

Este exemplar corresponde à redação final da Tese/Dissertação devidamente corrigida e defendida por: RODRIGO NEIVA PEDATELLA
e aprovada pela Banca Examinadora.
Campinas, 24 de maio de 2009
[Assinatura]
COORDENADOR DE PÓS-GRADUAÇÃO
CPG-IC

200407954

**Um Modelo de Negociação entre
Bandwidth Brokers para Oferta de QoS
Fim-a-Fim**
Rodrigo Neiva Pedatella
Dissertação de Mestrado

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE

Um Modelo de Negociação entre Bandwidth Brokers para Oferta de QoS Fim-a-Fim

Rodrigo Neiva Pedatella¹

28 de outubro de 2003

Banca Examinadora:

- Prof. Dr. Edmundo Roberto Mauro Madeira (Orientador)
- Prof. Dr. Djamel F. H. Sadok
Centro de Informática - Universidade Federal de Pernambuco
- Prof. Dr. Nelson Luis Saldanha da Fonseca
Instituto de Computação - Universidade Estadual de Campinas
- Prof. Dr. Paulo Lício de Geus
Instituto de Computação - Universidade Estadual de Campinas

¹ Suporte financeiro do CNPq.

IDADE BC
CHAMADA T/UNICAMP
P34m
EX
MBO BC/ 58210
DC 16-117-04
D X
EÇO R\$ 11,00
TA 28/5/04
CPD

CM00198026-0

18 10 316765

Pedatella, Rodrigo Neiva

P34m Um modelo de negociação entre Bandwidth Brokers para oferta de QoS fim-a-fim / Rodrigo Neiva Pedatella – Campinas, [S.P. :s.n.], 2003.

Orientador : Edmundo Roberto Mauro Madeira.

Co-orientador: Maurício Ferreira Magalhães.

Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

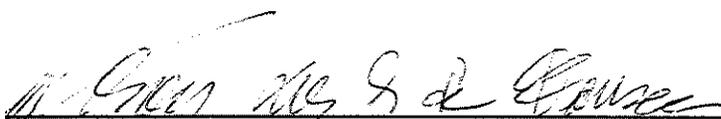
1. Internet (Redes de computação). 2. Programação (Computadores) - Gerência. 3. Negociação. I. Madeira, Edmundo Roberto Mauro. II. Magalhães, Maurício F.(Maurício Ferreira). III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

TERMO DE APROVAÇÃO

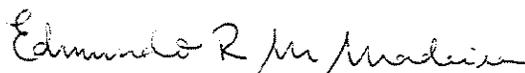
Tese defendida e aprovada em 26 de novembro de 2003, pela Banca examinadora composta pelos Professores Doutores:



Prof. Dr. Djamel Fawzi Hadj Sadok
UFPe



Prof. Dr. Nelson Luis Saldanha da Fonseca
IC - UNICAMP

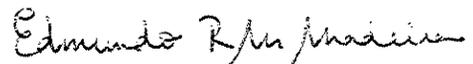


Prof. Dr. Edmundo Roberto Mauro Madeira
IC - UNICAMP

Um Modelo de Negociação entre Bandwidth Brokers para Oferta de QoS Fim-a-Fim

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Rodrigo Neiva Pedatella e aprovada pela Banca Examinadora.

Campinas, 26 de novembro de 2003.



Prof. Dr. Edmundo Roberto Mauro Madeira
(Orientador)



Prof. Dr. Maurício Ferreira Magalhães
(Co-orientador)

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

© Rodrigo Neiva Pedatella, 2003.
Todos os direitos reservados.

Resumo

Considerando a ascendente popularização que a Internet vem acumulando no papel de importante veículo de comunicação, muitas das atuais aplicações não mais têm encontrado neste ambiente uma infra-estrutura adequada para o funcionamento ideal. Deste modo, o estudo de soluções capazes de transformar a Internet em um ambiente adaptado ao tráfego de informações em diferentes níveis de qualidade de serviço (QoS) tem recebido grande destaque por parte da comunidade acadêmica. Neste contexto, uma proposta de negociação de serviços com requisitos distintos de QoS mostra-se bastante adequada na medida em que certamente contribui para garantir a QoS necessária às referidas aplicações de maneira permanente e controlada.

Este trabalho apresenta uma proposta para comunicação entre domínios distintos por meio de um modelo de negociação que visa automatizar as transações de QoS, inclusive entre domínios administrativos, numa busca por QoS fim-a-fim. Para realizar essa tarefa, o referido modelo de negociação a ser proposto é embutido em uma infra-estrutura de QoS simulada já existente. O objetivo principal é o estudo e a implementação de mecanismos de negociação que minimizem transmissões fim-a-fim realizadas através do melhor esforço, utilizando para tal reservas de recursos baseadas em políticas.

Para a especificação do modelo em nível operacional, serão utilizados conceitos presentes em protocolos como o *Common Open Policy System* (COPS), o *Border Gateway Routing Protocol Plus* (BGRPP) e o *Simple Inter-Domain Bandwidth Broker Signalling* (SIBBS). A idéia é que seja feita uma reunião de boas práticas existentes, tendo como foco a adaptação para o contexto de negociações dinâmicas e transparentes entre domínios. Como diferencial, o nível gerencial do modelo apresenta uma arquitetura de políticas que busca atribuir um maior grau de abstração para o administrador da rede, seguindo preceitos do paradigma de administração baseado em políticas de configuração.

Abstract

Considering the increasing Internet popularity as an important communication vehicle, many of the current applications don't find in this environment a suitable infrastructure for the ideal execution anymore. This way, the study of solutions that could be able to transform the Internet into an adapted environment to the information traffic with different levels of quality of service (QoS) has received great attention by the academic community. In this context, a service negotiation proposal with distinct QoS requirements shows up to be very appropriated since it certainly contributes to guarantee the necessary QoS to the mentioned applications in a permanent and controlled manner.

This work presents a proposal of negotiation model among distinct domains to automate QoS transactions on an effort for end-to-end QoS acquirement. To this, the proposed communication model is embedded in an existing simulated QoS infrastructure. The main goal is the study and implementation of negotiation mechanisms that can minimize end-to-end transmissions based on best effort service, making use of policy based resource reservation to achieve this.

For the model's operational level specification, concepts present on protocols like the Common Open Policy System (COPS), the Border Gateway Routing Protocol Plus (BGRPP), and the Simple Inter-Domain Bandwidth Broker Signalling (SIBBS) will be used. The idea is that a collection of good concepts can be made, focusing the adaptation to a dynamic and transparent negotiation context among domains. As a contribution, the model's managerial level presents a policy architecture which aims at the attribution of a higher abstraction degree to the network administrator, following rules of the administration paradigm based on configuration policies.

Agradecimentos

Em tão especial ocasião, não poderia deixar de exaltar a importância fundamental de meus queridos e saudosos pais no árduo caminho que vem me levando a mais essa grande conquista. Pai, mãe, a falta que me fazem certamente não pode ser descrita em palavras. Contudo, me conforta a certeza de que a alegria e satisfação de vocês, aonde quer que estejam, é tão grande quanto a gratidão e a saudade que sinto. Estejam certos de que a paixão, o que julgo ser um grande legado que nos deixaram, continuará presente em cada passo de minha vida.

Também gostaria de agradecer com muito carinho à minha noiva, Fernanda. Amor, você bem sabe o quanto foi difícil para nós ficarmos todo esse tempo separados por tantos quilômetros. Sem dúvidas devo grande parte do sucesso obtido nessa empreitada a você. Muito obrigado pelo apoio, paciência, compreensão, carinho, dedicação, enfim, pelo amor que sempre me oferece de maneira tão especial.

Aos meus queridos irmãos, Viviane e João Marcelo, agradeço pela maneira valente com que lidaram com a situação difícil pela qual passamos durante o período da doença de mamãe. Obrigado pela imensa compreensão, pelo carinho e pelas inúmeras palavras de incentivo e apoio. Ao meu cunhado Márcio, agradeço pela função conciliadora que mais uma vez soube tão bem exercer em nossa família durante dias tão difíceis.

Aos meus ex-companheiros de república, Henrique, Thaisa e Silvania, gostaria de agradecer pela convivência saudável com que me presentearam durante dois longos anos. A companhia de vocês certamente tornou a distância de casa menos dolorida.

Ao meu orientador Edmundo Madeira e ao meu co-orientador Maurício Magalhães, agradeço pelo apoio técnico de qualidade impecável. Aos colegas Alex Guillén, Leonardo Costa, José Carrilho, Tullius Ribeiro, Marcos Siqueira e Antônio Abelém sou muito grato pelo suporte prestado durante a execução deste trabalho, em especial durante o período da implementação.

Ao CNPq, agradeço pelo apoio financeiro.

“Mas é preciso ter força, é preciso ter raça, é preciso ter gana, sempre...”
Para uma corajosa, admirável e inesquecível Maria, minha mãe.

Sumário

Resumo	i
Abstract	ii
Agradecimentos	iii
Lista de Figuras	viii
Lista de Tabelas	ix
Lista de Acrônimos	x
1 Introdução	1
2 Qualidade de Serviço na Internet	5
2.1 Serviços Diferenciados (DiffServ)	6
2.2 Troca de Rótulos Multiprotocolar (MPLS)	8
2.3 Administração Baseada em Políticas	10
2.4 Bandwidth Broker (BB)	11
2.5 Qualidade de Serviço Fim-a-Fim	11
2.6 Trabalhos Relacionados	13
3 Negociação de Serviços com QoS na Internet	15
3.1 Protocolos de Negociação	16
3.1.1 O protocolo COPS	17
3.1.2 O Protocolo BGRPP	18
3.1.3 O Protocolo SIBBS	20
3.2 Comunicação entre Domínios	23
3.2.1 Modelo Estrela	23
3.2.2 Modelo Hierárquico	24
3.2.3 Modelo Cascata	25
3.3 Contratos de Serviços	26

3.3.1	Requisitos de um Contrato	26
3.3.2	Representação e Armazenamento de Contratos	27
3.4	Resumo	28
4	<i>Framework</i> para Obtenção de QoS Fim-a-Fim	31
4.1	Arquitetura do <i>Bandwidth Broker</i>	32
4.1.1	Repositório de Dados	33
4.1.2	Gerente de Recursos	33
4.1.3	Gerente de Políticas	33
4.1.4	Gerente de Negociação	34
4.1.5	Objeto de Configuração	34
4.1.6	Objeto de Aplicação	35
4.1.7	Objeto de Gerenciamento	35
4.2	A Plataforma MPLS-DS	35
4.2.1	O <i>Network Simulator</i> (NS)	36
4.2.2	O Pacote DS-Nortel	38
4.2.3	O Pacote MNS	39
4.2.4	Integração entre DS-Nortel e MNS	39
4.2.5	Monitoramento do Tráfego Simulado	40
4.3	Comunicação entre Componentes do <i>Framework</i>	40
4.4	Arquitetura de Políticas	41
4.4.1	Políticas de Inicialização	41
4.4.2	Políticas de Atuação	41
4.4.3	Políticas de Negociação	42
4.5	Resumo	42
5	Proposta de Modelo de Negociação	43
5.1	O Protocolo SLA-COPS	44
5.1.1	Fase de Inicialização	45
5.1.2	Fase de Configuração	45
5.1.3	Fase de Negociação	46
5.2	Negociação de Atacado/Varejo	46
5.3	Políticas de Negociação	48
5.3.1	Políticas de Admissão	48
5.3.2	Políticas de Restrição de Banda	49
5.3.3	Políticas de Renegociação	49
5.3.4	Políticas de Aceitação de Contratos	50
5.4	Especificação de Contrato de Serviços	51
5.4.1	Unidade Comum	51

5.4.2	Unidade de Topologia	52
5.4.3	Unidade de QoS	52
5.5	Implementação	53
5.5.1	Modificações Realizadas no Bandwidth Broker	54
5.5.2	Modificações Realizadas na Plataforma MPLS-DS	56
5.5.3	Repositório de Dados e Contratos de Serviços	57
5.6	Resumo	57
6	Validação do Modelo de Negociação	59
6.1	Tráfego Gerado	59
6.2	Classes de Serviço	60
6.3	Políticas de Atuação	61
6.4	Experimentos Realizados	61
6.4.1	Experimento 1	62
6.4.2	Experimento 2	66
6.5	Considerações Finais sobre as Simulações Realizadas	69
7	Conclusões e Trabalhos Futuros	71
	Referências Bibliográficas	75

Lista de Figuras

2.1	Campo ToS/CoS no pacote IP.	6
2.2	<i>Shim header</i> MPLS.	9
2.3	Arquitetura fundamental de um BB.	12
2.4	Cenário Típico de QoS Fim-a-Fim.	13
3.1	Troca de mensagens no protocolo COPS.	18
3.2	Cenário típico de utilização do protocolo BGRPP.	19
3.3	Funcionamento geral do protocolo SIBBS.	23
3.4	Modelo Estrela.	24
3.5	Modelo Hierárquico.	25
3.6	Modelo Cascata.	25
4.1	Arquitetura do <i>Bandwidth Broker</i>	32
4.2	A Plataforma MPLS-DS e o BB.	36
4.3	A arquitetura do NS.	37
5.1	Operação do protocolo SLA-COPS	47
5.2	Cenário ilustrativo para Políticas de Restrição de Banda	50
5.3	Implementação do BB.	55
6.1	Topologia da simulação	62
6.2	Gráficos do LSP 2000	63
6.3	Gráficos do LSP 2999	63
6.4	Gráficos do LSP 2001	64
6.5	Gráficos do LSP 2002	65
6.6	Gráficos de fim-a-fim	66
6.7	Topologia da simulação	66
6.8	Gráficos do LSP 2000.	67
6.9	Gráficos do LSP 2001.	68
6.10	Gráficos do LSP 2003.	68
6.11	Gráfico de atraso fim-a-fim.	69

Lista de Tabelas

6.1	Características dos tráfegos disponibilizados	60
6.2	Características das classes de serviços	60

Lista de Acrônimos

AF	Assured Forwarding
ASN.1	Abstract Syntax Notation One
BB	Bandwidth Broker
BE	Best Effort
BGP	Border Gateway Protocol
BGRP	Border Gateway Routing Protocol
BGRPP	Border Gateway Routing Protocol Plus
CIM	Class Information Model
COPS	Common Open Policy Service
CORBA	Common Object Request Broker Architecture
CoS	Class of Service
DiffServ	Differentiated Services
DII	Dynamic Invocation Interface
DSI	Dynamic Skeleton Interface
EF	Expedited Forwarding
FEC	Forwarding Equivalence Class
IETF	Internet Engineering Task Force
IntServ	Integrated Services

LDAP	Lightweight Directory Access Protocol
LDP	Label Distribution Protocol
LSP	Label Switched Path
LSR	Label Switched Router
MPLS	Multiprotocol Label Switching
NS	Network Simulator
PCIM	Policy Core Information Model
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PHB	Per-Hop Behavior
PIB	Policy Information Base
QoS	Quality of Service
QPIM	QoS Policy Information Model
RAA	Resource Allocation Answer
RAR	Resource Allocation Request
SE	Service Exchange
SIBBS	Simple Inter-domain Bandwidth Broker Signalling
SLA	Service Level Agreement
SLS	Service Level Specification
SNMP	Simple Network Management Protocol
TLS	Transport Layer Security
ToS	Type of Service
VINT	Virtual InterNetwork TestBed

XML	Extended Markup Language
WKS	Well Known Service
WRR	Weighted Round Robin

Capítulo 1

Introdução

A Internet tradicionalmente fornece a seus usuários o serviço do melhor esforço para a transmissão de dados. Neste serviço, o tráfego é processado o mais rápido possível, não havendo qualquer compromisso com requisitos de qualidade de serviço (QoS). Entretanto, com a rápida transformação da Internet em uma infra-estrutura comercial, e com o conseqüente ganho de importância das aplicações multimídia, a necessidade de QoS tornou-se fundamental para este ambiente.

Várias propostas têm sido apresentadas com objetivo de transformar a Internet em um meio adaptado ao tráfego de informações com QoS. A *Internet Engineering Task Force* (IETF) tem proposto modelos e mecanismos os quais buscam suprir esta necessidade. Dentre estes, devemos ressaltar o modelo de Serviços Integrados (IntServ) [4], o modelo de Serviços Diferenciados (DiffServ) [22, 26], o mecanismo de Troca de Rótulo Multiprotocolar (MPLS) [14], a Engenharia de Tráfego [11] e o Roteamento Baseado em Restrições [13].

Cada uma dessas abordagens possui características específicas e adequadas a situações peculiares. Todavia, duas propostas têm demonstrado maiores vantagens nas pesquisas recentes. O modelo DiffServ promete o oferecimento escalável de diversos serviços com requisitos de QoS específicos. Esta promessa está embasada em características como a simplicidade dos roteadores de núcleo e a facilidade de implementação e oferecimento de serviços. O MPLS, por sua vez, visa permitir um desempenho otimizado à rede através da utilização de técnicas de Engenharia de Tráfego. A característica essencial do MPLS é a modificação do processo de encaminhamento de pacotes através da análise de rótulos, em substituição ao cabeçalho IP. A atribuição de rótulos possibilita um melhor aproveitamento de recursos, minimizando o potencial aparecimento de congestionamentos.

Diversos trabalhos têm demonstrado a eficácia dos modelos DiffServ e MPLS. Os trabalhos [18] e [9] mostram a obtenção de QoS na Internet através do uso conjunto e intercalado das arquiteturas DiffServ e IntServ. O trabalho [29] apresenta uma platafor-

ma que combina DiffServ e MPLS, aliando a questão engenharia de tráfego da última com a questão diferenciação de tráfego da primeira. O trabalho [15] pioneiramente apresenta uma solução para a integração dos modelos MPLS e DiffServ. Uma plataforma que incorpore as características apresentadas por ambos os modelos certamente estará adaptada ao oferecimento de QoS.

A aplicação das idéias e conceitos previstos nos modelos DiffServ e MPLS pode ser realizada através de configuração manual de roteadores. Contudo, se feita desta maneira, a administração da rede se torna uma tarefa exaustiva e pouco eficiente. Por esse motivo é sugerida a figura de um *software* que seja capaz de configurar, gerenciar e distribuir os recursos disponíveis na rede. A esse *software* é dada a denominação de *Bandwidth Broker* (BB).

Dentre as funções exercidas por esse gerente, a função de negociação de recursos apresenta grande relevância no mecanismo de obtenção de QoS fim-a-fim. A negociação e provisão dinâmica de recursos constitui um importante passo na evolução da Internet [25]. A existência de um modelo de negociação fornece aos usuários a capacidade de especificar dinamicamente serviços e QoS desejados.

Um BB possui seu campo de ação limitado a um domínio administrativo, o qual inclui um certo número de roteadores que sofrem a influência do mesmo. Cada domínio possui suas próprias políticas de gerenciamento de recursos. Contudo, a transmissão de dados fim-a-fim pode atravessar dois ou mais domínios administrativos. Deste modo, é de clara percepção que um modelo de negociação deve prover um mecanismo de comunicação entre gerentes de domínios distintos, a fim de que a QoS fim-a-fim possa ser garantida também para estes casos.

Este trabalho propõe o desenvolvimento de um modelo de negociação de serviços em redes IP, além da especificação de um modelo de comunicação entre BBs de domínios administrativos distintos. O objetivo principal é o estudo e a implementação de mecanismos de negociação que minimizem transmissões fim-a-fim realizadas através do melhor esforço, utilizando para tal reservas de recursos baseadas em políticas.

Neste contexto, vale ressaltar a contribuição deste trabalho. A definição de um modelo de negociação entre BBs representa uma questão ainda em aberto, bem como a definição de mecanismos de comunicação entre os mesmos e as interfaces para utilização.

Para o alcance dos objetivos traçados, utilizamos um *framework* de QoS fim-a-fim composto, no nível operacional, por uma plataforma mista MPLS/DiffServ, adaptada ao oferecimento de serviços múltiplos com requisitos de QoS particulares. No nível gerencial utilizamos um BB baseado em políticas de configuração. A este BB é adicionada uma implementação do modelo de negociação proposto.

Pretende-se simular o modelo proposto a fim de que a performance do BB possa ser verificada para diferentes políticas. Para tal intento, é utilizado o software de domínio

público *Network Simulator* (NS) [38]. A *Common Object Request Broker Architecture* (CORBA) [31] é utilizada como ferramenta de implementação parcial do modelo.

Na sequência da dissertação, o Capítulo 2 apresenta uma abordagem sobre a questão QoS na Internet. Serão apresentados e discutidos tecnologias e conceitos utilizados atualmente. O Capítulo 3 discorre sobre negociação de serviços com QoS na Internet. O Capítulo 4 apresenta o *framework* de QoS utilizado neste trabalho, nos níveis gerencial e operacional. No Capítulo 5 é apresentada a proposta de modelo de negociação de que trata este trabalho. O Capítulo 6 é dedicado à validação do modelo proposto e, finalmente, o Capítulo 7 apresenta conclusões e algumas sugestões de trabalhos futuros.

Capítulo 2

Qualidade de Serviço na Internet

No contexto de redes de computadores, qualidade de serviço (QoS) se refere à idéia de que taxas de transmissão, taxas de erro, entre outras características do tráfego possam ser medidas, melhoradas e, com alguma extensão, previamente garantidas [35].

Provedores de serviços tradicionalmente oferecem o mesmo nível de performance a todos os seus consumidores através do serviço do melhor esforço. Contudo, o aumento no uso da Internet resultou em escassez de capacidade da rede, comprometendo o desempenho das aplicações de missões críticas. Ao mesmo tempo, novas aplicações surgiram, sobretudo aplicações multimídia, as quais passaram a exigir um desempenho superior da rede. Como resultado, QoS têm-se mostrado indispensável para a Internet.

QoS não é apenas uma questão de otimização. Pode significar, por exemplo, confiabilidade (mesmo que em detrimento de um desempenho superior). Enfim, QoS está intimamente ligada às características do tráfego considerado. Existem diversos parâmetros que podem ser utilizados em sua medição. Os mais importantes são atraso, variação de atraso, porcentagem de descarte e vazão média. Diferentes aplicações possuem diferentes valores adequados para os parâmetros citados. Garantir QoS a um determinado cliente significa criar mecanismos que mantenham os parâmetros em faixas aceitáveis para o tráfego do mesmo. A padronização de parâmetros promove a criação das chamadas classes de serviços.

A Internet deve grande parte de seu sucesso à simplicidade de idéias existentes em sua arquitetura. O conceito de datagramas, os quais atravessam a rede de maneira independente, sem garantias de ordenação ou entrega de pacotes, confere escalabilidade a esta rede. Deste modo, uma proposta de modelo de QoS deve respeitar tal característica a fim de que possa obter sucesso e aplicabilidade. Neste capítulo, estaremos detalhando as propostas e conceitos mais utilizados e referenciados no que tange a área de qualidade de serviço na Internet.

