Desenvolvimento de um Bandwidth Broker para a arquitetura de Serviços Diferenciados

Leonardo Costa

Dissertação de Mestrado

Instituto de Computação
Universidade Estadual de Campinas

Desenvolvimento de um Bandwidth Broker para a arquitetura de Serviços Diferenciados

Leonardo Costa

Fevereiro de 2001

Banca Examinadora:

- Prof. Dr. Edmundo Roberto Mauro Madeira (IC/UNICAMP) (Orientador)
- Prof. Dr. Maurício Ferreira Magalhães (DCA/FEEC/UNICAMP)
- Prof. Dr. Nélson Luis Saldanha da Fonseca (IC/UNICAMP)

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP

Costa, Leonardo

C823d Desenvolvimento de um Bandwidth Broker para a arquitetura de Serviços Diferenciados / Leonardo Costa -- Campinas, [S.P. :s.n.], 2001.

Orientador : Edmundo Roberto Mauro Madeira

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

1. Redes de computadores. I. Madeira, Edmundo Roberto Mauro. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

TERMO DE APROVAÇÃO

Tese defendida e aprovada em 02 de março de 2001, pela Banca Examinadora composta pelos Professores Doutores:

Prof. Dr. Maurício Ferreira Magalhães
FEEC – UNICAMP

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

Prof. Dr. Edmundo Roberto Mauro Madeira
IC - UNICAMP

Resumo

O controle da qualidade de serviço em diferentes níveis de granularidade permite idealizar uma série de novas aplicações com exigências críticas de desempenho. Este trabalho explora este novo cenário, apresentando a base desta discussão e propondo um modelo de desenvolvimento. Um protótipo foi implementado para validar estas novas estratégias de controle da qualidade de serviço.

A evolução das estratégias de diferenciação de serviços é apresentada, como referência da discussão da oferta de controle da qualidade de serviço através de políticas de alto-nível. Este trabalho se concentra na proposta de um modelo de camadas de atuação, discutindo as funcionalidades de cada camada e a interação necessária para oferecer o gerenciamento de recursos.

O conceito de Bandwidth Broker foi apresentado como solução do problema de gerência em diferentes níveis de atuação. Apresentou-se uma modelagem deste tipo de gerente, através da proposta de um modelo de desenvolvimento e da identificação das funcionalidades dos componentes do modelo. Diversos casos de uso discutem estas funcionalidades, identificando os participantes no processo de gerência, desde os níveis mais altos de administração até o nível de elemento de rede.

A implementação do protótipo foi realizada inserindo modificações no simulador de redes NS, que permitiram um agente externo distribuir informações de configuração e atuar no ajuste dos mecanismos de encaminhamento. Os experimentos simularam tráfegos com diferentes requisitos de qualidade de serviço, através de cenários que mostram o impacto positivo do controle em diferentes níveis de atuação, através da intervenção do Bandwidth Broker.

Através da definição de políticas simples, demonstrou-se uma estratégia na qual o Bandwidth Broker atua para estabilizar requisições de recursos que competem entre si. Ainda, o impacto que diferentes tipos de tráfego da rede inserem na utilização de recursos de rede foi discutido, assim como o Bandwidth Broker pode resolver este problema.

Abstract

The quality of service control on different granularity levels allows to perform of a set of new applications with performance demands. This work explores this new scenario, presenting the basis of this discussion and proposing a development model. A prototype has been implemented, to validate these new quality of service control strategies.

The evolution of service differentiation strategies is presented, as a reference for the discussion of quality of service control through high level policies. This work concentrates on the proposal of a provisioning tier model, discussing each level functionalities and their interactions to offer the resource management capability.

The Bandwidth Broker concept was presented as a solution to the management problem at different levels of provisioning. A model of this kind of manager was presented, through the proposal of a development model and the identification of the funcionalities of the model's components. Several use cases discuss these funcionalities, identifing the players on the management process, from highest levels of administration down to the network element.

