

Um *Framework* para Obtenção de QoS Fim-a-fim na Internet

Rodrigo Pedatella*

Edmundo Madeira*

Maurício Magalhães**

rodrigo.pedatella@ic.unicamp.br

edmund@ic.unicamp.br

mauricio@dca.fee.unicamp.br

* Instituto de Computação
Universidade Estadual de Campinas
Caixa Postal 6176
CEP 13083-970, Campinas - SP

** Fac. Eng. Elétrica e de Computação/DCA
Universidade Estadual de Campinas
Caixa Postal 6101
CEP 13083-970, Campinas - SP

Resumo

A obtenção de QoS fim-a-fim mostra-se como clara necessidade em uma Internet voltada aos interesses do mercado corporativo. Atualmente, o estudo de soluções que tornem este ambiente cada vez mais adaptado para as novas exigências do mercado tem recebido grande destaque. Neste contexto, este artigo apresenta um *framework* que visa garantir QoS fim-a-fim para a Internet através do uso conjunto dos modelos MPLS e DiffServ e da utilização de um *software* gerencial baseado em políticas de configuração.

Abstract

The acquirement of end-to-end QoS shows up to be a great necessity in an Internet worried with corporative market interests. Lately, the study of solutions that could adapt this environment to the new market needs has received great attention. In this context, this article introduces a framework which aims to assure end-to-end QoS to the Internet by integrating the models MPLS and DiffServ and by using a management software based on configuration policies.

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 notória 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 soluções que buscam suprir esta necessidade. Dentre estes, devemos ressaltar o modelo de Serviços Integrados (IntServ) [4], o modelo de Serviços Diferenciados (DiffServ) [15, 16], o mecanismo de Troca de Rótulo Multiprotocolar (MPLS) [12], a Engenharia de Tráfego [10] e o Roteamento Baseado em Restrições [11].

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.

Diversos trabalhos têm demonstrado a eficácia dos modelos DiffServ e MPLS. 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 citados acima 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* capaz de configurar, gerenciar e distribuir os recursos disponíveis. A esse *software* é dada a denominação de *Bandwidth Broker* (BB).

Este trabalho propõe o desenvolvimento de um *framework* que garanta QoS fim-a-fim para transmissões de dados realizadas em redes IP, notadamente a Internet. O *framework* a ser apresentado é composto, no nível operacional, por uma plataforma mista MPLS/DiffServ, adaptada ao oferecimento de serviços múltiplos com requisitos de QoS particulares. No nível gerencial apresentaremos um BB baseado em políticas de configuração.

Na sequência deste artigo, a Seção 2 apresenta o paradigma de administração baseado em políticas de configuração. Em seguida, a Seção 3 apresenta o nível gerencial do *framework*, representado pelo BB. A Seção 4 apresenta o nível operacional, representado pela plataforma MPLS-DS. A Seção 5 co-

menta detalhes de implementação dos componentes do *framework*. A Seção 6 mostra os resultados obtidos em sua aplicação. A Seção 7 apresenta alguns trabalhos relacionados e, finalmente, a Seção 8 reúne conclusões e sugestões de trabalhos futuros.

2 Gerência Baseada em Políticas de Configuração

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 novo esquema de gerenciamento que ofereça um nível mais alto de abstração.

O paradigma baseado em políticas de configuração mostra-se adequado ao novo contexto. 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”. Regras, por sua vez, são 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.

3 O *Bandwidth Broker* (BB)

A funcionalidade exercida por um BB consiste em oferecer ferramentas de suporte para o controle da qualidade de serviço em diferentes níveis de granularidade, efetuando o controle da admissão e da configuração dos dispositivos de rede e considerando políticas de utilização dos recursos disponíveis [6].

Um BB negocia contratos de serviços baseando-se nos recursos disponíveis em seu domínio. Um domínio é definido como o conjunto de todos os elementos de rede controlados pelo mesmo BB. 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 BB's, gerentes de domínios vizinhos, os quais desejam negociar recursos externos no intuito de garantir QoS fim-a-fim a clientes locais.

O *software* do BB representa o componente da camada gerencial do *framework* e 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 ou SNMP.

3.1 Comunicação entre Domínios

Um modelo de comunicação entre domínios define a maneira como os gerentes se relacionam com o objetivo de negociar serviços para os domínios que representam. Existem atualmente três propostas de modelos de comunicação entre domínios: Modelos Estrela, Hierárquico e Cascata.