2.1 Serviços Diferenciados (DiffServ)

O modelo DiffServ propõe o oferecimento de classes de serviços múltiplas com requisitos de QoS específicos para a Internet [22]. O modelo se encarrega de oferecer as classes de serviços com base em duas operações fundamentais: a diferenciação de tráfegos e a agregação de fluxos.

A diferenciação de tráfegos é realizada através de mecanismos de classificação. Tais mecanismos levam em consideração informações contidas nos pacotes. Tipicamente, é considerado o campo *Type of Service* (ToS), presente no cabeçalho de pacotes do protocolo IPv4¹. Outras informações podem ser consideradas para efeito de classificação, como endereço de origem, endereço de destino, entre outras. A Figura 2.1 mostra a disposição do campo ToS/CoS dentro de um pacote IP. São utilizados até 6 *bits* para a especificação de classes de serviços.

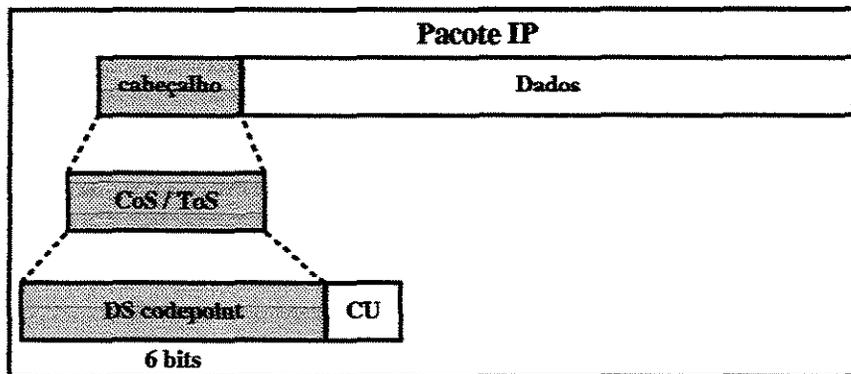


Figura 2.1: Campo ToS/CoS no pacote IP.

A agregação de fluxos tem por objetivo reunir pacotes pertencentes a uma mesma classe para que a eles seja dado o mesmo tratamento no repasse através do chamado *Per-Hop Behaviour* (PHB). Um PHB é definido por um conjunto de parâmetros de configuração que visam garantir os compromissos de QoS peculiares à(s) classe(s) de serviço que representam.

Outras importantes operações previstas pelo modelo são a marcação, o policiamento e o condicionamento do tráfego. A marcação busca atribuir uma classe ao pacote, de acordo com os resultados obtidos no processo de classificação. Mecanismos de policiamento são utilizados para garantir que qualquer tráfego de ingresso esteja dentro de padrões negociados. Tráfegos fora de padrão devem ser condicionados através da execução de algoritmos específicos, como o *Token Bucket* e o *Leaky Bucket*.

¹No caso do IPv6, o campo utilizado é o *Class of Service* (CoS).

A escalabilidade é a principal característica do modelo DiffServ. Através da agregação de diversos fluxos em classes, torna-se possível minimizar o esforço computacional proveniente da diferenciação de serviços. Os roteadores de um domínio DiffServ são divididos em duas categorias: roteadores de borda e de núcleo. Roteadores de borda são aqueles que servem de ingresso para qualquer tipo de tráfego. Realizam as operações de classificação, posicionamento, condicionamento, marcação e repasse dos pacotes. O grande esforço computacional fica restrito a tais roteadores. Roteadores de núcleo, bem mais numerosos, devem conhecer apenas o PHB a ser aplicado a cada tipo de tráfego a fim de que possam executar o repasse coerentemente.

As classes de serviços oferecidas por um domínio DiffServ são transformadas em PHBs através de diferentes parâmetros para as filas (físicas e virtuais) incorporadas aos roteadores do domínio. Pacotes pertencentes a diferentes classes recebem diferentes tratamentos nas filas. A definição e padronização de classes, através de PHBs, faz parte das atribuições do grupo de trabalho DiffServ do IETF. Atualmente, existem três PHBs de classes de serviços padronizados:

- **Expedited Forwarding (EF):** este PHB define o tratamento reservado para tráfegos com requisitos de baixo atraso, variação de atraso e descarte. Apresenta prioridade de tratamento sobre todos os demais.
- **Assured Forwarding (AF):** este PHB oferece baixo descarte e garantias de entrega de pacotes. É ideal para tráfegos que exijam confiabilidade. Define quatro classes de serviço, além de três níveis de descarte para cada classe.
- **Best Effort (BE):** o modelo DiffServ oferece um PHB para suporte ao tradicional serviço do melhor esforço. Para pacotes pertencentes a esta classe, não é dado qualquer tipo de tratamento preferencial.

A transmissão pelo serviço do melhor esforço é realizada para todos os pacotes que não pertençam a qualquer classe de serviço estipulada para o domínio.

Para que um cliente tenha acesso aos serviços de seu provedor DiffServ, ele deve negociar um contrato com o mesmo. Este contrato é denominado *Service Level Agreement* (SLA). Um SLA especifica o tráfego do cliente que deve ser diferenciado, as classes de serviço suportadas, a quantidade de tráfego permitida em cada uma delas e os detalhes técnicos da negociação. Do ponto de vista gerencial, os detalhes técnicos representam a parte mais importante de um SLA, recebendo a denominação de *Service Level Specification* (SLS). No Capítulo 3 entraremos em maiores detalhes a respeito do SLS.

2.2 Troca de Rótulos Multiprotocolar (MPLS)

O modelo MPLS faz uso de conceitos de Engenharia de Tráfego para promover a otimização do processo de roteamento e um melhor aproveitamento dos recursos existentes na rede.

A operação de roteamento tradicional de redes IP é baseada nas chamadas *Forwarding Equivalence Classes* (FECs). Dois pacotes pertencem a uma mesma FEC se existe um mesmo prefixo na tabela do roteador que representa o maior casamento possível com os endereços de destino de ambos os pacotes. Pacotes pertencentes a uma mesma FEC recebem o mesmo encaminhamento.

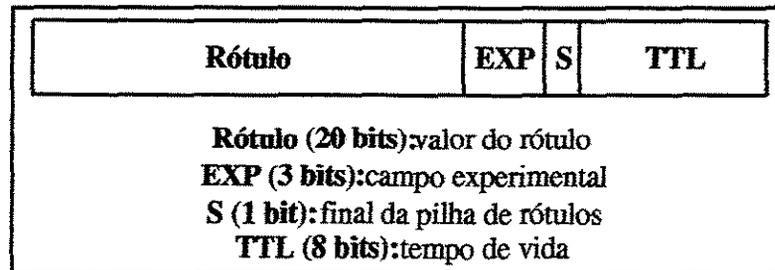
Durante o trajeto pelos nós da rede, cada roteador examina um pacote recebido e o atribui a uma FEC. Os pacotes têm sua rota definida com base em decisão individual e independente por parte de cada roteador a que são submetidos. Neste contexto, é de fácil percepção que o repasse usual de pacotes representa uma atividade de grande carga de processamento. O cabeçalho de um pacote IP possui bem mais informações do que o necessário para simplesmente realizar o repasse. O resultado é que o roteamento acaba por apresentar uma complexidade considerável.

O modelo MPLS busca diminuir esta complexidade através de uma técnica de atribuição de rótulos aos pacotes, trocados subsequentemente pelos roteadores. O rótulo passa, então, a ser a informação de análise em substituição ao cabeçalho dos pacotes. A atribuição de pacotes a uma determinada FEC é feita apenas uma vez, pelo roteador que recebe o pacote na rede. A FEC à qual o pacote foi atribuído é então codificada em um valor curto e de tamanho fixo, que representa o rótulo. Antes do pacote ser enviado ao próximo nó da rede, um rótulo é atribuído e anexado a ele. Após este primeiro momento, não mais ocorre análise do cabeçalho do pacote. Ao invés disso, o rótulo é utilizado como índice de uma tabela, a qual especifica o próximo nó e um novo rótulo. A cada roteador por que passa o pacote, o rótulo antigo é substituído por um novo, de acordo com a tabela de rótulos. A Figura 2.2 mostra a estrutura de um cabeçalho MPLS, denominado *Shim header MPLS*.

De maneira análoga ao que acontece em redes ATM, o conceito de rótulo possibilita a criação de circuitos virtuais sobre a topologia da rede. Tais circuitos são denominados *Label Switched Paths* (LSPs). A introdução de LSPs modifica as características das redes IP, possibilitando um melhor aproveitamento de recursos e minimizando o aparecimento de congestionamentos na rede.

Além do potencial melhor desempenho no roteamento e do melhor aproveitamento de recursos, o modelo MPLS apresenta uma série de outras vantagens sobre o roteamento tradicional:

- Como o pacote é atribuído a uma FEC quando ingressa na rede, o roteador de

Figura 2.2: *Shim header* MPLS.

ingresso pode utilizar qualquer informação sobre o pacote na atribuição do rótulo, mesmo que tal informação não pertença ao cabeçalho da camada de rede. No roteamento tradicional, apenas são consideradas informações contidas no cabeçalho do pacote para efeito de roteamento.

- Decisões de repasse que dependam do roteador de ingresso podem ser facilmente implementadas. Afinal, um pacote que ingresse na rede por um determinado roteador pode ser rotulado diferentemente, caso ele ingresse na rede por outro roteador. No roteamento tradicional isso não é possível, já que as informações de roteador de ingresso não são carregadas pelo cabeçalho.
- As condições que determinam como um pacote é atribuído a uma determinada FEC podem ser bastante complexas, sem que seja causado impacto aos outros roteadores, os quais irão apenas repassar os pacotes, devidamente rotulados.
- Em algumas situações, pode ser útil forçarmos que determinados pacotes sigam uma rota específica, escolhida no momento em que ele ingressa na rede. Este cenário ocorre, por exemplo, por razões de policiamento ou de engenharia de tráfego. No roteamento tradicional, a única solução seria obrigar o pacote a carregar consigo uma codificação da rota. No modelo MPLS, um rótulo pode ser utilizado para representar a rota, dispensando o uso da técnica tradicional.

O modelo MPLS originalmente não possui a noção de qualidade de serviço. Entretanto, o uso de conceitos de Engenharia de Tráfego apresenta como consequência a obtenção de QoS, através do melhor aproveitamento dos recursos da rede e da otimização do processo de roteamento.

Ainda sobre este assunto, o uso integrado de MPLS e DiffServ apresenta boas perspectivas para a obtenção de QoS. A combinação de Engenharia de Tráfego da primeira com

diferenciação escalável de fluxos da última mostra-se uma proposta bastante adaptada ao alcance dos objetivos traçados.

As modificações realizadas no modelo MPLS para suporte a DiffServ devem incluir a capacidade de diferenciação de tráfegos e o conhecimento dos PHBs por parte dos roteadores MPLS. O primeiro pode ser obtido através do uso do campo EXP contido no cabeçalho MPLS. Como tal campo contém três *bits*, é possível especificar até oito diferentes classes de serviços. Além disso, os roteadores MPLS devem ser capazes de implementar mecanismos de classificação, marcação, policiamento, condicionamento de tráfego e repasse de pacotes, de acordo com a sua posição dentro do domínio. Como acontece no modelo Diffserv, os roteadores devem ser separados em dois tipos: de borda e de núcleo, cabendo a estes último apenas a função de repasse de pacotes. Esta função só é possível se os roteadores conhecerem os PHBs a serem aplicados a cada classe de pacotes existente no domínio.

2.3 Administração Baseada em Políticas

O gerenciamento tradicional de redes de computadores, baseado na configuração voltada para os elementos de rede, tem se mostrado ineficiente nas redes atuais. A grande diversidade de fabricantes, equipamentos e novas tecnologias, principalmente no que tange a qualidade de serviço, é uma das razões dessa ineficiência. Desta forma, faz-se necessário um esquema complementar de gerenciamento que ofereça um nível mais alto de abstração.

O paradigma baseado em políticas de configuração mostra-se adequado ao novo contexto [24]. Nele, a gerência da rede foca serviços, e não mais os elementos de rede. A configuração destes últimos passa a ser transparente para o administrador, sendo realizada automaticamente a partir das políticas estipuladas para o domínio. Com isso, torna-se possível gerenciar tecnologias e equipamentos heterogêneos transparentemente.

Políticas descrevem os objetivos de uma empresa e são formadas por uma ou mais regras [21]. Regras, por sua vez, são tipicamente formadas pelo par condição/ação: uma vez ocorrido os acontecimentos previstos no campo de condição da regra as respectivas ações são disparadas.

A idéia deste paradigma é que os objetivos que as políticas representam sejam transparentemente transformados em parâmetros de configuração para os elementos de rede. Isso significa que, em princípio, tudo que o administrador precisa fazer é criar as políticas de configuração de acordo com as necessidades e objetivos da empresa.

2.4 Bandwidth Broker (BB)

A função de um BB é automatizar o processo de administração de uma rede através do gerenciamento dos recursos disponíveis e do monitoramento da quantidade de tráfego utilizada [7]. Ele deve oferecer ferramentas de suporte para o controle da qualidade de serviço em diferentes níveis de granularidade. Além disso, deve atender às políticas estabelecidas para o domínio, implementando-as através da configuração dos roteadores. Um domínio é definido como o conjunto de todos os elementos de rede controlados pelo mesmo BB.

Os parâmetros estabelecidos durante a negociação devem ser usados para controlar os mecanismos de classificação, marcação, policiamento, condicionamento e repasse dos pacotes nos roteadores. O BB atua como um gerenciador de políticas, realizando a interpretação destas em parâmetros de configuração para os roteadores que compõem o domínio.

O *software* do BB baseia-se no modelo clássico cliente/servidor. Para cada domínio existe um único BB. O principal componente do *software*, denominado *Policy Decision Point* (PDP), é responsável por converter as políticas definidas pelo administrador do domínio em parâmetros de configuração para os roteadores que controla. Cada roteador possui a parte cliente do *software* denominada *Policy Enforcement Point* (PEP), responsável por garantir a aplicação das configurações determinadas pelo PDP. A comunicação entre PDP e PEP é realizada pelo protocolo COPS [16] ou SNMP [17]. A Figura 2.3 ilustra a arquitetura descrita.

Um BB negocia contratos de serviços baseando-se nos recursos disponíveis em seu domínio. Os clientes de um BB podem ser locais ou externos ao domínio. Um exemplo de cliente local é um elemento de rede localizado na borda do domínio que deseja atravessá-lo com requisitos de QoS. Clientes externos são representados por outros BBs, gerentes de domínios vizinhos, os quais desejam negociar recursos externos no intuito de garantir QoS fim-a-fim a clientes locais.

2.5 Qualidade de Serviço Fim-a-Fim

Qualidade de serviço fim-a-fim diz respeito à manutenção de compromissos com parâmetros de QoS mesmo quando o tráfego considerado atravessa duas ou mais redes de domínios administrativos distintos e de tecnologias diversas. A Figura 2.4 ilustra esta situação.

Neste contexto, faz-se necessário definir o escopo a que se refere uma transação de fim-a-fim. Obviamente nenhum provedor de acesso pode prometer garantias de QoS a clientes localizados fora de suas redes de *backbone*. Afinal, a infra-estrutura física externa a estas

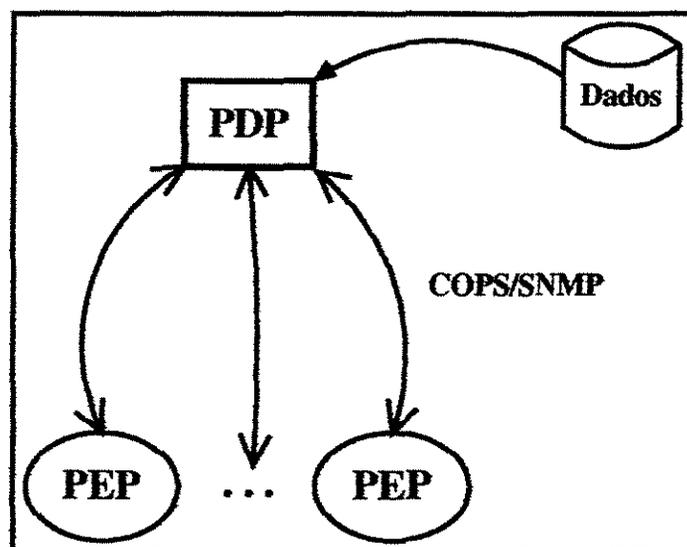


Figura 2.3: Arquitetura fundamental de um BB.

redes pode ser completamente aleatória. O cliente pode, por exemplo, estar se conectando ao provedor por meio de linha discada. Conseqüentemente, limitamos as garantias de QoS fim-a-fim ao escopo das redes controladas diretamente pelo provedor de acesso. Todos os clientes em negociação devem estar inseridos no contexto das redes de *backbone*. Fora deste escopo, nenhum requisito de QoS pode ser garantido.

Do ponto de vista do cliente, é desejável que não haja diferenças entre uma transmissão de dados com QoS que envolva um único domínio e outra que envolva vários domínios. O cenário ideal é que as diferenças certamente existentes entre os dois casos sejam transparentes para o cliente. Contudo, do ponto de vista operacional, existem vários complicadores para que isso se torne possível.

Domínios diferentes envolvem objetivos, recursos, serviços e filosofias diferentes. Portanto, antes de tudo, é necessário que haja interesse dos domínios em se relacionarem. Eles devem, neste caso, reservar recursos disponíveis para tráfegos externos. Ao mesmo tempo, devem ter a possibilidade de requisitar recursos externos aos outros domínios.

Além disso, é necessário que haja uma compatibilidade entre as classes de serviços oferecidas em cada um dos domínios. Essa compatibilidade pode ser obtida pelo oferecimento de serviços padronizados ou pela implementação de mecanismos de mapeamento de serviços. A primeira solução é mais simples e requer apenas um acordo entre os domínios a fim de existam classes de serviços equivalentes em cada domínio.

Existe ainda a necessidade de implementação de um modelo de comunicação entre os domínios, a fim de que estes possam negociar serviços com QoS entre si. Existem

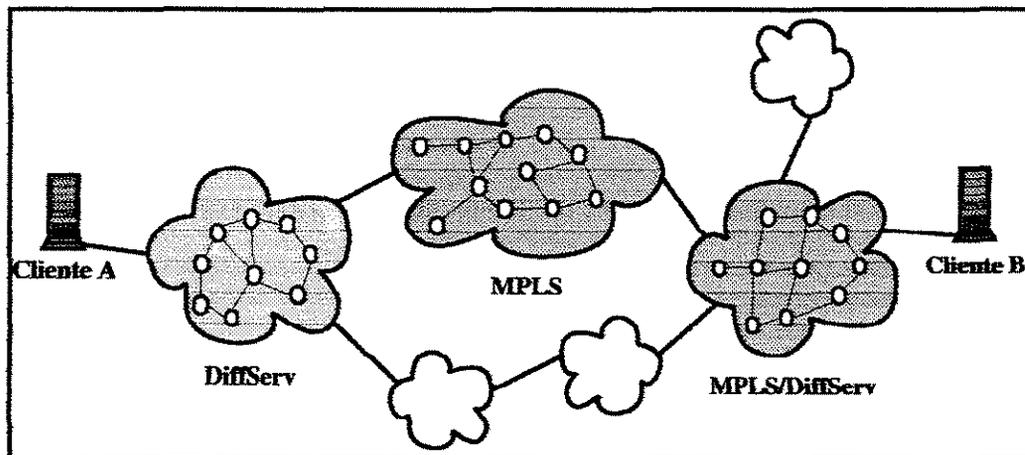


Figura 2.4: Cenário Típico de QoS Fim-a-Fim.

atualmente algumas propostas de modelos de comunicação, as quais serão apresentadas no próximo capítulo.

Por último, deve existir um modelo de negociação comum entre os domínios. Tal modelo deve ser capaz de negociar recursos externos para o domínio e garantir recursos locais para os outros domínios, mantendo compatibilidade e coerência com as negociações locais.

2.6 Trabalhos Relacionados

Muitos estudos têm sido realizados no sentido de garantir qualidade de serviço à Internet, notadamente através do uso dos modelos de Serviços Diferenciados e MPLS.

Diversos trabalhos sugerem o uso conjunto de algumas tecnologias de qualidade de serviço. Os trabalhos [18] e [9] propõem a obtenção de QoS na Internet através do uso conjunto e intercalado das arquiteturas de Serviços Diferenciados e Serviços Integrados. Em [30], encontramos a proposta de um modelo que realiza a administração transparente de domínios, quaisquer que sejam as arquiteturas utilizadas. O trabalho [29] apresenta uma plataforma que combina Serviços Diferenciados e MPLS, aliando a questão engenharia de tráfego da última com a questão diferenciação de tráfego da primeira.

Com relação ao gerenciamento de domínios, temos diversos trabalhos recentes que trazem implementações de *Bandwidth Brokers*. O trabalho [7] apresenta uma implementação de *Bandwidth Broker* baseado em políticas de configuração para a arquitetura de Serviços Diferenciados. De maneira análoga, o trabalho [36] propõe um gerente semelhante ao de [7], mas para a administração de plataformas MPLS. Temos ainda o trabalho [29], em

que, além da proposta da plataforma mista (MPLS/DiffServ), temos a adaptação do gerente de [7] para a referida plataforma. Contudo, em todos estes trabalhos, a abordagem adotada prevê o controle de um único domínio administrativo.

Finalmente, a preocupação com a obtenção de qualidade de serviço fim-a-fim também vem sendo demonstrada em diversos trabalhos recentes. A referência [3] mostra a especificação do projeto QBone/Internet2 de um modelo de comunicação entre *Bandwidth Brokers*, através do uso do *Simple Inter-Domain Bandwidth Broker Signalling* (SIBBS). Entretanto, tal modelo apresenta um nível mais alto de abstração, não especificando detalhes do modelo de negociação. Outros três grandes projetos tratam da questão obtenção de QoS fim-a-fim com grande riqueza de detalhes: os projetos TEQUILA [37], AQUILA [2] e CADENUS [5]. Em todos estes projetos, o objetivo fundamental é a obtenção de QoS fim-a-fim na Internet. Para tal, são apresentados modelos de negociação, além de propostas de representação e armazenamento de contratos.

Apesar dos esforços atualmente existentes, a definição de um modelo de negociação de recursos em redes IP que promova a comunicação entre *Bandwidth Brokers* vizinhos representa uma questão ainda em aberto, bem como a definição de interfaces para sua utilização e seu posicionamento dentro do domínio administrativo.

