nous avons choisi d'en étudier les différentes facettes de manière indépendante et progressive. La démarche et les résultats de cette étude peuvent s'appliquer à d'autres types de couches réseau à services différenciés que le modèle EDS. Ces travaux s'inscrivent dans une lignée des recherches nouvelles en rupture avec les protocoles de transport traditionnels dont nous avons montré certaines des limites au chapitre précédent. La démarche est la suivante :

- étudier dans quels domaines l'hypothèse 1 est vérifiée.
- identifier et spécifier les services de transport différenciés requis.
- isoler, concevoir et évaluer les mécanismes de la nouvelle couche transport.

Nous nous basons sur le modèle EDS qui propose des services dont les performances sont relatives et équivalentes. Une couche réseau EDS est vue par un protocole de bout en bout comme un ensemble de canaux plus ou moins rapides et moins et plus fiables. Dans la suite de sa thèse, Benjamin Gaidioz construit un service de transport différencié au dessus de la couche EDS en suivant la démarche proposée.

TCP et DiffServ

Pour vérifier l'hypothèse (1) nous avons entrepris un travail d'étude du comportement des protocoles de transport traditionnels TCP et UDP sur une couche EDS, afin d'étudier les propriétés de bout en bout obtenues par les flux.

Dynamique de TCP

Le travail sur TCP/EDS requiert une bonne compréhension de la dynamique de TCP [71]. Selon le modèle TCP/IP, le contrôle de congestion doit être assuré par les extrémités car il n'est pas pris en charge par les routeurs. Ce modèle suppose que toutes les extrémités réagissent à tout phénomène de congestion, de manière similaire par principe d'équité et le plus rapidement possible pour en limiter l'étendue. On distingue trois phases dans le comportement d'une connexion TCP : la phase de démarrage lent (slow start), la phase d'évitement de congestion (congestion avoidance) et la phase d'état stable (steady state). Le mécanisme d'évitement de congestion (AIMD : additional increase, multiplicative decrease) utilise une fenêtre de congestion dont la taille $cwnd$ est incrémentée de $1/cwnd$ à chaque réception d'acquittement ACK. La taille de la fenêtre décroît lorsqu'une perte est constatée. La valeur de cette diminution dépend de si la perte a été observée par une duplication de ACK ou un l'expiration d'un délai de garde. A l'ouverture de la connexion, le récepteur informe l'émetteur de la taille $rwin$ de sa fenêtre de réception. Ainsi, $cwnd$ pourra croître jusqu'à $rwin$ mais ne pourra pas aller au delà en l'absence de pertes.

Le comportement de TCP dépend donc beaucoup de la qualité du lien en termes de pertes de paquets. Par ailleurs, pour fonctionner sur différents types de réseaux, le timer de retransmission est calculé sur la base du délai d'aller-retour RTT . La capacité d'une connexion TCP est donc égale au produit $dbit * dlai$. Si ce produit devient trop élevé, les principes de fonctionnement de TCP - détection d'une congestion par perte - deviennent caduques -trop de paquets émis avant la detection -. Au cours de cette dernière décade, les chercheurs ont mené des études intensives et proposé des améliorations pour TCP/IP. Ces optimisations comprennent : des modifications dans les implémentations des piles TCP NewRENO, Vegas, Fast retransmit (FACK : avant time-out), SACK (selective ACK) [133], ainsi que de meilleurs algorithmes de gestion de files (par ex : AQM, RED, SRED, FRED, FPQ) [62] ou des schémas qui requièrent à la fois un support de bout en bout et dans le réseau [191], [180], [74]. Les améliorations de performances de toutes ces approches ont été mesurées en termes de

- évitement des expirations de délai de garde (timeout)
- meilleur filtrage des rafales de pertes pour déterminer les périodes de congestion et les réductions de fenêtre
- optimisation de la retransmission
- amélioration de l'équité entre les sessions TCP (courtes ou longues)

TCP sur AF

Dans un modèle à services différenciés, les routeurs sont capables de classer et de traiter les paquets différemment, et cela en particulier en cas de congestion. S'appuyer sur un protocole de transport effectuant du contrôle de congestion uniquement de bout en bout sans tenir compte des traitements spécifiques effectués dans le réseau est insatisfaisant. Le traitement différencié de niveau 3 est annulé par le niveau 4. C'est ce que montrent en effet un certain nombre de travaux sur TCP au dessus du service Assured [190]. La couche AF différencie le débit offert aux flux en éliminant plus ou moins les paquets selon le niveau de priorité. Dans [72] Feng a montré que le mécanisme de contrôle de congestion basé sur la détection des pertes de paquets, interfère de manière complexe avec ce service réseau. Ainsi, il existe des cas où certains flux, ayant négocié une priorité (et donc un tarif) plus élevée que d'autres, obtiennent des performances en terme de débit utile moins intéressantes ! Feng a proposé de modifier l'algorithme de contrôle de congestion afin de maintenir un débit garanti à chaque classe en manipulant la fenêtre de congestion.

$$cwnd = rwnd + ewnd$$

avec $rwnd$ la fenêtre correspondant à la bande passante réservée et $ewnd$ la fenêtre correspondant à la bande passante supplémentaire disponible. Plus la classe AF est élevée, plus la fenêtre $rwnd$ est grande. Lors d'une détection de congestion, l'algorithme d'adaptation interfère avec fenêtre $ewnd$ et laisse $rwnd$ non modifiée. Ceci permet d'assurer des garanties différentes aux flux appartenant à différentes classes AF en cas de congestion.

Les travaux de Sahu et al [189], ont aussi montré l'indéterminisme de performances lorsque TCP est projeté sur le service AF même avec un faible nombre de connexions. Cette incertitude provient du fait que AF introduit une nouvelle dimension dans la gestion des buffers qui est un marquage différentiel et des drops de paquets qui ne sont pas correctement pris en compte par TCP. Dans [?], les auteurs proposent d'ajouter des composants pour compenser ce gap de performances. Le principe est de voir le problème de la gestion des buffers comme deux sous-problèmes : le premier est le marquage *TCP-friendly* et le second un drop différentiel (RIO). L'objectif de cette proposition est de minimiser les expirations de délai de garde qui sont la source de la plus importante dégradation

des performances TCP sur le plan du débit utile moyen, du temps de transfert et de la variance d'équité de service. Pour atteindre ces objectifs, le marquage vise à protéger les petites fenêtres, pour maintenir un maximum d'espace entre les paquets ou de façon à disperser l'effet des pertes en rafales, protéger les paquets de retransmission. Le plus efficace est le marquage des paquets retransmis qui généralement sont envoyés pendant des périodes de congestion et ont une probabilité d'être éliminés très forte. C'est une solution de marquage adaptatif **sensible au flux** (flow aware marking).

TCP sur EDS

Dans le cas d'EDS, il nous a paru intéressant d'étudier le comportement de TCP puisque le taux de perte et le délai sont altérés simultanément. Nous avons donc cherché à isoler des situations dans lesquelles EDS apporterait un réel gain en terme de débit utile substantiel ou bien s'il était possible de dégager des valeurs de coefficients de proportionnalité intéressantes. Pour cela Benjamin Gaidioz a mené des expériences de simulation. Dans ces expériences, il utilise huit classes EDS dont quatre sont occupées par TCP et quatre par du trafic CBR sur UDP qui occupe 10% du débit disponible. Dans le cas d'un réseau au débit raisonnable et pour une latence faible, TCP a un bon fonctionnement en termes d'égalité des performances [177]. En effet, chaque connexion perd régulièrement des paquets, ce qui provoque une oscillation de la taille de fenêtre d'émission, donc du débit utile instantané. Ces oscillations relativement rapides font que d'une manière globale (et pas instantanée), les performances des connexions se valent. Dans le cas d'un réseau haut-débit et d'une latence élevée (cf. fig. 5.3), les "oscillations" (si on peut encore parler d'oscillations) sont beaucoup plus lentes et amples. Donc même d'un point de vue global, il est très courant qu'une connexion soit nettement désavantagée par rapport à une autre. La figure 5.3 à gauche correspond à un réseau classique et à 60 secondes d'exécution. A droite le réseau est haut-débit grande latence sur 10 secondes d'exécution. Dans le cas d'un réseau haut-débit, la progression d'un transfert est régulière mais certaines connexions sont sérieusement désavantagées par rapport aux autres car la perte d'un paquet a un impact très négatif dans un réseau haut-débit à latence élevée.

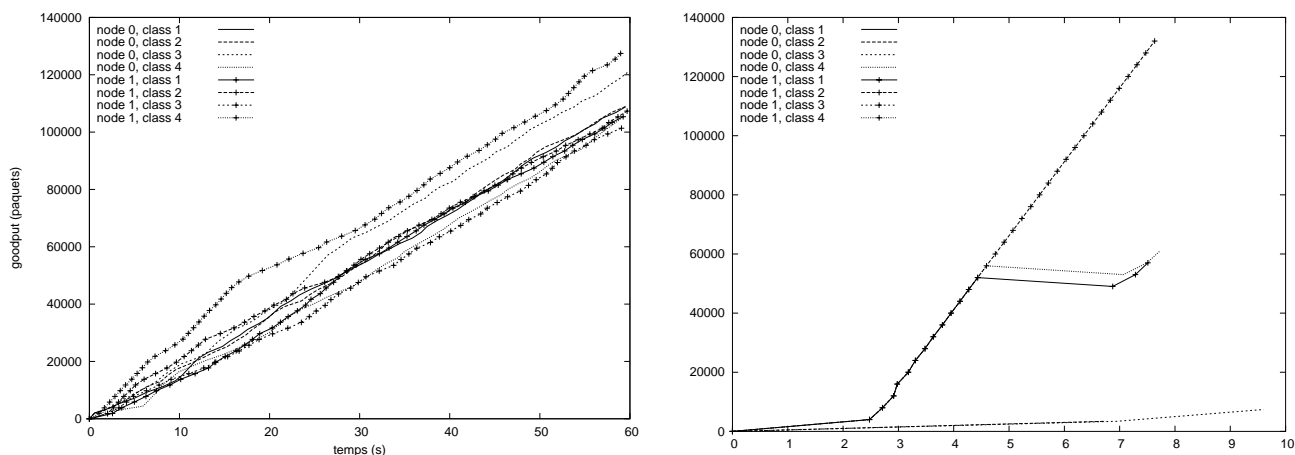


FIG. 5.3 – Performances de connexions TCP sur un réseau classique et sur un réseau haut débit à latence élevée

Nous avons conduit un certain nombre d'autres expériences où des connexions TCP de classes différentes se partagent un lien congestionné avec un trafic UDP concurrent réduit (10%). La différenciation sur une statistique a un impact sur les performances conforme aux résultats attendus. Dans le cas d'une différenciation sur les deux statistiques conformément au modèle EDS, l'impact est moins évident. Le cas particulier de la fig. 5.4 permet de constater qu'au sein d'une classe, deux connexions peuvent avoir des performances suffisamment différentes pour que le choix d'une classe n'ait pas d'impact absolu sur les performances. On a effectué un zoom sur les 10 dernières secondes quand les coefficients de EDS sont tous égaux (à gauche, les classes ont les mêmes performances) et sur EDS avec une différenciation linéaire sur les deux critères (à droite). Quand la différenciation a lieu, pour les flux du nœud 1 la connexion de la classe 2 obtient de moins bonnes performances que celle de la classe 4 alors que pour

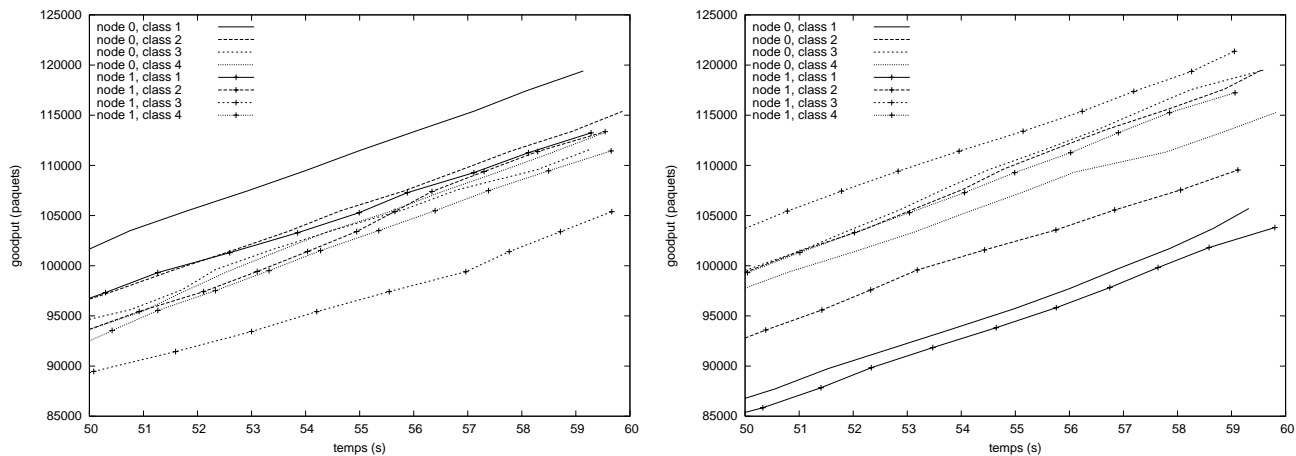


FIG. 5.4 – Performances de connexions TCP sur quatre classes

celles du nœud 0, c'est la classe 2 qui est la plus performante. Pourtant, les huit flux sont dans des conditions tout à fait semblables (caractéristiques des liens identiques d'un bout à l'autre du réseau, même trafic concurrent, etc.) L'algorithme d'adaptation de TCP interfère fortement avec EDS et cherche à rétablir l'équité de débit entre les flux. Par contre, les simulations avec des latences beaucoup plus importante (100 à 150ms) exhibent des propriétés plus marquantes. Une première analyse nous montre que dans ces conditions, ce sont les pertes de paquets pendant la phase de slow start qui ont la plus grande influence sur les performances de la connexion. En effet, une perte de paquet pendant cette phase ralentit gravement la croissance de la fenêtre de congestion et diminue le seuil de slow start (ssthreshold). Nous sommes en cours de dépouillement et d'analyse détaillée de ces résultats.

Nous poursuivons aussi le travail d'étude de TCP sur EDS avec Benjamin Gaidioz et Mathieu Goutelle par des expérimentations de marquage différencié sur les paquets en retransmission que l'on associe à une classe à taux de perte très faible. Nous redéployons aussi nos tests sur un réseau expérimental avec un émulateur de WAN.

Nouveaux services de transport sur EDS

L'hypothèse 1 est vérifiée : le protocole TCP n'exploite pas la différenciation offerte par EDS. Nous avons donc commencé à étudier quel type de modification devait être apporté à TCP pour cela. Nous explorons les solutions de marquage semi-statique (par flux) et de marquage dynamique (par paquets). En parallèle, nous réfléchissons à un nouveau protocole de transport valorisant au mieux les potentialités de la couche réseau EDS.

Le modèle EDS présente à la couche transport des datagrammes dont l'entête contient une nouvelle information qui est un identificateur de classe EDS. En manipulant l'identificateur de classe des paquets d'un même flux, la source peut directement influencer sur la couche inférieure. Ainsi, le problème revient à définir un algorithme de marquage adaptatif qui optimise un critère de performance spécifique au flux. Par exemple, dans le cas d'un flux temps-réel de type voix à débit de 64kb/s, si le critère cible est un délai de 150ms avec une variance de délai de 10ms, les paquets estampillés sont analysés à la réception. S'ils sont en retard, ils seront implicitement éliminés par l'application. Le récepteur renverra un message de rétroaction indiquant à l'émetteur de marquer les paquets avec un identifiant de classe plus rapide. Si les paquets sont en avance, il faut que le flux emprunte une classe plus lente. Dans la figure ci-dessous 5.5, nous donnons un exemple de simulation d'un tel protocole d'adaptation pour un flux temps-réel.

On voit ici l'intérêt de disposer d'un nombre élevé de classes afin de réaliser des ajustements progressifs qui ne rendent pas le système global instable. Si le critère de performance est un certain niveau de fiabilité dans l'intervalle $[0, 0.5]$ sans contrainte temporelle, le protocole pourra choisir une classe à faible taux de perte. De plus, il utilisera un mécanisme de type ARQ si le taux d'erreur requis est 0 ou de type FEC variable s'il est non nul. La souplesse apportée par EDS semble bien s'adapter à l'hétérogénéité des besoins de QoS que nous avons exprimés dans le chapitre 3. Notre objectif est de définir et d'évaluer les algorithmes adaptatifs permettant de construire au

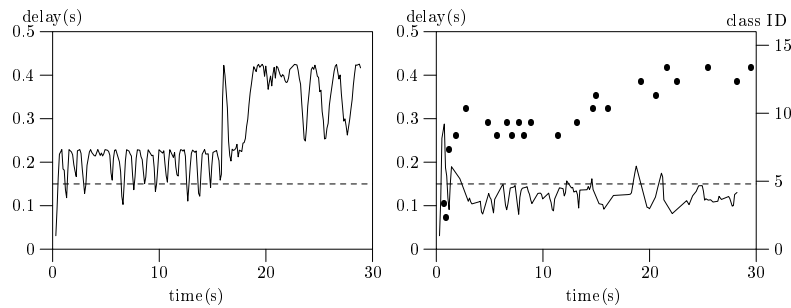


FIG. 5.5 – Flux temps réel sur IP Best effort et sur EDS

dessus de EDS des services de transport variés et utilisant tout le potentiel de EDS en réexploitant au mieux les briques élémentaires de TCP. Dans une première étape, nous relâchons la contrainte de contrôle de débit à *la TCP*. Nous pensons la réintroduire ensuite, de manière indépendante, en contrôlant le débit émis lors des congestions détectées. Dans le cas de EDS, l'indication de congestion ne sera pas a priori la détection d'une perte mais l'analyse du bit ECN [180] du paquet.

La modèle EDS nous offre des mécanismes pour manipuler la couche de réseau et définir des services différenciés simples à implémenter. Les travaux sur la couche transport au dessus de EDS, permettent une exploration des mécanismes de transport en rupture avec les approches traditionnelles. L'observabilité et la contrôlabilité pour surveiller et contrôler les performances, la programmabilité pour générer des services personnalisés à travers un formalisme abstrait flexible, l'auto-organisation et l'adaptabilité sont les propriétés requises par une telle couche de transport évoluée. Si l'approche est ambitieuse et n'assure pas des résultats certains, elle permet de remettre en cause des principes bien établis qui pénalisent parfois l'innovation et l'évolution des protocoles. Dans la continuité de cette démarche, nous étudions les possibilités de programmabilité du transport offert par l'approche réseaux actifs.

5.3 Approche Active pour la Qualité de Service

Parallèlement à mes travaux sur le modèle EDS, je me suis intéressée à la technologie active. Le concept des réseaux actifs a émergé en 1996 [206], [176], [35], [36]. L'objectif est d'augmenter les fonctionnalités du réseau pour inclure du traitement de données. Les réseaux actifs peuvent aussi accélérer le processus d'évolution et augmenter la souplesse du réseau en offrant des protocoles adaptatifs, des protocoles spécialisés, déployables dynamiquement et des réseaux reconfigurables. Le concept innovant des réseaux actifs offre cependant un grand nombre de défis en particulier en termes de performances et de sécurité, défis auxquels l'équipe RESO s'est attaquée. Considérant que le coeur de réseau est surdimensionné et que les routeurs traitent plusieurs dizaines de millions de paquets par secondes, l'intelligence et donc la complexité sont de plus en plus repoussées en bordure du réseau. Nous avons ainsi opté pour une architecture à noeuds actif dans les réseaux d'accès à la frontière des réseaux de coeur. La technologie active ouvre des voies nouvelles pour introduire de nouveaux services dans un réseau. Nous étudions les possibilités offertes par cette approche en nous focalisant plus particulièrement sur les services à valeur ajoutée relatifs à la Qualité de Service.

Pour mettre en oeuvre ces services, un environnement d'exécution doit être disponible. Dans chacun des noeuds actif, un tel environnement doit être prêt à recevoir et à interpréter les codes de services dynamiquement invoqués par les paquets. Actuellement, il n'existe pas de standard ni au niveau des langages de programmation de service [109], ni au niveau des environnements d'exécution.