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. A grande vantagem do Modelo Estrela é o controle administrativo que se ganha pela centralização das negociações no domínio solicitante. 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 de negociação pode crescer consideravelmente.

O Modelo Hierárquico promove a inserção de uma nova entidade com o objetivo de centralizar as negociações. Esta entidade, denominada *Service Exchange* (SE) [19], passa a ser responsável pela definição e negociação de serviços com base em informações fornecidas pelos gerentes de cada domínio. As informações compreendem serviços oferecidos, recursos disponíveis, topologia e características de enlaces entre domínios. Toda negociação ocorre por meio da SE. Periodicamente, os domínios enviam à SE informações sobre compra e venda de serviços. De posse destes dados, tal entidade realiza “rodadas de negociação”. O resultado é a permissão total, a permissão parcial, ou mesmo a negação dos serviços. O Modelo Hierárquico apresenta como vantagens escalabilidade e, principalmente, justiça na alocação de recursos. Como as decisões são tomadas por uma única entidade, pode-se

pensar em critérios sofisticados para uma concessão justa de recursos. Em contra-partida, a aplicação deste modelo apresenta problemas de ordem prática. É pouco provável que um provedor de acesso concorde em informar a uma entidade externa detalhes importantes sobre sua organização interna. Menos provável ainda é sua concordância com que essa entidade tome as decisões de alocação de recursos em nome de seu domínio.

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. O Modelo Cascata apresenta como vantagens escalabilidade e simplicidade. 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.

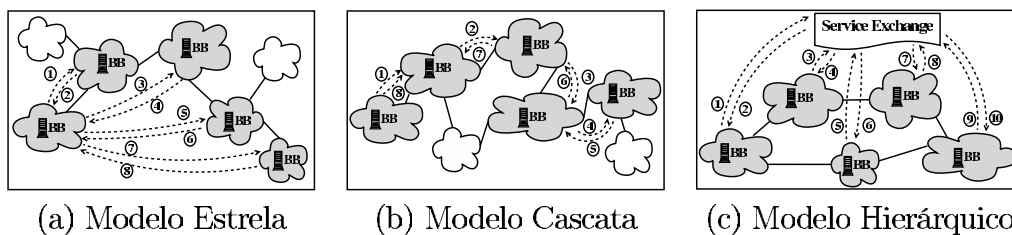


Figura 1: Modelos de comunicação entre domínios

3.2 Arquitetura de Desenvolvimento

A Figura 2 mostra a arquitetura de *Bandwidth Broker* proposta para compor o *framework* de que trata este artigo. Esta arquitetura é baseada nos trabalhos [6] e [18] com algumas modificações para a inclusão do modelo de negociação.

O **Repositório de Dados** armazena informações sobre as políticas estipuladas para o domínio. Além disso, deve armazenar informações sobre contratos negociados e sobre a configuração de todos os módulos gerenciais de um BB interagindo, portanto, com todos eles.

O **Objeto de Configuração** representa a entidade PEP. Instâncias deste objeto realizam as funções de monitoramento de tráfego nos enlances e configuração de roteadores, com base em instruções recebidas do Gerente de Políticas.

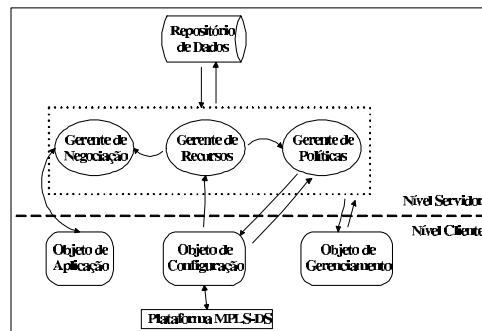


Figura 2: Arquitetura do *Bandwidth Broker*

O **Gerente de Recursos** é responsável por dimensionar recursos existentes na rede, além de monitorar o cumprimento de contratos vigentes. Ele realiza o monitoramento através de informações que recebe de instâncias do Objeto de Configuração. Além disso, interage com os outros dois módulos gerenciais: o Gerente de Políticas quando detecta o descumprimento de algum contrato para que este possa tomar as atitudes cabíveis, e o Gerente de Negociação no sentido de informá-lo a respeito de recursos disponíveis.