Capítulo 3

Negociação de Serviços com QoS na Internet

A disponibilização dos recursos de uma rede em diferentes níveis de qualidade introduz uma preocupação imediata com a maneira como estes serão oferecidos aos clientes. Espera-se que a distribuição seja realizada de forma justa, eficiente e flexível. Para que isso seja possível, é necessária a especificação de modelos e mecanismos comprometidos com estes objetivos. Neste contexto, a divisão dos recursos disponíveis na rede em serviços com características peculiares e posterior negociação por meio de contratos tem-se mostrado uma proposta adequada.

Atualmente, a negociação de serviços vem sendo tipicamente realizada de forma estática. Isso significa que existe a ação explícita do administrador da rede na reconfiguração dos equipamentos a cada contrato negociado. Os detalhes são tratados por meios tais como telefone, e-mail e outros métodos que implicam em intervenção manual. Este tipo de procedimento, além de tornar a administração complexa e pouco flexível, ainda apresenta problemas como falta de confiabilidade e aumento do tempo dedicado ao processo administrativo. Além disso, modelos de negociação estáticos não favorecem a escalabilidade pois um grande número de contratos negociados potencialmente agrava os problemas citados.

Por todos estes motivos, a proposta de um modelo de negociação automático e dinâmico tem sido recebida com grande interesse por parte da comunidade científica. Além de simplificar o processo de administração e configuração, a automatização certamente diminuiria o tempo necessário para a negociação. Uma outra vantagem é que métodos de configuração automáticos aumentam a flexibilidade do provedor de serviços em atender as necessidades dinâmicas de seus clientes.

A provisão automática e dinâmica de serviços exige a especificação de um protocolo para a definição e negociação de níveis de serviços [25]. Tal protocolo deve oferecer suporte

para diversos tipos de clientes e provedores de acesso, além de garantir a perfeita interação entre eles. Em adição, este protocolo deve privilegiar funcionalidade e confiabilidade para negociações não apenas no interior do domínio, mas também entre domínios distintos.

Uma negociação de serviços pode ser classificada como intra-domínio ou como inter-domínio. Uma negociação intra-domínio é aquela que reúne o programa de gerência do domínio solicitado e um programa cliente que não represente um domínio adjacente, mas sim uma entidade que negocie serviços para uso próprio. Em contra-partida, uma negociação inter-domínio reúne dois programas de gerência de domínios distintos, onde um cede recursos para o outro, que os distribuirá interna ou externamente, da maneira que melhor lhe convir.

Como a negociação de serviços é realizada por meio de contratos, é necessário criar padrões para diferentes perfis de serviços. A padronização mostra-se adequada na medida em que permite reunir as necessidades gerais de diferentes tipos de serviços. Contudo, um contrato padrão não deve restringir o modelo de negociação a um determinado tipo de protocolo ou tecnologia. Ao mesmo tempo, um protocolo de negociação deve permitir a negociação de diferentes padrões de contratos. Enfim, o cliente deve possuir uma maneira de obter informações sobre os tipos de contratos que seu provedor de acesso está apto a negociar.

A capacidade de negociação entre domínios é outra característica importante e útil a um modelo de negociação. Tal característica tem como função imprescindível garantir a QoS fim-a-fim. Portanto, é bastante desejável que um modelo de negociação contenha, entre seus componentes, um modelo de comunicação entre domínios.

Por último, um grupo de domínios que deseja negociar serviços entre si precisa concordar a respeito dos parâmetros de QoS garantidos para cada um deles. Isso pode ser feito de duas maneiras: pelo oferecimento de serviços padronizados ou pela implementação de mecanismos de mapeamento de serviços.

Neste capítulo, discutiremos propostas de soluções para os diversos requisitos destacados anteriormente e pertinentes a um modelo de negociação. Serão discutidas propostas para protocolos de negociação, modelos de comunicação entre domínios e contratos de serviços. Diversas idéias destes protocolos, modelos de comunicação e de contrato serão posteriormente utilizados para a confecção do modelo de negociação de que trata este trabalho.

3.1 Protocolos de Negociação

Nesta seção apresentaremos os conceitos que embasam os principais protocolos utilizados atualmente para a negociação de serviços na Internet. Alguns deles foram confeccionados especificamente para este objetivo. Outros, apesar de não confeccionados originalmente

para este objetivo, podem ser adaptados para tal com certa facilidade. A partir destes protocolos, discutiremos propostas de soluções para os diversos requisitos do modelo de negociação a ser proposto posteriormente.

3.1.1 O protocolo COPS

O *Common Open Policy System* (COPS) é um protocolo desenvolvido com o intuito de promover trocas de informações de políticas entre um servidor e seus clientes [16].

O protocolo é baseado no modelo cliente/servidor e possui um ambiente composto dos seguintes elementos:

- **Repositório de Dados:** armazena informações sobre as políticas de negociação.
- **Policy Enforcement Point (PEP):** ponto onde as decisões de configuração são aplicadas. Esta entidade reside no próprio cliente, recebendo deste solicitações de reserva e repassando-as ao PDP, para que este tome as decisões cabíveis.
- **Policy Decision Point (PDP):** residente no gerente, esta entidade é responsável por recuperar regras de políticas do repositório e gerar decisões de acordo com o que foi solicitado pelo cliente.

O uso do COPS como protocolo de negociação de serviços apresenta algumas motivações interessantes. Em primeiro lugar, o uso do *Transmission Control Protocol* (TCP) como protocolo de transporte garante confiabilidade na troca de mensagens de sinalização entre cliente(s) e servidor. Existe ainda a possibilidade de utilização de protocolos de segurança como o IPsec e o *Transport Layer Security* (TLS). Uma outra característica é a manutenção de estados parciais até o término do processo que a originou. Por último, o protocolo permite o transporte de objetos variados sem a necessidade de modificações.

Existem dois modelos de utilização do COPS: delegação e provisionamento. A diferença entre os dois está na maneira com que o PEP obtém a política do PDP. No primeiro, cada evento no PEP leva a uma decisão por parte do PDP indicando a política a ser aplicada ao evento. No segundo, as políticas são instaladas no PEP antes que este receba eventos.

A comunicação entre PDP e PEP ocorre através de trocas de mensagens. A Figura 3.1 mostra essa interação, ilustrando as principais mensagens do protocolo:

- **OPN:** mensagem enviada pelo PEP ao PDP com o intuito de iniciar uma sessão.
- **CAT:** mensagem enviada pelo PDP ao PEP com o objetivo de informar o primeiro sobre a aceitação da proposta de início de sessão.

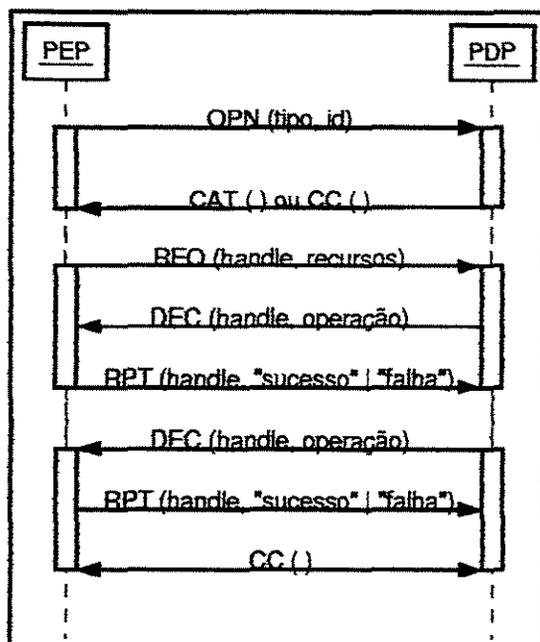


Figura 3.1: Troca de mensagens no protocolo COPS.

- **REQ:** mensagem enviada pelo PEP que reúne informações sobre algum tipo de requisição a ser feita ao PDP.
- **DEC:** mensagem de resposta a uma solicitação de requisição feita anteriormente.
- **RPT:** mensagens deste tipo carregam informações sobre sucesso ou falha na instalação de uma decisão. Toda mensagem DEC exige o envio de uma mensagem RPT por parte do PEP.
- **CC:** mensagem responsável por encerrar uma sessão entre PEP e PDP. Pode ser enviada a qualquer momento por ambas as partes.

O COPS originalmente não foi confeccionado para uso em negociações ou reservas de recursos. Contudo, sua adaptabilidade para o transporte de informações de políticas faz com que o protocolo apresente alguns conceitos interessantes para aplicações que tenham este objetivo.

3.1.2 O Protocolo BGRPP

O *Border Gateway Routing Protocol Plus* (BGRPP) [23] é uma arquitetura originada do BGRP [19] que oferece um protocolo de negociação de recursos baseado na agrega-

ção de reservas. As reservas são negociadas entre os chamados agentes BGRPP, os quais são posicionados nos roteadores de borda de cada domínio. Deste modo, cada domínio é responsável por realizar o controle de admissão levando em consideração os recursos disponíveis.

A Figura 3.2 mostra um cenário típico de utilização do BGRPP. O tráfego ingressa/egressa dos domínios através de dois diferentes tipos de roteadores:

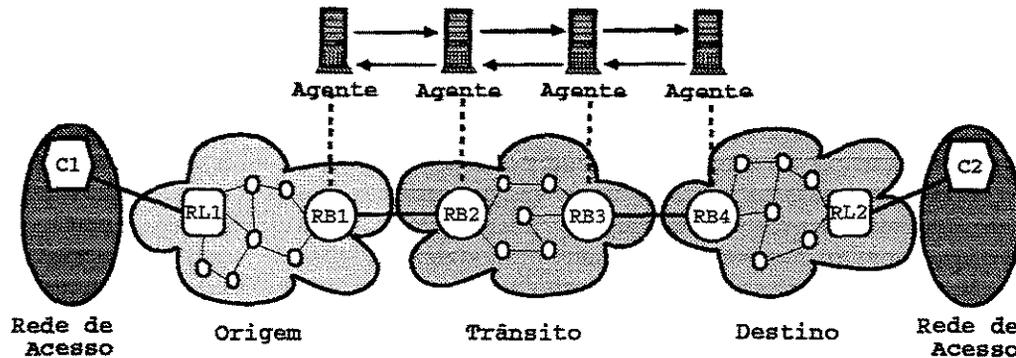


Figura 3.2: Cenário típico de utilização do protocolo BGRPP.

- **Roteador Limite (RL):** responsável por conectar um domínio a uma rede cliente.
- **Roteador de Borda (RB):** conecta um domínio a outro domínio que também faça parte da infra-estrutura BGRPP.

Uma vez efetuado o pedido de reserva, assume-se que o domínio de origem é capaz de executar controle de admissão levando em consideração os recursos disponíveis localmente. Contudo, ele não tem informações sobre a disponibilidade de recursos ao longo do caminho para o domínio de destino. Desta forma, o domínio de origem deve determinar um roteador de borda para a negociação com o respectivo agente BGRPP. Através do protocolo, os agentes são capazes de determinar se os recursos solicitados estão disponíveis em cada domínio do fim-a-fim. Cada agente é responsável pelos recursos de seus vizinhos imediatos. O roteamento entre os domínios é realizado pelo *Border Gateway Protocol* (BGP).

A sinalização do BGRPP é realizada através das seguintes mensagens:

- **PROBE:** disparada por um agente do domínio de origem, esta mensagem é repassada aos agentes de domínios intermediários até o domínio de destino. Ao receber esta mensagem, o agente certifica-se da validade do contrato e atualiza informações de roteamento na mesma, a fim de que o caminho de volta seja idêntico ao de ida.

- **GRAFT:** indica a disponibilidade de recursos até o domínio de destino. Esta mensagem é enviada em resposta a uma mensagem do tipo PROBE.
- **REFRESH:** contém a indicação de largura de banda necessária no momento. Esta mensagem é enviada por um agente a outro de domínio subsequente a fim de diminuir uma quantidade de recursos solicitada anteriormente. Uma mensagem do tipo REFRESH com valor de banda nulo é utilizada para cancelar a reserva.
- **ERROR:** indica a ocorrência de algum erro. Contém uma descrição do tipo de erro.

A principal característica do BGRPP é a escalabilidade. O protocolo faz uso de *sink trees*¹, conforme descrito pelo BGRP, para promover a agregação de reservas e uma consequente economia de memória necessária para a manutenção das reservas. Desta maneira, a quantidade de informações de estado armazenada pela rede pode ser reduzida proporcionalmente ao número de *sink trees* que possuam reservas ativas simultaneamente.

O BGRPP busca diminuir sobrecargas causadas pelo excesso de mensagens de sinalização através de um mecanismo de resposta prematura a mensagens de reserva. Este mecanismo é denominado *quiet grafting* e possibilita que as mensagens nem sempre tenham que trafegar por todos os domínios do fim-a-fim. O *quiet grafting* prevê que domínios intermediários guardem informações sobre recursos disponíveis em domínios vizinhos, dispensando em diversas situações o envio de mensagens de PROBE para todos os domínios do fim-a-fim e com isso criando uma perspectiva de grande diminuição no número de mensagens de sinalização.

3.1.3 O Protocolo SIBBS

O *Simple Inter-domain Bandwidth Broker Signalling* (SIBBS) [3] é o protocolo de sinalização proposto pelo grupo QBONE/Internet2 para promover a negociação de recursos em redes DiffServ.

O protocolo sugere que a negociação seja baseada nos BBs e busca a manutenção da simplicidade inerente ao modelo DiffServ. Ele define a troca de mensagens de sinalização do tipo requisição/resposta apenas entre BBs de domínios vizinhos imediatos. Tais mensagens carregam as informações essenciais sobre a negociação sendo executada. Existem dois tipos de mensagem no SIBBS:

- **Resource Allocation Request (RAR):** mensagem de solicitação de recursos. Carrega em seu interior as informações necessárias para a alocação.

¹*Sink trees* são estruturas de dados na forma de árvores que apresentam redes de computadores conectadas por caminhos ótimos.

- **Resource Allocation Answer (RAA):** representa a resposta a um pedido de alocação feito anteriormente. A resposta pode ser positiva, negativa, ou parcialmente positiva. Neste último caso, a mensagem carrega informações sobre a quantidade de recursos que pode ser disponibilizada (mesmo que não seja o que foi solicitado).

O processo de negociação é iniciado pela solicitação de recursos por parte do cliente. Ele envia uma RAR para o BB de seu domínio. A mensagem inclui um identificador do serviço, endereço de origem, endereço de destino, entre outras informações. Neste momento, o BB deve tomar algumas decisões, conforme segue:

- Verificar se o elemento de origem está autorizado para o serviço.
- Nomear um roteador de saída para o domínio.
- Especificar um caminho até o roteador de saída.
- Verificar se o fluxo requerido está de acordo com a capacidade do domínio e se está dentro dos padrões definidos para aquela classe de serviço.

Se todas estas verificações forem satisfatórias, o BB envia a RAR, devidamente modificada com as informações sobre o domínio de origem, para o(s) gerente(s) do(s) domínio(s) adjacente(s). Se alguma verificação for negativa, o gerente deve retornar uma RAA para o cliente, especificando os motivos da negação à solicitação de reserva.

A cada domínio de trânsito, o BB correspondente deverá realizar as seguintes avaliações:

- Certificar-se de que a requisição veio realmente do gerente de algum domínio adjacente.
- Nomear um roteador de saída para o domínio.
- Checar se os recursos solicitados não ferem a capacidade e/ou modelos de contratos utilizados pelo domínio anterior.
- Checar se os recursos solicitados não ferem a capacidade e/ou modelos de contratos utilizados pelo domínio posterior.
- Verificar se o fluxo requerido está de acordo com a capacidade disponível no domínio e se está dentro dos padrões definidos para aquela classe de serviço.
- Especificar um caminho até o roteador de saída.

Se todas estas avaliações forem satisfatórias, o BB do domínio de trânsito modifica novamente a RAR com as informações do domínio que ele representa e a envia para o próximo gerente. Estas operações se repetem a cada domínio de trânsito. Caso em algum desses domínios surja uma verificação negativa, o gerente correspondente deve enviar uma RAA para o gerente do domínio anterior, especificando os motivos da resposta negativa à solicitação de reserva. Neste caso, a RAA é propagada até o elemento de origem.

Quando finalmente chega ao domínio de destino, o BB deste domínio reconhece o endereço de destino contido na RAR. Novamente, ele deve realizar algumas avaliações:

- Certificar-se de que a requisição veio do gerente de algum domínio adjacente.
- Verificar se o fluxo requerido está de acordo com a capacidade disponível do domínio e se está dentro dos padrões definidos para aquela classe de serviço.
- Especificar um caminho até o elemento de destino.

Em caso negativo de alguma dessas verificações, acontece o mesmo procedimento dos domínios anteriores. Se todas estas avaliações forem satisfatórias, o BB do domínio de destino envia a RAR para o elemento de destino. Finalmente, tal elemento verifica se pode ou não receber o fluxo. Ele envia uma RAA para o gerente de seu domínio. Esta RAA possui sua autenticação e parâmetros para o fluxo que o elemento espera receber, os quais podem ser inclusive diferentes dos solicitados pelo elemento de origem. Em caso de rejeição, a RAA conterá os motivos da resposta negativa à solicitação de reserva.

Ao receber a RAA do elemento de destino, o gerente do domínio de destino a autentica e repassa para o domínio anterior correspondente. Neste momento, ele pode configurar condicionadores de tráfego no roteador de ingresso do domínio e possivelmente em outros roteadores no caminho até o elemento de destino. O mesmo procedimento é executado pelos gerentes dos domínios adjacentes, no caminho inverso em direção ao elemento de origem.

Ao receber uma RAA com resposta positiva do gerente de seu domínio, o cliente finalmente pode utilizar os recursos solicitados.

A Figura 3.3 mostra um exemplo de utilização do protocolo SIBBS. Também para este protocolo a principal preocupação é com a escalabilidade. A idéia é que a negociação exclusivamente entre domínios administrativos vizinhos consiga diminuir drasticamente o número de mensagens de sinalização. Deste modo, são especificadas apenas as interfaces para comunicação entre tais domínios, deixando-se em aberto detalhes sobre contratos de serviços, assim como os detalhes de como cada um dos domínios gerencia seus recursos. O objetivo é que diferentes implementações de modelos de gerenciamento de recursos e contratos possam ser utilizadas neste mesmo protocolo sem a necessidade de grandes modificações.

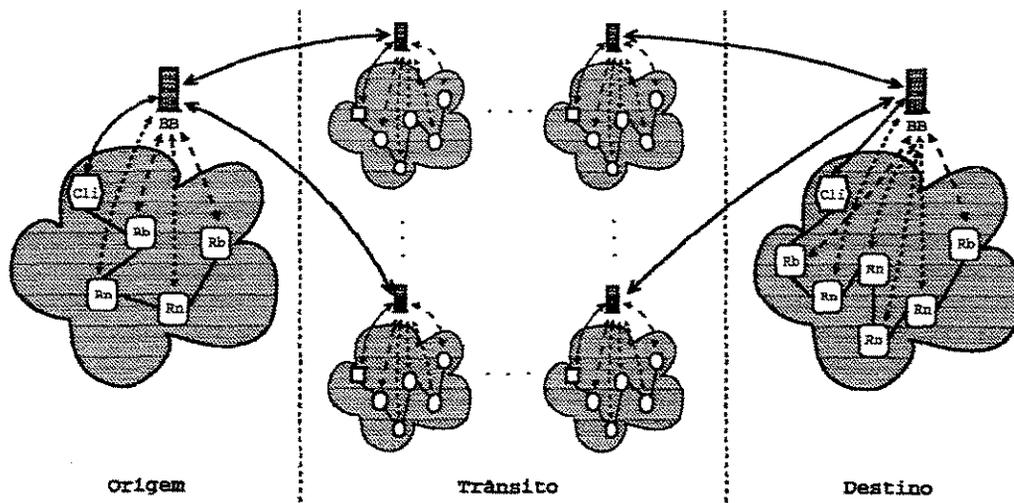


Figura 3.3: Funcionamento geral do protocolo SIBBS.

3.2 Comunicação entre Domínios

Um modelo de comunicação entre domínios define a maneira como os gerentes de cada domínio se relacionam, com o objetivo de negociar serviços para o domínio que representam.

Existem atualmente três propostas de modelos de comunicação entre domínios. Nas linhas abaixo, serão descritos o funcionamento geral e as principais características de cada um destes modelos.

3.2.1 Modelo Estrela

O Modelo Estrela caracteriza-se pela negociação centralizada realizada pelo domínio de origem. Neste caso, um gerente negocia contratos de serviços diretamente com os gerentes de domínios com os quais deseja manter relações.

Em um primeiro momento, o domínio de origem envia mensagens a cada um dos outros domínios, solicitando os serviços desejados. Posteriormente, ele receberá mensagens de resposta de cada um dos domínios contatados.

A grande vantagem do Modelo Estrela é o controle administrativo que se ganha pela centralização das negociações no domínio solicitante. Tal domínio depende apenas de si próprio para propor negociações, podendo fazê-lo da maneira que lhe for conveniente.

Por outro lado, este modelo apresenta um grande problema de escalabilidade. Dependendo da topologia entre os domínios de um determinado ambiente, o número de contratos

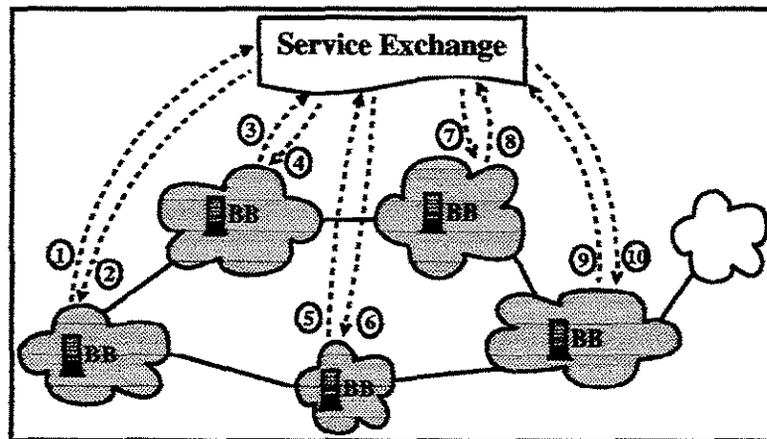


Figura 3.5: Modelo Hierárquico.

3.2.3 Modelo Cascata

O Modelo Cascata caracteriza-se pela negociação indireta de recursos. Neste modelo, o domínio de origem negocia apenas com domínios adjacentes. Quando é necessário negociar um contrato com um domínio não adjacente, essa tarefa é delegada a um domínio adjacente, que fará o mesmo se ainda não for o vizinho imediato do domínio de destino. Esse processo se repetirá até que o vizinho imediato do domínio de destino seja alcançado quando, neste caso, a negociação ocorre de fato e de maneira coletiva.