5.3.1 Environnement actif

Pour réaliser un environnement de réseau programmable, Java offre un modèle de développement et de déploiement d'application très attractif car basé sur une plate-forme portable et indépendante du système. Une des

premières implémentations standard de réseau actif ANTS (Active Node Transfer System) [215] s'appuie sur du code JAVA exécuté sur une Java Virtual Machine. Un accent particulier a été mis sur les questions de sécurité. Cependant, comme actuellement la JVM est émulée en logiciel, l'utilisation de Java induit un coût important en termes de performances. La seconde implémentation PAN (Practical Active Network) [67] a plutôt éludé les questions de sécurité et de portabilité au profit du problème des performances. Tout est développé en C, en mode utilisateur et en mode noyau et les performances finales sont plutôt bonnes. L'environnement **Tamanoir** développé par Jean-Patrick Gelas et Laurent Lefevre au RESAM [97] se focalise principalement sur les problématiques de performances et de portabilité. Ainsi Tamanoir s'appuie-t-il sur Java pour la portabilité et recherche à optimiser les performances en jouant sur deux fronts : au niveau de l'architecture du réseau actif et au niveau du langage de programmation. Selon les principes de Tamanoir, les noeuds actifs sont positionnés à la périphérie des réseaux de coeur très haut débit, afin d'éviter aux routeurs haute performance d'être ralentis par un surcoût de traitement des paquets dans le plan donnée. Par ailleurs, Tamanoir utilise l'approche compilée de Java par GCJ (GNU Compiler for Java) qui réalise l'exécution en mode natif et offre une augmentation très sensible des performances par rapport à l'exécution par une Machine Virtuelle Java (JVM). Les paquets actifs traités par les noeuds Tamanoir sont au format standard ANEP (Active Network Encapsulation Protocol). Les paquets ANEP créés sont directement encapsulés dans un segment UDP puis dans un datagramme IP. A terme, selon la philosophie des réseaux actifs, l'entête ANEP devra précéder l'entête du protocole de transport et être traitée en mode noyau dans les routeurs.

Le paquet ANEP est constitué de deux champs principaux :

- Le champ Payload contient les données utiles que le service doit interpréter.
- Le champ Options contient des entités TLV (Type, Length, Value) qui sont utiles au noeud actif pour traiter le paquet. Dans ces TLV, on trouve une référence au service qui va traiter la charge utile, l'émetteur et le destinataire du paquet ainsi que le dernier noeud actif traversé (utile si le noeud actif ne possède pas le service et veut le récupérer).

version	Flags	Type ID
ANEP header		ANEP packet length
Option		
Payload		

FIG. 5.6 – Structure du paquet ANEP

Dans Tamanoir, chaque service est une classe JAVA héritée d'une classe *Class Service* qui contient des méthodes génériques suivantes telles que *Send()* pour envoyer un paquet à un TAN, *Receive()* pour recevoir le paquet, *Start()* pour initialiser le service, *Stop()* pour terminer le service, *Process()* pour exécuter un code ; cette méthode est appelée par *receive()*.

```
public abstract class Service {
static final int SERVICE_NUMBER = xx ;
```

```

UDPnetworkTools udp ;
String localId ;
public Service()
udp = new UDPnetworkTools() ;
localId = udp.getLocalHostName() ;

public void recv( String srcId, destId, lastId, byte [] payload ) {}
public void send( String srcId, destId, lastId, byte [] payload ) {}
public void start() {}
public void stop() {}
public void process( byte [] payload ) {}
...
}
}

```

FIG. 5.7 – Code d’un service Tamanoir

Ces méthodes sont vides, elles sont surchargées par le créateur du service. Il est ainsi très facile de créer et de déployer un nouveau service.

Pour introduire de nouveaux services orientés QoS dans un réseau actif, on considère deux catégories de service actifs :

- les services actifs qui opèrent dans le plan contrôle.
- les services actifs qui opèrent dans le plan données.

5.3.2 QoS active dans le plan contrôle

L’objectif d’un service actif de contrôle est de modifier dynamiquement le comportement des équipements ou de véhiculer un protocole de signalisation personnalisé pour le contrôle du réseau ou du transport des données. Nous avons étudié le potentiel de la technologie active au niveau du plan contrôle de QoS, en développant un service de configuration dynamique de routeurs DiffServ basé sur le noyau Linux au dessus de la plate-forme active Tamanoir [172]. Cet outil permet de changer à *chaud* les seuils et les politiques de gestion de files d’attente dans le noyau des routeurs traversés. Nous avons ainsi montré qu’il était possible de propager simplement et efficacement une politique d’allocation de ressources le long d’un chemin actif. Le service de configuration dynamique est déployé à l’initialisation d’un flux. Ce service interprète la commande Linux **tc**, l’outil de conditionnement de trafic [20], [8]. Les paramètres de la commande sont insérés dans les paquets ANEP et activent les modifications dynamique des seuils d’écartement de paquets et de la taille des files d’attente lors de leur passage dans les routeurs actifs comme le montre la figure suivante. On peut aussi activer de nouvelles disciplines

FIG. 5.8 – Service actif de configuration dynamique de routeur DiffServ

de service. Cette proposition peut être vue comme une solution de Bandwidth Broker distribué dans laquelle le routeur actif joue simultanément le rôle de PEP (Policy Enforcement Point) et de PDP (Policy Decision Point). Le flux actif contient une requête d’allocation de ressource pour un agrégat de trafic et un chemin donné, comme le ferait le protocole RSVP pour un microflux dans une approche passive. Ainsi, au lieu d’avoir une vision par domaine, on gère les ressources par chemin avec une dynamique relativement élevée. Il est possible aussi d’utiliser cette technique pour programmer des routeurs actifs en bordure de domaine DiffServ. Ceci ouvre de nouvelles perspectives pour la gestion ressources dans les réseaux DiffServ. Nous étudions cette possibilité dans le cadre du projet VTHD++ pour exploiter de manière fine et flexible les 4 classes de services offerts par la plate-forme

VTHD++ : *EF, TCPAF, UDPAF et BE*. Nous avons par ailleurs initié une étude des services actifs de supervision et de métrologie. Les réseaux actifs, permettent en effet d'augmenter considérablement l'intelligence des éléments du réseau en leur donnant le pouvoir d'analyser eux-mêmes leurs variables et d'adapter leur comportement en fonction de ces variables. Par exemple, imaginons qu'un noeud actif a connaissance que dans son chemin vers l'Internet, un des équipements est incapable de traiter des MTU de 1512 octets et est donc obligé de fragmenter les paquets. Le noeud actif régénère automatiquement des paquets à la MTU la plus basse du réseau afin de soulager l'équipement de front end si ce dernier est surchargé. Dans son travail de fin d'étude, Sadek Smaili a développé un service de supervision actif qui remonte des informations pertinentes aux équipements précédents. Ce service est un équivalent de l'utilitaire Traceroute classique mais qui en plus de l'adresse IP et des délais, fournit des informations pertinentes relatives au trajet du paquet. Les informations identifiées sont la charge CPU du noeud traversé, le MTU de chaque interface, le temps de latence entre deux noeuds, le type de machine, le nombre de processus actifs, le nombre de paquets ANEP traités par unité de temps. D'autres variables pertinentes telles que l'état des files et les valeurs de seuils utilisés par les PHB dans un modèle DiffServ peuvent être rapportées. Cette approche active dans le plan contrôle est étudiée par plusieurs équipes française et en particulier dans le projet Amarrage car elle offre de nombreuses possibilités de gestion intelligente des équipements et ne nécessite pas des environnements actifs particulièrement performants.

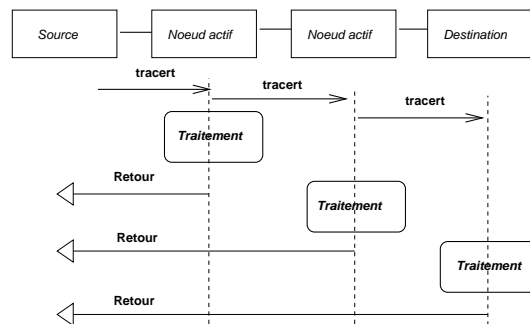


FIG. 5.9 – Service de supervision actif tracert

```

...
    if (attr.getType().equals("MTU")) {
try {
    int MTU_Size = 2000; //must be less than 10000
        char[] tab;
        String Instruction = new String();
        Instruction = "./ifconfig eth0";
        Process process = Runtime.getRuntime().exec( Instruction );
        InputStream in = process.getInputStream();
        tab = new char[MTU_Size];
        while((c=in.read())!= -1) {
tab[j] = (char)c;
        %j++;
        }// while
        String MTU = new String(tab);
        in.close();
        attr.setValue(MTU);
        Payload_back.setAttribut(attr);
        process.waitFor ();

```

FIG. 5.10 – Extrait du code du service Tracert

Les services actifs peuvent ou non être réentrants et conserver ou non des états. Nous avons étudié la problématique de la sauvegarde et de la restitution d'états dans le contexte Tamanoir. Ainsi nous avons développé un service de sauvegarde d'état, **statesaver**, pour Tamanoir. Ce service donne aux noeuds actifs la possibilité de sauvegarder des variables d'état dépendant du flux ou ne dépendant pas du flux. La gestion d'un tel type d'état global permet la gestion de la congestion par exemple.

```
\text Enregistrement dans une variable globale
public static void SaveValue(service, variable, value);
\text Enregistrement dans une variable dépendant du flux
public static void SaveValue(service, flux, variable, value);
\text Récupération d'une variable globale
public static String GetValue(service, variable)
\text Récupération d'une variable dépendant du flux
public static String GetValue(service, flux, variable);
```

FIG. 5.11 – API du service de sauvegarde d'états

5.3.3 QoS active dans le plan données : SQoS et ADS

Par ailleurs, nous nous sommes intéressés aux possibilités de traiter les paquets, c'est à dire de faire de la QoS active dans le plan données. Nous avons identifié deux approches possibles qui découlent directement des travaux présentés au chapitre 3 :

- soit descendre l'approche *bout en bout* (c'est à dire les techniques adaptatives) dans les noeuds actifs.
- soit remonter l'approche *dans le réseau* (c'est à dire IntServ/RSVP ou DiffServ) dans les noeuds actifs.

Notre objectif est de faire converger ces deux approches, comme nous avons proposé de le faire avec EDS dans le cas d'un réseau passif. Ici le contexte actif permet de dépasser les limites de traitement offert par la couche IP et d'étendre le principe à des fonctionnalités plus évoluées.

Déport des mécanismes adaptatifs dans le réseau

La première approche qui consiste à déporter l'adaptation dans le réseau est une extension des solutions adoptées par VideoGateway [9] ou Netstre@mer [79] (voir section 4.3.4). Elles consistent à faire de l'adaptation et du transcodage de flux à l'intérieur même du réseau. Dans son travail de DEA, Julien Rio a approfondi ce concept et étudié le potentiel supplémentaire offert par l'approche active. Ce potentiel se situe d'une part dans la possibilité de diminuer la latence de convergence de l'algorithme adaptatif et d'autre part dans la possibilité de prendre en compte l'hétérogénéité des besoins dans le temps et selon les récepteurs.

Dans l'approche adaptative classique (de bout en bout) comme l'adaptation est effectuée au niveau des extrémités, le protocole de régulation de bout en bout peut être lent à converger et entraîner des instabilités dans le système de communication surtout si le délai d'aller-retour est important. Si on rapproche l'adaptation au plus près du problème (congestion), on peut augmenter la rapidité de réaction. Dans [69] un protocole de contrôle de congestion programmable a été présenté selon ce principe pour faire face aux problèmes de latence de remontée des informations à l'application et de traitement unique des flux quels que soient leur nature. L'approche active offre une vision plus fine des flux globaux et la possibilité d'appliquer un algorithme d'écartement de paquets ou de stockage temporaire adapté aux types et aux besoins des flux en cas de congestion.

Dans notre modèle, nous utilisons la possibilité de propager une information de rétroaction dans deux directions : vers la source et vers les récepteurs (forward and backward feedback), pour les informer de la modification de la qualité, sans participer nécessairement au traitement. On peut ainsi obtenir une régulation plus rapide et donc plus efficace ainsi qu'une rétroaction (feedback) plus pertinente. On demande à la source de réduire son débit uniquement si la congestion persiste et/ou si tous les utilisateurs le désirent. Julien Rio s'est attaché à définir et mettre en oeuvre un service actif d'adaptation afin de mettre en lumière les problèmes de localisation ainsi que les mécanismes d'activation et de filtrage. Le protocole SqsS est basé sur un modèle source-agent-récepteurs. Les agents de surveillance des performances et d'adaptation spécifiques à l'application sont localisés et déployés au démarrage du flux dans différents noeuds actifs, points stratégiques entre la source et les récepteurs. Ces agents surveillent les métriques délai et pertes. Ils dialoguent entre eux avec un protocole spécifique et mémorisent des informations d'état du flux. Ils agissent directement sur le flux, il l'interceptent et le traitent. Un prototype de démonstration a été implémenté sur la plate-forme active Tamanoir sur un réseau local. Les aspects fonctionnels ont été étudiés. On a montré la simplicité de programmation d'un tel service au-dessus de Tamanoir. Le traitement adaptatif était restreint à un écartement de paquets en excès. De nombreuses études théoriques et expérimentales plus approfondies restent à mener sur ce modèle.

En particulier, le problème de l'approche adaptative active réside dans la problématique du codage du flux et des performances. Nous avons montré dans le projet Netstre@mer que les protocoles RTP et RTCP offraient des solutions pour implanter une approche adaptative modulaires dans le réseau, mais soumises à un certain nombre de contraintes au niveau du codage de l'information. Dans videogateway, le transcodage est effectué entre des formats H323 et MPEG qui ont des étapes de traitement analogues (Transformé en Cosinus Discrète par exemple). Ceci requiert un codeur logiciel. Dans Netstre@mer nous avons effectué le traitement d'adaptation sur le débit en manipulant le taux d'image, la taille et la couleur d'un flux M-JPEG. Pour chaque type de flux, il faut prévoir un codage qui permettent l'adaptation au débit de manière relativement modulaire tout en conservant des performances en délai de traitement compatibles avec la dynamique du flux. Nous déployons le module d'adaptation de débit de Netstre@mer dans un réseau Tamanoir expérimental afin de mesurer les performances d'un service de transcodage vidéo dans Tamanoir.

Approche DiffServ Actif

La deuxième approche consiste à généraliser les mécanismes réseau de QoS-IP. C'est celle que nous adoptons dans le modèle DiffServ Actif. Une des limites que nous avons rencontré dans le modèle EDS est le problème du stockage de l'identificateur de classe dans l'entête IP et la difficulté de préserver le marquage effectué par la source. Le champ TOS étant déjà utilisé par le DSCP et n'étant pas garanti de bout en bout par la rfc 2474 [142], nous requérons un champ spécifique. Soit nous utilisons un champ optionnel, soit nous empruntons le champ " fragment offset " qui est quasiment devenu obsolète. Ce sont dans les deux cas des solutions qui ne sont pas très élégantes. L'entête ANEP 5.6 offre la possibilité de rajouter des champs. Pour pousser la réflexion sur les services différenciés, nous avons étudié l'extension de l'approche DiffServ classique par un " marquage " des flux au niveau de l'entête ANEP et non pas dans le champ DSCP de l'entête IP. Ainsi l'identificateur de classe peut-il être protégé, il n'est traité que dans les noeuds actifs et sa longueur non limitée permet une grande extensibilité du concept de classes de services différenciés. Un autre inconvénient de l'approche DiffServ classique est la limitation des traitements que l'on peut effectuer sur les flux. Nous avons montré qu'il est possible d'accélérer ou de détruire des paquets en agissant sur les mécanismes de base de gestion des files d'attente et d'ordonnancement des paquets. Ces traitements influent uniquement le débit, le délai de transmission et les pertes. Dans le modèle ADS, les concepts de traitement et de mémorisation de paquets sont étendus. Les buffers d'un noeud actif peuvent être reconfigurés dynamiquement, et un espace disque peut être alloué à un flux. Des flux massifs non contraints temporellement peuvent être stockés temporairement ou bien un flux temps-réel peut être ré-encoder pour mieux s'adapter aux caractéristiques du réseau. Les traitements potentiels qu'un routeur actif peut appliquer peuvent l'être sur un paquet complet ou bien uniquement sur l'entête ou sur la charge utile. Au niveau paquet, le traitement sera :

- un écartement,
- un ralentissement (lissage)
- un stockage temporaire,
- une duplication

Sur l'entête on effectue un marquage ou une modification des adresses (reroutage). Sur le payload, on applique une compression, un transcodage ou un encryptage. Le principe de DiffServ actif est donc de transposer les techniques de marquage, de classification, de mesure, d'ordonnancement et d'écartement au niveau de la couche active afin de permettre une plus grande flexibilité. Des traitements plus sophistiqués peuvent être imaginés. En utilisant le service **statesaver**, des états spécifiques au flux sont conservés et des traitements adaptés effectués. Ainsi, contrairement à l'approche DiffServ classique, on ne distingue plus des noeuds de frontière et des noeuds internes. Les fonctions de conditionnement, de classification ainsi que les PHBs sont activés dans n'importe quels routeurs actifs du chemin et seulement lorsque cela est nécessaire. DiffServ actif n'est pas incompatible avec l'approche DiffServ classique. Les paquets ADS peuvent être véhiculés sur un service IP Best effort classique ou bien IP-QoS tel Premium Service, Assured Service ou EDS. Comme les traitements et les états ne concernent pas chaque routeur et chaque paquets, les problèmes de passage à l'échelle rencontrés dans IntServ ne sont pas aussi critiques.

Pour montrer la pertinence de cette approche, nous avons fait une mise en oeuvre dans la plate forme Tamanoir. Dans notre modèle, nous avons fait l'hypothèse que seuls certains routeurs, le long du chemin offraient des capacités **actives**. Nous considérons que la qualité de service n'est pas requise dans le coeur de réseau, mais est nécessaire dans les réseaux d'accès ou aux points d'engorgement. Dans ces lieux, il faut faire passer en priorité les flux contraints en terme de délai et au contraire ralentir les flux à contrainte de fiabilité. On doit connaître l'importance relative d'un paquet, une information sur ce qu'il transporte (sémantique du flux) mais aussi éventuellement sur ses contraintes d'acheminement (délai de bout en bout), son histoire (délai déjà consommé) pour prendre une décision de traitement en cours de route et en fonction de l'état du réseau. Ces informations sont stockées dans l'entête ANEP. Dans ces premiers travaux, nous avons mis de côté les aspects performance en se focalisant sur le marquage et le traitement des paquets en fonction de certains critères. Dans notre modèle, c'est le premier noeud actif traversé qui marque le flux avec un ADSCP (Active DiffServ Code Point). Ce champ est inséré avant les données utiles transmises par l'application. La taille de ce champ est variable et il est possible de créer autant de ADSCP différents sans être limité par des problèmes de taille d'en-tête. Les noeuds suivants se basent sur cet ADSCP, sur l'état général du réseau et sur leur état local pour adopter le meilleur PHB (Per Hop Behavior) étendu.

Ces travaux se poursuivent, notamment dans le cadre des projets VTHD++ et e-toile et ont pour objectif d'approfondir les problèmes de performances d'une part et les problèmes de sémantique de l'autre.

5.4 Perspectives de la QoS active

L'apport de l'actif au niveau de la QoS ne pourra pas se mesurer directement en termes d'amélioration des performances réseau de base mais plutôt au niveau des performances de bout en bout et des services à valeur ajoutée rendus. A ce niveau, on peut parler de qualité de service logique. Nous travaillons en collaboration avec le groupe de Micah Beck de l'Université du Tennessee qui travaille sur une dissociation l'acheminement des données contextuelles des données conversationnelles d'une session de communication. Ainsi, pour améliorer la performance globale, on achemine le contexte, qui peut être volumineux, avant l'ouverture de la session. Pendant la session, seules les données interactives sont échangées. Les contraintes pesant sur le réseau au moment critique sont moins lourdes. C'est donc la valeur ajoutée et les fonctionnalités nouvelles offertes qui sont susceptibles d'améliorer la qualité perçue. Les limites actuelles de la technologie des réseaux actifs au niveau performance sont encore importantes. Le défi est d'obtenir des délais de bout en bout faibles et un coût des traitements réalisés sur le chemin de données en adéquation avec les besoins. D'après les résultats de Jean-patrick Gelas et Laurent Lefevre sur Tamanoir [97], la traversée nominale d'un routeur Linux Tamanoir impose un surcoût de l'ordre de 2 à 3 par rapport à un routeur Linux classique. Ainsi, un routeur capable de relayer des paquets classiques quasiment à la vitesse d'un lien 100Mb/s obtient des performances de 30Mb/s à 50Mb/s avec Tamanoir, ce qui reste encore bien faible mais bien supérieur aux performances obtenues avec des environnements classiques tels que ANTS (2Mb/s) dans un même contexte. Dans son travail de thèse Marc Herbert étudie le problème d'architecture de routeur intelligent et de localisation des traitements afin d'optimiser les performances. La localisation des traitements supplémentaires que l'on veut intégrer au réseau concerne une dimension verticale : niveau utilisateur, niveau

noyau ou niveau hardware (carte NIC) et une dimension horizontale : aux extrémités, aux frontières du réseau, dans les routeurs de coeur. Ceci nous a amené à réfléchir aux équipements chargés d'effectuer les traitements sur les paquets pour obtenir de bonnes performances ainsi qu'une grande extensibilité. Dans cette optique là, nous avons entrepris un travail de conception de routeurs programmables haute-performance basés sur la technologie cluster (voir section 6.4.2).