O **Gerente de Políticas** representa a entidade PDP. Tem como responsabilidade cuidar da interpretação e do mapeamento das políticas corporativas em parâmetros de configuração para os elementos de rede.

O **Gerente de Negociação** implementa a funcionalidade de negociação de contratos de serviços, inclusive entre domínios administrativos. Este componente deve negociar, instanciar e gerenciar os contratos do domínio.

O **Objeto de Aplicação** oferece a interface para um cliente realizar requisições de reserva e, finalmente, o **Objeto de Gerenciamento** tem a função de permitir ao administrador da rede a execução de instruções explícitas no controle sobre a configuração dos componentes.

4 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.

A plataforma foi desenvolvida sobre o simulador de redes *Network Simulator* (NS). A opção pela simulação foi motivada pela maturidade e riqueza de recursos encontradas na ferramenta escolhida. Além disso, a simulação permite uma grande liberdade na criação de cenários de utilização.

A idéia utilizada para a confecção da plataforma foi a modificação de componentes já existentes no simulador. As modificações visaram o suporte às seguintes funcionalidades previstas nos modelos DiffServ e MPLS: diferenciação de fluxos de tráfego, criação de rotas explícitas (LSP's), roteamento por LSP's e configuração das filas dos roteadores. A Figura 3 ilustra a plataforma MPLS-DS e seu relacionamento com o BB.

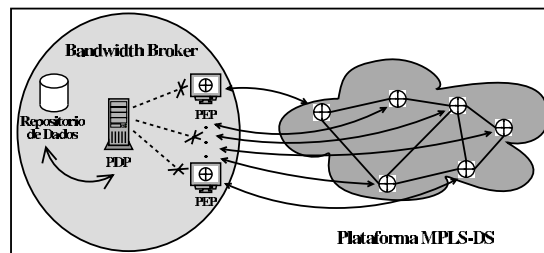


Figura 3: Plataforma MPLS-DS

As funcionalidades de diferenciação de fluxos e configuração das filas dos roteadores foram projetadas com base em conceitos do modelo DiffServ. A criação e roteamento dos LSP's foram projetados com base no modelo MPLS. Conforme já mencionado, a configuração e 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.

5 Implementação do *Framework*

A comunicação entre os elementos que compõem o *framework* é realizada por meio da *Common Object Request Broker Architecture* (CORBA), incluindo a comunicação entre elementos de rede e BB, assim como a comunicação entre clientes e BB no contexto da negociação de serviços. O objetivo é viabilizar a reutilização da implementação para um contexto real na medida em que permite criar uma independência da implementação do BB com relação ao simulador NS.

5.1 Implementação do *Bandwidth Broker*

A implementação do BB foi realizada em linguagem de programação Tcl, com uso das extensões iTcl e Combat para suporte, respectivamente, a orientação

a objetos e CORBA. A comunicação entre os elementos de rede e o BB foi implementada por meio de chamadas de métodos dos objetos que compõem a implementação. Estas chamadas de métodos emulam o funcionamento do protocolo COPS em diversas de suas mensagens de sinalização.

Foram implementadas quatro classes para a construção do BB: a primeira classe implementa as funcionalidades do Gerente de Recursos; a segunda implementa o Gerente de Políticas (PDP); a terceira implementa o Gerente de Negociação e a última classe implementa as funcionalidades previstas para o PEP. Objetos PEP, por estarem vinculados aos dispositivos de nível físico, são instanciados no próprio *script* de simulação. Os demais objetos do BB ficam completamente separados do simulador. Para todo domínio, é instanciado um objeto de cada classe gerencial do BB, além de um objeto PEP em cada elemento de rede.

O repositório de dados é responsável por armazenar informações de configuração de cada módulo gerencial do BB, além de informações sobre políticas para o domínio e sobre contratos de serviços. Foi implementado na forma de arquivos texto onde os objetos obtêm as informações.

A implementação do Gerente de Negociação contempla a necessidade de qualidade de serviço fim-a-fim através do uso de um modelo de comunicação entre domínios. Tal modelo foi embasado no Modelo Cascata apresentado anteriormente. A escolha deste se justifica pela junção de diversas características, em especial escalabilidade, maturidade e aplicabilidade. O exemplo mais notório de sua aprovação é a sua utilização no protocolo SIBBS [3], desenvolvido pelo Projeto QBone/Internet2.