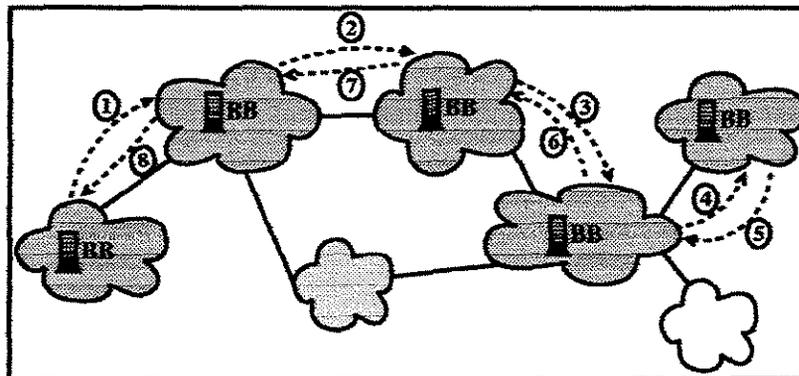


Figura 3.6: Modelo Cascata.

Este modelo apresenta grande similaridade ao modelo estrela. A diferença fundamental é que o primeiro possui potencialmente maior escalabilidade que o segundo. Tal

característica se deve ao fato de o número de negociações no modelo cascata ser potencialmente menor (ou, dependendo do caso, bem menor), quando comparado ao que ocorre tipicamente no modelo estrela.

A grande crítica ao modelo é a dificuldade de implementação de mecanismos de justiça na alocação de recursos, uma vez que as decisões são descentralizadas. Um outro problema é o pouco controle administrativo que se tem sobre o processo de negociação com domínios não adjacentes, já que isso é feito de maneira indireta.

O modelo cascata é o mais referenciado e aprovado entre todos. O exemplo mais notório de sua aprovação e utilização é o protocolo SIBBS [3], desenvolvido pelo projeto QBone/Internet2, o qual faz uso deste modelo de comunicação entre domínios.

3.3 Contratos de Serviços

Service Level Specification (SLS) pode ser definido como a representação de um conjunto de parâmetros técnicos os quais descrevem o tratamento a ser dispensado a um determinado fluxo de pacotes em um domínio de transporte, entre as interfaces de ingresso e egresso [28].

Na prática, tais entidades representam contratos realizados entre duas entidades em um modelo de negociação, sendo estas gerente/cliente ou gerente/gerente. Contratos armazenam os detalhes técnicos da negociação.

3.3.1 Requisitos de um Contrato

O projeto TEQUILA apresenta uma boa proposta sobre as informações que um contrato de serviços deve possuir. A idéia é especificar um contrato de serviços genérico o suficiente para se tornar um padrão. Com base nesse padrão, outros contratos poderiam ser criados para necessidades específicas.

Em [12], os requisitos de um contrato de serviços são divididos em: escopo, identificação de fluxo, parâmetros de conformidade de tráfego, tratamento de excesso, parâmetros de desempenho, disponibilidade e confiabilidade.

A informação de escopo de um SLS especifica a região topológica sobre a qual a qualidade de serviço deve ser garantida. O escopo é representado por informações de identificação dos nós de fim-a-fim: endereços IP ou mesmo qualquer informação capaz de identificar tais nós.

A identificação de fluxo de um SLS indica para quais pacotes o tratamento diferenciado deve ser executado. A informação de identificação é tipicamente representada pelo campo reservado à especificação da classe a que pertence o pacote. Este campo pode estar contido no cabeçalho do pacote IP de uma rede DiffServ ou no rótulo do pacote de uma

rede MPLS. Outras informações de identificação são: portas de origem/destino a que se referem os pacotes e versão do protocolo IP.

Os parâmetros de conformidade de tráfego são utilizados para o mecanismo de policiamento do mesmo. São representados por parâmetros que especificam valores referenciais a que um tráfego deve estar subordinado e por um algoritmo de conformidade de tráfego. Os mais utilizados são o algoritmo do Balde Furado e o do Balde de Fichas. Parâmetros para estes algoritmos são: taxa de pico (em bps), taxa de fichas do balde (em bps), profundidade do balde (em bits), unidades de transferência máxima e mínima (em bits).

O tratamento de excesso descreve como o gerente deve tratar excessos detectados nos mecanismos de policiamento. Existem três possíveis tratamentos: descarte, condicionamento ou remarcação. No caso dos dois últimos, deve haver informações adicionais sobre a operação. Por exemplo, a nova classe de pacotes para casos de remarcação, ou o tamanho do *buffer* dos condicionadores para casos de condicionamento.

Os parâmetros de desempenho descrevem o comportamento desejado, ou seja, as garantias de QoS que o gerente está oferecendo ao cliente. Os parâmetros de desempenho são: atraso (em ms), variação estatística de atraso (em ms), descarte (em percentual) e vazão (em Kbps). Garantias de desempenho podem ser quantitativas ou qualitativas. Se pelo menos um dos parâmetros de desempenho for quantificado, então as garantias de desempenho do contrato são ditas quantitativas. Caso contrário, elas são ditas qualitativas. Parâmetros qualitativos são especificados em três faixas: alta, média ou baixa. Apenas o parâmetro de vazão não pode ser qualitativo.

A disponibilidade indica o período de tempo em que determinado serviço estará acessível. A disponibilidade é representada pelos parâmetros de horários de funcionamento: durante o dia, a semana, o mês e o ano.

Finalmente, os parâmetros de confiabilidade indicam a quantidade máxima de tempo em que o serviço pode não estar disponível, assim como a quantidade máxima de enlaces e de nós que podem falhar simultaneamente.

3.3.2 Representação e Armazenamento de Contratos

A troca de informações de contratos no contexto da negociação de serviços exige a especificação de padrões para a representação e o armazenamento dos mesmos. Existem atualmente algumas propostas para a execução de ambas as tarefas.

Com relação à representação de informações de contratos, diversos modelos têm sido propostos. As informações contidas em um contrato podem ser representadas por estruturas de dados específicas através de *Policy Information Bases* (PIBs). Uma PIB deve ser descrita através da Abstract Syntax Notation One (ASN.1) para efeitos de padronização. Outras iniciativas, apoiadas pelo IETF, são o *Policy Core Information Model* (PCIM)

[10] e o *QoS Policy Information Model* (QPIM) [27]. Tratam-se de modelos de informação para uso específico de políticas e QoS, respectivamente. O PCIM estende o *Class Information Model* (CIM) para a representação de informações de políticas. De modo análogo, o QPIM estende o modelo PCIM para a representação de políticas aplicáveis a mecanismos de QoS.

Todos os modelos citados são baseados no paradigma de orientação a objeto e podem ser conceitualmente descritos como um conjunto de classes. Uma nova versão de modelo pode ser facilmente estendida a partir de um destes para a representação de informações de contratos de serviços.

Com relação ao armazenamento das informações de contratos, existem propostas para diferentes tipos de necessidades. A forma mais simples de se armazenar os dados de contratos é através de arquivos texto. Neste caso, o acesso e a interpretação das informações devem ser feitos por meio de um *parser*. Tal procedimento não possui grandes funcionalidades para o gerenciamento das informações e deve ser utilizado quando o número de contratos é pequeno e a quantidade de informações de cada contrato é reduzida.

Um proposta de solução mais robusta para o armazenamento de informações de contratos é a *Extended Markup Language* (XML). Através da criação de *tags* personalizadas pode-se facilmente implementar regras para o armazenamento de informações de políticas. Neste caso, também é necessária a utilização de um *parser*. Contudo, trata-se de um *software* já otimizado e específico para esta função. Além disso, a estrutura organizada do XML permite o gerenciamento de uma quantidade bem maior de contratos de serviços com informações diversas.

De qualquer modo, a proposta que tem recebido maior destaque no que diz respeito ao armazenamento de contratos segundo o IETF é o *Lightweight Directory Access Protocol* (LDAP). Diretórios LDAP são bastante apropriados para o armazenamento seguro, com acesso controlado e com alta eficiência para dados compartilhados na Internet [36]. Devido à sua estrutura distribuída, possuem capacidade de gerenciar grande quantidade de informações de forma robusta.

3.4 Resumo

Apresentamos neste capítulo alguns assuntos relacionados à negociação de serviços com QoS na Internet. Descrevemos os principais protocolos de negociação atualmente existentes. Estes protocolos servirão como base para a confecção do modelo de negociação que representa o principal produto deste trabalho. Além disso, discorreremos sobre os problemas da comunicação entre domínios, além da representação e do armazenamento de contratos.

Apresentaremos a seguir os componentes diretamente relacionados à solução de QoS

proposta neste trabalho. No próximo capítulo, falaremos sobre o *framework* que serve de infra-estrutura para o modelo de negociação entre Bandwidth Brokers.

Capítulo 4

Framework para Obtenção de QoS Fim-a-Fim

Um *framework* pode ser definido como um conjunto de classes estruturais relacionadas e define um modelo de interação entre objetos tomando por base os conceitos de especialização e herança. Mais especificamente, um *framework* representa a infra-estrutura de um ambiente de desenvolvimento padronizado e escalável. Neste contexto, tarefas específicas podem ser implementadas a partir de classes estruturais pré-estabelecidas, de acordo com as necessidades do desenvolvedor. O papel do *framework* é o de definir a interação entre estas classes, viabilizando o projeto e o desenvolvimento de módulos específicos de um sistema e livrando o desenvolvedor do conhecimento de detalhes de cunho estrutural.

As recentes pesquisas sobre QoS na Internet, além de terem demonstrado a necessidade de tecnologias que atuem diretamente sobre a configuração de equipamentos, também revelam benefícios advindos do uso de ferramentas que atuem no nível gerencial. A ausência de tais ferramentas tem resultado em contextos administrativos estáticos, pouco flexíveis e bastante suscetíveis a falhas. Deste modo, a proposta de um *framework* para a obtenção e garantia de QoS na Internet apresenta grande aplicabilidade no contexto gerencial/administrativo de domínios que façam uso de tecnologias como MPLS e DiffServ.

Neste capítulo será apresentado um *framework* que propõe simular um ambiente de QoS composto, no nível operacional, por uma plataforma mista MPLS/DiffServ, adaptada ao oferecimento de serviços múltiplos com requisitos de QoS particulares. No nível gerencial o *framework* apresenta um BB baseado em políticas de configuração. O último foi proposto em [29] e o primeiro em [7]. Originalmente estes trabalhos não prevêem mecanismos que garantam QoS fim-a-fim, especialmente para os casos em que existam dois ou mais domínios administrativos distintos. Desta forma, a idéia é utilizar o *framework* já existente como infra-estrutura para projeto e implementação do modelo de negociação de que trata este trabalho, adaptando-o ao oferecimento de QoS fim-a-fim.

Inicialmente mostraremos detalhes dos níveis gerencial e operacional do referido *framework*. Serão destacadas as principais funcionalidades existentes, bem como a maneira como elas foram projetadas. Logo a seguir, destacaremos as políticas de configuração criadas para uso do BB. O próximo capítulo será totalmente dedicado ao modelo de negociação.

4.1 Arquitetura do *Bandwidth Broker*

A Figura 4.1 mostra a arquitetura de *Bandwidth Broker* proposta para compor o *framework* de que este trabalho faz uso. Esta arquitetura é baseada nos trabalhos [7] e [29] com algumas modificações para a inclusão do modelo de negociação.

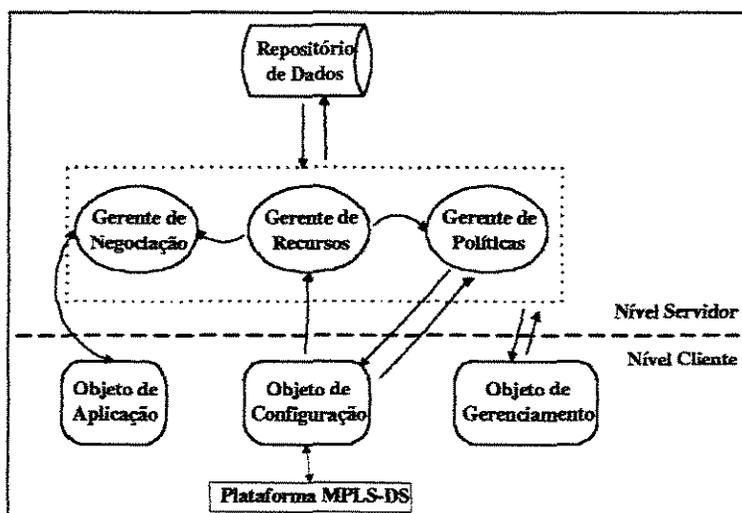


Figura 4.1: Arquitetura do *Bandwidth Broker*.

Como pode ser visto, a proposta divide os módulos funcionais do BB de acordo com o modelo clássico cliente/servidor. No nível servidor, temos os módulos de armazenamento de informação e de tomada de decisões. Tais componentes são centralizados e únicos no domínio. No nível cliente temos o componente de configuração dos roteadores, assim como aqueles que realizam o controle do nível servidor.

Nas próximas linhas descreveremos os diversos componentes da arquitetura proposta, assim como as possíveis interações existentes entre os mesmos.

4.1.1 Repositório de Dados

O Repositório de Dados representa o componente de armazenamento de informações. Ele é responsável por armazenar dados sobre as políticas estipuladas para o domínio, contratos negociados, além das informações sobre a configuração dos módulos gerenciais do BB.

Este componente participa do nível servidor da arquitetura e interage diretamente (e de forma independente) com cada um dos outros componentes do mesmo nível. Do ponto de vista gerencial, a interação objetiva a obtenção de dados de configuração. Particularmente com o Gerente de Negociação, o Repositório de Dados pode interagir a fim de prover/armazenar informações sobre os contratos negociados. Com o Gerente de Políticas a interação pode ocorrer com o objetivo de recuperar/armazenar informações de políticas. Com o Gerente de Recursos a interação pode visar a recuperação de informações sobre contratos, a fim de que a função de monitoramento possa ser satisfatoriamente realizada por este componente.

4.1.2 Gerente de Recursos

O Gerente de Recursos é responsável por dimensionar recursos existentes na rede, além de monitorar o cumprimento de contratos vigentes. Para executar corretamente suas atribuições, este componente interage com diversos outros.

Para a realização do monitoramento, o Gerente de Recursos recebe informações sobre o estado dos enlaces de instâncias do Objeto de Configuração. Além disso, recebe informações de contratos vigentes do Repositório de Dados. Com base nessas informações, ele tem condições de realizar os cálculos necessários ao monitoramento. Caso perceba o descumprimento de algum contrato, o Gerente de Recursos imediatamente notifica o Gerente de Políticas, a fim de que este tome as atitudes cabíveis.

Quanto ao dimensionamento de recursos, esta função é executada com base nas informações obtidas exclusivamente de instâncias do Objeto de Configuração. O dimensionamento é importante no relacionamento com o Gerente de Negociação, a fim de que este último tenha condições de oferecer e garantir recursos no contexto da negociação de contratos.

4.1.3 Gerente de Políticas

O Gerente de Políticas representa a entidade PDP também utilizada pelo modelo COPS descrito no capítulo anterior. Este componente tem como responsabilidade cuidar da interpretação e do mapeamento das políticas corporativas em parâmetros de configuração para os elementos de rede. Outras importantes funções são a tomada de decisões sobre a

configuração dos roteadores e a tomada de decisões para reconfiguração, de modo que o domínio mantenha o cumprimento dos contratos negociados.

A tradução de políticas em parâmetros de configuração é realizada com base nos recursos disponibilizados pela(s) tecnologia(s) de QoS utilizada(s). Quaisquer que sejam elas, o Gerente de Políticas deve manter a transparência proposta pelo paradigma de administração baseado em políticas.

A tomada de decisões sobre configuração de roteadores é realizada a partir do modo de provisionamento do COPS. Assim que iniciado, o roteador, por meio do Objeto de Configuração correspondente, solicita ao PDP parâmetros de configuração válidos.

A tomada de decisões de reconfiguração é realizada com base nas políticas estipuladas pelo administrador para o domínio.

4.1.4 Gerente de Negociação

O Gerente de Negociação implementa a funcionalidade de negociação de contratos de serviços, inclusive entre domínios administrativos distintos e com tecnologias de QoS diversas. Este componente deve negociar, instanciar e gerenciar os contratos do domínio [34].

A negociação de serviços sugere a especificação de políticas específicas para tal fim. Afinal, é desejável que o sistema seja verdadeiramente autônomo em termos de negociação de contratos. Estas políticas devem especificar clientes autorizados, quantidades máximas de recursos permitidas por cliente, situações de negociação parcial, entre outras informações. Tais políticas podem ser genericamente denominadas de **políticas de negociação**.

A fim de administrar a referida categoria de políticas, ao Gerente de Negociação foram incorporados um elemento PDP (denominado SLA-PDP) e outro PEP (denominado SLA-PEP). O SLA-PDP é responsável por gerenciar informações de políticas de negociação, no caso de o domínio que ele representa ser o solicitado. O SLA-PEP é responsável pelo repasse das solicitações de reserva feitas pelo cliente, no caso de domínio solicitante. No próximo capítulo descreveremos em detalhes o projeto e implementação deste componente.

4.1.5 Objeto de Configuração

O Objeto de Configuração representa a entidade PEP também utilizada pelo protocolo COPS. Instâncias deste objeto realizam as funções de monitoramento de tráfego nos enlaces e configuração de roteadores, com base em instruções recebidas do Gerente de Políticas.

Este componente representa a parte mais importante da arquitetura do ponto de vista do cliente. Existe um Objeto de Configuração para cada roteador participante do domínio.

É através dele que todas as informações de configuração são repassadas aos roteadores. Representa, portanto, o ponto de intersecção entre os planos gerencial e operacional do *framework*.

4.1.6 Objeto de Aplicação

O Objeto de Aplicação oferece a interface para um cliente realizar requisições de reserva de recursos. O cliente pode ser representado por uma rede (ou um nó) localizada à borda do domínio ou um gerente de domínio vizinho que deseja negociar recursos para o seu domínio.

Este componente representa o elemento SLA-PEP do Gerente de Negociação. É utilizado apenas quando o cliente é uma rede (ou um nó) que não participa do ambiente de QoS fim-a-fim. No caso de ser o gerente do domínio vizinho, o componente Objeto de Aplicação é parte do Gerente de Negociação do BB daquele domínio.

4.1.7 Objeto de Gerenciamento

O Objeto de Gerenciamento tem a função de fornecer uma interface para a execução de instruções explícitas no controle sobre a configuração dos módulos gerenciais.

Este componente representa a interface do administrador com o BB como um todo, permitindo a ele o controle detalhado sobre a configuração dos três módulos que compõem o nível gerencial da arquitetura proposta.

4.2 A Plataforma MPLS-DS

A plataforma MPLS-DS tem como objetivo validar, por meio de simulações, o uso de políticas de configuração para a obtenção de QoS fim-a-fim em redes IP através do emprego conjunto das tecnologias DiffServ e MPLS [29].

A plataforma foi desenvolvida sobre o simulador de redes *Network Simulator* (NS) [38]. A opção pela simulação foi motivada pela maturidade e riqueza de recursos encontradas na ferramenta escolhida. Além disso, simulações garantem uma grande liberdade na criação de cenários de utilização, o que em geral é complicado e trabalhoso de ser feito em aplicações reais.

A idéia utilizada para a confecção da plataforma foi a modificação de componentes já existentes no simulador NS. As modificações visaram o suporte às seguintes funcionalidades previstas nos modelos DiffServ e MPLS:

- Diferenciação de fluxos de tráfego.

- Criação de rotas explícitas (LSPs).
- Roteamento por LSPs.
- Configuração das filas dos roteadores.

A Figura 4.2 ilustra a plataforma MPLS-DS e seu relacionamento com o BB.

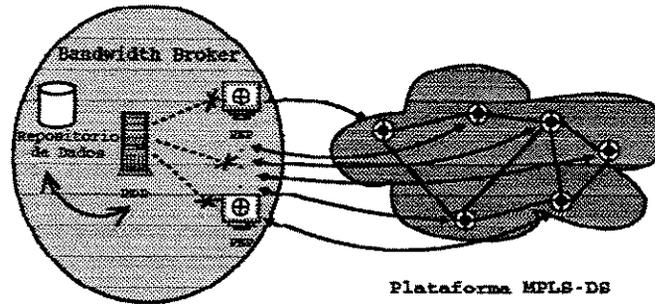


Figura 4.2: A Plataforma MPLS-DS e o BB.

As funcionalidades de diferenciação de fluxos e configuração das filas dos roteadores foram projetadas com base em conceitos do modelo DiffServ. Para suporte ao DiffServ foi utilizada a extensão DS-Nortel [20]. A criação e roteamento dos LSPs foram projetados com base no modelo MPLS. Para suporte ao MPLS foi utilizada a extensão MNS [1].

Conforme já mencionado, a configuração e o controle da rede são realizados através do BB. Suas decisões irão envolver uma ou mais das funcionalidades citadas e serão passadas pelo PDP ao PEP. O PEP, por sua vez, é responsável por efetivar as configurações no roteador que ele representa. Todo roteador da plataforma MPLS-DS incorpora as funcionalidades aqui destacadas.

4.2.1 O *Network Simulator* (NS)

O NS foi confeccionado a partir do projeto *Virtual InterNetwork Testbed* (VINT) [39] que tinha por objetivo a confecção de um simulador de redes que permitisse um estudo sobre a interação e escalabilidade de diversos protocolos de redes existentes.

Este simulador tem obtido grande destaque nas pesquisas recentes sobre redes de computadores graças a características como riqueza de funcionalidades e fidelidade a situações reais de uso de protocolos. Ele possui suporte variado e consistente para simulação de protocolos de roteamento, de transporte e *multicast*, em redes convencionais e sem fio.

O NS é baseado em eventos discretos e foi implementado sob o paradigma de orientação a objetos, através das linguagens C++ e OTcl. Como é um sistema de código aberto, é

possível modificar o código fonte, podendo-se criar ou mesmo estender classes existentes de acordo com as necessidades específicas da pesquisa em andamento. Deste modo, a criação/modificação dos protocolos e outras funcionalidades já existentes no simulador é bastante simplificada.

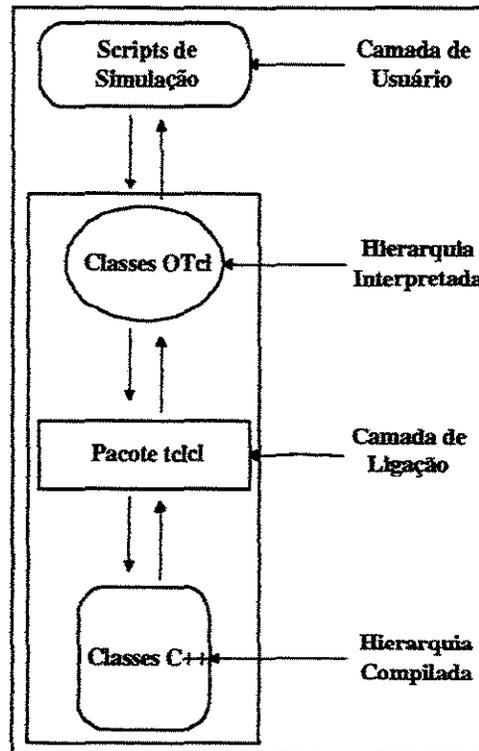


Figura 4.3: A arquitetura do NS.