The prototype implementation was done through modifications of the network simulator (NS), which allowed an external agent to distribute configuration information and to perform forwarding mechanism adjustments. The experiments simulated traffic with different QoS requirements, through scenarios that showed the positive impact of control at different levels of provisioning, through the Bandwidth Broker intervation.

Through the definition of simple policies, it was demonstrated a strategy that the Bandwidth Broker acts to stabilize competing resource requirements. Furthermore, we discussed the impact that different types of network traffic cause on the utilization of network resources and how the Bandwidth Broker can solve this problem.

Agradecimentos

- Gostaria de agradecer ao meu orientador, Edmundo Madeira, a quem considero um amigo, por sua excelente orientação e disponibilidade de tempo, além da serenidade com que lidou na abordagem dos problemas para desenvolver este trabalho. Ainda, gostaria de lhe agradecer a liberdade de criação que me foi permitida.

- Agradeço também a contribuição e apoio oferecidos pelo professor Maurício Magalhães durante toda a elaboração deste trabalho. Suas sugestões e críticas foram fundamentais em todo o desenvolvimento. A integração que me foi oferecida com seu grupo de pesquisa influenciaram grande parte das decisões de implementação, além de garantir o suporte logístico que permitiu minha tranquilidade na conclusão deste trabalho.

- Agradeço carinhosamente à minha família, por seu apoio incondicional durante todo esse período. A conclusão deste trabalho é minha retribuição para todo o carinho recebido e por terem acreditado no meu sucesso.

- Agradeço aos diversos amigos que fiz no Instituto de Computação, pela convivência em um ambiente apto à criação. Especialmente, aos amigos Paulo, Sandro, Marcelo, Ivan, Bruno e Nathan pelo companheirismo e pela descontração.

- Agradeço ao grupo de trabalho na Faculdade de Engenharia Elétrica e Computação, pela excelente convivência. Agradeço especialmente ao Vinicius, pelo suporte no desenvolvimento das simulações que validam este trabalho. Ainda, ao meu amigo Tulius, pela amizade e companheirismo de vários anos.

- A Deus, que me deu proteção e continua olhando por mim.

Sumário

Lista de Figuras

Capítulo 1

Introdução

O desenvolvimento tecnológico experimentado pela Internet e a diversidade de aplicações que a utilizam como infra-estrutura de comunicação têm despertado o interesse por novas formas de gerenciamento do uso dos recursos de rede. A motivação está principalmente em viabilizar que aplicações com exigências críticas de desempenho possam ser atendidas. A discussão envolve a definição de novas classes de serviço além do melhor-esforço tradicional, sendo que estas novas classes representam a prioridade pretendida por um ou mais fluxos de dados.

O tratamento diferenciado de tráfego nas redes de computadores permite oferecer às aplicações maiores garantias de previsibilidade no uso da infra-estrutura de rede. Sua implementação se dá através da utilização de mecanismos de controle do tráfego ao longo do caminho de transmissão, influenciando diretamente no processo de encaminhamento dos pacotes. Tradicionalmente, o comportamento de tais mecanismos é determinado através de configuração realizada manualmente sobre determinados parâmetros de controle.

A automatização dessa tarefa possibilita a idealização de um **sistema de gerenciamento** de um domínio administrativo. A visão lógica deste tipo de sistema é centralizada e permite traçar estratégias de otimização na utilização dos recursos disponíveis. O projeto e desenvolvimento deste tipo de gerenciamento é o foco deste trabalho. A administração dos recursos de rede implica na aplicação de uma série de políticas que controlam e/ou limitam sua utilização. Ainda, a avaliação e controle do impacto do tráfego na infra-estrutura de rede é determinante em qualquer abordagem de oferecimento de qualidade de serviço.