5.2 Implementação da Plataforma MPLS-DS

Para a implementação da plataforma MPLS-DS foram realizadas modificações em extensões já existentes dos modelos DiffServ e MPLS para o simulador de redes NS. Para suporte ao MPLS foi utilizada a extensão MNS [1] e para suporte ao DiffServ foi utilizada a extensão DS-Nortel [14]. As modificações realizadas tiveram por objetivo a integração dos dois modelos aproveitando-se o que cada um apresenta de adequado à obtenção de QoS.

A principal mudança no código do MNS foi a adição do campo EXP ao cabeçalho do rótulo MPLS. Este campo passou a ser responsável por armazenar a informação da classe de serviço a que pertence o pacote. Novas classes foram criadas para que o campo EXP pudesse ser interpretado e um tratamento diferenciado pudesse ser dado ao pacote.

Também foram realizadas algumas mudanças no código do DS-Nortel. As mais importantes foram no mecanismo de classificação de pacotes e na associação de fluxos não marcados à classe do melhor esforço.

Informações detalhadas sobre a implementação da plataforma MPLS-DS podem ser encontradas em [18].

6 Validação do *Framework*

A fim de demonstrar as funcionalidades do *framework* proposto, apresentamos um estudo de caso para obtenção de QoS fim-a-fim em um ambiente que inclui 3 domínios administrativos.

A Figura 4 mostra a topologia das redes simuladas. Nesta figura, 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. 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.

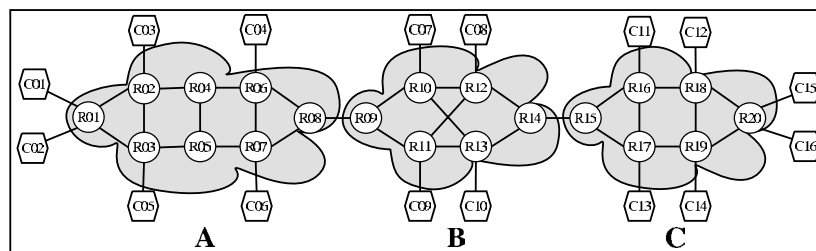


Figura 4: Topologia da simulação

6.1 Tráfego Gerado

Foram selecionados dois tipos de tráfegos para a simulação. O primeiro incorpora as características de tráfego de voz, como taxa de geração constante e tamanho de pacote reduzido. O segundo tipo simula as características de uma aplicação de tráfego exponencial. A Tabela 1 mostra detalhes sobre os tipos de tráfego.

Cada cliente da topologia mostrada gera um dentre os dois tipos de tráfegos. 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.

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

Tabela 1: Características dos tráfegos gerados

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 garantias de entrega de pacotes. É adequada ao tráfego de aplicações que exijam requisitos de segurança. 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 2 reúne informações sobre atraso e descarte máximo para cada classe.

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

Tabela 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.

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 LSP's. 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 [7].

6.4 A Simulação

Três LSP's 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.

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.

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 vêem 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 Kbps 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.

Em $t = 21,1$ s o BB-A é informado sobre excesso de descarte de tráfego AF nos roteadores do LSP 2000 (Figura 5-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

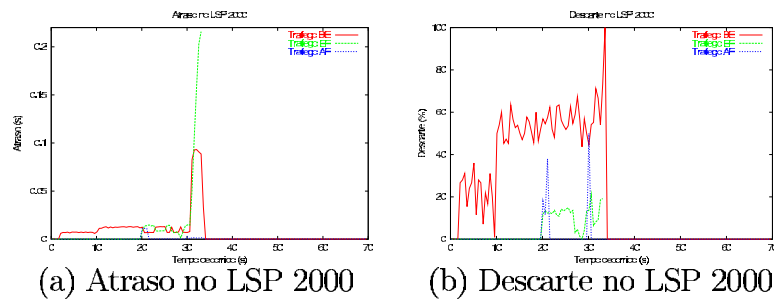


Figura 5: Gráficos do LSP 2000

de descarte e de atraso para as classes AF e EF. O BB-A é notificado sobre excesso de descarte dos tráfegos AF e EF, respectivamente, nos instantes $t = 30,1$ s e $t = 30,6$ s (Figura 5-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 5-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.