Faire de la QoS active implique de bien choisir le type de performance que l'on souhaite optimiser. Le gain envisageable se situe à priori plutôt au niveau de la dimension sémantique que de la dimension temporelle.

5.5 Conclusions

L'ensemble des travaux que nous avons mené sur EDS et ADS nous ont permis d'obtenir une validation de nos approches et de nos modèles sur le plan fonctionnel et au niveau de la couche réseau. Lorsque l'on étudie le problème de la Qualité de Service, le problème de la performance et du gain effectif obtenu par les flux de bout en bout est central. Les simulations de TCP sur EDS ou sur la différenciation proportionnelle nous ont enseigné qu'une couche réseau et une couche transport conçues de manière indépendantes et avec des objectifs différents ne pouvait interagir de manière vraiment efficace. Cela nous pousse à conclure que pour faire évoluer IP il faut faire évoluer en profondeur la couche transport et les mécanismes de contrôle de congestion d'Internet. Ce travail risque d'être très long, mais il paraît inévitable. Aussi Benjamin Gaidioz explore-t-il une voie nouvelle de transport adaptatif qui donne la capacité aux extrémités d'avoir une influence sur l'acheminement des paquets à travers le réseau. Il s'agit d'une forme primitive et simpliste d'approche active. Dans ces développements, nous ne visons pas une couche de transport universelle, mais seulement à montrer comment une couche de transport adaptatif peut interagir de manière optimale avec une couche IP à services différenciés. Par ailleurs, nous pensons que le déploiement, incontournable à moyen terme, de la technologie active aura aussi une influence sur l'évolution de la couche transport. C'est dans cette perspective que nous menons nos travaux de manière coordonnée et en étroite collaboration avec nos développeurs de l'environnement Tamanoir. Nous considérons que les propositions EDS et ADS s'intègrent et ne s'opposent pas aux autres solutions en cours de déploiement telles le sur-dimensionnement du coeur de réseau, la gestion active des files dans les routeurs, Premium Service ou les techniques adaptatives. L'ensemble de ces techniques sont à même de fournir des éléments de réponse à cette problématique complexe et ne doivent pas s'exclure les unes des autres. Internet nous a montré à maintes reprises sa capacité à absorber petit à petit les meilleures idées et les bonnes méthodes d'optimisation. Outre le travail conceptuel et théorique qui reste à mener sur nos modèles EDS et ADS, plusieurs autres questions demeurent encore sombres. Les outils de spécification des besoins hétérogènes et dynamiques des applications et le problème du déploiement de ces architectures nouvelles dans l'Internet sont les principales autres voies que je souhaite explorer. C'est le sens de ma très forte implication dans le domaine en pleine explosion des grilles de calcul et de données.

Troisième partie
Grilles de calcul

Chapitre 6

Réseaux et grilles de calcul

La problématique réseau des Grilles de calcul et de données devient un domaine d'application majeur des réseaux haut débit. C'est à mon avis un cadre favorable au développement de recherches fructueuses aussi bien sur les applications et les systèmes répartis que sur les réseaux. Mes responsabilités scientifiques dans deux projets de construction de plate-formes expérimentales : le projet Européen **DataGRID** et le projet RNTL français **e-toile**, ainsi que ma participation dans le projet Européen **DataTAG** et deux projets d'**ACI-GRID** me conduisent à coordonner des activités multiples dans la thématique réseaux et communications dans une grille. Dans le cadre de ce chapitre, je cerne les problématiques réseaux que nous avons identifiées, puis je me focalise sur les aspects liés aux performances de bout en bout des communications de la grille. Je développe d'abord l'architecture de mesure de performances et de surveillance du réseau que nous avons conçu et mettons en place dans les plate-formes expérimentales. Cette architecture est un outil préalable et indispensable au traitement des performances et de la qualité de service dans un environnement distribué. J'étudie ensuite les problèmes de transport haute performance sur lesquels nous travaillons avec Marc Herbert dans sa thèse de doctorat. Je présente finalement le concept de " grille active " que nous explorons et cherchons à promouvoir dans le cadre du projet e-toile.

6.1 Communication dans les grilles de calcul

6.1.1 Concept de grille

La grille de calcul est un concept quelque peu délicat à définir et qui évolue chaque jour. Il s'agit d'une collection complexe de ressources partagées (traitement, stockage, communication et information) dont les applications sont essentiellement le calcul scientifique intensif [88]. Cependant, le spectre des applications des grilles s'élargissant, il devient de plus en plus difficile d'identifier précisément ce que chacun sous-entend en employant le terme grille. Dans le rapport [41], la grille est identifiée à *des systèmes répartis haute-performance dont certains des constituants sont des calculateurs parallèles*. Ce sont les progrès dans la technologie réseau, notamment les réseaux gigabits, et les infrastructures de calcul qui ont rendu possible la construction de tels environnements distribués. Les grilles de calcul se distinguent des systèmes distribués conventionnels par le fait que la distance géographique entre les ressources peut être très large et que le réseau d'interconnexion est Internet. La dimension réseau a un impact important en particulier à cause de la variabilité des performances. Le concept classique de transparence des systèmes distribués en est donc ébranlé.

Comme l'objectif d'une grille est de fournir un accès cohérent et sûr de fonctionnement à un très grand nombre de ressources distribuées et partagées, [90], la grille doit fournir des mécanismes et des services pour la découverte, l'allocation et la surveillance des ressources et la sécurité. Le logiciel intermédiaire, intergiciel ou *middleware*, est le logiciel système qui masque les détails matériels de l'infrastructure répartie aux utilisateurs et qui leur en permet un accès sécurisé. Le *middleware* doit assurer trois fonctions principales : gérer des ressources, gérer des utilisateurs et permettre la communication entre les composants de la grille au travers d'un certain nombre de services. Par exemple, Globus [89] propose un service de gestion des ressources (GRAM), responsable de l'allocation des ressources et de la gestion des processus, un service sécurité (GSI) qui réalise l'authentification et le

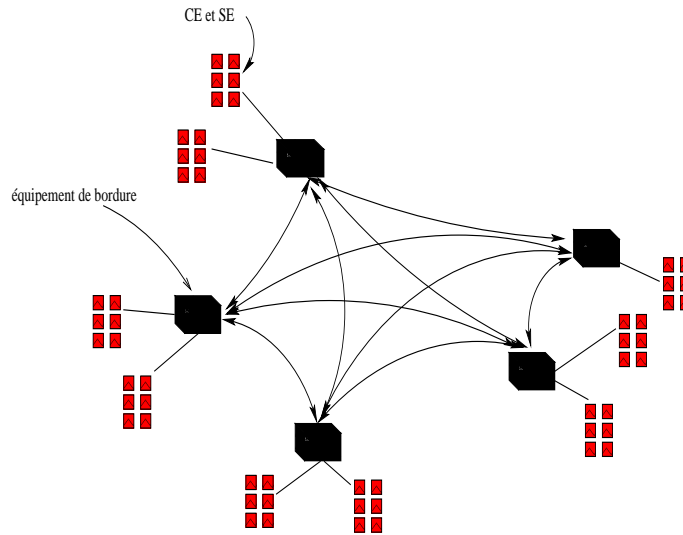


FIG. 6.1 – Vue logique d'une grille

contrôle d'accès, un service d'information (MDS) qui permet un accès distribué à l'information de structure et d'état et deux services de communications unicast et multicast (Nexus et GlobusI/O). D'autres services tels que le service de supervision des composants système (HBM), d'accès aux données distantes (GASS), ou de gestion des exécutables (GEM), ont été proposés. Ces services moins critiques et généraux ne sont pas systématiquement déployés sur les plate-formes expérimentales.

6.1.2 Problématique réseaux des grilles de calcul

Chaque instance de grille possède ses caractéristiques propres et est construite au dessus d'une infrastructure de communication particulière qui utilise les protocoles de l'Internet pour ce qui concerne les communications longue distance. La relation qui lie Internet au concept de grille de calcul est très forte et mérite toute notre attention. L'étude de la problématique réseau des grilles passe tout d'abord par une analyse des besoins de communication des applications et du *middleware* ainsi que des caractéristiques de l'interconnexion sous-jacente. Nous avons initié ce travail et défini une méthodologie d'analyse dans le cadre du projet DataGRID [28]. Il s'agit, par un processus itératif d'analyser les flux des applications et les liens existants puis de mesurer les performances afin de vérifier l'adéquation du dimensionnement des liens avec les contraintes.

Les besoins de communication des applications ou du *middleware* d'une grille sont très divers et les paradigmes de communication peuvent être très variés. Nous distinguons les flux de données des applications des flux de contrôle du *middleware*. Les premiers environnements tels que Globus sont basés sur la soumission de commandes distantes et le transferts de fichiers. D'autres environnements plus évolués sont bâtis sur l'invocation de méthodes d'objets ou de services distants. Certaines applications ou composants *middleware* requièrent des mécanismes de passage de messages tels que MPI ou bien des communication multicast fiable ou non fiable. Souvent, plusieurs modes de communications doivent être simultanément offerts. Une des caractéristiques importantes des transferts est que le spectre des volumes échangés couvre au moins 6 ordres de magnitude, de quelques octets à plusieurs gigaoctets.

Parmi les aspects réseau, nous avons identifié les problèmes de la *performance* et de la *sécurité* nous semblent être les deux verrous majeurs des réseaux de grille. En ce qui concerne la sécurité, de nombreux aspects plus ou moins liés au réseau sont critiques : l'authentification, l'autorisation, la comptabilité (accounting) (AAA) à la frontière du réseau public, la confidentialité, mais aussi la continuité et la disponibilité des services, c'est à dire la prévention des dénis de services (DoS). Dans [87], des solutions relatives à l'authentification et au contrôle d'accès basé sur une infrastructure à clés publiques (PKI) sont proposées. Nos travaux sur la sécurité sont encore dans un état embryonnaire tant au niveau du projet DataGRID que de celui des travaux théoriques de Julien Laganier en stage

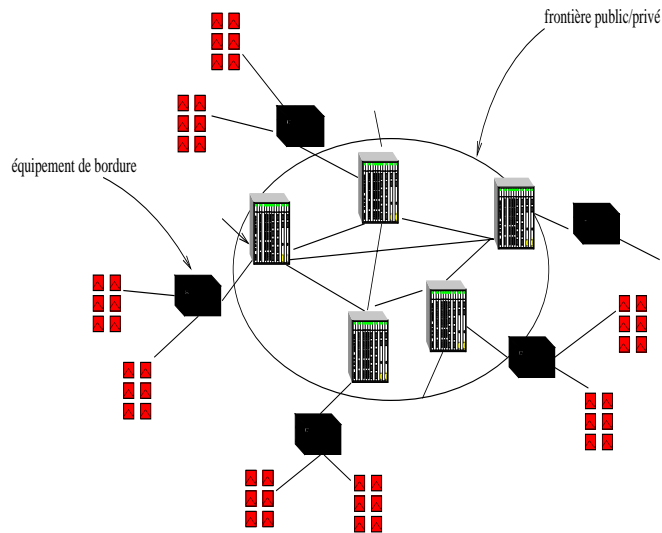


FIG. 6.2 – Vue IP de la grille

de DEA et ceux autour du concept de réseau privé virtuel de grille que nous avons mené avec Franck Bonnassieux [27]. Je ne développerai pas ces dimensions dans ce document et je m'attacherai uniquement aux aspects relatifs aux performances.

Les performances du réseau

Un des défis auxquels doit faire face la communauté grille est de concilier deux opposés : d'un côté un réseau longue distance ou Internet qui exhibe une extrême hétérogénéité de performance et de fiabilité et de l'autre des mouvements de données qui peuvent être des déterminants critiques des performances des applications. Ainsi les applications très couplées peuvent être très sensibles aux caractéristiques de communication [70]. En particulier, la charge et la disponibilité des liens réseau utilisés pendant les transferts de données peuvent être difficiles à prédire, ce qui peut handicaper l'ordonnanceur de travaux en charge de déterminer l'enchaînement efficace des travaux. Ainsi, bien que les grilles offrent une performance potentiellement considérable par l'agrégation de ressources, la performance d'exécution des applications peut être délicate à obtenir en pratique à cause de l'interconnexion réseau. Il est nécessaire d'identifier le degré de couplage et de parallélisme compatible avec une instance particulière de grille. Pour cela des modèles de performance calcul/communication/stockage adéquats doivent être définis. Je citerai les travaux de [213] et de Frederic Desprez [51]. Vu du côté réseau, le problème des performance peut être étudiée selon deux angles différents mais complémentaires.

- mettre en place un système de mesure et de surveillance de performances adéquat pour caractériser les liens qui relient l'ensemble des sites. De là, il faut fournir des fonctions de calcul du coût du réseau permettant aux niveaux d'abstraction supérieurs de développer des algorithmes d'optimisation et d'adaptation.
- faire évoluer l'infrastructure de communication et ses services pour servir au mieux les besoins spécifiques de la grille.

Dans DataGRID et DataTAG, nous avons entrepris des études dans ces deux directions. J'ai par ailleurs initié un travail de redéfinition du concept Network Element pour modéliser la ressource réseau dans une grille [160]. Ce concept souvent utilisé dans la littérature reste en effet un terme flou comparativement aux concepts de *Computing Element* et de *Storage Element*, dont on comprend, à peu près bien les rôles au sein d'une grille.

6.1.3 Modélisation du réseau de la grille : network element

L'infrastructure de communication de la Grille est une interconnexion entre des réseaux locaux privés et l'infrastructure Internet. Les domaines publics et privés sont généralement délimités par des routeurs d'accès et des

étages de pare-feu. Pour fournir aux utilisateurs et au *middleware* une vue abstraite et homogène de cet ensemble complexe de ressources interconnectées, le nuage de réseau peut être représenté au niveau logique par un graphe complet reliant tous les éléments physiques de la grille. Ainsi, le protocole IP, le service DNS et les passerelles offrent-ils la possibilité d'adresser et donc d'accéder virtuellement à tout ordinateur connecté. Je propose de modéliser une infrastructure de grille par un quadruplet :

$$\mathcal{G} = (\mathcal{D}, \mathcal{C}, \mathcal{S}, \mathcal{N})$$

où $\mathcal{D} = \{D_1, D_2, \dots, D_n\}$ est un ensemble de sites (domaines), $\mathcal{C} = \{CE_1, CE_2, \dots, CE_m\}$ est un ensemble de CE (computing elements), $\mathcal{S} = \{SE_1, SE_2, \dots, SE_p\}$ est un ensemble de SE (storage elements), $\mathcal{N} = \{NE_1, NE_2, \dots, NE_n\}$ est un ensemble de NE (network elements).

Il existe une bijection entre \mathcal{D} , l'ensemble des sites et \mathcal{N} , l'ensemble des **Network Elements**. En effet, un NE est une ressource partagée par l'ensemble des ressources de calcul et de stockage d'un même site. Un NE est en charge de la fonction de communication sur une interconnexion de réseaux longue distance. Un NE regroupe un ensemble de liens IP orientés auxquels on peut associer une classe de service réseau mais aussi un niveau de sécurité. Un **Network Element** est donc défini de la manière suivante : $NE = L_1, L_2, \dots, L_x$ un ensemble de liens L_i avec $1 \geq i \geq n$ où n est le nombre de sites de la grille. Chaque lien possède des attributs tels que la destination et la classe de service ou le niveau de sécurité. Un lien est caractérisé par des valeurs mesurées de métriques délai, débit, taux de perte. La ressource **Network Element** doit être quantifiable, optimisée et d'un usage facile, sûr et flexible. L'objectif d'une architecture de mesure de performance de réseau de grille est de quantifier chaque arc de ce graphe avec des métriques simples et significatives pour les niveaux supérieurs. Un allocateur de ressources de grille s'appuiera sur les valeurs fournies par le système de mesure de performances pour calculer l'ordonnancement optimal et choisir le bon lien, un outil de gestion des politiques de sécurité et d'approvisionnement de bande passante gèrera les attributs Cos ou sécurité des liens du NE. Le concept de NE est en cours de discussion au sein de différents groupes de travail auxquels je participe (groupe réseau de DataGRID et GGF).

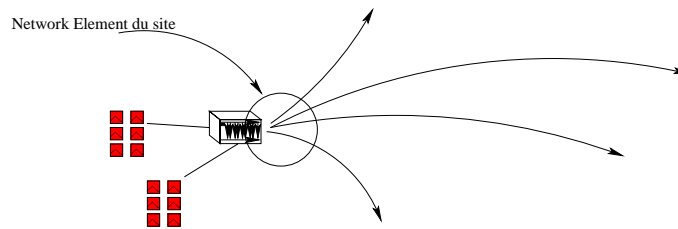


FIG. 6.3 – Modèle logique d'un Network Element

6.2 Mesure des performances réseau de la grille

La mesure et la surveillance des performances des liens et du trafic est une fonctionnalité indispensable, si non la première, à mettre en place dans une infrastructure réseau de grille. Elle permet de caractériser le *nuage d'interconnexion*, modélisé par les **Network Elements** et doit s'adapter à toutes les évolutions du réseau et des services.

6.2.1 Architecture de mesure de performance

Problématique générale

Dans un environnement de grille, les données de mesure sont requises pour déterminer un problème de performance ou pour mieux ajuster le système afin d'obtenir de meilleures performances. La nature très distribuée et étendue de la grille requiert le développement de méthodes appropriées pour le stockage à court et moyen terme

de ces informations et le développement de moyens effectifs de présentation des données multivariées. De nombreux systèmes de surveillance et de mesure de performance ont été proposés pour les applications parallèles et les systèmes distribués. Dans [6] et [77] un certain nombre de ces systèmes ont été analysés. Les objectifs de ces systèmes sont souvent très différents : certains visent à éclairer le programmeur ou l'utilisateur sur les performances d'une application, d'autres s'adressent à l'administrateur d'un système et surveillent la disponibilité des ressources. De plus, l'infrastructure de communication des systèmes surveillés est très hétérogène. Dans certains cas, il s'agit d'un réseau SAN bien contrôlé, dans d'autres un réseau local Ethernet privé surdimensionné ou bien le réseau est Internet aux comportements imprévisibles. Les outils diffèrent aussi au niveau des résultats qu'ils produisent : certains peuplent une base de données d'information, d'autres génèrent des pages Web enfin certains activent des processus qui exécutent des commandes en réponse à certains états du système (dépassement de seuils...) Ces outils sont généralement fermés et ne peuvent accueillir des outils externes. Netlogger [32], Gloperf [126] sont des exemples de tels systèmes de supervision fermés. Pour fournir à la communauté grille un cadre ouvert et inciter les développeur à construire des outils plus ouverts, le groupe *Grid Performance* du Global Grid Forum a proposé l'architecture GMA : Grid Monitoring Architecture [15] Les systèmes tels que Autopilot [183], NWS [217] et R-GMA [77] suivent ce modèle architectural, mais soit ils n'intègrent pas de composants de mesures de performance réseau (R-GMA) soit les outils ne sont pas assez précis (NWS).

Objectifs et architecture

Mesurer les performances du réseau d'une grille vise deux objectifs :

- fournir des informations de type " coût du réseau" aux composants d'optimisation du *middleware* de la Grille (courtier de ressource ou gestionnaire des replicats). Pour ce type d'usage, la détermination et la publication des métriques significatives est de première importance. La prédiction des performances à court terme est aussi critique.
- fournir des statistiques de performances du point de vue externe pour identifier tout goulet d'étranglement, des points de vulnérabilité ou pour réguler et approvisionner la qualité de service. Des mécanismes de détection et de recouvrement de panne réseau sont aussi requis.

L'architecture d'un système de mesure de performances du réseau d'une grille doit être, comme tout service de grille : simple, extensible, robuste, modulaire, sûre et facile à utiliser. La simplicité et le passage à l'échelle sont de première importance car le nombre d'entités et de liens peut être considérable. La robustesse est requise car les ressources sont très dynamiques et peuvent changer au cours du temps. Pour permettre l'intégration de nouveaux composants (capteur, actuateurs) l'architecture doit être modulaire.

Dans le cadre du projet DataGRID, nous avons conçu une architecture conforme à l'architecture GMA et basée sur quatre types de composants qui sont : les capteurs de mesure, les processus de traitement des données brutes et d'extraction des statistiques, le système d'archivage des données historiques et des données traitées, les composants de sortie qui permettent d'extraire les données de mesure afin de les afficher, les exploiter dans le *middleware* ou activer des processus de contrôle [204],[?]. Nous avons exploré un grand nombre d'outils de mesure de performance dédiés à Internet et nous avons déployé notre propre architecture de mesure sur une plate-forme expérimentale européenne dédiée. Nous avons analysé les métriques, défini le schéma logique de description du **network element** dans le système d'information de la grille (MDS), développé les interfaces pour la publication des métriques dans cette base de données et développé des interfaces de visualisation.