A classe de serviço amplamente utilizada nos dias de hoje é a de **melhor-esforço**. Esta classe não oferece nenhum tipo de garantia ou previsibilidade no uso da rede, consistindo na transmissão dos pacotes na medida em que haja disponibilidade do meio de transmissão. A grande maioria das aplicações é atendida satisfatoriamente por esse tipo de classe, pois não possue exigências de tempo-real (correio eletrônico, newsgroups, ftp)

ou, pelo menos, é pouco prejudicada em situações de congestionamento.

Algumas soluções têm sido discutidas com o objetivo de explorar a priorização de tráfego. Estas soluções são direcionadas principalmente para a capacidade de **selecionar** tráfego sendo transmitido e oferecer um tratamento diferenciado através de filas de prioridade [1, 2]. Além disso, a atual estrutura tem de ser mantida, sendo que as aplicações usuárias de melhor-esforço não devem ser prejudicadas. Outro ponto importante está relacionado ao fato que a **diferenciação de serviços** tende a implicar também em tarifação diferenciada, oferecendo novas formas de exploração da infra-estrutura de rede.

A funcionalidade exercida por um *Bandwidth Broker* consiste em oferecer ferramentas de suporte para este tipo de gerenciamento, efetuando o controle de admissão e da configuração dos dispositivos de rede considerando políticas definidas de utilização dos recursos. Estas políticas refletem as decisões corporativas, podendo ser até mesmo parte do modelo de negócios de uma empresa.

Este trabalho apresenta as questões envolvidas na implementação de classes de serviço além do melhor-esforço e contribui na compreensão dos requisitos necessários para o desenvolvimento de uma arquitetura com esta funcionalidade. O objetivo principal está na discussão e no desenvolvimento de um modelo de Bandwidth Broker, responsável pela coordenação e controle da infra-estrutura de rede. Ainda, nosso modelo considera a utilização do uso de políticas para acesso aos recursos de rede. Exploramos características de distribuição em nossa proposta, avaliando ainda as vantagens de se utilizar a arquitetura distribuída CORBA na implementação do modelo.

O desenvolvimento de um protótipo foi realizado para validar este cenário de controle do Bandwidth Broker. Os experimentos realizados obtiveram resultados que comprovam o ganho de desempenho na priorização de tráfegos com diferentes requisitos de transmissão. Foram apresentados cenários que mostram a aplicação gradativa do oferecimento de qualidade de serviço, justificando a necessidade do controle dos mecanismos para obtenção de um comportamento compatível com as políticas definidas. Os experimentos também apresentam uma avaliação do impacto que o Bandwidth Broker pode exercer ao gerenciar, além do controle baseado em políticas, informações sobre o tipo de tráfego de diferentes fontes.

O Capítulo 2 apresenta as abordagens de oferecimento de qualidade de serviço atualmente sendo discutidas. A apresentação dos mecanismos tradicionais é realizada, mostrando a evolução até as estratégias mais atuais de arquiteturas de qualidade de serviço. O Capítulo 3 apresenta o conceito de Bandwidth Broker como gerenciador de recursos de rede, discutindo suas funcionalidades básicas na aplicação do controle da qualidade de serviço. O Capítulo 4 apresenta uma proposta de modelagem para um Bandwidth Broker, discutindo os componentes básicos que este tipo de gerenciamento tem de possuir. Ainda, a modelagem é apresentada em alto-nível através de casos de uso, identificando os atores

que participam do controle e seu relacionamento. O Capítulo 5 discute a implementação de um protótipo para validar o modelo apresentado, realizada através de simulações com diferentes tipos de tráfego. O Capítulo 6 apresenta a conclusão sobre o trabalho realizado e apresenta propostas de continuidade deste trabalho.

Capítulo 2

Abordagens de QoS: Melhor-Esforço e Estratégias Atuais

O serviço de melhor-esforço impõe uma série de dificuldades na manutenção da qualidade do serviço de transmissão do tráfego. Diferentes aplicações possuem diferentes exigências de transmissão, sendo bastante comum que pacotes de fluxos de aplicações críticas estejam intercalados com aplicações de menor exigência de transmissão. O que se observa é que o serviço de melhor-esforço não oferece nenhum tipo de otimização no uso dos recursos de rede.