Em muitos casos, para que uma simulação tenha alto índice de fidelidade a situações reais, é necessário que a ferramenta utilizada seja capaz de proporcionar o gerenciamento eficiente de informações pertinentes em níveis realmente básicos como, por exemplo, no nível de *bytes* do cabeçalho de um pacote. Neste cenário, os algoritmos devem estar preparados para manipular grandes quantidades de informações. Por outro lado, a configuração de muitas redes possui um caráter altamente dinâmico, de modo que a ferramenta deve estar preparada para rapidamente modificar parâmetros como topologia, configuração de filas dos roteadores, aplicações geradoras de tráfego, entre outros. O NS consegue atender a esses requisitos através do uso de duas hierarquias de execução: uma hierarquia interpretada, baseada na linguagem OTcl; e outra compilada, baseada na linguagem C++. A hierarquia compilada é adequada à execução que exija rapidez e alto desempenho. A hierarquia interpretada, por sua vez, possui grande adaptabilidade,

apesar de baixo desempenho. Através do uso conjugado de ambas é possível obter-se os resultados esperados.

A hierarquia compilada está estruturada sobre classes criadas em C++, enquanto que a hierarquia interpretada possui uma estrutura similar utilizada por um interpretador OTcl. Por meio de *scripts*, os usuários criam objetos de simulação, os quais são instanciados dentro do interpretador OTcl e espelhados para objetos correspondentes na hierarquia compilada. O NS consegue prover a ligação transparente entre as duas hierarquias por meio do pacote *tccl*. Por meio dessa ligação, variáveis dos dois níveis podem ser relacionadas, objetos do nível compilado podem ser criados automaticamente, entre outras funcionalidades. A Figura 4.3 mostra a arquitetura completa do simulador NS.

Em geral, uma boa prática de programação é que operações que exijam desempenho sejam executadas em nível compilado, enquanto que operações de configuração e passagem de parâmetros são melhor executadas pela hierarquia interpretada.

Existem cinco componentes básicos de simulação no NS: o nó de rede, o enlace, a fila de roteador, o gerador de tráfego e o monitor de dados. Filas, escalonadores e classificadores de tráfego são implicitamente criados quando da instanciação de um nó. Monitores de dados são utilizados para medição dos parâmetros de QoS como atraso, descarte e largura de banda utilizada.

4.2.2 O Pacote DS-Nortel

O pacote DS-Nortel, desenvolvido por um grupo de pesquisa da *Nortel Networks*, apresenta uma implementação da arquitetura de Serviços Diferenciados para o simulador NS. A implementação busca oferecer as principais funcionalidades da arquitetura DiffServ através da implementação de classes para a realização de tarefas típicas.

A diferenciação de serviços baseia-se na utilização de quatro filas físicas, sendo que cada uma pode ser organizada em até três filas virtuais. A implementação conta ainda com mecanismos para classificação, marcação, policiamento e adequação do tráfego. Estes mecanismos são implementados em forma de filas, as quais se localizam nos enlaces entre os roteadores no contexto do NS.

O pacote disponibiliza os dois tipos de filas compatíveis com a arquitetura DiffServ: de borda e de núcleo. Filas de borda são incorporadas aos roteadores de borda e realizam a classificação, marcação, policiamento, adequação e encaminhamento dos pacotes. Filas de núcleo, mais simples, são incorporadas aos roteadores de núcleo e realizam apenas o encaminhamento dos pacotes com base nos parâmetros atribuídos às mesmas, de acordo com o PHB da classe à qual o pacote pertence.

4.2.3 O Pacote MNS

O *MPLS Network Simulator* (MNS) apresenta uma implementação do MPLS para o simulador NS. Em resumo, o pacote atribui dois novos elementos ao ambiente do NS: o *Label Switched Router* (LSR), um nó de rede especializado em realizar roteamento por LSPs; e o agente LDP, especializado em promover a distribuição de rótulos aos roteadores por meio do *Label Distribution Protocol* (LDP).

De uma maneira geral, o MNS oferece ao simulador a possibilidade de configurar nós que recebam rótulos de um agente LDP e façam uso dos valores recebidos para tomar decisões a respeito do encaminhamento dos pacotes, tomando por base ainda os dados do cabeçalho MPLS do mesmo. Além disso, o MNS ainda permite a criação de rotas explícitas (LSPs) que podem ser utilizadas por tipos específicos de tráfego. Descrevemos a seguir o funcionamento geral de um nó LSR e de um agente LDP:

- **Nó LSR:** realiza o encaminhamento do pacote baseado no rótulo contido no cabeçalho MPLS do mesmo. Quando recebe um pacote, inicialmente verifica se está rotulado. Caso esteja, verifica em suas tabelas o rótulo correspondente. Caso não esteja rotulado, verifica se existe um rótulo associado à sua FEC e o atribui ao pacote. Caso não exista, especifica um rótulo específico para este caso.
- **Agente LDP:** realiza a distribuição de rótulos entre os nós LSR da rede simulada. Possui algumas opções interessantes, como distribuição sob demanda, especificação dos vizinhos LDP de um LSR, entre outras.

4.2.4 Integração entre DS-Nortel e MNS

A integração entre os pacotes DS-Nortel e MNS objetiva a obtenção de uma implementação que reúna os conceitos de QoS contidos nos modelos DiffServ e MPLS com o intuito de prover uma infra-estrutura bastante adaptada ao oferecimento de QoS.

Com a integração, todos os pacotes passam a ser rotulados na entrada do domínio a fim de que possam receber um tratamento diferenciado no interior do mesmo. Os roteadores ficam responsáveis por interpretar o campo EXP contido no rótulo dos pacotes. A classificação leva em conta o código referente a alguma classe de serviço oferecida pelo domínio.

A integração também possibilita especificar dois tipos diferentes de LSPs: E-LSPs e L-LSPs. A referência [15] propõe que um E-LSP transporte pacotes de até oito classes distintas, enquanto que o L-LSP transporte apenas uma classe [29]. A combinação dos pacotes DS-Nortel e MNS fornece roteadores capazes de diferenciar pacotes e encaminhá-los pelos LSPs.

4.2.5 Monitoramento do Tráfego Simulado

Importante atribuição da plataforma MPLS-DS, a realização de monitoramento do tráfego simulado é realizado no contexto dos modelos DiffServ/MPLS. O NS nativamente conta com objetos especializados em acumular informações sobre requisitos de QoS, em especial porcentagem de descarte e atraso médio. Para o descarte, cada roteador apresenta mecanismos de controle de pacotes recebidos e não encaminhados. Para o atraso, podem ser configurados mecanismos de medição entre dois roteadores quaisquer da simulação.

Esta infra-estrutura pré-existente foi sofisticada para permitir o monitoramento de atraso e descarte por fluxo de pacotes que passam por um determinado LSP. Todos os compromissos firmados por um BB tomam por referência os resultados do monitoramento. A cada meio segundo, os valores de porcentagem de descarte e atraso médio acumulados pelos monitores, localizados nos próprios elementos de rede, são repassados ao BB. O descarte de um LSP é calculado com base na média dos descartes sofridos por cada um dos roteadores que o compõem. Analogamente, o atraso de um LSP significa o atraso médio calculado entre todos os pares de roteadores deste.

Sempre que porcentagem de descarte ou atraso médio de determinado fluxo ultrapassam o máximo estipulado nos contratos negociados, uma política de atuação é disparada pelo BB a fim de que estes parâmetros voltem a valores toleráveis. Políticas de atuação fazem parte da arquitetura de políticas do *framework* e serão apresentadas na subseção 4.4.

4.3 Comunicação entre Componentes do *Framework*

A comunicação entre os elementos que compõem o *framework* é realizada por meio de emulação das mensagens de sinalização do protocolo COPS. Inclui-se nesta especificação a comunicação entre Objeto de Configuração e Gerente de Políticas, Objeto de Aplicação e Gerente de Negociação e entre Gerentes de Negociação.

O uso de conceitos do COPS apresenta algumas motivações interessantes. Em primeiro lugar, estamos tratando do gerenciamento de redes com base em políticas de configuração. Portanto, nada mais coerente utilizar um protocolo construído para promover a troca de informações de políticas. Uma outra motivação: o COPS permite a especificação de diferentes tipos de objetos em suas mensagens. Deste modo, torna-se possível a interação de um mesmo servidor com múltiplos tipos de clientes. Por último, além de suporte a decisões por sincronização de políticas, o COPS também oferece suporte à configuração de equipamentos (através do modo de provisionamento) e confiabilidade (através do uso de TCP e mensagens do tipo KA).

4.4 Arquitetura de Políticas

Um mecanismo de gerência automatizada baseada em políticas possibilita o controle dinâmico dos mecanismos da rede, viabilizando intervenções na distribuição do tráfego e nos mecanismos de prioridades [36].

Apresentaremos a seguir a arquitetura de políticas confeccionada para uso e configuração do BB. As políticas estão divididas em 3 categorias distintas: políticas de Inicialização, de Reação e de Negociação. Estas políticas são suficientes para atender as necessidades do *framework*. Contudo, se para algum caso específico for preciso novas políticas, a idéia é que elas possam ser implementadas na forma de *plugins* e carregadas dinamicamente ao *software* que o implementa.

4.4.1 Políticas de Inicialização

Reúne políticas para a configuração inicial dos roteadores. Estabelece os padrões para os diversos parâmetros dos roteadores, como peso de filas, disciplinas, LSPs pré configurados, entre outras informações. São tipicamente representadas pelos PHBs, os quais devem ser associados aos diferentes roteadores do domínio. Os PHBs são transparentemente transformados pelo Gerente de Políticas nos parâmetros de configuração citados anteriormente.

4.4.2 Políticas de Atuação

Definem o comportamento do BB para os casos em que algum contrato de serviços seja violado. Sempre que acionadas, as políticas são automaticamente traduzidas pelo Gerente de Políticas em parâmetros de configuração para os roteadores do domínio. São definidos nativamente dois tipos de políticas de atuação para o *framework*:

- A primeira categoria refere-se à atualização do peso das filas dos roteadores. Esta política é acionada sempre que um determinado fluxo de dados sofre descarte maior que o estipulado para a sua classe.
- A segunda atua sobre a criação de novos LSPs. Esta política é disparada em duas situações: caso o atraso de um fluxo extrapole o estipulado para a sua classe ou caso ocorra excesso de descarte após três atuações consecutivas da primeira política. O novo LSP deve ser criado com base em recursos disponíveis no domínio.

4.4.3 Políticas de Negociação

Reúne políticas utilizadas pelo Gerente de Negociação e/ou Objeto de Aplicação no contexto da negociação de serviços. Estas políticas definem clientes autorizados, quantidades máximas de recursos permitidas por cliente/serviço e tempo mínimo de renegociação para cada cliente.

Do lado do cliente podem ser feitas restrições para a quantidade mínima de recursos que deve ser aceita no caso de contra-proposta do servidor. São disponibilizados três possíveis comportamentos: (1) o cliente aceita apenas aquilo que foi pedido, (2) o cliente aceita menos do que foi pedido (nesse caso, é preciso especificar o mínimo aceitável) e (3) o cliente aceita qualquer contra-proposta do servidor.

Esta categoria de políticas será melhor detalhada no capítulo a seguir.

4.5 Resumo

Apresentamos neste capítulo um *framework* para o oferecimento de QoS baseado no simulador de redes NS. Esta solução, apesar de funcional, apresenta como restrição o oferecimento de QoS para o contexto de um único domínio administrativo. Entretanto, situações em que os clientes do fim-a-fim estão localizadas em domínios distintos não são raras, de modo que a adição de um modelo de negociação entre domínios representa um grande avanço para a infra-estrutura já existente.

Neste contexto, o próximo capítulo apresenta uma proposta de modelo de negociação que, complementando o *framework* aqui detalhado, pretende garantir QoS fim-a-fim especialmente para casos em que os clientes estejam localizados em domínios distintos.

Capítulo 5

Proposta de Modelo de Negociação

O modelo de negociação a ser apresentado neste capítulo representa a contribuição deste trabalho no que tange o estudo de soluções para a obtenção e garantia de QoS fim-a-fim na Internet. Para a formulação do modelo, aproveitou-se conceitos encontrados em soluções já existentes, empregados sobretudo nos protocolos apresentados no Capítulo 3. A idéia é que seja feita uma junção das boas práticas encontradas em cada um deles, tendo como foco a adaptação destes para o contexto da negociação dinâmica e transparente entre domínios. O diferencial do modelo é a proposta de uma arquitetura de políticas que busca atribuir um maior grau de abstração para o administrador da rede, seguindo preceitos do paradigma de administração baseado em políticas de configuração. O dinamismo citado se refere à capacidade do modelo em modificar, em tempo de execução e com base em resultados obtidos no monitoramento dos requisitos de QoS, os parâmetros de configuração dos roteadores.

Do protocolo COPS foi utilizada toda a arquitetura básica, com as entidades PEP, PDP e o repositório de dados. O conceito de administração baseada em políticas de configuração foi igualmente incorporado ao modelo, assim como formato e semântica das mensagens de sinalização. O uso de políticas garante à gerência do sistema grande automatização, simplificando a tarefa administrativa essencialmente à especificação de regras de negócio. Além disso, a idéia de transparência nas informações transportadas pelas mensagens possibilita que contratos de diferentes padrões sejam utilizados, o que garante adaptabilidade ao novo modelo.

Com relação ao BGRPP, a principal herança diz respeito a escalabilidade. A estratégia do *quiet grafting* contribui de forma decisiva para a negociação escalável entre diferentes domínios. Deste modo, um esquema semelhante foi incorporado ao modelo, recebendo a denominação de **negociação de atacado/varejo**. Por estar bastante vinculado ao nível gerencial do modelo, esta estratégia será detalhada na Seção 5.2 deste capítulo.

Para a comunicação entre domínios distintos foi utilizado o Modelo Cascata apresen-

tado no Capítulo 3. Este modelo é igualmente utilizado pelo protocolo SIBBS, o qual apresenta uma proposta detalhada para sua utilização (ver [3]). O modelo a ser proposto se valeu desta especificação para implementar a comunicação entre domínios e viabilizar a QoS fim-a-fim almejada.

A negociação de que iremos tratar neste capítulo se refere à largura de banda máxima permitida para uma determinada classe de serviço, no contexto dos modelos DiffServ e MPLS. A idéia é que garantias sobre parâmetros de QoS sejam oferecidas a fluxos dos serviços negociados com a limitação de banda correspondente. Obviamente, o modelo deve permitir que sejam negociadas bandas máximas para diferentes classes de serviços, como se fossem diferentes cláusulas de um mesmo contrato. Entretanto, além de largura de banda, existem alguns outros parâmetros, menos relevantes do ponto de vista comercial, passíveis de negociação. Entre eles, podemos destacar a negociação de número máximo de roteadores intermediários, número máximo de LSP's utilizados, atraso e descarte independentes de largura de banda.

Podemos dividir a especificação do modelo de negociação em quatro partes. Inicialmente, trataremos da questão operacional, representada por um protocolo de envio e recebimento de mensagens denominado SLA-COPS. Em seguida, será descrito o plano gerencial do modelo, representado pela estratégia da negociação de atacado/varejo. Logo a seguir serão apresentados detalhes sobre as políticas de configuração propostas para uso no modelo, as quais foram divididas em diversas categorias com objetivos específicos. Por último será apresentado o formato de contratos de serviços escolhido para esta proposta.

5.1 O Protocolo SLA-COPS

Como toda a infra-estrutura para envio e recebimento de mensagens foi retirada do protocolo COPS, o protocolo de negociação que compõe o modelo proposto recebeu a denominação de **SLA-COPS**. As entidades PEP e PDP foram igualmente herdadas do COPS, sendo denominadas, respectivamente, **SLA-PEP** e **SLA-PDP** [33].

As novas entidades possuem funções análogas às das entidades originais. Deste modo, o SLA-PDP representa um gerente de políticas, assim como o SLA-PEP representa um agente de aplicação de políticas. A diferença fundamental é que estas novas entidades, além de se orientarem pelas políticas de negociação estipuladas para o domínio, baseiam-se ainda na utilização de contratos de serviços (SLAs), devidamente estabelecidos com clientes, para a tomada de decisões sobre concessão de recursos.

É importante notar que, para o protocolo SLA-COPS, não existem diferenças operacionais na negociação de um cliente SLA-PEP que representa um simples nó de rede e outro que representa um gerente de um domínio adjacente. As diferenças entre negociações intra-domínio e inter-domínio ficam restritas ao plano de gerência.

Um outro detalhe que merece destaque é que, apesar das similaridades, o SLA-COPS apresenta um princípio funcional diferente da arquitetura de políticas em três camadas tradicional utilizada pelo COPS. O SLA-PEP não representa um aplicador de políticas nos elementos de rede, mas sim um intermediário que pode ser encarado como um “configurador de contratos” para o gerente de negociação. A partir dos contratos negociados, a idéia é que o gerente de negociação do domínio solicitante reconfigure seus roteadores, utilizando para tanto um PEP tradicional.

A operação do SLA-COPS está dividida em três fases: inicialização, configuração e negociação. Efetuadas estas três etapas satisfatoriamente, o processo de negociação está finalizado. SLA-PDP ou SLA-PEP podem, a qualquer momento, cancelar o contrato enviando ao outro uma mensagem do tipo CC. As penalidades para estes procedimentos podem ser negociadas em contrato.

5.1.1 Fase de Inicialização

O SLA-PEP, após receber pedido do cliente, envia ao SLA-PDP uma mensagem de abertura de negociação (OPN). Esta mensagem, bastante simplificada, tem o intuito de identificar o cliente o qual deseja obter recursos do domínio representado pelo SLA-PDP. Este último verifica a validade da requisição recebida através da busca por políticas restritivas ao cliente em questão. Caso estas políticas não sejam encontradas, o SLA-PEP está autorizado a negociar com o domínio e recebe uma mensagem de aceitação de cliente (CAT). Caso contrário, recebe uma mensagem de fechamento de cliente (CC) e encerra-se o processo de negociação.

5.1.2 Fase de Configuração

A fase de configuração tem por objetivo definir padrões importantes para a fase seguinte do protocolo, ou seja, a fase de negociação propriamente dita. Estes padrões incluem:

- **Modo de negociação:** define quem irá restringir parâmetros técnicos do contrato sendo negociado, tais como algoritmo de policiamento, parâmetros de QoS, serviços reconhecidos, tratamento de excesso, entre outros. No modo predefinido o gerente restringe os parâmetros técnicos do contrato. No não predefinido o próprio cliente restringe estes parâmetros do contrato.
- **Tempo para renegociação:** define em quanto tempo pode ser realizada uma renegociação deste mesmo contrato. Este parâmetro é um instrumento importante na busca por escalabilidade.

- **Contratos predefinidos:** define padrões de contrato possivelmente utilizados na negociação. O cliente deve utilizar-se deste instrumento para descobrir que padrões de contratos são reconhecidos pelo domínio solicitado.

Caso tenha sido aceito na fase anterior, o SLA-PEP envia uma mensagem do tipo REQ ao SLA-PDP. Esta mensagem inclui propostas para os parâmetros acima destacados. O SLA-PDP analisa os parâmetros, realiza as modificações que lhe forem convenientes e envia uma mensagem do tipo DEC à entidade SLA-PEP com possíveis novos valores. O SLA-PEP envia ao SLA-PDP uma mensagem do tipo RPT para confirmar o recebimento de DEC.

5.1.3 Fase de Negociação

O SLA-PEP envia ao SLA-PDP uma nova mensagem do tipo REQ, desta vez salientando os recursos que o cliente deseja obter. Os recursos são representados pela especificação de serviços desejados e largura de banda máxima para cada um deles.

O SLA-PDP analisa o pedido do SLA-PEP, aplica as políticas relacionadas àquele cliente e envia uma resposta, numa mensagem do tipo DEC, frisando o que pode ser oferecido a partir do que foi pedido (concessão total, parcial ou negação da solicitação). O SLA-PEP analisa a resposta do SLA-PDP, comparando os resultados obtidos com o que foi pedido e, finalmente, envia uma mensagem do tipo RPT ao SLA-PDP informando-o se o cliente concorda ou não com os termos propostos para a negociação.

5.2 Negociação de Atacado/Varejo

A estratégia de negociação proposta para este trabalho é denominada **negociação de atacado/varejo**. Esta estratégia visa garantir o provisionamento escalável de QoS especialmente para casos em que os nós de fim-a-fim pertençam a domínios distintos.

A estratégia de negociação diz respeito ao plano gerencial do modelo e está organizada da seguinte maneira: para obter recursos de outros domínios, cada gerente negocia contratos apenas com seus vizinhos imediatos (segundo o Modelo Cascata), pedindo a estes recursos de todos os domínios para os quais deseja ter garantias de QoS. Em contra-partida, cada gerente deve garantir a outros domínios uma quantidade de recursos adequada, a fim de que consiga atender satisfatoriamente a propostas de contratos de seus vizinhos imediatos. Busca-se com isso a montagem de “túneis” de QoS entre os domínios e uma substancial redução do número de mensagens de sinalização. Contratos desta natureza compõem a chamada **negociação de atacado**, uma vez que um único contrato simboliza a necessidade de vários clientes.

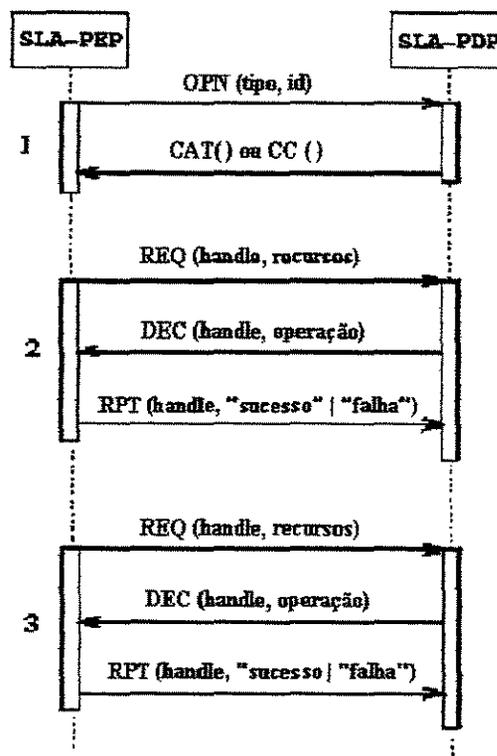


Figura 5.1: Operação do protocolo SLA-COPS

De posse de recursos provenientes de outros domínios garantidos pelos contratos de atacado, cada gerente negocia contratos locais, oferecendo a seus clientes, além de recursos próprios, os recursos obtidos na negociação de atacado. Estes contratos locais compõem a chamada **negociação de varejo**.