Dans [?], [161] nous avons montré que la conception d'une architecture de supervision posait un certain nombre de questions au niveau du

- choix des métriques
- choix de la méthodologie et des outils de mesure
- choix de la localisation et du mode de déploiement des capteurs
- choix de la coordination des mesures

Choix des métriques, des méthodes et des outils

Les métriques classiques de délai et taux d'erreur étudiées au chapitre 2 sont importantes dans le cas des grilles de calcul. La métrique débit est souvent la plus importante dans le cas d'une grille de calcul ou de données, la connaissance du débit utile d'une connexion et principalement le débit du goulet d'étranglement d'un lien est

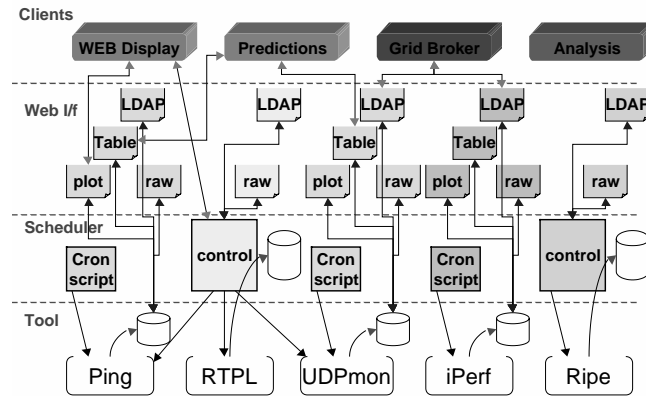


FIG. 6.4 – Architecture de supervision de grille

donc nécessaire. Nous avons montré que cette métrique empirique était difficile à obtenir. Le gestionnaire de réplication de EDGI, le *middleware* du projet DataGRID, requiert une fonction f de coût du réseau, telle que pour un volume V de données à écouler entre deux points i et j , f calcule le temps de transfert T requis. Dans la formule de calcul d'une valeur approximative du coût, nous utilisons la bande passante du goulet d'étranglement B pour calculer T tel que $T = V/B_{i,j}$

Pour mesurer les performances d'un réseau de grille, on s'appuie sur des outils spécialisés, dédiés et très contrôlés ou bien sur l'instrumentation des applications. Il existe, comme pour la mesure classique des performances réseau, deux types de méthodes : les méthodes actives et les méthodes passives.

Le principe d'une méthode active est d'injecter du trafic dans le réseau de manière contrôlée et d'enregistrer les paquets retournés (ping, traceroute...) Les mesures actives de performances sont utilisées traditionnellement pour diagnostiquer des problèmes réseau, pour analyser le comportement du trafic sur des liens donnés, pour analyser les performances d'un ISP particulier

Une méthode passive consiste à observer et à analyser les paquets reçus sur un système terminal ou à collecter des informations de trafic en un point du réseau tel un routeur, un commutateur ou un équipement dédié. Les résultats collectés permettent d'étudier la composition du trafic par application, la distribution de la taille des paquets, les intervalles de temps entre les arrivées de paquets, la performance et la longueur des liens, la matrice des flux. Ils permettent aussi d'identifier et de traquer les attaques de sécurité dans une infrastructure, de vérifier l'efficacité des algorithmes de congestion, d'établir si l'augmentation du trafic est due à l'augmentation du nombre d'utilisateurs ou au volume par usager...

Une revue et une évaluation des outils disponibles pour la mesure des performances de l'interconnexion a été établie et rapportée dans le document [203]. Nous avons sélectionné les outils Pinger pour la mesure du délai RTT et du taux de perte ainsi que Iperf [111] pour la mesure du débit utile TCP. Un outil spécifique de mesure active du débit UDP qui donne aussi le taux de perte et la variance de délai, UDPmon, a été développé dans le cadre du projet. La méthode utilisée est similaire à celle de Bolot [22] et est basée sur le fait que si deux paquets voyagent ensemble comme une paire dans le goulet, sans aucun paquet entre eux, l'intervalle entre les paquets sera proportionnel au temps requis par le routeur le moins puissant (goulet d'étranglement) pour traiter le second paquet de la paire. Cet outil est par ailleurs très utile pour calculer l'espace mémoire total alloué aux files d'attente sur un lien vide et pour mesurer l'état de congestion d'un chemin. Dans les réseaux très haut-débit, les méthodes actives classiques pour le calcul de la bande passante ou du débit du goulet d'étranglement d'un lien peuvent être très intrusives et impossibles. Dans Iperf, il est en effet nécessaire de générer un flux qui remplisse complètement le tuyau pendant un temps suffisamment long. Sur les liens à 1Gb/s, il faut que le capteur soit à même de générer un flux d'au moins 1Gb/s pendant x secondes. Ceci requiert par exemple une machine Linux dédiée avec un processeur à au moins 1GHz et un bus PCI à 66MHz et 64 bits. On obtient une sonde résultante, dont l'unique travail est de générer du "bruit" qui est plus perfectionnée que beaucoup de stations émettrices. Nos partenaires hollandais du projet DataGRID travaillent sur la conception de sondes plus pertinentes dans ce contexte. On peut

aussi citer les travaux de Laurent Toutain à l'ENST Bretagne avec la sonde Saturne pour le projet VTHD.

Déploiement des capteurs

Les mesures actives ne peuvent pas être réalisées entre tout couple d'unité de calcul (CE) ou de stockage (SE) de la grille. Un graphe complet entre tout élément de la grille n'est pas envisageable du simple point de vue de l'extensibilité. Par conséquent, il est plus pertinent d'effectuer les mesures entre les couples de sites, c'est à dire sur les liens des NE. Cela donne une bonne approximation des caractéristiques du nuage Internet vu par les unités de ressources. Le lien entre deux éléments de calcul $CE_i - CE_j$ des sites S_i et S_j représenté par les capteurs C_i et C_j , peut être approximé par la composition :

$$f(CE_i, CE_j) \leq f(CE_i, C_i) + f(C_i, C_j) + f(C_j, CE_j) \quad (6.1)$$

Si f est la fonction délai d , on choisira une borne supérieure de délai local D_{max} . Ainsi

$$d(CE_x, CE_j) \leq d(CE_x, CE_k) \iff d(C_x, C_j) \leq d(C_x, C_k) \text{ avec } (j, k) \in \{1, ..n^2\} \text{ et } j \neq i \text{ et } j \neq i \quad (6.2)$$

Pour le taux de pertes, comme la valeur locale tend vers 0, seule la valeur sur le lien distant sera considérée dans l'approximation. Pour le débit utile, il faut vérifier que le goulet d'étranglement ne se situe pas sur le réseau local. Pour cela, il faut tester les liaisons locales et les comparer à la liaison distante.

$$b(CE_i, CE_j) \leq \min(b(C_i, C_j), b(CE_i, C_i), b(C_j, CE_j)) \quad (6.3)$$

Les mesures passives réalisées à partir des journaux (logs) des applications instrumentées (gridFTP avec Netlogger [86]) reflètent les performances entre les systèmes d'extrémité : les CE et les SE. Grâce à une combinaison des mesures passives et actives, l'approximation du coût du réseau peut être améliorée. Le déploiement dynamique des capteurs de mesure dans la grille, leur localisation et le calcul de la fonction coût du réseau est un problème encore largement ouvert et que nous étudions intensivement dans les projets DataGRID et E-Toile.

6.2.2 PCP et l'ordonnancement des mesures

Un certain nombre de problèmes apparaissent lors du déploiement d'une architecture de mesure de performances réseaux sur un plate-forme grille. Par exemple, l'activation et la gestion d'un nombre considérable de capteurs est complexe, la coordination des mesures actives doit être étudiée soigneusement. Deux stratégies sont possibles. La stratégie optimiste laisse le système simple et robuste en estimant que la probabilité de collision entre les mesures reste relativement faible, si par exemple elles sont espacées dans le temps. Mais on peut aussi adopter une stratégie pessimiste en ne négligeant pas le fait que la probabilité de collision entre tests et que l'effet du trafic de mesure croit de manière quadratique avec le nombre de capteurs. Ce phénomène sera encore plus important dans des structures de grille de type hiérarchique telle que la plate-forme DataGrid organisée en arborescence à multiples tiers, calquée sur la structure des applications de physique des hautes énergies. L'outil RTPL (Remote Throughput Ping Load) [150] s'appuie sur l'approche optimiste pour tester de manière périodique et centralisée les performances entre un ensemble de sites Internet. Dans l'infrastructure NIMI [112] des agents spéciaux sont responsable de la configuration des capteurs et de la coordination des tests. Avec Robert Harakaly, nous avons proposé un protocole PCP de coordination des tests qui est basé sur un algorithme distribué selon une topologie en anneau et un passage de jeton de synchronisation [105] comme illustré à figure 6.6. Cet algorithme permet de gérer avec précision la périodicité des mesures, ce qui est fondamental si on veut exploiter les séries de mesures en entrée d'un modèle de prédiction (voir figure 6.7). Cet outil est en cours de déploiement dans la grille DataGRID.

6.2.3 MapCenter et la visualisation de l'état de la grille

Avec Franck Bonnassieux, nous avons conçu et développé un outil original et flexible de visualisation de l'état de la grille qui permet aux administrateurs et aux utilisateurs d'organisations virtuelles de surveiller les éléments et les services de la grille selon différentes vues logiques [26]. La figure 6.8 donne le positionnement de MapCenter par rapport aux différentes entités d'une grille.

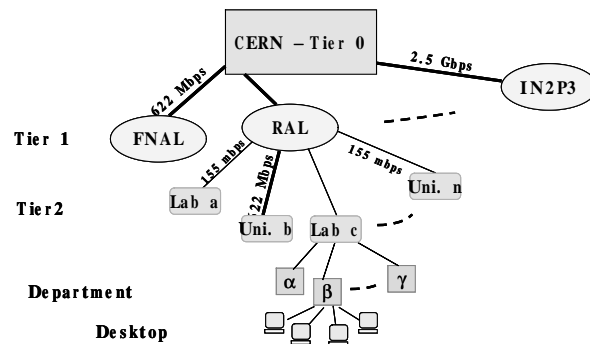


FIG. 6.5 – Structure arborescente de DataGRID
 MONARC report: <http://home.cern.ch/~barone/monarc/RCArchitecture.html>

Ainsi un responsable d'application préfère visualiser les équipements et les processus relatifs à son organisation virtuelle, un responsable administratif préfère vérifier l'usage des ressources de son institution, un responsable de site souhaite visualiser le détail des ressources locales ou un historique de l'état des services.

Le modèle de présentation est basé sur 4 entités :

- **Objet** : l'objet est l'élément de base ; en générale il modélise une machine ou un ensemble de services qui sont examinés à intervalles réguliers.
- **Symbole** : Un symbole est la représentation visuelle d'un objet.
- **Vue** : une vue contient des sous-vues, des symboles et des liens. Ces vues permettent la construction de représentations hiérarchiques.
- **Lien** : un lien est l'abstraction d'une interconnexion logique entre vues.

Une représentation logique de l'arborescence des composants est donnée dans la figurefig :model suivante.

MapCenter est utilisé quotidiennement dans le cadre du projet DataGRID et est nous travaillons à son intégration dans la suite Globus.

6.2.4 Prédiction des performances réseau : amélioration de NWS

S'il est important de caractériser les performances du réseau et de surveiller l'état des services, il peut être encore plus utile, pour un ordonnanceur, de pouvoir prévoir le comportement du réseau dans un avenir proche.

Dans le domaine de la prédiction des performances d'une grille de calcul, la plupart des travaux se réfèrent au système NWS (Network Weather Service) [218]. Un tel service de prédiction utilise des données de mesures actives ou passives en entrée d'un modèle de prédiction. La technique utilisée dans NWS s'appuie sur les performances passées de l'application obtenue sur le système global plutôt que sur une construction des prédictions au niveau de chaque composant. Dans [70], une méthode originale de prédiction de performances réseau est introduite pour déterminer le temps de transfert des fichiers. Cette méthode appelée AQdRM : Adaptive Regression modeling, s'appuie à la fois sur NWS et sur Netlogger. Dans [210] pour résoudre le problème de la sélection de réplicat, les auteurs proposent d'instrumenter GridFTP, pour récupérer les informations de performance de tous les transferts de fichiers, puis d'injecter ces informations dans un outil de prédiction. Ils comparent les résultats de mesures avec ceux obtenus avec l'outil NWS et observent une différence importante en faveur de GridFTP.

Nous avons réalisé des expérimentations avec l'outil NWS entre un certain nombre de sites européens) [173]. Nous avons évalué l'erreur de prédiction à environ 5% sur les réseaux très chargés et jusqu'à 20% sur les réseaux bien approvisionnés. Selon nos analyses, les modèles statistiques de prédiction de NWS ne s'appliquent pas bien sur des réseaux faiblement chargés car le trafic ne peut pas être bien décrit par un simple modèle statistique. Dans un réseau encombré au contraire, les modèles statistiques sont beaucoup plus pertinents. L'étude des modèles

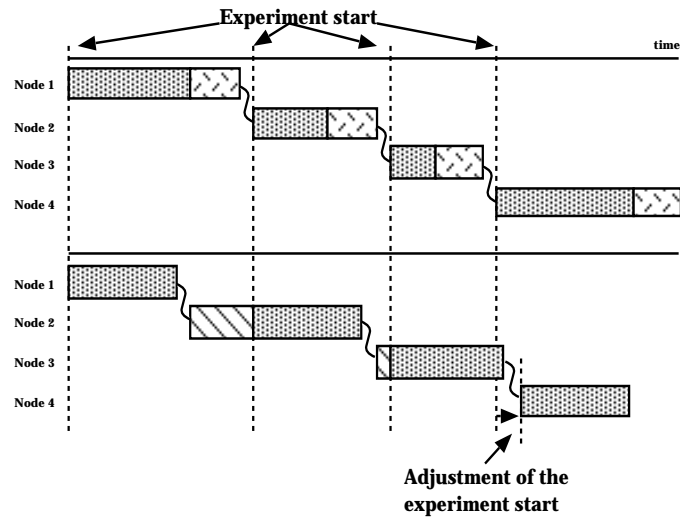


FIG. 6.6 – Passage du jeton et ajustement de la période de rotation

de prédiction dynamiquement choisis montre en effet une extrême instabilité dans le cas des réseaux faiblement chargés et au contraire un comportement beaucoup plus stable dans le réseau très chargé. Nous avons par ailleurs comparé les résultats des mesures de débit TCP produits par NWS et ceux d'un outil très populaire dans Internet pour mesurer le BTC (Bulk Transfer Capacité) Iperf [111]. Nous avons aussi constaté une importante différence, que nous expliquons en partie par l'influence du **slow start** de TCP dans les tests fréquents et de courte durée de NWS (300 ms sur des connexions à 2Mbps). Selon nous les conclusions sur l'inadéquation de NWS à la prédiction des performances réseau serait dû à ce défaut de la méthode de mesure utilisée plus qu'au module de prédiction qui est relativement correct pour les réseaux chargés. Mais cette méthode de mesure a le mérite de n'être pas intrusive et de générer des séries temporelles avec une périodicité précise. Nous avons donc proposé un modèle de calcul permettant de rendre la mesure NWS plus réaliste tout en demeurant beaucoup moins intrusive que les méthodes classiques de calcul du BTC telles que Iperf ou Netperf. Ce modèle est présenté dans le document d'annexes au chapitre 7.

6.3 Optimisation des performances des communications

Une infrastructure de mesure permet d'analyser des problèmes de performances réseau. En analysant les statistiques produites, on est ainsi à même de déceler des problèmes d'approvisionnement à certain endroit du réseau de la grille. Une évolution des liens ou des services réseau peut donc s'avérer nécessaire. Selon cette direction, plusieurs niveaux sont à considérer. Le dimensionnement des liens réseaux concerne les couches 1 et 2, l'utilisation de services différenciés de niveau IP touche la couche 3, l'optimisation des services de transport concerne le niveau 4, mais aussi un ajustement fin des paramètres au niveau application et du système. Je me suis principalement concentrée sur ces problématiques aux niveaux réseau et transport.

6.3.1 Grille et QoS réseau

Nos travaux sur la qualité de service réseau sont centrés sur l'étude d'une méthodologie de définition des besoins des applications puis une expérimentation des techniques de différenciation. Ce cadre doit nous permettre de déterminer

- quelles applications/logiciels peuvent bénéficier d'un traitement prioritaire des paquets ?
- s'il faut-il protéger certains trafics de transferts très volumineux ?
- quels types de protocoles réseau/application sont requis par les applications ?

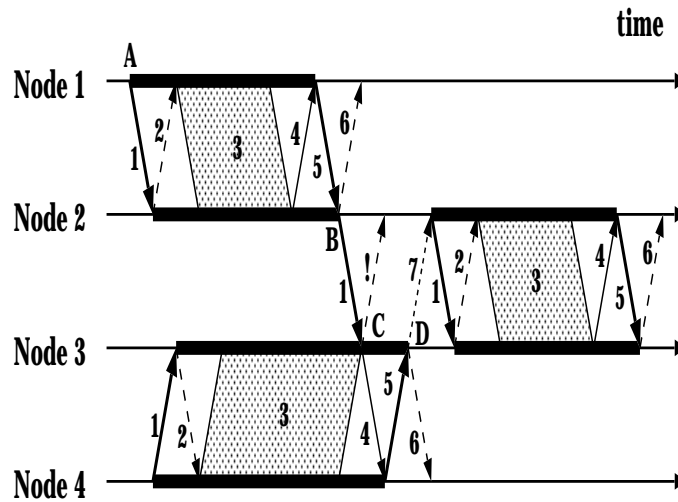


FIG. 6.7 – Periodicite du protocole de clique

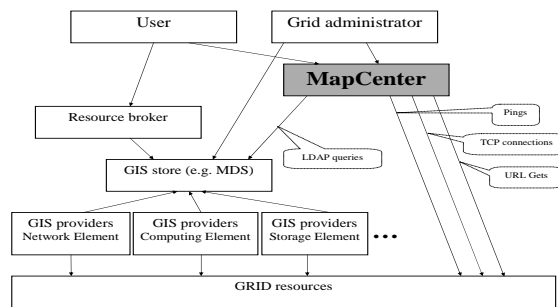


FIG. 6.8 – MapCenter overview

- quelle est la fréquence des sessions réseau et la sensibilité au trafic arrière plan
- quelle est la quantité et nature des données transportées dans la grille
- si l'application est interactive, quels sont les temps de réponse attendus ?

Dans le cadre de notre accord de collaboration DataGRID-GEANT [165] et en collaboration avec le projet IST SEQUIN, j'ai initié un travail expérimental de test d'un service Premium pilote pour la grille (c'est à dire de bout en bout). Pour cela, j'ai spécifié avec Johan Montagnat du laboratoire Creatis (INSA-Lyon) une application grille type dans le domaine de l'informatique médicale. Nous travaillons aussi sur ces aspects avec Christophe Blanchet dans le cadre d'un projet de l'ACI-Grid (Grips) pour la gridification d'algorithmes de bio-informatique sur la plate-forme E-Toile/VTHD. Il s'agit dans les deux cas de montrer l'apport d'un service Premium de bout en bout pour des applications interactives parallélisable à fort besoin de calculs et de transfert de données. Nous voulons aussi comparer comment différents modèles de calcul/communication peuvent justifier ou non l'emploi d'un service garanti. Ces travaux en sont à leur balbutiements, aussi, je ne les développerai pas ici. Par ailleurs avec Benjamin Gaidioz et Mathieu Goutelle nous étudions quels sont les apports et les contraintes des modèles proportionnels, EDS et scavenger pour les flux grille dans le cadre du projet DataTAG.

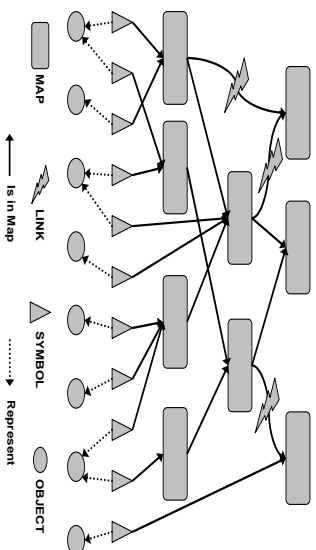


FIG. 6.9 – Modèle relationnel entre les quatre entités de la couche présentation

6.3.2 Transport haute performance

De nombreux composants du middleware tels que l'authentification, la réplication de bases de données ou l'échange de jobs et de données d'entrée-sortie, requiert des transmissions faibles et volumineuses dans une grille. Les applications de DataGRID vont par exemple produire d'importants flux de données, qui, selon nos estimations seront de l'ordre de 200Mbps à 400Mb/s en continu. La taille des fichiers à transférer peut aller de quelques mégaoctets à quelques dizaines de gigaoctets. Ces valeurs seront amenées à croître dans les années futures.