O surgimento de novas aplicações tem implicado no aumento das exigências de transmissão, principalmente por serviços de tempo-real de conteúdo multimídia. Novos serviços interativos, por exemplo telefonia sobre IP, exigem um comportamento mais previsível da infra-estrutura de rede e baixos atrasos de transmissão. Essa discussão converge, portanto, para um gerenciamento inteligente dos recursos de rede em um domínio administrativo.

O oferecimento de serviços com diferentes níveis de qualidade tradicionalmente está relacionado à tarifação, relativo ao tipo de cliente (usuários domésticos, empresas, corporações, etc) e a qualidade do tipo de conexão (acesso discado, linhas dedicadas, etc). O foco de oferecimento de qualidade de serviço nesse trabalho será relacionado à capacidade de gerenciamento da transmissão dos pacotes de dados. Esta é uma abordagem importante de qualidade de serviço.

O aumento de tráfego nas redes de computadores, notadamente na Internet, rapidamente utiliza os recursos computacionais, comprometendo o desempenho. A capacidade de priorizar determinados tipos de tráfego oferece um importante suporte para usuários e aplicações, sendo um conceito importante quando se depende de um comportamento mais previsível da infra-estrutura de rede.

Diversas visões da infra-estrutura de rede podem ser realizadas, de acordo com a conveniência: visão de negócios, visão corporativa, estrutura de acesso ou de *backbone* de

suporte. Entretanto, o serviço de entrega fim-a-fim de um pacote se reduz ao problema de encaminhamento roteador-a-roteador, da origem ao destino.

Ainda, pode ser observado que o **melhor** caminho de transmissão não é necessariamente o **menor** caminho. Um caminho com um maior número de *hops* pode estar apto a garantir os requisitos de transmissão de uma determinada aplicação, enquanto o menor caminho pode estar congestionado ou não ter recursos suficientes.

Portanto, o fator determinante do sucesso de um ambiente habilitado para qualidade de serviço é definir o comportamento dos roteadores ao longo do caminho de transmissão do par origem/destino. Algumas soluções têm sido utilizadas (individualmente ou em conjunto) para implementar esse conceito:

Precedência de pacotes: a identificação de pacotes é determinante no sucesso de qualquer abordagem de priorização de tráfego. Através de um ou mais campos do cabeçalho IP, por exemplo, os pacotes podem ser direcionados para filas com diferentes prioridades de transmissão;

Adequação de tráfego (*traffic shaping*): A adequação de tráfego busca garantir que a taxa de um ou mais fluxos mantenha-se dentro de parâmetros pré-estabelecidos, impondo um comportamento previsível;

Controle de admissão: a inserção de tráfego na infra-estrutura influi diretamente na sobrecarga dos recursos de rede. Este tipo de controle é importante na solução de problemas de congestionamento;

Gerência de congestionamento: o tratamento diferenciado em situações de congestionamento constitui uma garantia importante para aplicações críticas, garantindo a transmissão de determinadas classes de tráfego com maior prioridade. Este é um conceito fundamental na abordagem de soluções de engenharia de tráfego;

Engenharia de Tráfego: a engenharia de tráfego propõe o controle do caminho a ser utilizado para a transmissão de um pacote. Através da determinação dos possíveis caminhos entre um par origem/destino, uma seleção pode ser realizada que atenda os requisitos de transmissão. Exemplos desta estratégia são novos algoritmos de roteamento baseados em critério de QoS e de roteamento explicíto, além da tecnologia MPLS (*MultiProtocol Label Switching*) (Seção 2.4);

A utilização de classes de serviço tem por objetivo, através do tratamento diferenciado do tráfego, garantir os valores dos parâmetros exigidos como estimadores de QoS pelas aplicações. A definição dos serviços deve estabelecer a prioridade de cada classe, esclarecer o tipo de serviço implementado e os mecanismos necessários. A tarefa de diferenciação

também depende da identificação da classe a qual cada pacote pertence. Uma vez identificado, o pacote pode ser direcionado para filas nos roteadores, que enviam os pacotes seguindo uma política de prioridade.