Cabe a cada domínio organizar coerentemente seus contratos e recursos. Enfim, é necessário um bom relacionamento entre os domínios. Eles devem se organizar com base em mecanismos de histórico de dados e perfis de tráfego. A decisão sobre concessão de recursos é feita com base em análise de políticas de negociação, as quais apresentaremos na próxima subseção. Tais políticas permitem variações diversas para uma concessão de recursos completamente adaptável às necessidades do domínio.

É desejável que os contratos negociados entre os domínios apresentem uma grande quantidade de recursos, a fim de que a renegociação seja feita com uma frequência mínima. Deste modo, a duração dos contratos entre domínios tende a ser grande, apesar disso não ser uma regra. Sempre que os recursos do varejo estiverem escassos, significa que uma nova rodada de negociação de atacado deve ser realizada. Enfim, é clara a necessidade de constante correlação entre atacado e varejo.

Uma característica importante dessa estratégia de negociação é que, apesar de contratos de atacado representarem pedidos de diversos clientes, o domínio solicitado só necessita conhecer o domínio solicitante. Cabe a este último dividir os recursos obtidos entre os seus clientes. A tarefa de cada domínio é, portanto, reunir os recursos de que necessita (interna e externamente) a fim de poder realizar inteligentemente a negociação de atacado.

5.3 Políticas de Negociação

As políticas de negociação servem ao nível gerencial do modelo com a infra-estrutura para configuração das negociações que sejam interessantes para o domínio. É através destas políticas que o administrador de rede dividirá os recursos disponíveis em seu domínio da maneira que melhor lhe convir. A especificação de políticas permite distinguir cliente internos/externos aos domínios, atribuindo maior poder de barganha à parte de interesse e de maneira completamente configurável.

A implementação do modelo de negociação proposto disponibiliza minimamente as seguintes políticas de negociação de recursos: políticas de admissão, políticas de restrição de banda, políticas de renegociação e políticas de aceitação de contratos.

As três primeiras categorias de políticas fazem parte da configuração do provedor do(s) serviço(s) (domínio solicitado). A última categoria faz parte da configuração do cliente: sempre que estipular um contrato, ele deve escolher o comportamento desejado a partir do recebimento de uma contra-proposta.

Estas políticas são suficientes para atender as necessidades do modelo de negociação. Contudo, se para algum caso específico for preciso novas políticas, a idéia é que elas possam ser implementadas na forma de *plugins* e carregadas dinamicamente ao *software* que implementa o modelo.

5.3.1 Políticas de Admissão

Definem os clientes autorizados a negociar com o domínio. Esta categoria tem por objetivo restringir o acesso aos recursos do domínio para clientes de interesse efetivo. Caso um cliente não esteja presente na lista dessa categoria de políticas, o protocolo de negociação o rejeita logo no início da primeira fase do protocolo SLA-COPS.

Com o uso dessas políticas, clientes indesejáveis são rapidamente descartados e o domínio evita gastos desnecessários de largura de banda com mensagens de sinalização, numa clara preocupação com a escalabilidade do modelo.

5.3.2 Políticas de Restrição de Banda

Complementares às políticas de admissão, estas políticas restringem a quantidade máxima de recursos que pode ser concedida a cada cliente autorizado a negociar com o domínio. Esta categoria está dividida em duas subcategorias:

- **Políticas de Retrição Total:** limitam, para cada serviço, a quantidade máxima geral de recursos por cliente do domínio. Isto é, a limitação vale para a somatória de recursos de todos os domínios que um determinado cliente solicita a um mesmo domínio, seguindo o Modelo Cascata apresentado no Capítulo 3.
- **Políticas de Restrição por Domínio:** limitam, para cada serviço, a quantidade máxima de recursos para domínios específicos por cliente, também seguindo o Modelo Cascata de comunicação entre domínios.

Estas duas subcategorias de políticas estão intimamente ligadas. Na verdade, a primeira restringe o limite superior para a somatória da segunda. Suponha o exemplo da Figura 5.2. Nesta figura, temos o domínio B como intermediário de A no acesso a C e D. Uma boa prática de B, caso queira negociar contratos de QoS com A, é restringir a quantidade máxima de recursos cedidos (largura de banda) para cada serviço e para cada domínio. Caso não implemente essas políticas e ainda assim permita a admissão de A, o domínio B corre o risco de ter seus recursos esgotados por A.

Ainda tomando por base o exemplo dado, suponha que A solicite a B recursos do próprio B, de C e de D. Políticas de restrição por domínio poderiam ser criadas por B com o intuito de limitar o uso de recursos individuais de cada domínio solicitado por A. Com isso, o domínio B não correria o risco de representar um simples intermediário de A no acesso a C e D. Por outro lado, todo tráfego de A com destino a B, C ou D passa necessariamente por B. Portanto, B possui neste caso um tráfego de trânsito potencial grande que deve ser diferenciado. Para evitar esse tipo de gargalo por tráfego de trânsito, B pode também implementar políticas de restrição total para o domínio A. Deste modo, contratos de A com B não poderiam ultrapassar os limites estipulados nestas políticas, mesmo que as políticas de restrição por domínio assim permitissem.

Caso as políticas de restrição estejam ausentes para um cliente admitido no domínio, isso significa que ele não possui qualquer restrição de banda, a não ser a capacidade do domínio. Esta certamente não é uma boa prática na maioria dos casos.

5.3.3 Políticas de Renegociação

Também relativas a cada cliente autorizado a negociar com o domínio, definem o tempo mínimo para renegociação de um contrato. Deste modo, têm por principal objetivo a

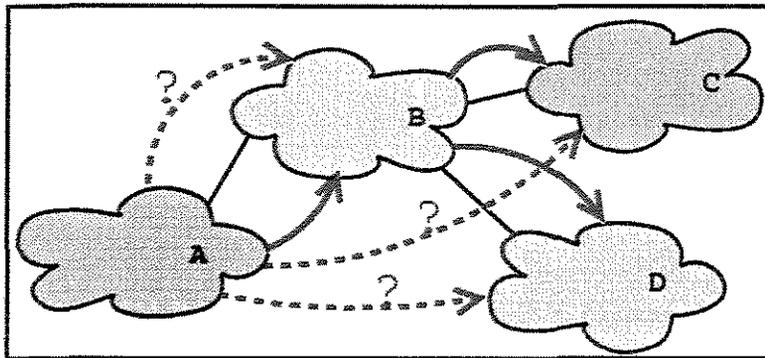


Figura 5.2: Cenário ilustrativo para Políticas de Restrição de Banda

diminuição do número de contextos de negociação. Como as políticas de admissão, esta categoria também é executada na primeira fase do protocolo SLA-COPS. Caso existente para o cliente em negociação e caso o cliente tenha um contrato negociado no momento, a política verifica se o contrato atual já está expirado. Apenas nesse contexto o cliente está autorizado a renegociar o contrato atual. Se ausentes para um determinado cliente, significa que ele é livre para renegociar o contrato a qualquer momento.

A idéia desta categoria de políticas é auxiliar a estratégia de atacado/varejo apresentada anteriormente. Através da aplicação dessa política, cabe ao administrador da rede fazer com que os contratos de atacado com o domínio destino sejam negociados com pequena frequência. Para os contratos de varejo a renegociação é menos drástica e pode ocorrer com uma frequência maior.

5.3.4 Políticas de Aceitação de Contratos

Esta última categoria diz respeito à configuração da parte cliente do modelo de negociação. Executada na terceira etapa do protocolo SLA-COPS, ela define o que um cliente está disposto a aceitar em termos da contra-proposta feita pelo domínio.

Existem três possíveis comportamentos:

- O cliente aceita qualquer contra-proposta, independentemente do que foi concedido pelo domínio a partir do solicitado.
- O cliente aceita apenas o que foi originalmente pedido. Se o domínio propõe conceder algo diferente do solicitado pelo cliente, este último não aceita a contra-proposta e o contrato não é estabelecido.

- O cliente aceita condicionalmente uma contra-proposta. Neste último caso, é necessário especificar a quantidade mínima de recursos que serão aceitos por serviço e por domínio.

5.4 Especificação de Contrato de Serviços

O modelo de contrato a ser proposto foi projetado com base no levantamento de requisitos apresentado no Capítulo 3. Tais requisitos foram divididos em unidades lógicas que reúnem parâmetros semanticamente relacionados.

Alguns dos parâmetros do contrato podem ser especificados pelo cliente. Outros, exclusivamente pelo provedor de serviços. A decisão sobre parâmetros de responsabilidade do cliente está intimamente ligada ao modo de negociação especificado na fase de configuração do protocolo SLA-COPS. Em qualquer situação, a unidade de QoS deve ser preenchida pelo cliente minimamente no que diz respeito a serviços desejados e largura de banda máxima para cada um deles.

As próximas subseções apresentam o projeto detalhado de contratos de serviços a ser utilizado nas simulações que validam o modelo aqui proposto e que serão apresentadas no próximo capítulo. Mais uma vez, é importante destacar que o protocolo SLA-COPS possui suporte a múltiplos formatos e padrões de contrato. Caso isso aconteça, o formato utilizado na negociação deve ser definido também na fase de configuração do protocolo SLA-COPS.

5.4.1 Unidade Comum

A Unidade Comum foi idealizada com o objetivo de reunir informações de identificação e controle do contrato. Mais especificamente, essas informações compreendem:

- Formato do contrato: identifica o tipo de contrato utilizado na negociação.
- Especificação de serviços: identifica os serviços negociados no contrato. Essa identificação é útil quando da utilização de serviços bem conhecidos (WKS - *Well Known Services*)¹, pois neste caso dispensa a especificação dos parâmetros de cada um deles.
- Identificação do provedor: identifica o domínio solicitado, ou seja, o domínio que concede os recursos a que se refere o contrato.
- Identificação do cliente: identifica o solicitante.

¹Um serviço bem conhecido é aquele para o qual existe uma definição clara e precisa sobre quais garantias de desempenho um provedor oferece ou espera receber quando está negociando um contrato [30].

- Tipo de cliente: identifica se um cliente é o gerente de um outro domínio ou se é um cliente local.
- Data de criação do contrato.
- Data de validade.

5.4.2 Unidade de Topologia

A Unidade de Topologia reúne informações sobre os nós de fim-a-fim envolvidos na negociação. Formam em conjunto um grafo parcial onde os vértices intermediários não são especificados, e sim apenas os vértices das extremidades, representando os nós do fim-a-fim. O cliente deve preencher as informações de nós de origem e o provedor deve preencher as de nós de destino. Deste modo, o cliente não precisa conhecer a topologia do provedor, e vice-versa.

Podem ser especificados diversos pares de nós de origem/destino. Cada par recebe uma identificação e é posteriormente utilizado em cláusulas do contrato, conforme será demonstrado na subseção a seguir.

5.4.3 Unidade de QoS

Esta última unidade é responsável por agregar as informações sobre as “cláusulas” do contrato. Entenda-se por cláusula todos os detalhes técnicos que envolvem a negociação, devidamente divididos por domínio e por serviço. Existe uma Unidade de QoS para cada cláusula do contrato. Deste modo, um contrato possui potencialmente uma lista de Unidades de QoS.

Cada uma dessas unidades é formada por diversas subunidades:

- Subunidade de Escopo: carrega informações relativas à topologia a que se refere a cláusula, ou seja, sobre os nós de fim-a-fim a que ela se refere. Na verdade, esta subunidade armazena uma ou mais referências para elementos da Unidade de Topologia.
- Subunidade de Descritor de Tráfego: possui informações sobre o tipo de tráfego que será enquadrado nesta unidade de QoS, ou seja, nesta cláusula do contrato. Essas informações compreendem código de serviço (de acordo com o campo ToS do cabeçalho IP ou equivalente), portas de origem/destino e protocolo de rede utilizado.
- Subunidade de Descritor de Carga: armazena informações sobre parâmetros de condicionamento e policiamento de tráfego. Representa uma parte bastante técnica do contrato. Seus parâmetros reúnem tipo de descritor (mecanismo de policiamento,

com os algoritmos de condicionamento e policiamento a serem utilizados), parâmetros relativos ao descritor (tamanho da rajada, vazão média, entre outros²) e tratamento de excesso (condicionamento, remarcação ou descarte).

- Subunidade de Parâmetros de QoS: possui as informações fundamentais para a garantia da QoS propriamente dita. Os parâmetros compreendem: atraso, variação de atraso e descarte máximos permitidos para a cláusula do contrato. Caso serviços WKS estejam sendo utilizados no contrato, o preenchimento das informações dessa subunidade é dispensável.
- Subunidade de Disponibilidade de Serviços: armazena informações sobre horários de disponibilidade da QoS prometida na cláusula em foco.
- Largura de banda: especificada à parte de qualquer subunidade por uma questão de simplicidade, toda cláusula deve obviamente destacar a largura de banda máxima que o cliente deve poder usufruir com a QoS prometida e delimitada pelas outras subunidades.

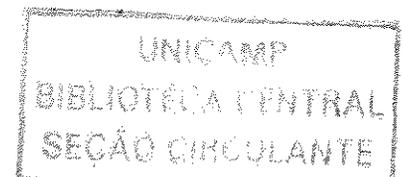
5.5 Implementação

A implementação do modelo de negociação foi realizada como uma extensão às funcionalidades do *framework* apresentado no Capítulo 4. De [7] foi utilizado o *software* de BB, com os gerentes de recursos e de políticas, bem como o componente de aplicação das políticas nos roteadores. Já de [29] foi utilizada a plataforma de QoS adaptada ao uso conjunto das tecnologias MPLS e DiffServ.

O BB é originalmente composto por três módulos estruturais: o primeiro implementa as funcionalidades do gerente de recursos; o segundo implementa o gerente de políticas; o terceiro módulo implementa as funcionalidades previstas para os pontos de aplicação de políticas, vinculados aos dispositivos de nível físico. A estes componentes foi integrado um quarto módulo, denominado gerente de negociação e implementado segundo o modelo apresentado neste capítulo.

O *software* do BB apresenta uma implementação em linguagem de programação Tcl. O uso das extensões iTcl e Combat visam o suporte, respectivamente, à orientação a objetos e à arquitetura CORBA [32]. O mapeamento de CORBA para iTcl, realizado pelo pacote Combat, é obtido através das interfaces *Dynamic Invocation Interface* (DII) e *Dynamic Skeleton Interface* (DSI). Por uma questão de compatibilidade e reaproveitamento de código, este conjunto foi mantido para a implementação do modelo de negociação.

²Variáveis de acordo com o descritor escolhido



A comunicação entre cliente e servidor na negociação de serviços foi implementada por meio de chamadas de métodos no contexto da arquitetura CORBA. Estas chamadas de métodos emulam o funcionamento do protocolo SLA-COPS apresentado anteriormente. Com o uso de CORBA espera-se viabilizar a reutilização da implementação para um contexto real na medida em que esta arquitetura permite criar uma independência da implementação do BB com relação ao simulador NS.

A plataforma MPLS-DS foi desenvolvida sobre o simulador de redes *Network Simulator* (NS) [38], com o uso das extensões DS-NORTEL [20] e MNS [1] para suporte a DiffServ e MPLS. Ela tem como objetivo validar, por meio de simulações, o uso de políticas de configuração para a obtenção de QoS fim-a-fim em redes IP. A opção pela simulação proporciona grande liberdade na criação de cenários de utilização.

A implementação do modelo de negociação, além de adicionar componentes ao *framework* já estabelecido, envolveu ainda algumas modificações dos componentes originais. Todavia, as modificações visaram sempre a manutenção da arquitetura herdada a fim de que o trabalho realizado estivesse completamente voltado à questão da negociação de serviços propriamente dita.

A Figura 5.3 ilustra o *framework* com as devidas modificações para inclusão do modelo de negociação. É importante ressaltar que a implementação do modelo foi realizada de maneira parcial. Deste modo, o produto obtido não configura uma ferramenta comercial, mas sim um protótipo bem projetado que objetiva a validação das soluções propostas para os problemas discutidos ao longo deste trabalho.

5.5.1 Modificações Realizadas no Bandwidth Broker

O *software* original do BB prevê a negociação estática de contratos de serviços. Todo o processo é realizado na inicialização do sistema e qualquer mudança exige a reinicialização do mesmo. Portanto, o principal esforço na adição do modelo de negociação foi a adaptação da infra-estrutura existente para o contexto dinâmico e automático almejado às negociações de serviços.

Para atingir esse objetivo foram necessárias mudanças de pequeno impacto aos componentes existentes. Os módulos dos gerentes de recursos e de políticas tipicamente pesquisam por contratos de serviços vigentes no Repositório de Dados. É em torno destes contratos que ocorre todo o funcionamento do BB. No novo contexto, os contratos passam a ser dinamicamente modificados pelo gerente de negociação. Este componente recebe livre acesso ao Repositório de Dados e negocia contratos com base no modelo de atacado/varejo e nas políticas estipuladas pelo administrador da rede. Entretanto, a ação do novo componente no gerenciamento de contratos ocorre de maneira completamente transparente aos demais componentes, de maneira que os gerentes de recursos e de

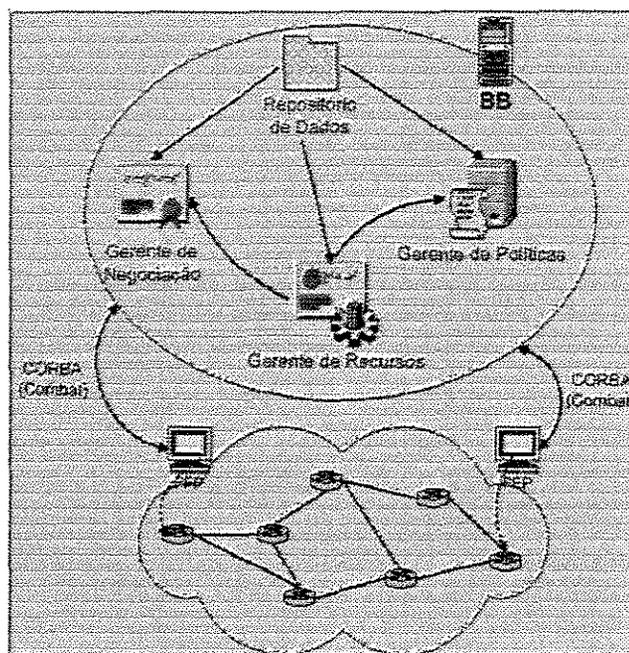


Figura 5.3: Implementação do BB.

políticas sequer tomam conhecimento da existência de um gerente de negociação.

A única integração necessária ao gerente de negociação com relação aos outros componentes diz respeito à obtenção da informação sobre quantidade de recursos disponível no domínio. Essa informação, obtida a partir do gerente de recursos, é imprescindível no mecanismo de negociação de serviços, tendo por objetivo avaliar o gerente de negociação a respeito da quantidade máxima de recursos por serviço que é possível ser negociada. Deste modo, o novo *software* do BB adiciona ao gerente de recursos uma interface de comunicação para ser utilizada pelo gerente de negociação. Esta interface configura a maior modificação realizada no *software* original. Foi implementada na forma de chamadas de métodos da arquitetura CORBA, de maneira análoga ao que foi feito na comunicação entre cliente e provedor de serviços no processo de negociação.

Outras modificações realizadas no *framework* original incluem:

- A aplicação das configurações sobre algoritmos de policiamento e de condicionamento de tráfego adquire caráter dinâmico. Originalmente os algoritmos eram escolhidos na inicialização do sistema para cada serviço e não podiam mais ser alterados. Esta nova funcionalidade foi incorporada ao gerente de políticas. Na nova tarefa, ele obtém as informações necessárias a partir do Repositório de Dados e as aplica nos roteadores de borda por meio dos pontos de aplicação de políticas (PEPs).

- Serviços passam a ser modificados e adicionados também em caráter dinâmico. Esta funcionalidade exigiu pequenas modificações nos gerentes de recursos e de políticas, os quais passam a ler periodicamente os contratos do domínio a fim de garantir que novos valores para os parâmetros dos serviços existentes, e mesmos novos serviços incorporados ao domínio, possam ser tratados de maneira adequada.

No *software* do BB original, as negociações de contratos são todas locais. Contudo, a adição de contratos e negociações entre domínios não implicou em qualquer modificação aos componentes existentes. Para estes, ficam completamente transparentes as diferenças entre contratos locais e remotos. A distinção entre estas duas categorias fica a cargo do gerente de negociação. É implementado, no mesmo módulo, uma parte cliente e outra provedora de serviços. O gerente de negociação utiliza a parte cliente da implementação para solicitar recursos externos aos gerentes de outros domínios e a parte servidora para prover recursos a clientes locais e a gerentes de outros domínios.

5.5.2 Modificações Realizadas na Plataforma MPLS-DS

Para a inclusão do modelo de negociação também foram realizadas modificações na plataforma MPLS-DS original especificada em [29]. As principais mudanças dizem respeito ao mecanismo de classificação de pacotes e à existência de múltiplos domínios no contexto do NS.

A classificação era realizada unicamente pela análise de informações de origem e destino do pacote. Esta solução mostrou-se deficiente para a negociação entre domínios. Neste contexto, roteadores de borda devem considerar, além de origem e destino, o endereço do último roteador que processou o pacote. A partir disso é possível descobrir qual foi o último domínio pelo qual ele passou e, conseqüentemente, honrar os contratos no contexto do modelo cascata de negociação entre domínios.

Na implementação do novo método de classificação de pacotes os roteadores LSR-DS de borda receberam uma nova subdivisão:

- **Roteadores de Ingresso:** reúne roteadores que servem como entrada para o domínio. Representam os roteadores de borda tradicionais, com a função típica de realizar a classificação e marcação dos pacotes. A diferença destes novos roteadores é que eles realizam a classificação com base nos novos critérios: endereço de origem, endereço do último roteador em que esteve o pacote e endereço de destino.
- **Roteadores de Egresso:** reúne roteadores que servem de saída do domínio. Estes novos componentes possuem como atribuição marcar pacotes com a informação do último roteador que processou o pacote, a fim de que o roteador de ingresso do

próximo domínio, se existente, saiba por qual domínio passou o pacote e possa, adequadamente, honrar possíveis contratos entre domínios.