La première partie de nos travaux sur le transport haut débit consiste en l'étude des limites et l'optimisation de TCP pour des accès WAN longue distance à 1Gbps. Pour cela nous développons un émulateur de réseau WAN, textiGigWAN, à 1Gb/s puisque l'émulateur Nistnet [149] ne supporte que du FastEthernet. Nous redéployerons nos batteries de tests sur le réseau VTHD lorsque la connexion lyonnaise sera en production. L'étude de l'optimisation du transport sur une couche IP-QOS et IP/EDS, présentée au chapitre précédent et que je même avec Benjamin Gaudioz dans la deuxième partie de sa thèse vient en complément de ces travaux. Une troisième partie est relative à l'étude et à la mise en oeuvre de nouveaux protocoles de transport optimisés et des possibilités de fragmentation des connexions TCP et leur traitement dans les équipements de bordures de type routeur-cluster intelligent que nous développons dans la thèse de Marc Herbert.

Un protocole de transport fiable et qui utilise de manière efficace les ressources réseaux est un élément critique de la grille. Le seul protocole candidat actuellement est TCP. Cependant, il a été montré qu'il n'est pas très efficace pour des transferts de données massifs (plusieurs gigaoctets) sur des connexions à produit débit-délat élevé [127], [99], [154]. TCP a été amélioré pour les haut débits dans la rfc 1311 [7] pour des réseaux offrant des débits allant jusqu'à 45Mb/s, ce qui est bien inférieur à ce que l'on peut obtenir de nos jours. Le service fournit par TCP vise à offrir un taux de pertes résiduelles nul. Les performances de TCP ne dépendent pas seulement du débit de transfert mais du produit délat-débit. Ce produit correspond à la quantité de données qui remplit le tuyau, c'est la quantité de tampons mémoire chez l'émetteur et le récepteur pour obtenir le débit maximum sur une connexion TCP, i.e. la quantité de données non acquittées que TCP doit gérer pour maintenir le tuyau plein. Les problèmes de performances de TCP arrivent lorsque ce produit est trop important.

Selon la caractérisation analytique du débit de TCP issue des travaux de Padhye [148] le débit utile r_{tcp} d'une connexion TCP dans sa phase d'état stable est :

$$r_{tcp} = \min\left(\frac{r_{win}}{r_{th}}, \frac{M}{r_{th}\sqrt{l}}\right)$$

où r_{win} est la taille maximale annoncée par le récepteur, r_{th} le délat d'aller-retour, M la taille des segments de données et l le taux de pertes. Selon cette équation, le débit utile de TCP est dépendant du taux de perte l , du délat r_{th} , de la taille de la fenêtre de réception r_{win} . Ainsi, pour améliorer le débit d'une connexion TCP, il faut soit :

- Diminuer le taux de pertes. C'est pour cela que les mécanismes de gestion active de file d'attente (RED, WRED) ont été développés dans les routeurs.
- Diminuer le délat. Diminuer les attentes dans les files des routeurs en sur dimensionnant les réseaux ou en offrant des garanties temporelles strictes agit ainsi directement sur la partie variable du délat de bout en bout.

- Augmenter la taille de la fenêtre de réception (chez le récepteur), en augmentant l'espace mémoire par exemple.

Dans le contexte TCP, il existe une importante connaissance de ce problème et de nombreux mécanismes d'ajustement automatique ou non (autotuning) ont été décrits [192] et certains intégrés dans les piles protocolaire (telles celles de FreeBSD et de Linux).

En fait l'adaptation à faire n'est pas identique sur un LAN et sur un WAN. En particulier à cause du délai rtt . Sur un LAN où le délai rtt et le taux de perte l sont très faibles, le débit utile r_{tcp} est généralement très bon. Sur un WAN, où le rtt est bien plus important, il faut considérablement augmenter la taille de la fenêtre de réception. Mais augmenter la taille des fenêtres fait simultanément croître la probabilité d'erreur. Si le taux d'erreur est très important, il aura un impact encore plus important si le rtt est long. Il faut donc vraiment éviter les pertes. Les algorithmes de Retransmit and Fast Recovery algorithms de Jacobson [80], [198] permettent de récupérer une erreur par fenêtre sans purger le *tuyau*. Par ailleurs, l'option SACK, Selective acknowledgments est indispensable pour les longs et larges *tuyau* (LFN : long fat network ou éléphant !). Mais il faut savoir qu'elle est peu performante pour les régimes non LFN. Pour obtenir des performances TCP correctes avec une connexion sur un réseau de latence supérieure à 10ms, il faut obligatoirement disposer d'une mémoire d'au moins 1Go. On peut alors escompter un débit de l'ordre de 500Mb/s. Les dernières expériences dont j'ai eu les résultats sur un lien transatlantique entre Amsterdam et Chicago, en Gigabit sur WDM, avec apparemment un goulet d'étranglement de 622Mb/s dans un des équipements intermédiaires, révélaient un débit plafonné à 80Mb/s pour une configuration d'extrémité optimale !

Deux autres approches pour augmenter le débit utile de bout en bout sont souvent utilisées dans les grilles [104]. L'une utilise UDP comme protocole de transport de base. Le recouvrement d'erreur ou de perte ainsi que le contrôle de congestion revient à l'application ou doit être intégré dans le nouveau protocole de transport. L'autre approche consiste à ouvrir plusieurs connexions TCP en parallèle puis à répartir les données d'une manière similaire à ce que l'on fait dans les disques RAID. Ces techniques mettent en échec l'algorithme d'évitement de congestion de TCP et sont donc mal reçues par la communauté Internet [83]. Face à ces limites importantes de TCP, la communauté s'active autour de la définition de nouveaux protocoles de transport [80].

6.4 La grille active

6.4.1 Convergence des problématiques grilles et réseaux actifs

Aujourd'hui, le développement des grilles s'avère limité non par les capacités des équipements matériels eux-mêmes, mais par les abstractions logicielles et les services. D'un côté les outils réseaux classiques se sont focalisés sur la communication et non sur le calcul et de l'autre les systèmes distribués ne se sont pas orientés vers la performance et se sont basés sur le modèle client-serveur qui n'est pas très adapté au calcul intensif. Le foisonnement de travaux autour du *middleware* de grille témoigne d'une activité intense autour des abstractions et des services. Comme on assiste à une mise en place de multiples grilles très diverses quant à leur objectifs et aux applications qu'elles supportent, les travaux se dirigent de plus en plus vers des technologies ouvertes pour permettre l'interconnexion aisée des grilles hétérogènes. Tel est par exemple l'un des objectifs du projet DataTAG, dont nous sommes partenaires ou bien de la technologie OGSA, Open Grid Service Architecture [1]. Cela n'est pas sans nous rappeler une certaine histoire dans le domaine des réseaux... Avec le regard d'une spécialiste réseau, le problème des grilles ouvertes peut en effet être comparé à celui de la recherche d'une solution d'interconnexion de ressources de calcul et de stockage aussi puissante, simple et robuste que l'a été le protocole IP pour l'interconnexion des réseaux hétérogènes de transmission de données. A l'opposé du modèle OSI qui tentait de résoudre une multitude de problèmes, ce qui a fait la puissance du modèle IP fut son extrême simplicité. Selon ce point de vue, on constate le protocole IP a été conçu pour réaliser un nombre très restreint de fonctions : la fonction de routage des paquets (forwarding) et celle de l'adressage. Les autres problèmes étant repoussés à la périphérie de l'interconnexion et notamment aux extrémités [83]. Dans les chapitres précédents je me suis attachée à montrer qu'il était extrêmement difficile de faire évoluer IP pour lui ajouter de nouvelles fonctionnalités. Le déploiement de nouveaux protocoles tels que Ipv6 ou IpMulticast sont très lents car le nombre d'équipements concernés par les mises à jour est gigantesque. C'est une des raisons qui pousse la communauté réseau à étudier aujourd'hui la technologie des réseaux actifs dont le but est de permettre d'introduire de nouvelles fonctionnalités dans un ré-

seau tel qu'Internet de manière dynamique et flexible. En rapprochant l'idée de réseaux actifs de la problématique de l'architecture ouverte de services pour la grille, nous avons fait émerger le concept de **grille active** que nous cherchons à promouvoir [128], [167] et qui paraît une voie prometteuse pour la *grille ouverte*. Nous travaillons d'une part au niveau des services actifs pour la grille dans le cadre du projet RNTL e-toile avec Fabien Chanussot et d'autre part à l'architecture des équipements actifs aptes à honorer les contraintes de performance des grilles de calcul intensif avec Marc Herbert dans le cadre de sa thèse. Le concept de grille active me semble en effet bien plus ambitieux que simplement l'application de la technologie des réseaux actifs au domaine des grilles. Il faut fournir un environnement permettant le déploiement dynamique de nouveaux protocoles et de services de grille.

6.4.2 Middlehardware intelligent pour la grille

La modélisation de la grille proposée précédemment met en évidence la frontière entre l'ensemble des sites et le réseau longue distance ou l'Internet. Avec Marc Herbert nous nous intéressons à cette frontière. C'est en effet là que nous pensons que la partie *active* de la grille sera localisée. Nous analysons en particulier les limitations de performances et de fonctionnalités des architectures classiques d'équipements de bordure de grille. L'objectif de son travail de thèse est de trouver des solutions architecturales et protocolaires pour accroître les performances du transport de bout en bout et la complexité des traitements effectués sur les flux et les paquets au niveau de cette frontière.

Dans son travail de DEA [108], Marc Herbert a analysé les limites des architectures de commutateurs Ethernet et a montré qu'il est possible de concevoir, selon des principes d'architecture récents et issus du domaine du calcul haute performance, un nouveau type de matériel de commutation distribué capable d'émuler un équipement de type Ethernet, incontournable standard des réseaux locaux. Une comparaison des réponses apportées par Ethernet et Myrinet aux problèmes généraux de la commutation dans les réseaux locaux a été préalablement réalisée. Nous avons proposé une architecture de commutation distribuée Ethernet [107] basée sur la technologie *cluster*. La figure 6.10 donne un schéma de principe de cette architecture. Cette architecture n'est pas limitée en bissection et est totalement extensible. Les premières étapes de la réalisation d'un prototype ont été effectuées ainsi que la réalisation d'outils de développement spécifiques et notamment un environnement de programmation de carte intelligente (MCP), illustré par la figure 6.11. Ce travail a permis d'amorcer la réflexion sur la problématique de répartition des traitements dans le réseau et d'étudier un certain nombre de verrous relatifs aux architectures de commutation distribuées basées sur des réseaux *whormhole* tels que les problèmes de topologies redondantes et irrégulières et les problèmes de diffusion de groupe.

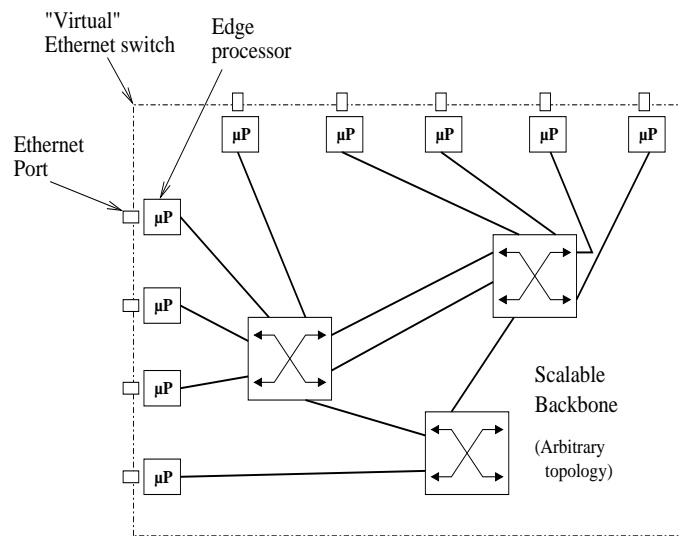


FIG. 6.10 – Principe de l'architecture distribuée d'équipement réseau

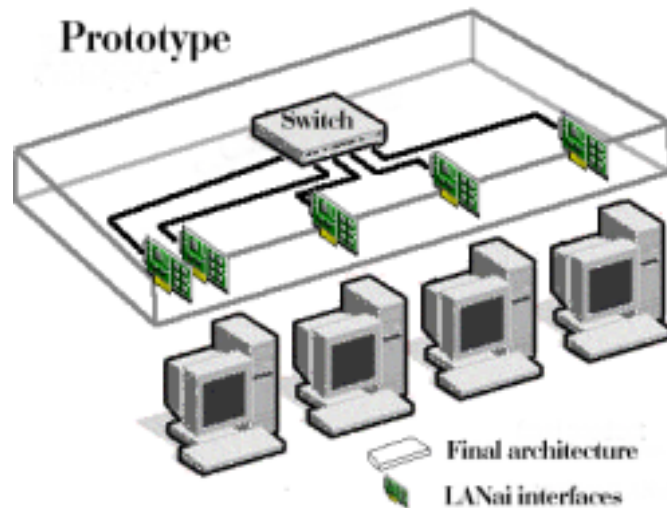


FIG. 6.11 – Prototype développé

Nous poursuivons cette recherche et étudions les possibilités d'intégration de fonctions évoluées dans les équipements de bordure d'un réseau haut débit tel que VTHD basés sur ces mêmes concepts architecturaux. Ce type d'équipement doit permettre de réaliser, dans un contexte très haut débit, c'est à dire supérieur à 1 Gb/s de part et d'autre de la frontière :

- la gestion de l'hétérogénéité des communications de la grille
- le transport haute-performance sur un réseau de coeur très haut débit
- le contrôle de la performance de bout en bout
- le support efficace d'un environnement actif tel que Tamanoir
- les fonctions de surveillance de trafic
- les fonctions de sécurité distribuées (AAA)

La réalisation pratique et l'évaluation de la plupart des services intelligents requis par la grille active à des débits de l'ordre du Gb/s peut nécessiter une forte puissance de traitement et de stockage. La seule réponse matérielle peu coûteuse connue à ce jour qui répond à ce genre de problématiques est la grappe de PC, qui répartit les traitements entre différents noeuds. Cependant, deux contraintes allant à l'encontre de ce type de parallélisme doivent être prises en compte :

- du côté interne, les services fournis aux applications utilisatrices doivent l'être de manière **transparente**, avec une interface **unique** selon une sémantique de type socket ou analogue ("endpoint").
- du côté externe, les paquets arrivent en un point **unique** : le routeur d'accès.

Il est nécessaire, pour s'adapter à une architecture distribuée, d'éclater/assembler le trafic en ces deux points.

Côté externe (WAN), cette tâche peut être réalisée en profitant de la capacité de routage haute performance des routeurs actuels : chaque nœud de la grappe est reliée directement au routeur, à un débit qu'il est capable de gérer. Une correspondance longue distance de nœud à nœud sera établie afin que le routeur puisse répartir le trafic arrivant entre les nœuds de la grappe.

Côté interne (LAN), le problème est beaucoup plus ouvert et dépend des applications, services souhaités, logiciels et matériels réseaux utilisés. Des modifications logicielles des bibliothèques de communication sont nécessaires.

Nous envisageons différentes stratégies de répartition du trafic, du tourniquet à une distribution associée à la différenciation en service offerte sur le WAN.

L'architecture d'équipement proposée sera validée avec la mise en oeuvre d'un protocole de transport fragmenté et l'adaptation de la bibliothèque de communication *Madeleine* développée par l'équipe REMAP du LIP. L'objectif est d'obtenir des performances de bout en bout supérieures à 500Mb/s et une latence de l'ordre de la latence du réseau VTHD, c'est à dire inférieure à 10ms pour l'ensemble des liens du réseau.

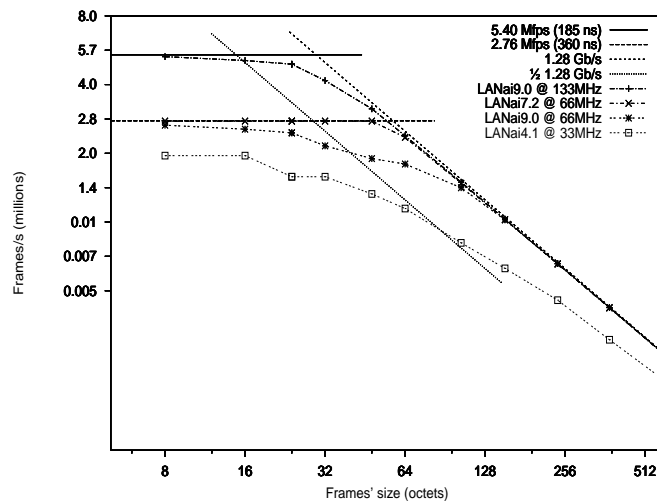


FIG. 6.12 – Performances en émission obtenues sur le prototype

6.5 Déploiement de plate-formes expérimentales

Dans cette dernière section, je développe rapidement les activités que je mène dans le cadre de la conception et du déploiement de plate-formes expérimentales de grande envergure. Les objectifs d'une plate-forme expérimentale (testbed) sont de développer, de tester et de raffiner une technologie (ou un ensemble de technologies) [91]. Dans certains cas il faut étendre les capacités d'un système existant : plus grand, plus rapide, plus simple. Dans d'autres, ce qui est prévu va à l'encontre des pensée établies et ne peut être accompli par une simple extension des technologies courantes. Pour la création de grilles de calcul, le déploiement de plate-formes expérimentales est critique à trois points de vue

- échelle d'intégration : diverses technologies doivent être intégrées, déployées et tester de manière totalement nouvelle.
- construction de communautés : le calcul distribué permet de construire des communautés d'utilisateurs et de développeurs autour des ressources de calcul. Les plates-formes permettent d'accélérer la formation de ces communautés
- mitiger les risques : permet de quantifier et de qualifier les résultats évolutionnaires afin de permettre aux utilisateurs de mesurer les nouvelles opportunité et les risques correspondants. Ainsi, les plates-formes doivent supporter des mesures et doit choisir avec attention ses objectifs.

Une plate-forme est une combinaison complexe de technologies et d'utilisateurs. Il est donc important de souligner l'importance des aspects organisationnels d'une telle plate-forme.

6.5.1 Support réseau du projet DataGRID

Dans le cadre du projet européen DataGRID, nous avons développé une méthodologie de dimensionnement [27], étudié les besoins des applications de la grille européenne [166]. Nous avons exploré les différentes solutions actuelles de construction de VPN ainsi que l'évolution des technologies réseau des infrastructure des réseaux nationaux et européen de la recherche [27]. Nous avons pu constater que la technologie ATM fortement déployée jusqu'à aujourd'hui et utilisée pour construire des réseaux privés tels que celui de l'IN2P3 dans l'infrastructure RENATER disparaissait au profit de la technologie IP sur SDH et rendait la construction de VPN aujourd'hui impossible. Les seules solutions que se profilent à l'horizon sont MPLS et IPSec. Cette problématique reste ouverte tant pour les aspects performance que sécurité des grilles de calcul. Dans [42], une analyse très pertinente des similarités entre les VPN et les protocoles de communication de groupe est menée. C'est à mon sens une voie très intéressante à explorer dans le cadre des grilles de calcul où la notion de groupe est centrale.