A capacidade de priorização de tráfego é o mecanismo que permite, no nível físico, implementar qualquer estratégia de qualidade de serviço além do melhor-esforço. Esta estratégia é obtida através do gerenciamento das filas nos roteadores. Diversos algoritmos de gerenciamento têm sido propostos para otimização da ocupação das filas. Entretanto, a configuração ótima é um objetivo difícil de ser alcançado, devido à grande heterogeneidade de enlaces, aos protocolos que interagem nesses links e à diversidade de diferentes aplicações e de níveis de congestionamento que causam nessas enlaces. Este é notadamente o cenário percebido na Internet [3].

Diversos mecanismos são utilizados para diferenciação de serviços. Algumas arquiteturas de QoS especificam o comportamento dos roteadores, atribuindo diferentes tarefas de tratamento dos fluxos de dados, principalmente no que se refere ao encaminhamento de pacotes. Cada roteador possui um **comportamento**, denominado *per-hop behaviour*, que é a ação aplicada em um fluxo ou em um agregrado de fluxos de dados. Estes mecanismos e o papel de cada elemento de rede são a essência da diferenciação de serviços e serão apresentados nas seções seguintes. Os critérios variam conforme a arquitetura adotada.

A Seção 2.1 apresenta alguns algoritmos que têm sido utilizados como estratégia de diferenciação de serviços. A Seção 2.2 descreve a arquitetura de Serviços Integrados, que tem sido bastante discutida através do grupo de trabalho Integrated Services, do IETF (*Internet Enginnering Task Force*) [4]. A Seção 2.3 apresenta a arquitetura de Serviços Diferenciados, também alvo de discussões do grupo de trabalho Differentiated Services, do IETF.

2.1 Estratégias de gerência de filas

O enfileiramento de pacotes consiste na armazenagem de pacotes nos roteadores, para que possam ser processados e finalmente encaminhados para o próximo roteador, até seu destino. Basicamente, um roteador possui um fila de entrada e uma fila de saída, sendo que a ação de enfileiramento pode ocorrer em ambas as filas, ou separadamente. Esta ação é determinante no funcionamento dos protocolos de roteamento, e ainda mais determinante no oferecimento de qualidade de serviço.

A gerência das filas é, portanto, a estratégia adotada para a transmissão de dados fim-a-fim, com ou sem qualidade de serviço. A técnica de **armazenagem-encaminhamento** reflete o tratamento aplicado aos pacotes em nível físico. Entretanto, diversos algoritmos podem ser aplicados em diferentes fases do enfileiramento, viabilizando o controle sobre características do tráfego (valor de *jitter*, taxa de descarte, etc). Este controle é a base

da diferenciação de serviços.

Os algoritmos tradicionalmente utilizados no processo de enfileiramento compreendem:

Algoritmo FIFO (*First-In, First-Out*)

Este é o algoritmo padrão de gerenciamento de filas, uma abordagem *store-and-forward* simples. Este método pode ser considerado o que oferece o maior troughtput em pacotes por segundos, principalmente ao fato de ser de simples implementação, podendo tirar grande proveito de otimizações de hardware. A desvantagem deste algoritmo ocorre em situações de congestionamento, quando não é possível oferecer prioridade para aplicações críticas.

Algoritmo de prioridade de enfileiramento (*Priority Queuing*)

A priorização simples de pacotes surgiu como a primeira variação de enfileiramento da estratégia FIFO. Consiste em selecionar antecipadamente os pacotes que devem ser transmitidos. Sua desvantagem é que a necessidade de se reordenar a fila de pacotes degrada o desempenho, aumentando o esforço computacional.