A inexistência do conceito de domínios no NS provocou impacto nos mecanismos de monitoramento de LSPs e de tratamento de tráfego. Para superar esta deficiência, foram adicionados aos roteadores informações sobre o domínio do qual faziam parte. Além disso, os roteadores passam a conhecer, por meio dos contratos negociados, os roteadores de borda autorizados pelo domínio vizinho e que, portanto, devem ser interpretados como roteadores de tal domínio.

5.5.3 Repositório de Dados e Contratos de Serviços

O repositório de dados foi implementado na forma de arquivos texto de onde o gerente de negociação obtêm os dados de contratos negociados pelo domínio. Para tanto, foi construído um *parser* bastante simplificado que, a partir de uma sintaxe pré-definida, consegue traduzir as informações contidas no arquivo, as quais estão divididas em seções, em parâmetros úteis para o perfeito funcionamento do BB. Cada seção do contrato, conforme proposto neste capítulo, é representada por uma etiqueta diferente. Dentro das etiquetas existem informações particulares da unidade em questão.

Contratos de serviços são, portanto, armazenados de modo permanente na forma de arquivos texto. Do ponto de vista do gerente de negociação, estes arquivos são interpretados e carregados para estruturas especialmente confeccionadas para uso do gerente de recursos no mecanismo de análise de cumprimento dos contratos.

É importante notar que estas não configuram soluções ideais para a representação e armazenamento de contratos e foram realizadas desta maneira por uma questão de simplicidade e por não representarem o foco principal deste trabalho. Uma análise de soluções mais sofisticadas é discutida no Capítulo 3.

5.6 Resumo

Apresentamos neste capítulo uma proposta de modelo de negociação entre BBs de domínios distintos. Este modelo visa o oferecimento escalável de QoS fim-a-fim, inclusive para situações em que os dados tenham que trafegar entre domínios de administrações distintas e que implementem tecnologias de QoS quaisquer.

A fim de validar o modelo proposto, o próximo capítulo será totalmente dedicado a simulações. Apresentaremos em detalhes dois cenários de utilização que fazem uso do *framework* de QoS completo, destacando sobretudo as funcionalidades adquiridas a partir da incorporação do modelo de negociação ao mesmo.

Capítulo 6

Validação do Modelo de Negociação

A fim de demonstrar as funcionalidades do modelo de negociação proposto neste trabalho, apresentamos neste capítulo dois estudos de caso representados por simulações realizadas sobre o *framework* de QoS modificado para obtenção de QoS fim-a-fim. Os experimentos foram modelados para simular cenários que, apesar de simples, apresentam tipos de tráfegos em situações onde o oferecimento de QoS é claramente bem-vindo.

O primeiro experimento apresenta um ambiente que inclui 3 domínios administrativos distintos. O principal objetivo da simulação realizada neste primeiro caso é validar o modelo de negociação no que diz respeito ao oferecimento de soluções capazes de garantir o cumprimento de contratos, locais e externos aos domínios, em situações de descarte e atraso excessivos. O segundo experimento faz uso de quatro domínios. Esta simulação pretende mostrar adaptações realizadas por um domínio quando da ineficiência de uma rota entre domínios inicialmente escolhida.

Nas próximas páginas detalhamos os experimentos citados acima. Inicialmente apresentamos os tipos de tráfegos utilizados nas simulações. Logo a seguir comentamos sobre as classes de serviços oferecidas em cada um dos domínios. Em seguida, tratamos das políticas de atuação disponibilizadas para o BB e, finalmente, mostramos as simulações passo a passo e apresentamos os resultados obtidos.

6.1 Tráfego Gerado

Foram selecionados dois tipos de tráfegos para os experimentos. Estes tráfegos visam emular o comportamento típico dos tráfegos de algumas aplicações reais as quais se beneficiam ou até mesmo exigem a utilização de QoS em situações de congestionamento e escassez de recursos, a fim de que sejam executadas satisfatoriamente.

O primeiro tipo de tráfego incorpora as características de aplicações que envolvem tráfego de voz. Estas características incluem taxa de geração constante e tamanho de

pacote reduzido. O tamanho do pacote escolhido é de 66 *bytes* e a taxa de geração é de um pacote a cada vinte milissegundos.

O segundo tipo de tráfego simula as características de uma aplicação de tráfego exponencial. A idéia de sua utilização é emular aplicações que não exijam baixo atraso, mas sim confiabilidade na entrega dos pacotes. Este é o cenário típico de uma aplicação de *e-business*, por exemplo. Neste caso, o tamanho escolhido para os pacotes é de 512 bytes e a taxa de geração é de 432 Kbps.

A Tabela 6.1 apresenta detalhes sobre os tráfegos apresentados.

Tipo de tráfego	Tamanho do pacote	Taxa
Exponencial	512 bytes	432 Kbps
Voz	66 bytes	66 bytes/20ms

Tabela 6.1: Características dos tráfegos disponibilizados

6.2 Classes de Serviço

Foram estipuladas três classes de serviços concordadas entre os domínios: *Best Effort* (BE), *Assured Forwarding* (AF) e *Expedited Forwarding* (EF).

A classe AF oferece baixo descarte e alta probabilidade de entrega de pacotes. É adequada ao tráfego de aplicações que exijam requisitos de segurança e confiabilidade. A classe EF apresenta prioridade de tratamento sobre as demais classes. Suas características são baixos atraso, descarte e variação de atraso. A classe BE acomoda o tráfego que não precisa receber tratamento diferenciado. A Tabela 6.2 reúne informações sobre atraso e descarte máximo para cada classe.

Classe	Atraso	Descarte
<i>Best Effort</i> (BF)	-	-
<i>Assured Forwarding</i> (AF)	4 s	20%
<i>Expedited Forwarding</i> (EF)	210 ms	20%

Tabela 6.2: Características das classes de serviços

As classes de serviço foram simuladas através de 4 filas do tipo *Droptail*, utilizando o escalonador *Weighted Round Robin* (WRR). São três filas físicas, sendo que uma dessas

filas é dividida logicamente em duas filas virtuais. Cada classe de serviço ocupa uma fila. A classe AF ocupa as duas filas virtuais.

No início da simulação, todo tráfego pertence à classe BE e recebe o mesmo tratamento, independente de tipo ou origem. Durante a simulação ocorrem os processos de negociação, levando a um tratamento diferenciado dos tráfegos referenciados em contratos.

É importante destacar a utilização de serviços WKS em todos os domínios simulados nos experimentos.

6.3 Políticas de Atuação

Foram definidas duas políticas de atuação para a simulação. Sempre que acionadas, as políticas são automaticamente traduzidas pelo BB em parâmetros de configuração para os roteadores do domínio.

A primeira política se refere à atualização do peso das filas dos roteadores para uso do WRR. Esta política é acionada sempre que um determinado fluxo de dados sofre descarte maior que o estipulado para a sua classe. No início da simulação, as filas de todas as classes possuem o mesmo peso.

A segunda política atua sobre a criação de novos LSPs. Esta política é disparada em duas situações: caso o atraso de um fluxo extrapole o estipulado para a sua classe ou caso ocorra excesso de descarte após três atuações consecutivas da primeira política. O novo LSP é criado com base em recursos disponíveis no domínio. Uma nova rota é escolhida por um agente que implementa o algoritmo de roteamento baseado em banda residual [8].

6.4 Experimentos Realizados

Nas próximas subseções apresentamos as simulações realizadas destacando as ocorrências importantes, além dos resultados obtidos e algumas análises sobre estes últimos. Nas figuras sobre topologias de simulação, cada círculo representa um roteador MPLS/Diffserv. Os hexágonos representam os clientes. Todo enlace da topologia mostrada possui capacidade total de 3.456 Kbps. Os roteadores MPLS/DiffServ possuem 4 filas para cada enlace com o qual possuem ligação. As filas são capazes de armazenar até 1000 pacotes, independente de seu tamanho.

As medições de atraso e descarte médios mostradas nos gráficos de resultados foram realizadas com base em acumuladores e medidores disponibilizados pelo NS. O descarte de um LSP representa a média de descartes de todos os roteadores que o compõem. De maneira análoga, o atraso de um LSP é calculado com base no atraso médio dos pacotes

no percurso realizado entre pares de roteadores do LSP. Estas medições foram realizadas com frequência de 0,1 segundo.

6.4.1 Experimento 1

A Figura 6.1 mostra a topologia das redes simuladas no primeiro experimento. O BB do domínio A (BB-A) atua sobre os roteadores R01 a R08, o do domínio B (BB-B) atua sobre os roteadores R09 a R14 e o do domínio C (BB-C) atua sobre os roteadores R15 a R20.

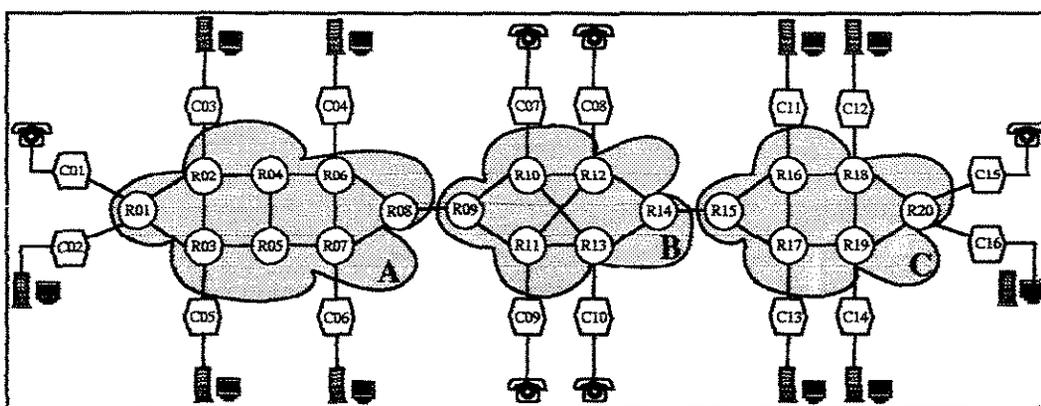


Figura 6.1: Topologia da simulação

Três LSPs pré-configurados são criados no início da simulação. O LSP 2000 é criado pelo BB-A e segue a rota R01-R02-R04-R06-R08. O LSP 2001 é criado pelo BB-B, seguindo a rota R09-R10-R12-R14. O BB-C cria o LSP 2002, que segue a rota R15-R16-R18-R20.

Cada cliente da topologia mostrada gera um dentre os dois tipos de tráfego. O cliente C01 gera tráfego de voz para o cliente C015 e o cliente C02 gera tráfego exponencial para o cliente C16. Os demais clientes geram apenas tráfego local. Os clientes C03, C04, C05 e C06 trocam tráfego exponencial. Já os clientes C07, C08, C09 e C10 trocam tráfego de voz. Os cliente C11, C12, C13 e C14 também trocam informações por meio de tráfego de voz.

Em $t = 1$ s, o BB-B negocia satisfatoriamente um contrato com o BB-C. Ele negocia, com destino ao domínio C, 1780 Kbps para fluxos de tráfego da classe AF e 890 Kbps para a classe EF. No instante $t = 1,5$ s, é a vez do BB-A negociar um contrato com o BB-B. Neste contrato, o BB-B assegura 1000 kbps para fluxos da classe AF e 600 Kbps

para fluxos de tráfego EF provenientes do domínio A e com destino ao domínio C. Ainda no instante $t = 1,5$ s, inicia-se o tráfego dos clientes C01 a C15 e C02 a C16.

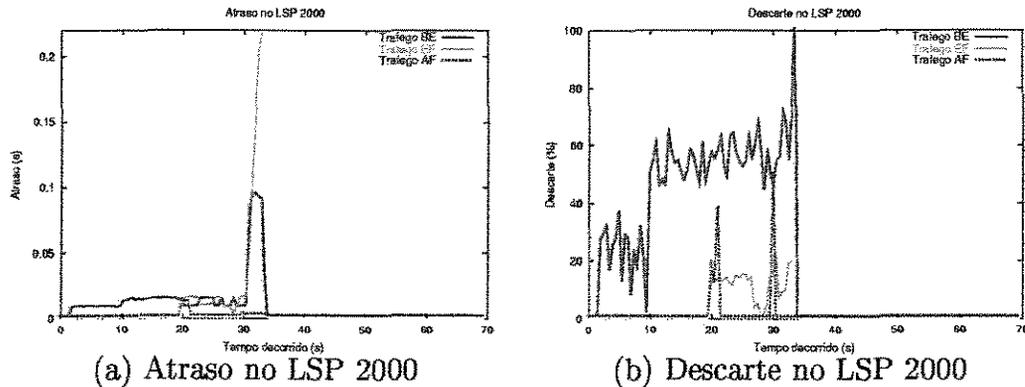


Figura 6.2: Gráficos do LSP 2000

Aos 10 segundos, inicia-se o tráfego exponencial entre os clientes C03, C04, C05 e C06. O domínio A começa a ter seus enlaces saturados e os clientes C01 e C02 vêm seus tráfegos ameaçados de atraso e descarte. No instante $t = 20$ s, os clientes C01 e C02 realizam negociações de contratos com o BB-A. O cliente C01 negocia 500 Kpbs de fluxos EF com destino a C15. Enquanto isso, o cliente C02 negocia 480 Kbps de fluxos AF para C16. O tráfego de C01 passa a pertencer à classe EF e o de C02 à classe AF.

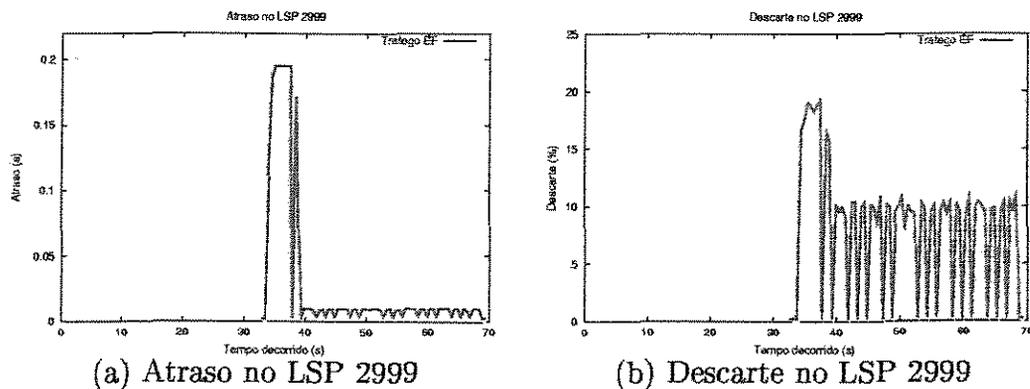


Figura 6.3: Gráficos do LSP 2999

Em $t = 21,1$ s o BB-A é informado sobre o excesso de descarte de tráfego AF nos roteadores do LSP 2000 (Figura 6.2 -b). Neste momento, ele executa a política de

reconfiguração das filas referentes ao tráfego AF nos roteadores do LSP 2000. O descarte para este tráfego volta, então, a índices aceitáveis.

Quando a simulação alcança os 30 segundos, o cliente C03 negocia um contrato com o BB-A. O contrato de C03 prevê tratamento diferenciado para 400 Kbps de seu fluxo através da classe AF. A partir desse momento, observa-se um aumento significativo do tráfego diferenciado no domínio A. O tráfego diferenciado está acumulado nos mesmos enlaces e o resultado é o aumento de descarte e de atraso para as classes AF e EF. O BB-A é notificado sobre o excesso de descarte dos tráfegos AF e EF, respectivamente, nos instantes $t = 30,1$ s e $t = 30,6$ s (Figura 6.3-b). Sua ação é disparar a política de reconfiguração de filas para ambas as classes. Contudo, essas ações não são suficientes para estabilizar o atraso do tráfego EF. No instante $t = 33,1$ s o atraso deste tráfego alcança 215 ms, extrapolando o máximo permitido (Figura 6.2-a). A ação do BB, desta vez, é disparar a segunda política definida para a simulação: é criado um novo LSP, identificado como LSP 2999. O novo LSP segue a rota R01-R03-R05-R07-R08. O tráfego EF proveniente de C01 passa a ser roteado para o novo LSP. Com isso, os índices de atraso e descarte de todos os fluxos de tráfego do domínio A voltam a valores aceitáveis.

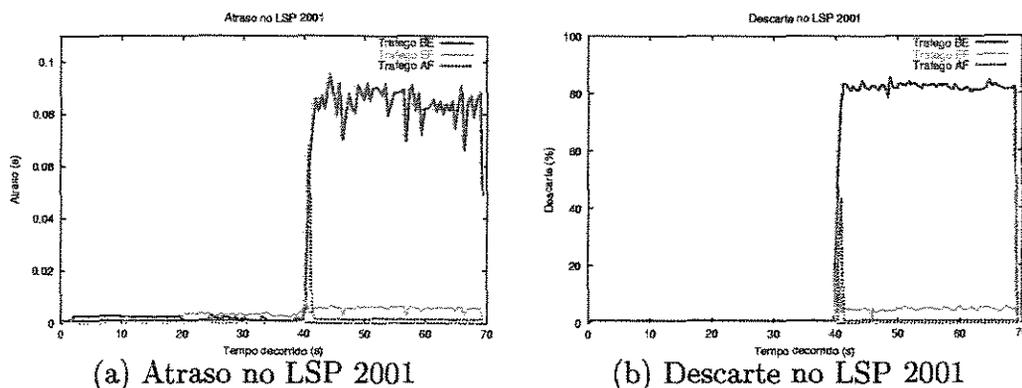


Figura 6.4: Gráficos do LSP 2001

No instante $t = 40$ s, é iniciado o tráfego de voz entre os clientes C07, C08, C09 e C10. É a vez do domínio B ter seus enlaces saturados (Figura 6.4). Em $t = 40,5$ s, o tráfego EF do LSP 2001 sofre descarte de 22,14%. Em $t = 41$ s, o tráfego AF sofre descarte de 25,92%. O BB-B executa, então, a política de reconfiguração de filas para ambas as classes. Essas duas medidas são suficientes para trazer de volta a estabilidade ao domínio B.

Aos 50 segundos da simulação é iniciado o tráfego de voz entre os clientes do domínio C. Os roteadores do LSP 2002 passam a sofrer descartes em excesso (Figura 6.5-b). Em t

– 50,89 s os monitores detectam descarte de 24,21% para o tráfego EF. Em $t = 51,89$ s é detectado descarte em excesso (24,13%) para o tráfego AF. O BB-C executa a política de reconfiguração de filas e o domínio C volta a uma situação de estabilidade de seus fluxos de tráfego. Em $t = 70$ s, a simulação é finalizada.

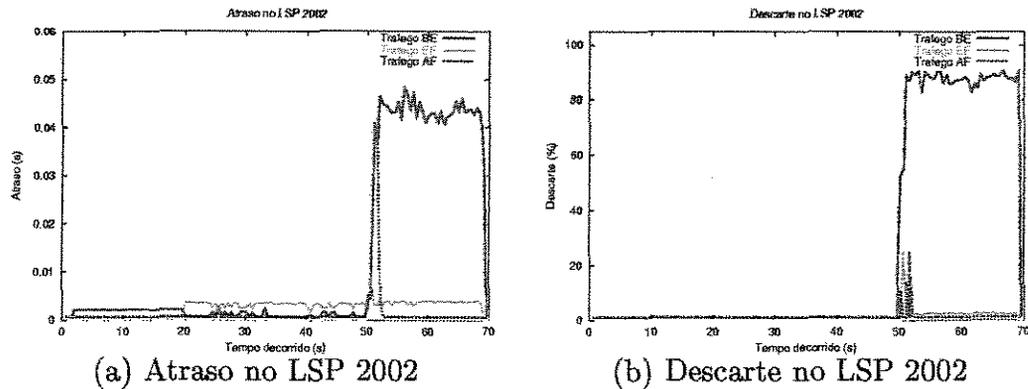


Figura 6.5: Gráficos do LSP 2002

Através dos gráficos, podemos observar a eficiência e coerência de ação dos BBs. Mais que isso, podemos comprovar a eficácia das políticas no sentido de garantir os índices combinados para atraso e descarte.

Nos LSPs 2001 e 2002, observamos a existência de tráfego da classe BE mesmo antes da inicialização do tráfego dos clientes locais (Figuras 6.4-a e 6.5-a). A existência deste tráfego confirma a eficiência dos métodos de policiamento existentes na borda do domínio A. Tal policiamento é responsável por garantir uma largura de banda máxima aos clientes C01 e C02, conforme estipulado em contrato. A ação dos policiadores é remarcar o tráfego em excesso para o código referente à classe BE, para que não recebam tratamento diferenciado.

Os gráficos da Figura 6.6 comprovam a eficiência do *framework* em garantir QoS fim-a-fim. Nestes gráficos, observamos que atraso e descarte se mantiveram em índices aceitáveis durante toda a simulação. É importante notar que, no caso de QoS fim-a-fim, os índices para os parâmetros de QoS obviamente tendem a ser mais altos. Afinal, o atraso e o descarte do fim-a-fim são a soma dos atrasos e descartes de todos os domínios. Portanto, quanto maior o número de domínios, maior tende a ser o valor máximo dos parâmetros de QoS.

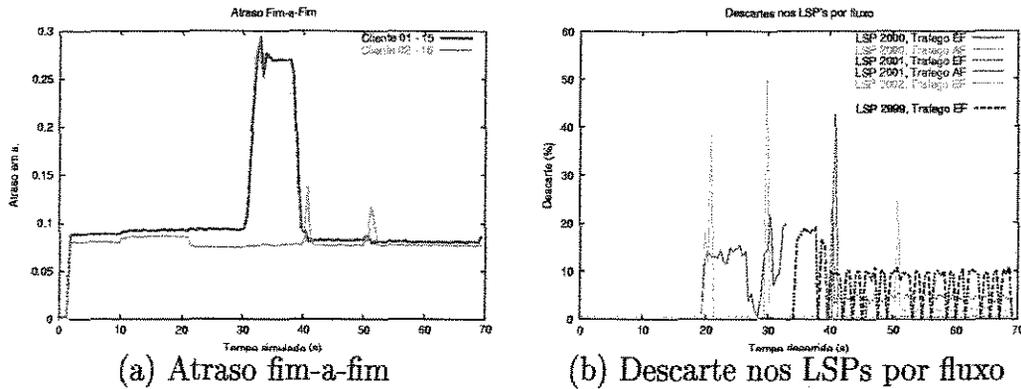


Figura 6.6: Gráficos de fim-a-fim

6.4.2 Experimento 2

A Figura 6.7 mostra a topologia das redes simuladas no experimento de número 2. O BB do domínio A (BB-A) atua sobre os roteadores R01 a R06, o do domínio B (BB-B) atua sobre os roteadores R07 a R10, o do domínio C (BB-C) atua sobre os roteadores R11 a R14 e o domínio D (BB-D) sobre R15 a R20.