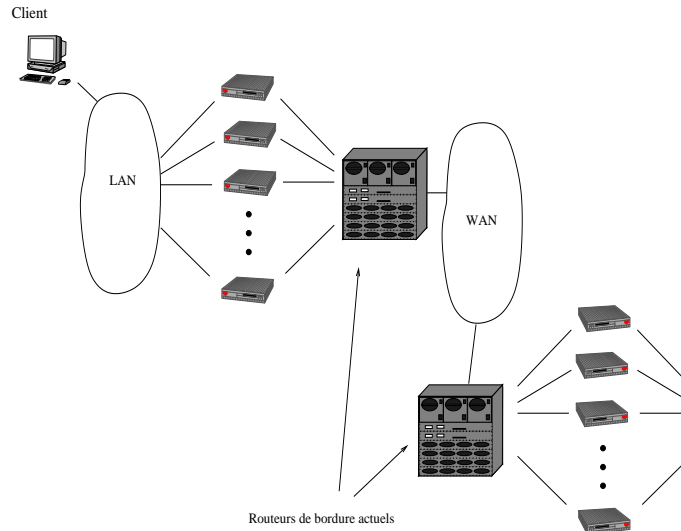


FIG. 6.13 – Principe du middle-hardware distribué de grille

Nous avons par ailleurs défini et participé à la mise en oeuvre d'une architecture de mesure des performances de l'infrastructure réseau basée sur l'épine dorsale européenne (GEANT). Cette architecture est composée d'un ensemble d'outils sélectionnés pour leur précision et leur faible intrusivité. Une arborescence spécifique pour le stockage des informations dynamiques de mesure du réseau a été définie dans le système d'information standard de l'environnement Globus, MDS2, et des outils de présentation dédiés ont été intégrés. Avec Franck Bonnassieux nous avons conçu et déployé l'outil MapCenter qui est utilisé quotidiennement dans la plate-forme. Avec Robert Harakaly, nous explorons les aspects mesure et prédiction de performances à court et moyen terme avec l'outil NWS (Network Weather Service). Nous avons montré et identifié la cause des problèmes de précision de l'outil de mesure et avons proposé des évolutions. Nous avons conçu l'algorithme de coordination des tests PCP qui est en cours de déploiement dans DataGRID. Nous évaluons et déployons un service de multicast fiable pour la grille en collaboration avec Moufida Maimour et CongDuc Pham de Resam. Avec l'ensemble du groupe réseau, j'ai restructuré les activités du workpackage de la manière suivante :

- Etude détaillée des besoins et de la nature des communications des applications de la grille dans le plan données et du *middleware* dans le plan contrôle, en collaboration étroite avec les utilisateurs.
- Collaboration technique avec le réseau européen GEANT(10Gb/s) et les réseaux nationaux (NRENTs)et en particulier RENATER pour le test de services réseaux avancés et l'analyse des mesures de performance des liens. Ceci nous conduit à participer aux expérimentations du projet IST SEQUIN.
- Déploiement d'une infrastructure ouverte de supervision, de mesure de performance efficace du réseau de la grille européenne.
- Test et développement d'outils de prédiction de performances et intégration dans les composants d'ordonnancement et d'optimisation du *middleware*.
- Développement d'une fonction *coût du réseau*.
- Etudes et optimisation des transferts très volumineux sur des liens longue distance très haut débit et à latence élevée.
- Etude et Développement d'un service de Multicast fiable pour la Grille.
- Conception de l'architecture de sécurité de la grille.

6.5.2 Le projet E-toile

L'objectif d'E-Toile est de développer une plate-forme expérimentale de grille de calcul basé sur un réseau très haut débit. Des applications dans le domaine du calcul intensif et du traitement de grandes masses de données servent à démontrer l'intérêt de la technologie grille sur des applications types. Un *middleware* prototype, intégrant

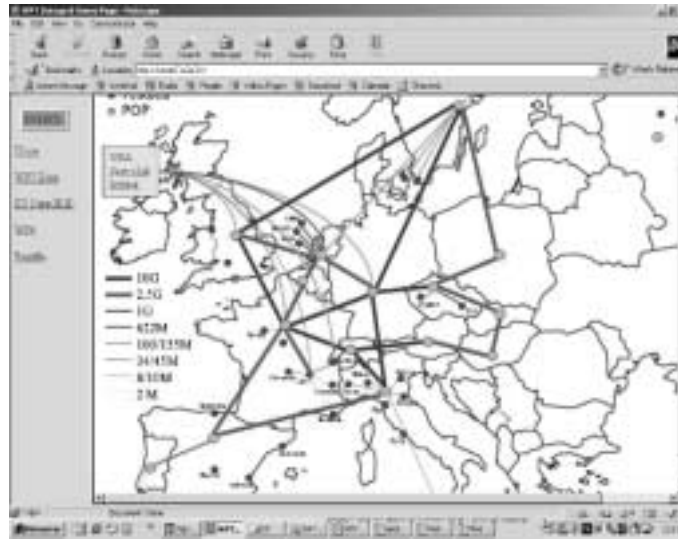


FIG. 6.14 – Vue logique du réseau de DataGRID (avec Mapcenter)

les travaux de recherche des différents laboratoires impliqués notamment sur les thèmes des réseaux actifs et des communications haute performance est développé.

J'assure la coordination scientifique de ce projet. Avec les chercheurs de Resam nous focalisons nos travaux sur le concept de Grille Active et sur les besoins réseaux des applications et du *middleware* [167]. La démarche consiste à

- Récupérer l'expérience acquise à travers le projet DataGRID (méthodologies, outils...)
- Participer à la mise en place de la plate-forme expérimentale
- Identifier les limites des services réseau existants et les problèmes de performances de bout en bout
- Identifier les aspects dynamiques requis au niveau réseau
- Déployer des routeurs actifs haute performance dans la Grille
- Développer des services actifs spécifiques.
- Intégrer et valider des services actifs prototypes dans le *middleware* de la Grille.

J'approfondis plus particulièrement les thèmes de recherche suivants :

- l'étude d'une architecture de supervision réseau dynamique et ouverte basée sur la technologie active et des méthodes de mesure peu intrusives, permettant de personnaliser le contrôle des performances de bout en bout.
- la conception d'équipements d'accès au réseau très haut débit, distribués et intelligents et permettant le transport optimisé des flux grid et l'exécution efficace de services actifs.
- la conception d'un service actif de transport offrant aux flux hétérogènes une qualité de service réseau appropriée aux besoins des utilisateurs et aux conditions du réseau. Ces travaux approfondissent et expérimentent les approche EDS à *services équivalents* et ADS à *services différenciés actifs* pour développer un protocole de transport flexible et adaptable.

Nous en sommes au démarrage de ce projet et je compte y concentrer une grande part de mes activités de recherche dans le proche avenir.

6.6 Conclusion

Nous avons mené des études sur les spécificités des modèles des applications de la grille ainsi que sur leurs besoins de transport de données. J'ai initié un travail de réflexion sur le concept de Network Element et en propose

une spécification. Dans le cadre du projet DataGRID, nous avons conçu et développé une architecture et des outils de surveillance de l'état et des performances réseaux de la grille. Un certain nombre de problématiques liées à la mesure et à la prédiction des performances de l'interconnexion réseau sont apparues et nécessitent des investigations plus approfondies notamment en ce qui concerne la coordination des campagnes de mesure et la précision et l'intrusivité des méthodes de mesure active sur des réseaux très haut-débit. Par ailleurs, nous démarrons des activités expérimentales sur le déploiement de nouveaux services tels que premium service et IPmulticast pour les applications de grille dans les réseaux européens et internationaux en collaboration avec les consortium européens tels que DANTE, SEQUIN mais aussi les équipementiers. Chaque jour nous voyons émerger de nouveaux problèmes de déploiement et de nouveaux protocoles. La mise à jour des différents sites des plate-formes expérimentales et la validation des solutions nouvelles deviennent de plus en plus complexes au fur et à mesure de l'expansion de la grille. Cette tendance risque de s'alourdir dans les années futures et pour anticiper ce phénomène, nous explorons le concept de **grille active** dont un des objectifs est de favoriser le déploiement dynamique de nouveaux services et de nouveaux protocoles évolués. Nous étudions aussi les architectures des équipements de bordure de grille qui seront très certainement en charge de ces fonctions évoluées. La technologie *cluster* nous semble être la plus prometteuse. Je coordonne le déploiement de plusieurs plate-forme expérimentales, convaincue que cette expérience pratique nous conduira vers une meilleure compréhension des problématiques et orientera nos travaux théoriques vers des solutions originales et pragmatiques.

Chapitre 7

Conclusion

7.1 Bilan et perspectives scientifiques

Dans ce document j'ai développé le concept de réseau sensible aux flux et ouvert aux applications comme réponse aux problèmes de transport de flux hétérogènes dans une infrastructure globale complexe. L'idée fédératrice est de concevoir un réseau capable de réagir dynamiquement à des *couleurs* de paquets afin de les acheminer de manière appropriée vers leur destination et ce sans altérer la philosophie initiale des protocoles de l'Internet. J'ai proposé deux approches qui illustrent deux manières différentes

- de colorer les flux ou les paquets dès leur génération
- de réagir dynamiquement et de manière flexible aux couleurs de ces paquets pendant leur transfert.

7.1.1 Du besoin des applications...

Pour arriver à ce concept, je me suis appuyée sur une profonde expérience des applications et des systèmes coopératifs multimédia et de la technologie ATM. J'ai tout d'abord montré quels étaient les besoins et les contraintes des applications interactives multi-utilisateurs et multimédia. La conception et le développement de la boîte à outils CoTools ainsi que du modèle d'architecture AMF-C nous a permis de cerner les problématiques de production et d'exécution des applications coopératives synchrones, mais au delà, celle des intergiciels de construction et de support d'applications réparties. **J'ai proposé une nouvelle approche ainsi qu'une abstraction de programmation pour articuler de manière souple les différents niveaux d'abstraction et apporter de la flexibilité dans les logiciels interactifs répartis.** J'ai étudié l'interaction collaborative multimédia sur un réseau ATM dans le cadre d'un projet de télé-ingénierie de conception et de réalisation. Nous avons montré que la qualité du réseau n'était pas le seul facteur déterminant la qualité d'un dispositif de collaboration. **La chaîne de la qualité de service est très complexe et l'interdépendance de multiples éléments à la fois sur le plan horizontal d'un bout à l'autre du réseau et sur le plan vertical de haut en bas de la pile protocolaire rend l'obtention et la compréhension des problèmes de performances aux extrémités délicate.**

L'expérience P2-ATM a mis en évidence l'hétérogénéité des besoins pour un même flux dans le temps et selon les usages et les usagers. Face à cette disparité des applications et des besoins, des questions fondamentales et récurrentes se posent :

- quelles performances sont réellement nécessaires ?
- quand et pour quoi faire ?
- quels types de garanties veut-on offrir ?
- comment capturer les besoins des applications ?
- comment permettre aux applications de les exprimer ?
- quels mécanismes faut-il inventer pour permettre une adaptation dynamique à des facteurs aussi variés que la tâche à effectuer, le support technique disponible mais aussi l'expertise des acteurs, la maturité du groupe, l'aspect financier ?

La multitude des besoins et des usages requiert des modèles et de disciplines de services flexibles

pour allouer dynamiquement différents profils de performances aux flux.

7.1.2 ...aux modèles d'architecture et de services différenciés

Au niveau réseau IP, les mécanismes de gestion active de file d'attente et le surdimensionnement permettent de limiter les congestions dans Internet. Mais ces mécanismes ne fournissent pas explicitement des services évolués aux applications. Les propositions IntServ/RSVP et DiffServ visent à apporter des services différenciés avec des garanties de QoS tout en conservant un service Best Effort bien éprouvé. Mais le déploiement aussi bien de l'architecture IntServ/RSVP et dans une moindre mesure celui de DiffServ s'avèrent complexes et peu concluants. D'un autre côté, les techniques adaptatives se sont beaucoup développées dans les applications audio-vidéo pour Internet. Une bonne connaissance de la problématique des application multimédia est aujourd'hui acquise. L'inconvénient majeur est qu'elles ajoutent de la complexité au développement de l'application et qu'elles sont peu réutilisables et extensibles. Nous avons cherché à dissocier les mécanismes d'encodage et d'émission des mécanismes de mesure de performances et d'adaptation dynamique pour le cas particulier de la transmission vidéo. L'objectif visé de **construire un mécanisme générique qui s'adapte d'une part aux variations de performances du réseau mais aussi aux variations des besoins des applications** paraît atteignable mais plusieurs verrous restent encore à lever.

A l'issu de ces travaux sur les application réparties, les techniques adaptatives et les protocoles réseau, j'ai acquis l'intime conviction qu'il est nécessaire de combiner les approches proposées à différents niveaux du modèle architectural pour obtenir un compromis final flexibilité-performance intéressant. **Les solutions dans le réseau et aux extrémités ne s'excluent pas mutuellement.** Nous avons exploré les limites du modèle DiffServ et étudié le domaine des réseaux actifs pour la qualité de service. Nous avons analysé et expérimentons des solutions de QoS plus simples telles que les services *différents mais égaux* proposés dans ABE ou QBSS et la proposition du modèle Balanced Forwarding. L'expérience acquise avec Balanced Forwarding et Netstre@mer ont servi de base à nos recherches sur les services différenciés équivalents et actifs et les protocoles de transport programmables. *La qualité de Service traditionnellement associée à la notion de garantie est un objectif très élevé qui s'avère délicat voire impossible à mettre en oeuvre dans le contexte Internet.* J'oppose à cet objectif strict une cible plus flexible de fourniture de *services appropriés*. Conformément à la philosophie IP le but est d'offrir *du mieux qu'il est possible de faire* des services réseaux et transports différenciés. **L'expression sensibilité aux flux apporte toute la nuance nécessaire au traitement de ce problème dans une interconnexion complexe.** L'ensemble des travaux que nous avons menés sur le modèle de services différenciés équivalents EDS et les services différenciés actifs ADS sont très prometteurs. Cependant, les simulations de TCP sur divers modèles de différenciation IP nous ont enseigné qu'**une couche réseau et une couche transport conçues de manière indépendantes et avec des objectifs différents ne peuvent interagir de manière élégante et vraiment efficace.** Cela nous pousse à conclure que pour faire évoluer IP il faut faire évoluer en profondeur la couche transport et les mécanismes de contrôle de congestion d'Internet. C'est la raison pour laquelle je concentre nos efforts à présent sur le niveau transport et l'approche réseaux actifs. Il s'agit de montrer les limites des protocoles TCP et RTP/UDP existants et d'ouvrir de nouvelles pistes plus novatrices. La couche de transport adaptatif à laquelle nous réfléchissons donne la **capacité aux extrémités d'avoir une influence sur l'acheminement des paquets lors de leur traversée du réseau.** Il s'agit d'une forme primitive mais immédiatement déployable d'approche active. Dans ces développements, nous ne visons pas une couche de transport universelle, mais seulement à **montrer comment une couche de transport adaptatif peut interagir de manière optimale avec une couche IP à services différenciés.** Par ailleurs, nous pensons que le déploiement, incontournable à moyen terme, de la technologie active aura aussi une influence sur l'évolution de la couche transport. Les propositions EDS et ADS s'intègrent et ne s'opposent pas aux autres solutions telles le sur-dimensionnement du coeur de réseau, la gestion active des files dans les routeurs, le déploiement de Premium Service ou des techniques adaptatives.

Outre le travail d'approfondissement théorique qui reste à mener sur les modèles EDS et ADS, plusieurs autres questions demeurent encore sombres. **Comment spécifier formellement les besoins hétérogènes et dynamiques des applications et comment déployer les architectures nouvelles de QoS dans l'Internet** sont les principales voies que je souhaite explorer. C'est le sens de ma très forte implication dans le domaine en pleine explosion des grilles de calcul et de données.

7.1.3 D'une grille *passive*

Nous avons initié des études sur les spécificités des modèles des applications de la grille, sur leurs besoins de transport de données et sur le concept de l'élément réseau dans la grille. Nous avons aussi **conçu et développé une architecture et des outils de surveillance de l'état et des performances réseaux de la grille**. Un certain nombre de problématiques liées à la mesure et à la prédiction des performances du réseau étendu sont apparues et nécessitent un investissement important. Par ailleurs, nous démarrons des activités expérimentales sur le déploiement de nouveaux services de QoS et de multicast pour les applications de grille dans les réseaux européens et internationaux en collaboration avec les opérateurs académiques mais aussi les équipementiers.

7.1.4 ... à un réseau sensible à la grille et une grille active

Chaque jour nous voyons émerger dans la grille de nouveaux besoins, de nouveaux protocoles et de nouveaux problèmes de déploiement. La mise à jour des logiciels sur les plate-formes expérimentales et la validation des solutions deviennent de plus en plus complexes au fur et à mesure que la grille se déploie. Cette tendance risque de s'alourdir dans les années futures et pour anticiper ce phénomène, nous explorons le **concept de grille active dont un des objectifs est de favoriser le déploiement dynamique de nouveaux services et de nouveaux protocoles évolués sur une grille**. Nous étudions et concevons une **architecture d'équipements de bordure de grille distribuée en charge de ces fonctions évoluées**.

En qualité de domaine d'application gros consommateur de ressources de communication, la grille pose des défis intéressants à la communauté réseau et requiert une compétence informatique multidisciplinaire. Je souhaite, dans les années futures montrer qu'au delà de la simple fourniture d'un *tuyau*, le réseau est de nature à contribuer de manière riche et pertinente au domaine des grilles de calcul. **Les expériences les plus solides mais aussi les modèles les plus évolutionnaires du domaine des réseaux doivent être confrontés aux concepts du métacomputing, des systèmes distribués et de la programmation parallèle pour aboutir à une architecture ouverte de services de grille**.

Dans le futur, je pense que les principales limitations pour la construction des systèmes et des applications réparties sera la complexité des programmes d'application. Les protocoles réseaux devront donc être conçus de façon à simplifier cette programmation. Les programmeurs se concentreront sur la sémantique des traitement plutôt que sur les communications. Nous devons continuer à faire le maximum pour **rendre le réseau de plus en plus invisible aux usagers. Pour qu'il le soit, il doit être le plus intelligent et efficace possible et offrir des services à valeur ajustée**. Il devra intégrer des services de mesure et de surveillance des performances affinés ainsi que des fonctions de prise de décisions pertinentes vis à vis de l'acheminement des flux. Nous sommes limités d'un côté par les traitements que l'on peut faire au niveau réseau, dans les routeurs du chemin, d'un autre nous sommes contraints par la simplicité de construction des applications communicantes.

Les protocoles réseaux sont critiques pour l'avenir de l'informatique, car tous les traitements futurs seront des traitements répartis. Ces protocoles fourniront la glue qui maintient ensemble les entités dispersées. Vraisemblablement, aucune solution unique et fédérant l'ensemble des besoins n'apparaîtra. Il faut plutôt s'orienter vers des **boîtes à services de transport plus ou moins sophistiqués, flexibles, adaptables et reliés dynamiquement au cours de l'exécution des applications**. L'utilisation de ses services sera opaque au programmeur et à l'utilisateur, ou bien translucide pour celui qui voudra optimiser son code en reliant des composants de plus bas niveau.

7.2 Conclusion personnelle

Mes travaux ont couvert un large spectre de la problématique des réseaux allant des protocoles aux outils de développement d'applications distribuées interactives en passant par des aspects architecturaux. Je me suis donc intéressée aux niveaux 3 à 7 du modèle OSI et j'ai navigué au sein de trois communautés qui n'échangent pas toujours leurs modèles et leurs préoccupations : réseaux, systèmes et interfaces homme-machine. Les réseaux, maillon de base de nos systèmes d'information, sont aujourd'hui vitaux pour la société moderne. Depuis que j'ai plongé dans ce domaine passionnant de la communication, je pressens à la fois des bouleversements profonds tout en ressentant un immobilisme exaspérant. En dépit des apparences, le modèle Internet n'aime pas les révolutions

et se complait vraiment dans l'évolution en douceur. J'ai assisté à l'explosion des débits d'accès aux réseaux longue distance puisqu'en quelques années d'intervalle j'ai eu successivement accès à une prise RNIS à 128Kb/s, puis à une prise ATM à 2Mb/s et aujourd'hui je peux brancher ma machine sur un réseau optique à 1Gb/s et faire des expérimentations sur un backbone à 10Gb/s!! J'ai eu la chance de travailler intensément avec des outils de vidéoconférence et de collaboration en profitant d'une QoS garantie et je continue de croire qu'un accès de type CBR à 384kb/s associé à un canal moins rapide mais plus fiable de quelques Mb/s est nécessaire et mais aussi suffisant pour permettre le déploiement de services de travail collaboratif efficaces.

Au fil de ce document, j'ai montré l'intérêt et l'importance d'établir un lien étroit entre les utilisateurs et les chercheurs en réseau aussi bien pour l'analyse des besoins que pour la validation des solutions proposées. Cet échange est proposé comme démarche méthodologique. **Un des points forts de mon parcours personnel est d'avoir abordé la problématique des réseaux et de la qualité de service par ses deux côtés : côté application (utilisateur de service) et côté réseau (fournisseur de service). J'ai ainsi acquis une compétence pluri-disciplinaire précieuse.** Cette démarche *bout en bout* a été initiée il y a plus de cinq ans dans le cadre de mes travaux sur les applications coopératives et se poursuit aujourd'hui dans mes recherches sur le support réseaux haute performance aux grilles de calcul. L'expérience du développement d'un *middleware* coopératif et de la définition d'un modèle architecture d'applications coopératives, me permet de cerner l'interdépendance entre le réseau et les applications qui l'utilisent. Je pense que sans dialogue et coopération concrète entre les communautés, les avancées dans le domaine de l'informatique répartie seraient très lentes. Par exemple, le fossé entre la vision des usagers et celle des fournisseurs de service demeure important. D'un côté les utilisateurs sont mécontents car ils n'obtiennent pas toujours les performances escomptées et de l'autre, les opérateurs mesurent un faible trafic sur des liens qu'ils considèrent - à juste titre - surdimensionnés. Depuis plusieurs années, cette question me préoccupe et **j'ai orienté mes travaux dans cette direction de l'étude de la performance de bout en bout.**

L'ensemble de mes recherches théoriques a été inspiré et nourrit par des expérimentations en vraie grandeur menées dans le cadre de projets nationaux ou internationaux. Ainsi les modèles définis et les expérimentations réalisées sur des réseaux locaux contrôlés et très haut débit, sont redéployées sur des réseaux longues distances et mises en perspectives. Ce travail permet d'affiner les modèles et d'en étudier leur extensibilité, ce qui a l'avantage de conduire à des solutions que je pense pertinentes, utiles et utilisables.