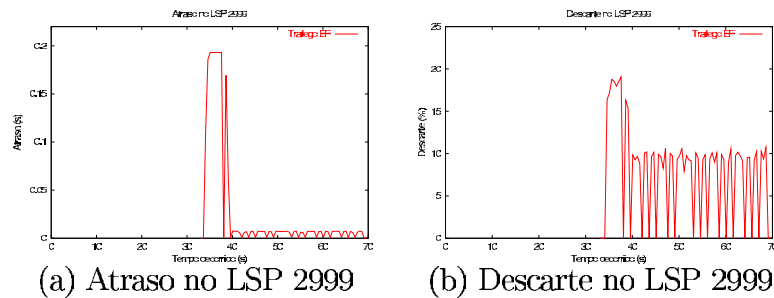


Figura 6: Gráficos do LSP 2999

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 7). 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.

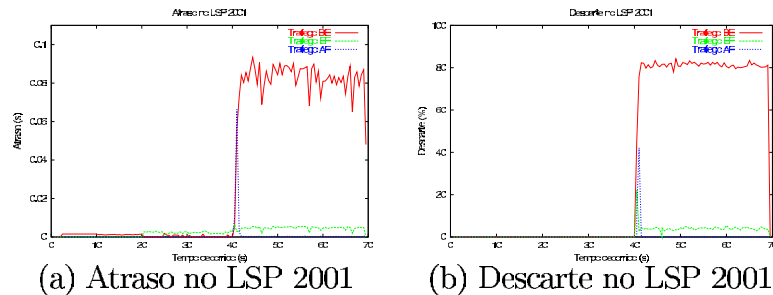


Figura 7: Gráficos do LSP 2001

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 8-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.

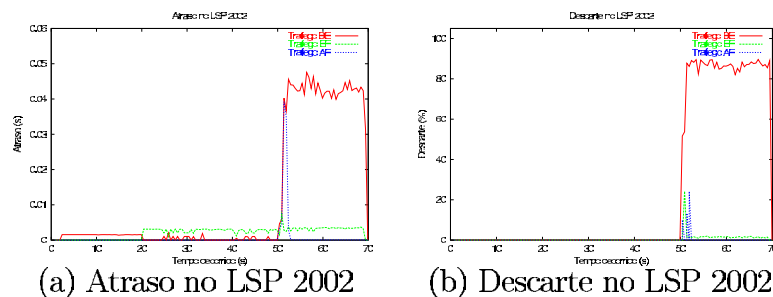


Figura 8: Gráficos do LSP 2002

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

Nos LSP's 2001 e 2002, observamos a existência de tráfego da classe BE mesmo antes da inicialização do tráfego dos clientes locais (Figuras 7-a e 8-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 políciadores é remarcar o tráfego em excesso para o código referente à classe BE, para que não recebam tratamento diferenciado.

Os gráficos da Figura 9 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.

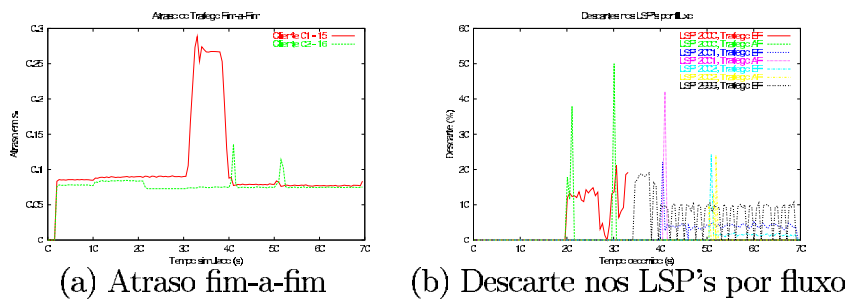


Figura 9: Gráficos de fim-a-fim

7 Trabalhos Relacionados

Muitos estudos têm sido realizados no sentido de garantir QoS à Internet através do uso dos modelos DiffServ, MPLS e, em alguns casos, IntServ.