Class-Based Queuing (CBQ)

O algoritmo CBQ é uma variação do algoritmo de prioridade simples. Nesta variação, diversas filas de saída são definidas, com diferentes níveis de prioridade para emissão dos pacotes que elas contêm. A vazão destas filas é determinada em bytes por ciclo, sendo que cada ciclo retira, no máximo, a quantidade de bytes configurada para cada fila. Desta maneira, pode-se configurar diferentes níveis de prioridade, sem no entanto monopolizar os recursos do roteador para uma determinada classe de tráfego.

A configuração deste algoritmo não é geralmente determinística, sendo que o ajuste ideal dos parâmetros normalmente é encontrado empiricamente. Uma vantagem desta técnica é justamente resolver o problema de *resource starvation*, uma vez que a abordagem não consiste mais em um fila de prioridade absoluta, mas sim de diferentes ordens de precedência. Entretanto, o esforço computacional necessário para reordenação de pacotes e para o escalonamento das filas não lhe fornece escalabilidade, limitando sua utilidade em *links* de maior velocidade.

Weighted Fair Queuing (WFQ)

A estratégia adotada pelo algoritmo WFQ consiste em priorizar fluxos que possuam baixos volumes de tráfego e permitir que fluxos com grande volume de tráfego compartilhem os recursos restantes. A técnica consiste em indexar e intercalar pacotes por fluxo,

enfileirando-os de acordo com o volume de tráfego que possuem, evitando a ocorrência de *resource starvation.*

Entretanto, este algoritmo possui as mesmas deficiências dos algoritmos de prioridade e baseado em classes, ou seja, pouca escalabilidade, principalmente devido ao grande esforço computacional necessário para implementá-lo.

RED (*Random Early Detection*)

O algoritmo RED foi documentado por Sally Floyd e Van Jacobson em 1993 [5]. O objetivo é prover um mecanismo que permita evitar situações de congestionamento, através do descarte de pacotes de acordo com uma dada probabilidade, sendo que a escolha dos fluxos para este descarte é aleatória. Através desta estratégia, busca-se evitar a situação na qual todos os fluxos TCP experimentam o congestionamento ao mesmo tempo, causando uma sincronização global de ajuste das janelas de transmissão. Este algoritmo permite configurar o gatilho e o nível de descarte. Quanto maior o congestionamento, maior a quantidade de pacotes descartados e a quantidade de fluxos selecionados.

RIO (*Random with In-and-Out*)

A estratégia adotada pelo algoritmo RIO é diferenciar o tipo de tráfego sendo escolhido para o descarte. O mecanismo de escolha é o mesmo do algoritmo RED. Entretanto, considera-se que os pacotes foram previamente verificados quanto a sua adequação às taxas de tráfego permitidas. Os pacotes identificados como excedentes são descartados com maior agressividade. Isto possibilita amenizar o descarte de um fluxo, privilegiando os fluxos que mantêm sua taxa de transmissão de acordo com o permitido.

2.2 Arquitetura de Serviços Integrados (IntServ)

A arquitetura de Serviços Integrados [6] estabelece o tratamento diferenciado de fluxos de tráfego a partir de cada nó da rede pertencente ao caminho de transmissão. Através da manutenção do estado dos fluxos que atravessam o nó, este pode realizar operações tais como:

- **controle de admissão**: o controle de admissão permite ao nó da rede avaliar a possibilidade de oferecer o serviço requisitado pelo fluxo. Quando isto não é possível, os pacotes devem ser enviados como tráfego melhor-esforço;
- **classificação**: esta função avalia os pacotes que chegam por determinada interface e, baseando-se na identificação deste pacote, direciona-o para uma fila de transmissão de acordo com sua classe de serviço;