Je souhaite ardemment poursuivre mes recherches dans le domaine des réseaux, des applications réparties et des grilles de calcul avec l'ensemble des chercheurs et des ingénieurs passionnés qui m'entourent, mais aussi avec toutes les équipes nationales et européennes ou plus lointaines avec lesquelles nous collaborons. Il nous restent encore beaucoup de pain sur la planche pour mieux maîtriser nos idées et les partager avec la communauté internationale. L'ensemble des collaborations que j'ai tissé ces dernières années laisse entrevoir de riches et enthousiasmantes perspectives.

Bibliographie

- [1]
- [2] *The Grid : Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers, Inc., 1998.
- [3] London, England, June 1999. IEEE/IFIP.
- [4] Pittsburgh, PA USA, June 2000. IEEE/IFIP.
- [5] Karsenty A. and M. Beaudouin-Lafon. An algorithm for distributed groupware applications. In *ICDCS'93 International Conference on Distributed Computing Systems*, 1993.
- [6] J. George A. Waheed, W. Smith and J. Yan. An infrastructure for monitoring and management in computational grids. *IEEE International Symposium on High-Performance Distributed Computing*.
- [7] Le Boudec J.Y. Ferrari T. Almersberger, W. Scalable resource reservation for the internet. november 1997.
- [8] Werner Almesberger, Alexey Kuznetsov, and Jamal Hadi Salim. Differentiated services on linux. In *Proceedings of Globecom'99*, pages 831–836, Rio de Janeiro, December 1999.
- [9] E. Amir, S. McCannne, and H Zhang. An application-level video gateway. In *Proc. of the ACM Multimedia*, San Francisco, November 1995.
- [10] G.R ANDERSON, T.C.N. GRAHAM, and T.N. WRIGHT. Dragonfly :linking conceptual and implementation architectures of multiuser interactive system. In *proceedings of the ICSE'2000*, 2000.
- [11] L. Andre, P. Primet, B. Grandjean, and S. Pierre. Civic : cite virtuelleconnaissance - cours atm. *coopération CNET/LICEF/ECL 06/98* . Copyright (c) 1998 - Centre de recherche LICEF, Télé-Université, Montréal, Québec, Canada, 1998.
- [12] Audio-Video Transport Working Group, Henning Schulzrinne, Steve Casner, Ron Frederick, and Van Jacobson. RTP : A transport protocol for real-time applications. Internet Request For Comments RFC 1889, Internet Engineering Task Force, January 1996.
- [13] C. Aurrecochea, A. Campbell, and L. Hauw. A survey of qos architecture. 1997.
- [14] Teitelbaum B. Future priorities for internet2 qos. Technical report, Internet2/Qbone, 2001.
- [15] D. Gunter W. Smith V. Taylor R. Wolski M. Swany B. Tierney, R. Aydt. Technical report.
- [16] M. Beaudouin-Lafon. Computer-supported cooperative work. John Wiley and Sons Ltd, trends in software series edition, 1999.
- [17] R. Bentley and P. Dourish. Medium versus mecanism : supporting collaboration through customisation. In *ECSCW'95*. Kluver Academic Press, 1995.
- [18] Samrat Bhattacharjee, Kenneth L. Calvert, and Ellen W. Zegura. An architecture for active networking. Technical Report GIT-CC-96-20.
- [19] Steven Blake, David Black, Mark Carlson, Elwyn Davies, Zheng Wang, and Walter Weiss. An architecture for differentiated services. Internet Request For Comments RFC 2475, Internet Engineering Task Force, December 1998.
- [20] Roland Bless and Wehrle Klaus. Evaluation of differentiated services using an implementation under linux. In *Proceedings of IWQoS'99* [3].
- [21] J Bolliger, T. Gross, and U. Hengartner. Bandwidth modelling for network-aware applications. In *Infocom*.

- [22] J. Bolot. Characterizing end to end packet delay and loss in the internet. *Journal of High Speed Networks*.
- [23] J.C. Bolot and T. Turletti. A rate control for packet video in the internet. In *IEEE Infocom*, pages 1216–1223, Toronto, 1994.
- [24] J.C. Bolot and T. Turletti. Experience with control mechanisms for packet video in the internet. 1998.
- [25] Jean-Chrysostome Bolot and Andres Vega-Garcia. Control mechanisms for packet audio in the internet. In *INFOCOM (1)*, pages 232–239, mar 1996.
- [26] F. Bonnassieux, R. Harakaly, and P. Primet. Mapcenter : an open grid status visualization tool. <http://ccwp7.in2p3.fr/mapcenter>.
- [27] F. Bonnassieux, P. Primet, and P. Clarke. Network provisioning for the datagrid testbed1. Technical report, EU DATAGRID report Deliverable D7.1 - Approved by the EC January 2001, 2001.
- [28] Franck Bonnassieux, Peter Clark, and Pascale Primet. Network requirements and network infrastructure. Technical report, IST EDG DataGrid project, 2002.
- [29] J. Boyle and al. The cops (common open policy service) protocol. Technical Report 2748, ietf.
- [30] Robert Braden, David Clark, and Scott Shenker. Integrated services in the internet architecture : an overview. Internet Request For Comments RFC 1633, Internet Engineering Task Force, June 1994.
- [31] P. Brady. Effect of transmission delay on conversational behavior on echo-free telephone circuits. In *Bell System Technical Journal*, volume 50, pages 115–134, January 1971.
- [32] B. Tierney and al. The netlogger methodology for high performance distributed systems performance analysis. *IEEE International Symposium on High-Performance Distributed Computing*.
- [33] I. Busse, B. Deffner, and Henning Schulzrinne. Dynamic qos control of multimedia application based on rtp. 1995.
- [34] G. Calvary, J. Coutaz, and L. Nigay. From single-user architectural design to pac*. In *CHI'97 proceedings*, pages 242–249. ACM Press, 1997.
- [35] Calvert. Direction in active networks. *IEEE Communication*, 36(10) :72, Octobre 1998.
- [36] T.M. Chen and A Jackson. Active and programmable networks. *IEEE Network*, 12(3) :10–16, may-june 1998.
- [37] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke. The data grid : Towards an architecture for the distributed management and analysis of large scientific datasets. *Journal of Network and Computing Applications*, 23 :187–200, 2001.
- [38] D Clark and D. Tennenhouse. Architectural considerations for a new generation of protocols. In ACM, editor, *Proc. of the SIGCOMM*, Philadelphia, September 1990.
- [39] David D. Clark. The design philosophy of the DARPA internet protocols. In *ACM SIGCOMM*, pages 106–114, 1988.
- [40] David D. Clark and Wenjia Fang. Explicit allocation of best-effort packet delivery service. *IEEE/ACM Transactions on Networking*, 6(4) :362–373, 1998.
- [41] M. Cosnard, T. Priol, F. Desprez, P. Primet, V. Alessandrini, D. Vandrome, and C. Roucairol. Rapport de la mission française “grilles de calcul et réseaux haut débit. Technical report, rapport de Mission à Washington - Internet2 - NSF - DOE - Novembre 2001 pour le Ministère Français de la Recherche, 2001.
- [42] L. Costa, S. Fdida, and O. Duarte. An introduction to virtual private networks. In *Network and Information Systems Journal*, volume 2, pages 83–92.
- [43] J. Coutaz. Architecture models for interactive software : Failures and trends. In G. Cockton, editor, *Engineering for Human-Computer Interaction*, pages 137–153. Elsevier Sc. Publ., 1990.
- [44] J. Coutaz, D. Salber, M. Riveill, and al. Etude de cas n°1 : contrôle d'accès. Technical Report 94-95, Groupe de Travail GT. SCOOP du GDR-PRC Communication Homme-Machine, dec 1995.
- [45] T. and P. MILLAZZO CROWLEY, E. BAKER, and al. Mmconf : an infrastructure for building shared multimedia applications. In *proceedings of the ACM Conference on Computer Supported Collaborative Work*, pages 329–342, October 1990.

- [46] Rene L. Cruz. SCED+ : Efficient management of quality of service guarantees. In *INFOCOM (2)*, pages 625–634, 1998.
- [47] F. Dabeck and al. Building peer-to-peer systems with chord, a distributed lookup service. 2002.
- [48] Groupe de Recherches sur les réseaux actifs et programmables. Les réseaux actifs. mars 2002.
- [49] S. Deering and R. Hinden. Internet protocol, version 6 (ipv6) specification. Internet Request For Comments RFC 1883, Internet Engineering Task Force, 1995.
- [50] A. Derycke. Le c.s.c.w. au delà de l'i.h.m. : taxinomie et dimension sociale. Technical report, Communication au Groupe de Travail GT. SCOOP - du GDR-PRC Communication Homme-Machine, Lyon, 1994.
- [51] F. Desprez. Contribution à l'algorithmique parallèle. In *Document d'Habilitation à Encadrer des recherches*, Lyon, Juillet 2001. Université Claude Bernard.
- [52] P. Dewan. Tools for implementing multi-user user interfaces. In Bass and Dewan, editors, *User Interface Software*, pages 149–174. 1993.
- [53] P. Dewan. Architectures for collaborative applications. In Michel Beaudouin-Lafon, editor, *Computer-Supported Cooperative Work*, chapter 7, pages 169–194. John Wiley and Sons Ltd, trends in software series edition, 1999.
- [54] P. Dewan and R. Choudhary. A high-level and flexible framework for implementing multiuser user interfaces. *ACM Transactions on Information Systems*, 10(4), oct 1992.
- [55] P. and R. CHOUDARY DEWAN. A high-level and flexible framework for implementing multiuser user interfaces. *ACM Transactions on Information Systems*, 10(4), 1992.
- [56] M. Diaz. Protocoles et réseaux. In *Laas Report 00090*, janvier 2000.
- [57] P. Dourish. Using metalevel techniques in a flexible toolkit for cscw applications, 1998.
- [58] P. Dourish. Developing a reflective model of collaborative systems. *ACM Transactions on Computer-Human Interaction*, 2(1) :40–63, March 1995.
- [59] Constantinos Dovrolis and Parameswaran Ramanathan. A case for relative differentiated services and the proportional differentiation model. *IEEE Network*, 13(5) :26–34, September 1999.
- [60] Constantinos Dovrolis and Parameswaran Ramanathan. Proportional differentiated services, part ii : Loss rate differentiation and packet dropping. In *Proceedings of IWQoS'00* [4], pages 52–61.
- [61] Constantinos Dovrolis, Dimitrios Stiliadis, and Parameswaran Ramanathan. Proportional differentiated services : Delay differentiation and packet scheduling. *SIGCOMM 99*, 1999.
- [62] A. Drashinchi and S. Fdida. Congestion avoidance for unicast and multicast traffic. In *ECUMN 2000, IEEE Conference on Universal Multiservice Networks*, october 2000.
- [63] K. Drira, F. Gouezec, and M. Diaz. Design and implementation of coordination protocols for distributed cooperating objects. a general graphbased technique applied to corba. In *Third IFIP International Conference on Formal Methods for Open Objectbased Distributed Systems*, 1999.
- [64] K EDWARDS. Policies and roles in collaborative applications. In *proceedings of the ACM Conference on Computer Supported Collaborative Work*, pages 11–20, Cambridge MA USA, 1996. ACM.
- [65] C.A. ELLIS. Concurrency control in groupware systems. *ACM SIGMOD*, 18(2) :399–407, 1989.
- [66] C.A. and S.J.GIBBS ELLIS and G.L. REIN. Groupware, some issues and experiences. *Communications of the ACM*, 34(1) :38–58, 1991.
- [67] Stephen J.Garland Erik L.Nygren and M.Frans Kaashoek. Pan : A high-performance active network node supporting multiple mobile code systems. In *IEEE OpenArch'99*, March 1999.
- [68] Primet P. and Tarpin-Bernard F. A framework to support run time flexibility in synchronous collaborative applications. *submitted Journal of the Computer Supported Collaborative Work*.
- [69] T. Faber. Acc : Using active networking to enhance feedback congestion control mechanism. pages 61–65, May 1998.

- [70] Marcio Faerman, Alan Su, Richard Wolski, and Francine Berman. Adaptive performance prediction for distributed data-intensive applications. Technical Report CS1999-0619, 18 1999.
- [71] W. Feng and al. Understanding tcp dynamics in an integrated services internet. In *NOSSDAV'97*.
- [72] W. Feng, D. Kandlur, D. Saha, and K. Shin. Adaptive packet marking for maintaining end to end throughput in a differentiated services internet. In *IEEE/ACM Transaction on Networking*, number 5, pages 685–697, april 1999.
- [73] Paul Ferguson and Geoff Huston. *Quality of Service, Delivering QoS on the Internet and in Corporate Networks*. Wiley Computer Publishing, New-York, January 1998.
- [74] A. Feroz, S. Kalyanaraman, and A. Kumar. A tcp-friendly traffic marker for ip differentiated services. In *IWQoS'2000*, Pittsburgh, june 2000.
- [75] Tiziana Ferrari and Philip F. Chimento. A measurement-based analysis of expedited forwarding phb mechanisms. In *Proceedings of IWQoS'00* [4].
- [76] V. Firoiu and X. Zhang. Best effort differentiated services :trade-off service differentiation for elastic applications. In *Proceedings of IEEE ICT'01*, June 2001.
- [77] Steve Fisher. Datagrid.information and monitoring wp3 architecture report : Desing, requirements and evaluation criteria. Technical report, PPARC, 2002.
- [78] A. FLADENMULLER. *Gestion de la qualité de service des applications multimédias dans les environnements sans garantie de ressources*. Thèse de doctorat, Université Pierre et Marie Curie- Paris - France, 1997.
- [79] J.F. Fleury, A. Weil, C. Deprez, J. Laganier, J.B. Rios, and P. Primet. Netstreamer : diffusion multimédia auto-adaptative. Technical report, Technical Report RESAM, 2001.
- [80] S. Floyd. A report on recent developments in tcp congestion control. 39 :84–90, April 2001.
- [81] S. Floyd and V. Paxson. Difficulties in simulating the internet.
- [82] S. al Floyd. Internet research : comment on formulating the problem. In *Technical Note*, january 1998.
- [83] Sally Floyd and Kevin R. Fall. Promoting the use of end-to-end congestion control in the internet. *IEEE/ACM Transactions on Networking*, 7(4) :458–472, 1999.
- [84] Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4) :397–413, August 1993.
- [85] Sally Floyd, Van Jacobson, Steven McCanne, Ching-Gang Liu, and Lixia Zhang. A reliable multicast framework for light-weight sessions and application level framing. *IEEE/ACM Transactions on Networking*, 5(6) :784–803, 1997.
- [86] Global Grid Forum. Gridftp introduction. Technical report, Global Grid Forum.
- [87] I Foster and al. A security architecture for computational grids. In *ACM COnference on Computer and Security*.
- [88] Ian Foster and Carl Kesselman. Globus : A metacomputing infrastructure toolkit. *The International Journal of Supercomputer Applications and High Performance Computing*, 11(2) :115–128, Summer 1997.
- [89] Ian Foster and Carl Kesselman. Globus : A metacomputing infrastructure toolkit. *The International Journal of Supercomputer Applications and High Performance Computing*, 11(2) :115–128, Summer 1997.
- [90] Ian Foster and Carl Kesselman. The globus project : A status report. In IEEE, editor, *The Globus project : A status report*, pages 4–18. Heterogeneous Computing Workshop (HCW '98), Mar 1998.
- [91] Ian Foster and Carl Kesselman, editors. *The Grid : Blueprint for a New Computing Infrastructure*, chapter 22, Testbed : Bridges from Research to Infrastructure. Morgan Kaufmann Publishers, San Francisco, California,, 1998.
- [92] B. Gaidioz and P. Primet. Eds : a new scalable architecture for service differentiation in the internet. Technical Report 4387, INRIA Research Report, feb 2002.
- [93] B. Gaidioz, P. Primet, and B. Tourancheau. The balanced forwarding model for the multimedia internet. Technical report, Research Report RESAM, 2001.

- [94] B. Gaidioz, P. Primet, and B. Tourancheau. Differentiated fairness : a new soft differentiated service model. In *Conférence internationale IEEE High Performance Switching and Routing*, may 2001.
- [95] M. Galvao. Intégration dun modeleur mono-utilisateur dans lenvironnement coopératif ecoop. Master's thesis, mémoire DEA, Ecole Centrale de Lyon, France, 1997.
- [96] M. Galvao and P. Primet. *GEO : un modeleur 3D coopératif*.
- [97] J.P. Gelas and L. Lefevre. Tamanoir : A high performance active network framework. *Active Middleware Services*, August 2000.
- [98] S. Greenberg and M. Roseman. Groupware toolkits for synchronous work. In Michel Beaudouin-Lafon, editor, *Computer-Supported Cooperative Work*, chapter 6, pages 135–168. John Wiley and Sons Ltd., trends in software series edition, 1999.
- [99] Pittsburgh Supercomputer Center Networking Group. Enabling high performance data transfers on hosts. Technical report, Pittsburgh Supercomputer Center Networking Group.
- [100] J. GRUDIN. Eight challenges for developers. *Communications of the ACM*, 37(1) :93–104, 1991.
- [101] J. Grundy. Engineering component-based, user-configurable collaborative editing systems. In *proceedings of EHCI'98, IFIP Working Conference on Engineering for HCI*, pages 111–126. Kluwer Academic Publishers.
- [102] R. Guerin and V. Peris. Quality of service in packet networks : Basic mechanisms and directions. In *Computer Networks*, volume 31, pages 169–189, February 1999.
- [103] Roch Guerin and Henning Schulzrinne. *The Grid : Blueprint for a New Computing Infrastructure*, chapter Network quality of service. Morgan Kaufmann Publishers, San Francisco, California, 1998.
- [104] T.J. Hacker and D. Athey. The end to end performance effects of parallel tcp sockets on a lossy wide-area network.
- [105] R. Harakaly, P. Primet, F. Bonnassieux, and Gaidioz B.
- [106] Juha Heinanen, Fred Baker, Walter Weiss, and John Wroclawski. Assured forwarding PHB group. Internet Request For Comments RFC 2597, Internet Engineering Task Force, June 1999.
- [107] M. Herbert, P. Primet, B. Tournacheau, and L. Lefevre. A distributed architecture for a scalable ethernet switch based on myrinet technology. In *to appear in proceedings of the IEEE High Performance Switching and Routing HPSR*, 2002.
- [108] Marc Herbert. Vers une architecture de commutation distribuée. Master's thesis, École Normale Supérieure de Lyon, Lyon, France, July 2001.
- [109] M. Hicks, P. Kakkar, J. T Moore, and S. Gunter, C. A. and Nettles. PLAN : A programming language for active networks. *ACM SIGPLAN Notices*, 34(1) :86–93, 1999.
- [110] QBSS home page. <http://qbone.internet2.edu/qbss>. <http://qbone.internet2.edu/qbss>.
- [111] <http://dast.nlanr.net/Projects/Iperf>.
- [112] <http://www.ncne.nlanr.net/nimi>. National internet measurement infrastructure home page.
- [113] <http://www.ripe.net/cgi-bin/gttm/pod>. Ripe ncc home page.
- [114] Paul Hurley and Jean-Yves Le Boudec. A proposal for an asymmetric best-effort service. In *Proceedings of IWQoS'99* [3], pages 132–134.
- [115] Paul Hurley, Jean-Yves Le Boudec, Maher Hamdi, Ljubica Blazevic, and Patrick Thiran. The asymmetric best-effort service. Technical Report SSC/1999/003, EPFL-DI-ICA, January 1999.
- [116] ITU-T. Transmission systems and media, general recommendation on the transmission quality for an entire international telephone connection ; one way transmission time. Technical report, Geneva Switzerland, march 1993.
- [117] ITU-T. Video codec for audiovisual services at p*64kb/s. In *Recommendation H.261*, March 1993.
- [118] V. Jacobson. Congestion avoidance and control. In *ACM Computer Communication Review*, volume 18.
- [119] Van Jacobson, Kathleen Nichols, and Kedar Poduri. An expedited forwarding phb. Internet Request For Comments RFC 2598, Internet Engineering Task Force, June 1999.