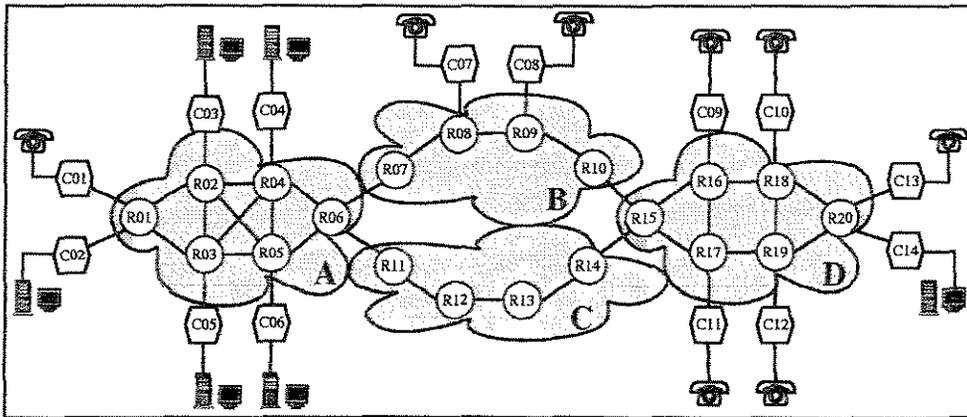


Figura 6.7: Topologia da simulação

Quatro LSPs pré-configurados são criados no início da simulação. O LSP 2000 é criado pelo BB-A e segue a rota R01-R02-R04-R06. O LSP 2001 é criado pelo BB-B, seguindo a única rota possível: R07-R08-R09-R10. Similarmente, o BB-C cria o LSP 2002, que segue a rota R11-R12-R13-R14. Finalmente, o BB-D cria o LSP 2003, que segue a rota

R15-R16-R18-R20.

Cada cliente da topologia mostrada gera um dentre os dois tipos de tráfego. O cliente C01 gera tráfego de voz para o cliente C13 e o cliente C02 gera tráfego exponencial para o cliente C14. Os demais clientes geram apenas tráfego local. Os clientes C03, C04, C05 e C06 trocam tráfego exponencial. Já os clientes C07 e C08 trocam tráfego de voz. Os clientes C09, C10, C11 e C12 também trocam informações por meio de tráfego de voz.

Em $t = 1$ s, o BB-B negocia satisfatoriamente um contrato com o BB-D. Ele negocia 1780 Kbps para fluxos de tráfego da classe AF. No instante $t = 1,5$ s, é a vez do BB-C negociar um contrato com o BB-D. Neste contrato, ele assegura 890 Kbps para fluxos da classe EF com destino ao domínio D. Ainda no instante $t = 1,5$ s, inicia-se o tráfego dos clientes C01 a C13 e C02 a C14. Aos 2 segundos da simulação, o BB-A negocia um contrato com o domínio B. Este contrato prevê 1000 Kbps para a classe AF com destino ao domínio D. Finalmente, o ciclo de negociações de atacado se encerra aos 2,5 s de simulação, quando o BB-A negocia 600 kbps de serviço EF para o domínio D com o BB-C.

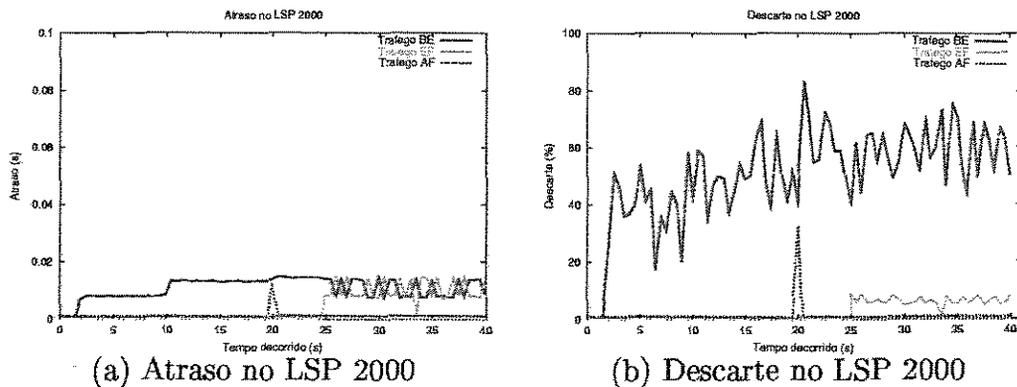


Figura 6.8: Gráficos do LSP 2000.

Aos 10 segundos, inicia-se o tráfego exponencial entre os clientes C03, C04, C05 e C06. Em $t = 15$ s, inicia-se o tráfego de voz entre os clientes C07 e C08. Os domínios A e B começam a ter seus enlaces saturados e os clientes C01 e C02 vêem seus tráfegos ameaçados de atraso e descarte. No instante $t = 20$ s, o cliente C02 decide realizar uma negociação com o BB-A. Ele negocia 500 Kbps de fluxos AF com destino a C14. O tráfego de C02 passa pelo domínio B no caminho a C14. Em $t = 20,1$ s o BB-A é informado sobre excesso de descarte de tráfego AF nos roteadores do LSP 2000 (Figura 6.8-b). Neste momento, ele executa a política de reconfiguração das filas referentes ao tráfego AF nos roteadores do LSP 2000. O descarte para este tráfego volta, então, a índices aceitáveis.

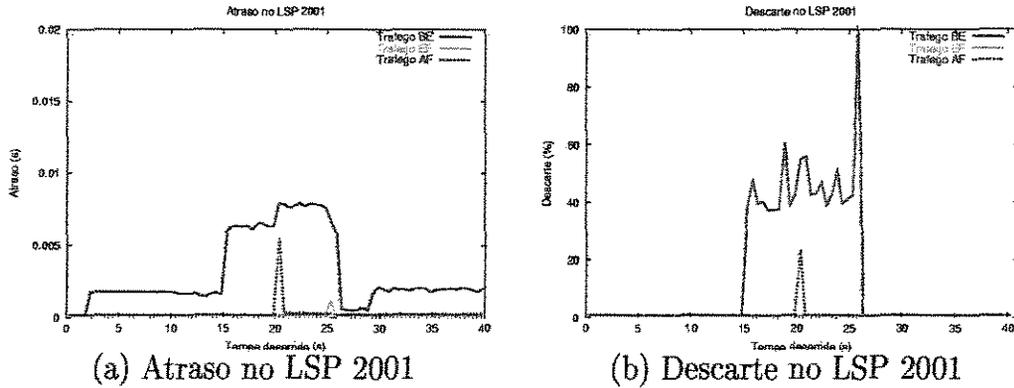


Figura 6.9: Gráficos do LSP 2001.

Em $t = 20,5$ s, o BB-B é notificado sobre um descarte de 22,72% em seu tráfego AF, ou seja, o tráfego de C02. Para honrar o contrato de atacado realizado com o BB-A, o primeiro executa a política de reconfiguração de filas para a classe AF, o que faz com que o descarte neste domínio volte a valores compatíveis com os contratos existentes (Figura 6.9-b).

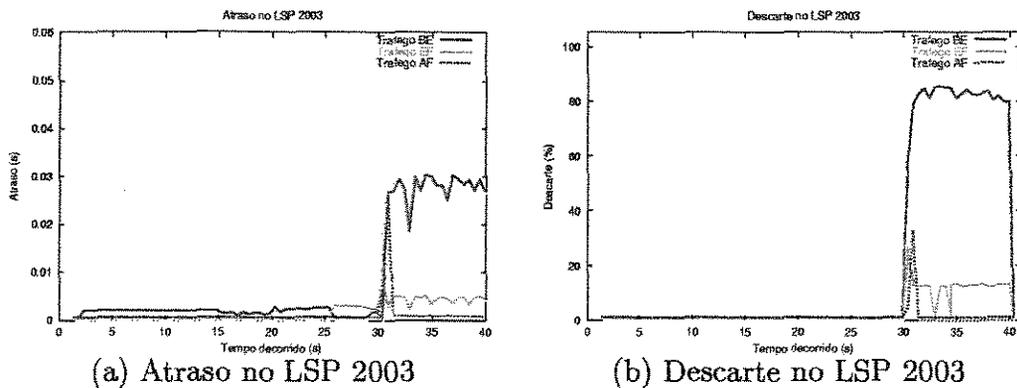


Figura 6.10: Gráficos do LSP 2003.

Aos 25 segundos de simulação, o cliente C01 negocia com o BB-A 480 Kbps de fluxos EF para C13. O tráfego de C01 é desviado para o domínio C no caminho até o destino. Como o domínio C estava ocioso até o momento, o grande descarte que vinha ocorrendo no tráfego de C01 cessa imediatamente, restabelecendo para este cliente a qualidade de serviço contratada.

Quando a simulação alcança os 30 segundos, é iniciado o tráfego de voz entre os clientes C09, C10, C11 e C12. É a vez do domínio D ter seus enlaces saturados (Figura 6.10-b). Em $t = 30,5$ s, o tráfego EF do LSP 2003 sofre descarte de 24,79%. Em $t = 31$ s, o tráfego AF sofre descarte de 31,81%. O BB-D executa, então, a política de reconfiguração de filas para ambas as classes. Essas duas medidas são suficientes para trazer de volta a estabilidade ao domínio D. Em $t = 40$ s, a simulação é finalizada.

O gráfico da Figura 6.11 confirma o sucesso da simulação ao apresentar valores de atraso reduzidos para os monitores de fim-a-fim durante todo o tempo. Estes resultados se mostram compatíveis com os atrasos parciais apresentados nas Figuras 8-a, 9-a e 10-a. Para o LSP 2002, os parâmetros de QoS permaneceram anulados durante toda a simulação. Este era o comportamento esperado, uma vez que a capacidade dos enlaces e das filas dos roteadores superava consideravelmente a necessidade real do tráfego transportado neste LSP.

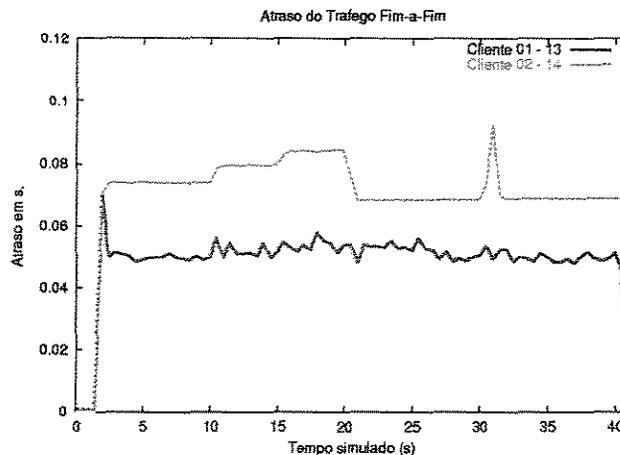


Figura 6.11: Gráfico de atraso fim-a-fim.

Um outro fato que merece ser notado é a estabilidade alcançada pelo sistema através da aplicação das políticas. Parâmetros que extrapolaram os contratos foram em todos os casos rapidamente corrigidos pela aplicação das mesmas.

6.5 Considerações Finais sobre as Simulações Realizadas

Apresentamos neste capítulo dois estudos de casos demonstrando o funcionamento do *framework* de QoS, sobretudo no que se refere ao modelo de negociação proposto neste

trabalho. Vale destacar que os cenários demonstrados visaram simular cenários onde a obtenção de QoS seja de boa utilidade, além de validar os diversos conceitos apresentados para o modelo de negociação.

Desta forma, as duas simulações apresentadas servem como cenários iniciais e típicos, ainda que simples, para validar estes conceitos. Os cenários são típicos porque simulam o comportamento desejado de alguns tráfegos largamente encontrados nas redes atuais e que não encontram na Internet convencional uma infra-estrutura adequada para tal.

Capítulo 7

Conclusões e Trabalhos Futuros

Apresentamos neste trabalho uma proposta de comunicação entre BBs de domínios administrativos distintos que foi executada sobretudo através da especificação de um modelo de negociação de serviços. O modelo proposto visa substituir as negociações estáticas praticadas na maioria dos sistemas que oferecem algum tipo de QoS, numa busca por automação e dinamismo, inclusive para a comunicação entre gerentes de diferentes domínios. A adição deste modelo ao *framework* de QoS mostrou-se decisiva para o alcance da qualidade de serviço fim-a-fim desejada.

O modelo de negociação proposto é formado em nível operacional pelo protocolo SLA-COPS, baseado em políticas de configuração e adaptado ao transporte de contratos heterogêneos. No nível gerencial, a proposta de negociação de atacado/varejo busca garantir transparência de configuração para o administrador da rede, também através do uso de políticas de configuração, além de oferecer escalabilidade para o contexto de negociações entre domínios.

Diversas pesquisas relacionadas a este tema têm feito uso do COPS para a confecção de modelos de negociação, de maneira análoga ao que foi realizado neste trabalho. Nossa contribuição é a especificação de um modelo para o contexto referido anteriormente que oferece garantias QoS fim-a-fim através do uso de políticas de configuração aplicadas ao contexto da negociação entre domínios distintos. As negociações de atacado/varejo, em última instância, reúnem políticas as quais facilitam a negociação de fim-a-fim, ao mesmo tempo em que permitem um crescimento escalável do número de domínios e contratos de serviços.

O uso combinado dos modelos DiffServ e MPLS apresenta boas perspectivas no que diz respeito à obtenção de QoS fim-a-fim para redes IP. A combinação de engenharia de tráfego da última com diferenciação escalável de fluxos de tráfego da primeira mostrou-se uma solução bastante adequada ao alcance dos objetivos traçados. Os resultados obtidos nas simulações realizadas estimulam o aprofundamento das soluções estudadas neste trabalho.

A infra-estrutura baseada no protocolo COPS possibilita o tratamento transparente de contratos, fato que confere portabilidade ao modelo. Além disso, a utilização de CORBA para a comunicação entre BB e rede simulada possibilita, em tese, a exportação deste *software* para um ambiente de rede real, onde as políticas implementadas podem ser reaproveitadas.

O uso de políticas certamente contribui para a obtenção da abstração e do dinamismo almejados. No modelo proposto, as tarefas administrativas ficam restritas à especificação de políticas. Elas definem o comportamento esperado do sistema quando do recebimento de propostas de contratos. Nos experimentos realizados, o sistema mostrou-se hábil na modificação dinâmica de parâmetros de configuração, tais como peso de filas dos roteadores, sem que houvesse ação externa do administrador da rede. Seguindo este pensamento, uma extensão natural para o trabalho seria a sofisticação dos níveis de abstração oferecidos. Mais especificamente, seria desejável que outros parâmetros, como tempo médio de atraso e porcentagem de descarte, fossem igualmente determinados pelo sistema, simplificando ainda mais a tarefa do administrador da rede. Uma sugestão seria a utilização de QoS qualitativo, o qual prevê a especificação de faixas ou níveis para os parâmetros citados por meio de interfaces gráficas intuitivas.

Com relação aos resultados obtidos, a ênfase mais imediata foi a de identificar os cenários de políticas que fazem sentido para tipos de tráfegos que se beneficiam da aplicação de conceitos de QoS e aprimorar o contexto da negociação de atacado/varejo. Trabalhos futuros certamente compreendem testes em cenários mais complexos de simulação e de ambiente real. Mais um vez, o uso de CORBA oferece um certo grau de transparência do ambiente de simulação a qual espera-se possibilitar que no futuro as técnicas desenvolvidas possam ser testadas em ambientes de rede reais.

Além de cenários mais complexos, um problema que certamente merece ser atacado em termos de trabalhos futuros é a questão modelagem e armazenamento de informações de políticas. O IETF tem sugerido alguns modelos de informação para uso específico de políticas, como o PCIM [10] e o QPIM [27]. Em termos de armazenamento de informações, a solução mais adequada parece ser o uso de servidores LDAP. Ressaltamos neste trabalho sucintamente os principais conceitos que compõem estas soluções. Linhas de pesquisa nesta área podem explorar técnicas capazes de evitar que um número grande de contratos e políticas possam se tornar um gargalo para o crescimento do sistema.

Um outro campo de interesse para aplicação em trabalhos futuros é a questão do gerenciamento de políticas, sobretudo no que diz respeito à detecção de conflitos. É praticamente impossível evitar que conflitos ocorram uma vez que em grandes redes muitas políticas podem ser especificadas por diversas pessoas em tempos diferentes e permanecerem válidas simultaneamente, sendo que algumas delas podem entrar em conflito entre si [6].

Finalmente, a questão justiça na concessão de recursos, deficiente no modelo cascata de comunicação entre domínios, representa também um bom campo para pesquisa em trabalhos futuros. O uso de políticas na admissão de clientes foi a alternativa utilizada neste trabalho no sentido de minimizar o problema.

Referências Bibliográficas

- [1] G. Ahn e W. Chun. Overview of MPLS Network Simulator: Design and Implementation, 1999. [Http://flower.ce.cnu.ac.kr/fog1/mns/index.htm](http://flower.ce.cnu.ac.kr/fog1/mns/index.htm).
- [2] AQUILA Project. <http://www-st.inf.tu-dresden.de/aquila>, 2000.
- [3] QBone Bandwidth Broker Architecture. <http://www.internet2.edu/qos/wg/documents-informational/20020709-chimento-et-al-qbone-signaling/>, 2001.
- [4] R. Braden, D. Clark, e S. Chenker. Integrated Services in the Internet Architecture: an Overview. Internet RFC 1633, IETF, Junho 1994.
- [5] CADENUS Project. <http://www.cadenus.org>, 2000.
- [6] J. A. Carrilho. *Detecção e Resolução de Conflitos em Políticas para Oferecimento de QoS na Internet*. Proposta de tese de doutorado, Unicamp, Dezembro 2002.
- [7] L. Costa. *Desenvolvimento de um Bandwidth Broker para a arquitetura DiffServ*. Dissertação de mestrado, Unicamp, Março 2001.
- [8] V. L. Costa. *Algoritmos de Roteamento*. Dissertação de mestrado, Unicamp, Abril 2002.
- [9] A. Abelém et al. QoS Fim a Fim Através da Combinação entre Serviços Integrados e Serviços Diferenciados. *Simpósio Brasileiro de Redes de Computadores (SBRC 2001)*, Maio 2001.
- [10] B. Moore et al. Policy Core Information Model - Version 1 Specification. Internet RFC 3060, IETF, Fevereiro 2001.
- [11] D. Awduche et al. Requirements for Traffic Engineering over MPLS. Internet draft, IETF, draft-ietf-mpls-traffic-eng.00.txt, Outubro 1998.
- [12] D. Goderis et al. Service Level Specification Semantics and Parameters. Internet draft, IETF, draft-tequila-sls-00.txt, Novembro 2000.

- [13] E. Crawley et al. A Framework for QoS-based Routing in the Internet. Internet RFC 2386, IETF, Agosto 1998.
- [14] E. Rosen et al. Multiprotocol Label Switching Architecture. Internet RFC 3031, IETF, Janeiro 2001.
- [15] F. Faucheur et al. MPLS Support of Differentiated Services. Internet draft, IETF, draft-ietf-mpls-diff-ext-09.txt, Abril 2001.
- [16] J. Boyle et al. The COPS (Common Open Policy Service) Protocol. Internet RFC 2748, IETF, Janeiro 2000.
- [17] J. D. Case et al. Simple Network Management Protocol (SNMP). Internet RFC 1157, IETF, Maio 1990.
- [18] O. Mota et al. Uma Arquitetura Adaptável para Provisão de QoS na Internet. *Simpósio Brasileiro de Redes de Computadores (SBRC 2001)*, Maio 2001.
- [19] P. Pan et al. BGRP: Sink-Tree-Based Aggregation for Inter-Domain Reservations. *Journal of Communications and Networks*, Junho 2000.
- [20] P. Piedad et al. A Network Simulator Differentiated Services Implementation. Internet draft, Nortel Networks, draft-tequila-sls-00.txt, Julho 2000.
- [21] R. Yavatkar et al. A Framework for Policy-based Admission Control. Internet RFC 2753, IETF, Janeiro 2000.
- [22] S. Blake et al. An Architecture for Differentiated Services. Internet RFC 2475, IETF, Dezembro 1998.
- [23] S. Salsano et al. Inter-domain QoS Signaling: the BGRP Plus Architecture. Internet draft, IETF, draft-salsano-bgrpp-arch-00.txt, Maio 2002.
- [24] T. Braun et al. Management of Quality of Service Enabled VPNs. *IEEE Communications Magazine*, Maio 2001.
- [25] T. Nguyen et al. COPS-SLS: A Service Level Negotiation Protocol for the Internet. *IEEE Communications Magazine*, páginas 158–165, Maio 2002.
- [26] Y. Bernet et al. A Framework for Differentiated Services. Internet draft, IETF, draft-ietf-diffserv-framework-00.txt, Maio 1998.
- [27] Y. Snir et al. Policy Framework QoS Information Model. Internet draft, IETF, draft-ietf-policy-qos-info-model-01.txt, Abril 2000.

- [28] Y T'Joens et al. Service Level Specification and Usage Framework. Internet draft, IETF, draft-manyfolks-sls-framework-00.txt, Outubro 2000.
- [29] A. M. S. Guillén. *MPLS-DS: Uma plataforma para validação de políticas no contexto das redes MPLS/DiffServ*. Dissertação de mestrado, Unicamp, Dezembro 2001.
- [30] C. Kamienski e D. Sadok. Chameleon: uma Arquitetura para Serviços Avançados Fim a Fim na Internet com QoS. *Simpósio Brasileiro de Redes de Computadores (SBRC 2001)*, páginas 606–621, Maio 2001.
- [31] Object Management Group. <http://www.omg.org>, 2001.
- [32] R. Orfali, D. Harkey, e J. Edwards. *Instant CORBA*. Wiley Computer Publishing, 1997.
- [33] R. Pedatella, E. Madeira, e M. Magalhães. Um Modelo de Negociação de Serviços com QoS Fim-a-Fim na Internet. *Revista da Sociedade Brasileira de Telecomunicações (RevSBRT)*, Outubro 2003.
- [34] R. Pedatella, E. Madeira, e M. Magalhães. Um *Framework* para Obtenção de QoS Fim-a-Fim na Internet. *Simpósio Brasileiro de Redes de Computadores (SBRC 2003)*, Maio 2003.
- [35] Search Networking. <http://searchnetworking.techtarget.com>, 2003.
- [36] M. A. Siqueira. *Uma Arquitetura de Políticas Para Gerência de Redes MPLS*. Dissertação de mestrado, Unicamp, Abril 2002.
- [37] TEQUILA Project. <http://www.ist-tequila.org>. 2000.
- [38] The Network Simulator. <http://www.isi.edu/nsnam/ns>, 2001.
- [39] VINT Project. <http://www.isi.edu/nsnam/vint>, 1996.