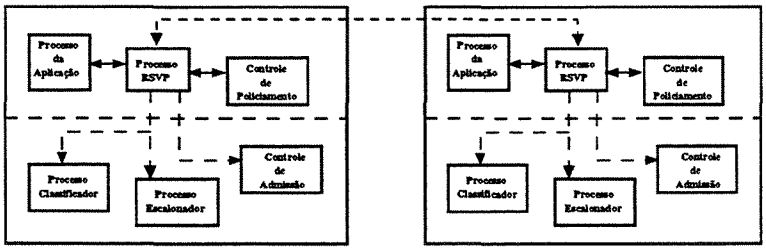

Figura 2.1: Arquitetura de Serviços Integrados utilizando RSVP

- **escalonamento**: a função de escalonamento implementa as prioridades das filas de transmissão. Quanto maior a prioridade das filas, maior a vazão experimentada pelos pacotes ali presentes;

- **reserva de recursos**: a alocação de recursos nos nós da rede é um mecanismo fundamental nesta arquitetura. Através de um protocolo de sinalização, o RSVP (Resource reSerVation Protocol) [7, 8], a aplicação determina quais os valores dos parâmetros de qualidade de serviço necessários para atender as expectativas de sua transmissão;

Além do melhor-esforço, a arquitetura de Serviços Integrados define as seguintes classes de serviço:

- **serviço garantido**: A classe de serviço garantido destina-se a aplicações com requisitos explícitos de transmissão, tais como taxas de atraso baixas, perda de pacotes e *jitter*;

- **serviço de carga-controlada**: Esta classe de serviço tem a função de suportar aplicações menos exigentes em relação ao tratamento de seu fluxo de dados. Entretanto, o oferecimento desta classe de serviço é estabelecido como sendo "melhor que o melhor-esforço", ou seja, o oferecimento de um ambiente no qual a rede está com pouca carga de transmissão e sem congestionamentos;

Os mecanismos desta arquitetura têm de ser implementados em cada nó pertencente ao caminho percorrido pelo fluxo. A manutenção do estado de cada fluxo (identificação do fluxo, parâmetros de tráfego, etc) é necessária para efetuar o controle de admissão. A Figura 2.1 mostra o modelo de funcionamento dos Serviços Integrados, utilizando como protocolo de sinalização o RSVP.

A sinalização é realizada entre os processos RSVP, sendo que algumas informações operacionais são fornecidas pelo controle de policiamento. O processo de classificação identifica os fluxos já admitidos para serem controlados pelo roteador. O controle de

admissão avalia as requisições contidas nas mensagens de reserva. O processo escalonador controla a emissão de pacotes nas interfaces de rede, implementando a diferenciação de serviços no nível físico.

Esta arquitetura determina que a implementação de diversos mecanismos de gerenciamento do tráfego tem de existir em todos os roteadores. Considerando o crescimento do tráfego (uso da Internet, por exemplo), certamente existe um momento em que o gerenciamento das informações (estados) de todos os fluxos e a utilização dos mecanismos supera a capacidade computacional dos roteadores. Portanto, esta arquitetura possui como desvantagem a pouca escalabilidade, principalmente pelo fato de necessitar ser implementada em todos os roteadores.

2.3 Arquitetura de Serviços Diferenciados (DiffServ)

A escalabilidade é o foco principal na Arquitetura de Serviços Diferenciados [1]. Através da agregação de diversos fluxos em classes de serviço, torna-se possível minimizar o esforço computacional da diferenciação de serviços. Nesta arquitetura, um **serviço** é compreendido pelas características de transmissão aplicadas aos pacotes em uma determinada direção, através de um ou vários caminhos de uma rede.

A implementação é realizada através de blocos funcionais, que especificam o tratamento a ser aplicado no tráfego e percebido externamente pelas aplicações. Estes blocos funcionais constituem a **definição do comportamento de encaminhamento**, sendo denominados na arquitetura DiffServ, como PHBs (*Per-Hop Behaviour*). Nestes blocos funcionais estão presentes os mecanismos de classificação e condicionamento de pacotes, tais como medição (auditoria), marcação, adequação e policiamento.