- [120] Lea A. Maeda C. Kiczales G., DeLine R. Open implementations analysis and design. tutorial notes. In *Proceedings of the ACM conference OOPSLA'95*, 1995.
- [121] L. Kleinrock. *QQueing Theory*, volume 2. 1976.
- [122] G.E. Krasner and S.T. Pope. A cookbook for using the model-view controller user interface paradigm in smalltalk-80. *Journal of Object-Oriented Programming*, 1(3) :26–49, 1988.
- [123] V. Kumar. The mbone information web homepage.
- [124] C. Lauwers, T. Joseph, K. Lantz, and A. Romanow. Replicated architectures for shared window systems : A critique. In *Proceedings Conference on Office Information Systems*, pages 249–260, apr 1990.
- [125] T. Lavian, R. Jaeger, and J. Rey. Open programmable architecture for java-enabled network devices. In *In Proc. of the Seventh IEEE Workshop on Hot Interconnects*, Stanford University CA, August 1999.
- [126] Craig A. Lee, Rich Wolski, James Stepanek, Carl Kesselman, and Ian Foster. A network performance tool for grid environments. pages??–??, 1999.
- [127] J Lee and al. Applied techniques for high bandwidth data transfers accross wide area networks. Technical report, Lawrence Berkeley National Laboratory, dec 2000.
- [128] L. Lefevre, C Pham, P. Primet, B. Tourancheau, B Gaidioz, J.P. Gelas, and Maimour M. Active grid. In *IWAN : International Workshop of Active Networking*, october 2001.
- [129] M. Leung and al. Characterization and performance evaluation for proportional delay differetiated services, 1997.
- [130] D.and R. MUNTZ LI. Coca : Collaborative objects coordination architecture. In *proceedings of the ACM CSCW'98*, Seattle, 1998.
- [131] J. S. Madhavi. Tcp-friendly unicast based flow control. In *Technical Note*, <http://ftp.ee.lbl.gov/floyd/papers.html>, june 1997.
- [132] M. Mathis and S. Floyd. Tcp friendly unicast rate based flow control. Technical report, Technical note, Lbl, june 1997.
- [133] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. Tcp selective acknowledgement options. Proposed Standard 2018, IETF, april 1996.
- [134] Martin May, Jean-Chrysostome Bolot, Christophe Diot, and Bryan Lyles. Reasons not to deploy RED. In *Proceedings of IWQoS'99* [3].
- [135] Martin May, Jean-Chrysostome Bolot, Alain Jean-Marie, and Christophe Diot. Simple performance models of differentiated services schemes for the internet. In *INFOCOM (3)*, pages 1385–1394, 1999.
- [136] S. McCanne. Scalable compression and transmission of internet multicast video. In *PhD thesis*, University of California, 1996.
- [137] S. McCanne, V. Jacobson, and M. Vetterli. Receiver-driven layered multicast. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, volume 26,4, pages 117–130, New York, August 1996. ACM Press.
- [138] Steve McCanne and Van Jacobson. Vic : A flexible framework for packet video. In *ACM Multimedia*, nov 1995.
- [139] J.P. Munson and P. Dewan. A concurrency control framework for collaborative systems. In *Proceedings of ACM Conference on Computer-Supported Cooperative Work*, pages 278–287, 1996.
- [140] A. Nakajima. Telepointing issues in desktop conferencing systems. *Computer Communications*, 16(9) :603–610, sep 1993.
- [141] R. Newman-Wolfe, M. Webb, and M. Montes. Implicit locking in the ensemble concurrent object-oriented graphics editor. In *Conference on Computer Supported Collaborative Work ACM*, pages 265–272, Toronto, nov 1992.
- [142] Kathleen Nichols, Steven Blake, Fred Baker, and David Black. Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers. Internet Request For Comments RFC 2474, Internet Engineering Task Force, December 1998.

- [143] Network simulator ns-2 web site. <http://www.isi.edu/nsnam/ns/>.
- [144] T. O'Grady. Flexible data sharing in a groupware toolkit. M.sc. thesis, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada, nov 1996. 126 pages.
- [145] T. O'GRADY. *Flexible Data Sharing in a Groupware Toolkit*. M.sc. thesis, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada, 1998.
- [146] Primet P. Qualité de service et flexibilité dans les applications coopératives : l'approche cotools. *Calculateurs Parallèles*, 2001.
- [147] Primet P. and Akkouche S. Telepointer : Analysis and propositions. *Technique et Science Informatiques*, (10), 1998.
- [148] J. Padhye, V. Firiou, D. Towsley, and J. Kurose. Modeling tcp throughput : a simple model and its empirical validation. In *ACM SIGCOMM*. ACM press, september 1998.
- [149] NistNet Home page. <http://qbone.internet2.edu/qbss>.
- [150] RTPL Home Page. <http://fseven.phys.uu.nl/blom/rtpl/index.htm>.
- [151] P. Pan and H. Schulzrinne. Lightweight resource reservation signaling : Design, performance and implementation.
- [152] Cisco White Paper. Ip qos in cisco routers. Technical report, <http://www.cisco.com>, 2000.
- [153] A. Parekh and R. Gallager. Generalized processor sharing approach to flow control in integrated services networks : the single node case. 1(3).
- [154] Steve Parker and Chris Schmechel. Some testing tools for TCP implementors. Internet Request For Comments RFC 2398, Internet Engineering Task Force, August 1998.
- [155] J.F. PATTERSON and al. Rendez-vous : an architecture for synchronous multi-user applications. In *proceedings of the ACM Conference on Computer Supported Collaborative Work*, pages 317–328. ACM, 1990.
- [156] J.F. Patterson, M. Day, and J. Kucan. Notification servers for synchronous groupware. In *CSCW'96 Proceedings*, pages 122–129, Cambridge MA USA, 1996.
- [157] V. Paxson and S. Floyd. Why we don't know how to simulate the internet. In *Winter Simulation Conference*, pages 1037–1044, 1997.
- [158] Vern Paxson, Guy Almes, Jamshid Mahdavi, and Matthew Mathis. Framework for IP performance metrics. Internet Request For Comments RFC 2330, Internet Engineering Task Force, May 1998.
- [159] W.G. Phillips. Architectures for synchronous groupware. Technical Report 1999-425, Department of Computing and Information Science, Queen's University, Kingston, Ontario, Canada, may 1999. 53 pages.
- [160] P. Primet. Grid networking : the network element concept. In *4th DataGRID workshop ; Paris, Mars 2002*.
- [161] P. Primet. Monitoring réseau pour la grille. In *Journée Grid@INRIA - ENS Lyon - Janvier 2002*.
- [162] P. Primet. Ecoop : une infrastructure d'accueil et d'évaluation d'applications coopératives. Technical report, Research Report - GRACIMP-ECL, feb 1995.
- [163] P. Primet. Contrôle de concurrence dans les collecticiels. Technical report, Research Report - GRACIMP ECL, jul 1996.
- [164] P. Primet. Contrôle de concurrence dans les collecticiels : mise en uvre de la flexibilité. In *Actes de CRAC96-Contrôle Réparti dans les Applications Coopératives*, pages 31–36, Paris, jun 1996.
- [165] P. Primet. The datagrid and geant project collaboration agreement. *DataGRID Newsletter*, (1), oct 2001.
- [166] P. Primet. High performance grid networking : the experience of datagrid project. In *to appear in the proceedings of the European Conference TERENA*, Limerick, june 2002.
- [167] P. Primet. Le projet e-toile : une plate-forme grille haute-performance. In *Conférence invitée au Séminaire Aristote*. Ecole Polytechnique, Paris, apr 2002.
- [168] P. Primet and S. Akkouche. Le télépointeur. In *Actes IHM94. Lille septembre 94*.
- [169] P. Primet and S. Akkouche. Telepointer : Analysis and propositions. october 1998.

- [170] P. Primet and R. Chalon. Collaborative engineering on atm :a case study. In *Proceedings of the International Conference on Computer Supported Collaborative Work in Design (CSCWD 99)*, Compiègne, sep 1999.
- [171] P. Primet, D. Drif, J. Rio, and J.F. Fleury. *CoTools :a flexible groupware toolkit*. <http://www.ens-lyon.fr/pprimet/CoTools>, 2000.
- [172] P. Primet, B. Gaidioz, J.P. Gelas, and L. Lefevre. Dynamic configuration of diffserv routers with active technology. In *soumis International Conference INET2002*, august 2002.
- [173] P. Primet and R. Harakaly. Experiment of the nws (network weather service) network forecasting for grid networking. In *to appear in the proceedings the proceedings of the IEEE Conference on Cluster Computing and Grid2002*, Berlin, june 2002.
- [174] P. Primet, F. Tarpin-Bernard, D. Drif, P. Lacaze, and S. Akkouche. *ECoop : une plate-forme coopérative*. Laboratoire ICTT, Ecole Centrale de Lyon.
- [175] Network Time Propocol. rfc, ietf, 1998.
- [176] K. Psounis. Active networks : Applications, security, safety, and architectures. *IEEE Communications Surveys*.
- [177] L. Qiu and al. On individual and aggregate tcp performance. In *International Conference on Network Protocols*, number 7.
- [178] Primet P.and Chalon R. Collaborative engineering on atm :a case study. In *proceedings of the International Conference on Computer Supported Collaborative Work in Design*, Compiègne, Sept 99.
- [179] R. Rajan and al. A policy framework for integrated and differentiated services in the internet. 13(5).
- [180] K. Ramakrishnan and S. Floyd. A proposal to add explicit congestion notification (ecn) to ip. rfc 2481, ietf, june 1999.
- [181] M. Raynal. Gestion des données réparties : problèmes et protocoles. In Eyrolles, editor, *Introduction aux systèmes répartis*, France.
- [182] Huan Ren and Kihong Park. Toward a theory of differentiated services. In *Proceedings of IWQoS'00* [4].
- [183] R. Ribler and al.
- [184] J. Roberts. Traffic theory and internet. Technical report, IEEE communication magazine, jan 2001.
- [185] T. Rodden. Cscw and distributed systems : the problem of control. In *ECSCW '91 Proceedings*, Amsterdam, 1991. Kluwer Academic Press.
- [186] M. Roseman and S. Greenberg. Groupkit, a groupware toolkit for building real-time conferencing applications. In *ACM 1992 - Conference on Computer Supported Collaborative Work*, pages 43–50, Toronto, nov 1992.
- [187] M. Roseman and S. Greenberg. Building real-time groupware with groupkit, a groupware toolkit. *ACM TOCHI*, 3(1) :66–106, mar 1996.
- [188] M. ROSEMAN and S. GREENBERG. Building real-time groupware with groupkit, a groupware toolkit. *ACM TOCHI*, 3(1) :66–106, 1996.
- [189] S. Sahu. On achievable service differentiation with token bucket marking for tcp. In *ACM SIGMETRICS'00*, Santa Clara, CA, june 2000.
- [190] V. Sander, I. Foster, and L. Winkler. A differentiated services implementation for high performamnce tcp flows. *Computer Networks*, 34(, pages =).
- [191] N. Seddigh, B. Nandy, and P. Piedad. Bandwidth assurance issues for tcp flows in a differentiated services network. In *IEEE Globecom*, december 1999.
- [192] Jeffrey Semke, Jamshid Mahdavi, and Matthew Mathis. Automatic TCP buffer tuning. In *SIGCOMM*, pages 315–323, 1998.
- [193] S. Shenker. Fundamental design issues for the future internet. 13(7).
- [194] Scott Shenker, Craig Partridge, and Roch Guerin. Specification of guaranteed quality of service. Internet Request For Comments RFC 2212, Internet Engineering Task Force, September 1997.
- [195] Traffic Specification. rfc 3148, ietf, jun 1999.

- [196] W. Stalling. Isdn and broadband isdn. In *MacMillan Publishing Company*. 1992.
- [197] R. Steinmetz and K. Nahrstedt. *Multimedia : computing, communications and applications*. Prentice Hall, 1995.
- [198] W. Stevens. Tcp slow start, congestion avoidance, fast retransmit and fast recovery algorithms. Technical report.
- [199] I. Stoica, S. Shenker, and H. Zhang. Corestateless fair queuing : Achieving approximately fair bandwidth allocations in high speed networks. In *ACM Computer Communication Review*, volume 28, pages 118–130. ACM press, september 1998.
- [200] F. Tarpin-Bernard. *Travail coopératif synchrone assisté par ordinateur : Approche AMF-C*. PhD thesis, Thèse de doctorat, Ecole Centrale de Lyon, 1997.
- [201] F. Tarpin-Bernard and B.T. David. Amf a new design pattern for complex interactive software? In *International HCI97 Proceedings, Design of Computing Systems*, pages 351–354, San Francisco, aug 1997. Elsevier.
- [202] F. Tarpin-Bernard, B.T. David, and P. Primet. Frameworks and patterns for synchronous groupware : Amf-c approach. In Chatty S. and Dewan P., editors, *proceedings of EHCI'98, IFIP Working Conference on Engineering for HCI*, pages 225–242. Kluwer Academic Publishers, 1998.
- [203] R. Tasker, P. Primet, F. Bonnassieux, and P. Meador. Network monitoring architecture. Technical report, EU DATAGRID report Deliverable D7.2 - Approved by the EC January 2002, 2002.
- [204] Robin Tasker, Pascale Primet, and Franck Bonnassieux. Network monitoring architecture. Technical report, IST EDG DataGrid project, 2002.
- [205] B. Teitelbaum and al. Internet2 qbone : Building a testbed for differentiated services. 13(5) :8–16, october 1999.
- [206] Jonathan M. Tennenhouse, David L. and Smith, W. David Sincoskie, and Gary J Wetherall, David J. and Minden. A survey of active network research. *IEEE Communications Magazine*, 35(1) :80–86, 1997.
- [207] TOLONE, S.M.KAPLAN, and G. FITZPATRICK. Specifying dynamic support for collaborative work within worlds. In *proceedings of the ACM Conference on Organizational Computing Systems (COOCS'95)*, Milpitas CA., 1995. ACM.
- [208] TSpec. Tspec. rfc 2215, IETF, 1999.
- [209] J. VACHERAND-REVEL. Le travail coopératif médiatisé et distant : ressources et contraintes pour l'interaction interhumaine. *Psychologie du travail et des organisations*, 5(1-2) :206–224, dec 1999.
- [210] Sudharshan Vazdkudai, Jennifer J. Schopf, and Ian Foster. Predicting the performance of wide area data transfers.
- [211] L. Vicisano and, L. Rizzo and J. Crowcroft. Tcp-like congestion control for layered multicast data transfer. In *conference on Computer Communication, IEEE Infocom*, San Francisco, March 1998.
- [212] A. Vogel, B Kerhervé, G. von Bochmann, and J. Gecsei. Distributed multimedia and qos : a survey. In *IEEE Multimedia*, pages 10–18. IEEE, july 1995.
- [213] F. Vraalsen, R.A. Aydt, C.L. Mendes, and D.A. Reed. Performance contracts : Predicting and monitoring grid application behavior. In *International Workshop on Grid Computing*, volume GRID2001.
- [214] web100 home page. <http://www.web100.org/>. <http://www.web100.org/>.
- [215] John V. end Tennenhouse David L. Wetherall, David T. end Gutttag. Ants : A toolkit for building and dynamically deploying network protocols. In *In Proceedings of IEEE Openarch'98*, April 1998.
- [216] F. Wilson, I. Wakeman, and W. Smith. Quality of service parameter for commercial application of video telephony. In *Human Factors In Telecommunication Symposium*, march 1993.
- [217] Richard Wolski. Dynamically forecasting network performance using the network weather service. *Cluster Computing*, 1(1) :119–132, 1998.
- [218] Richard Wolski, N. Spring, and Jim Hayes. The network weather service : A distributed resource performance forecasting service for metacomputing. In *Future Generation Computer Systems (to appear)*, 1998.

- [219] John Wroclawski. Specification of the controlled-load network element service. Internet Request For Comments RFC 2211, Internet Engineering Task Force, September 1997.
- [220] John Wroclawski. The use of RSVP with IETF integrated services. Internet Request For Comments RFC 2210, Internet Engineering Task Force, September 1997.
- [221] C. Wu and D. Irwin. *Emerging Multimedia Computer Communication Technologies*. Prentice Hall, 1998.
- [222] N. Yeom, I. Reddy, D. Wetherall, and G. J. Minden. Impact of marking strategy on aggregated flows in a differentiated services network.
- [223] W. Yeong, T. Howes, and S. Kille. Lightweight directory access protocol. Technical Report 1777, March 1995.
- [224] Hui Zhang. Service disciplines for guaranteed performance service in packet-switching networks. *Proceedings of the IEEE*, 83(10) :1374–1396, October 1995.
- [225] W. Zhao, D. Olshefski, and H. Schulzrinne. Internet quality of service : an overview. Technical report, Columbia.
- [226] H. Zimmerman. OSI reference model - the ISO model of architecture for open systems interconnection. *IEEE trans. on commun.*, pages 425–432, 1980.

Table des figures

1.1	Modèle de réseau actuel avec des routeurs haute performance dans le coeur et des routeurs d'accès en périphérie	9
1.2	Evolution des performances et de la complexité dans l'interconnexion réseau	10
2.1	Modèle de l'environnement coopératif CoTools	16
2.2	Architecture générale de CoTools	17
2.3	Modèle architectural de CoTools	18
2.4	Exemple de la scène 3D animée construite	19
2.5	Principe d'un contrôle optimiste et d'un contrôle pessimiste	20
2.6	Le protocole de contrôle et de notification NCP	20
2.7	Automate du protocole NCP	21
2.8	Le modèle AMF	22
2.9	Couplage du modèle AMF-C et de CoTools	22
2.10	Vues multiples du modeleur géométrique 3D coopératif GEO	23
2.11	Localisation de nos contributions dans le domaine du CSCW	24
2.12	L'outil télépointeur dans un modeleur 3D coopératif	25
2.13	Utilisation de la fonction masque de téléPTR	26
3.1	Infrastructure de communication du projet P2-ATM	29
3.2	Dispositif de collaboration	30
3.3	Synchronisation intermédia dans P2-ATM	30
4.1	Architecture actuelle d'un routeur Internet type	42
4.2	Architecture DiffServ	47
4.3	Définition du champ TOS (A) et du Définition du DSCP (B)	47
4.4	Evolution du délai et du taux de perte pour maintenir l'équité	50
4.5	Ordonnanceur de BF	51
4.6	Mise en évidence de la différenciation en délai	51
4.7	Réseau mis en place pour les expérimentations	52
4.8	Comportement des flux TCP-QBSS en présence de trafic Best Effort (A), en l'absence de trafic Best Effort (B)	53
4.9	Principe de traitements des flux vidéo dans JMF	58
4.10	Architecture générale de netstre@mer	60
5.1	Complémentarité des approches <i>sensibles</i>	62
5.2	L'ordonnanceur de EDS	67
5.3	Performances de connexions TCP sur un réseau classique et sur un réseau haut débit à latence élevée	69
5.4	Performances de connexions TCP sur quatre classes	70
5.5	Flux temps réel sur IP Best effort et sur EDS	71
5.6	Structure du paquet ANEP	72
5.7	Code d'un service Tamanoir	73
5.8	Service actif de configuration dynamique de routeur DiffServ	73

5.9	Service de supervision actif tracer	74
5.10	Extrait du code du service Tracert	75
5.11	API du service de sauvegarde d'états	75
6.1	Vue logique d'une grille	81
6.2	Vue IP de la grille	82
6.3	Modèle logique d'un Network Element	83
6.4	Architecture de supervision de grille	85
6.5	Structure arborescente de DataGRID	87
6.6	Passage du jeton et ajustement de la période de rotation	88
6.7	Periodicite du protocole de clique	89
6.8	MapCenter overview	89
6.9	Modèle relationnel entre les quatre entités de la couche présentation	90
6.10	Principe de l'architecture distribuée d'équipement réseau	92
6.11	Prototype développé	93
6.12	Performances en émission obtenues sur le prototype	94
6.13	Principe du middle-harware distribué de grille	95
6.14	Vue logique du réseau de DataGRID (avec Mapcenter)	96

Liste des tableaux

2.1	Scénario de production coopérative d'une scène 3D et mode de couplage associés	19
3.1	Domaines d'application des catégories de services ATM (source ATM forum)	34
3.2	Classification des flux selon la dimension temporelle	35
3.3	Classes AAL et exemples d'utilisation (source ATM forum)	36
3.4	Paramètres de spécification des services ATM selon les dimensions (source ATM forum)	36
4.1	Récapitulatif des mécanismes de QoS	42
4.2	Utilisations du champ TOS	47
4.3	Tableau comparatif des approches de QoS IP traditionnelles	48
4.4	Tableau comparatif des approches de QoS IP relatives et de IP Best Effort	49
4.5	Propositions de services Best Effort Amélioré	49
5.1	Deux type d'initialisation des coefficients de proportionnalité	66