Alguns trabalhos sugerem o uso conjunto de tecnologias de QoS. O trabalho [8] propõe a obtenção de QoS na Internet através do uso conjunto e intercalado de DiffServ e IntServ. Em [19], encontramos a proposta de um modelo que realiza a administração transparente de domínios, independente das arquiteturas utilizadas. O trabalho [13] pioneiramente apresenta uma solução para a integração dos modelos MPLS e DiffServ.

Com relação ao gerenciamento de domínios, o trabalho [6] apresenta uma implementação de BB baseado em políticas de configuração para o modelo DiffServ. Finalmente, a preocupação com a obtenção de qualidade de serviço fim-a-fim também vem sendo demonstrada. A referência [3] mostra a especificação do projeto QBone/Internet2 de um modelo de comunicação entre

BB's, 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 a questão obtenção de QoS fim-a-fim com grande riqueza de detalhes: os projetos TEQUILA [20], AQUILA [2] e CADENUS [5]. Em todos estes projetos, o objetivo fundamental é a obtenção de QoS fim-a-fim na Internet.

8 Conclusões

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.

Do ponto de vista gerencial, o emprego de um *software* baseado em políticas de configuração confirmou as expectativas de maior simplicidade e controle sobre tarefas administrativas.

A utilização de CORBA para a comunicação entre BB e rede simulada possibilita a exportação deste *software* para um ambiente de rede real, onde as políticas implementadas poderiam ser reaproveitadas.

Existem alguns problemas interessantes que certamente merecem ser atacados em termos de trabalhos futuros. Um destes problemas 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 [9] e o QPIM [17]. Além disso, em termos de armazenamento de informações, a solução mais adequada parece ser o uso de servidores LDAP. Por último, não existem atualmente estudos conhecidos sobre a questão do gerenciamento de contratos, especialmente para casos em que estes são numerosos. Estas seriam continuações naturais do trabalho aqui apresentado.

Referências Bibliográficas

- [1] G. Ahn and W. Chun. Overview of MPLS Network Simulator: Design and Implementation. <http://flower.ce.cnu.ac.kr/fog1/mns/index.htm>, 1999.
- [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, and 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] L. Costa. Desenvolvimento de um Bandwidth Broker para a arquitetura Diff-Serv. Dissertação de mestrado, Unicamp, Março 2001.
- [7] V. L. Costa. Algoritmos de Roteamento. Dissertação de mestrado, Unicamp, Abril 2002.
- [8] A. Abelém et al. QoS Fim a Fim Através da Combinação entre Serviços Integrados e Serviços Diferenciados. *SBRC 2001*, Maio 2001.
- [9] B. Moore et al. Policy Core Information Model - Version 1 Specification. Internet RFC 3060, IETF, Fevereiro 2001.
- [10] D. Awduche et al. Requirements for Traffic Engineering over MPLS. Internet draft, IETF, draft-ietf-mpls-traffic-eng.00.txt, Outubro 1998.
- [11] E. Crawley et al. A Framework for QoS-based Routing in the Internet. Internet RFC 2386, IETF, Agosto 1998.
- [12] E. Rosen et al. Multiprotocol Label Switching Architecture. Internet RFC 3031, IETF, Janeiro 2001.
- [13] F. Faucheur et al. MPLS Support of Differentiated Services. Internet draft, IETF, draft-ietf-mpls-diff-ext-09.txt, Abril 2001.
- [14] P. Piedad et al. A Network Simulator Differentiated Services Implementation. Openip, Nortel Networks, draft-tequila-sls-00.txt, Julho 2000.
- [15] S. Blake et al. An Architecture for Differentiated Services. Internet RFC 2475, IETF, Dezembro 1998.
- [16] Y. Bernet et al. A Framework for Differentiated Services. Internet draft, IETF, draft-ietf-diffserv-framework-00.txt, Maio 1998.
- [17] Y. Snir et al. Policy Framework QoS Information Model. Internet draft, IETF, draft-ietf-policy-qos-info-model-01.txt, Abril 2000.
- [18] A. M. S. Guillén. MPLS-DS: Uma plataforma para validação de políticas no contexto das redes MPLS/DiffServ. Dissertação de mestrado, Unicamp, Dezembro 2001.
- [19] C. Kamiński and D. Sadok. Chameleon: uma Arquitetura para Serviços Avançados Fim a Fim na Internet com QoS. *SBRC 2001*, Maio 2001.
- [20] TEQUILA Project. <http://www.ist-tequila.org>. 2000.