A Arquitetura de Serviços Diferenciados determina como estratégia de oferecimento de qualidade de serviço [2]:

- simplificar o processo de identificação e encaminhamento de pacotes;
- isolar, o máximo possível, a complexidade de identificação e adequação de tráfego. Esta tarefa, de grande esforço computacional, deve ser mantida nos elementos de rede (roteadores) presentes na fronteira do domínio administrativo;
- implementar políticas de utilização dos recursos que permitam o gerenciamento de requisições de longa e curta duração na geração de tráfego;
- manter a compatibilidade com o modelo clássico da Internet que permanece dependente do serviço de melhor-esforço.

A simplificação do processo de identificação utiliza uma abordagem oposta à classificação por múltiplos campos adotada na arquitetura de Serviços Integrados. O processo adotado considera somente o campo de qualidade de serviço TOS (*Type of Service*) [9] do cabeçalho IP. A arquitetura de Serviços Diferenciados define novas codificações para este campo, denominando-o como **campo de serviços diferenciados**, ou **campo DS**. A codificação presente neste campo indica a classe de serviço a qual o pacote pertence e qual o tratamento a que deve ser submetido.

O roteador tem papel determinante na diferenciação de serviços. De acordo com sua localização, pode assumir a seguinte classificação:

- **Roteador de Borda (ou fronteira)**: este roteador está posicionado na fronteira de um domínio, ou seja, faz ligação com outros roteadores que não pertecem ao mesmo domínio administrativo. Dependendo da direção do fluxo, o roteador de borda pode ser responsável por receber o tráfego que chega em um determinado domínio (roteador de entrada), ou enviar tráfego interno para um domínio vizinho (roteador de saída);

- **Roteador folha**: um roteador-folha está posicionado como sendo o primeiro elemento de rede (*hop*) para uma aplicação. Este roteador recebe o tráfego gerado e aplica os mecanismos necessários para identificar os pacotes de acordo com os critérios estabelecidos no controle de admissão. A fase de admissão implica na configuração deste tipo de roteador com os valores dos parâmetros requeridos pela aplicação. Caso a aplicação envie os pacotes já marcados, o roteador-folha deve garantir que o tráfego gerado está de acordo com o controle de admissão;

- **Roteador interno (de núcleo)**: os roteadores internos são todos aqueles que não são de borda (ou fronteira) nem folhas de uma aplicação.

O processo de classificação depende do tipo de roteador. Os roteadores de borda classificam um pacote através de múltiplos campos do cabeçalho. Os roteadores de núcleo identificam os pacotes somente pelo campo DS. Este campo determina a classe do pacote, sendo representado nos pacotes IP através do campo TOS (*Type of Service*, no protocolo IPv4) ou do campo *Traffic Class* (no protocolo IPv6).

O núcleo de um domínio administrativo se constitui, na sua maioria, de conexões de alta velocidade, interligando os pontos periféricos utilizados pelas diversas sub-redes para inter-comunicação. O esforço computacional dos elementos de rede do núcleo deve, portanto, ser o mínimo possível, otimizando o processo de encaminhamento dos pacotes.

Nenhuma consideração é feita em relação à carga de tráfego sendo introduzida, tarefa a ser realizada pelos nós de rede presentes na fronteira do domínio. Procura-se, dessa maneira, garantir que o encaminhamento entre os pontos interligados pelo núcleo seja o mais

rápido possível, mas que a carga sendo introduzida seja mantida sob controle, evitando congestionamentos que implicariam em perdas de pacotes e consequente instabilidade na QoS percebida fim-a-fim.

Além da simplificação do encaminhamento, nenhuma sinalização é realizada entre os roteadores, além da informação de classe contida no próprio pacote. Todo processamento é realizado pacote-a-pacote. Esta estratégia contribui para a escalabilidade e desempenho do sistema, uma vez que temos granularidade em nível de classes e não de fluxos.

A complexidade de gerenciamento do tráfego não cresce na medida em que o tráfego aumenta. A disponibilidade de ferramentas de gerência e configuração é o fator que viabiliza a disponibilidade destes serviços nos roteadores, atuando de maneira coordenada em todo o domínio administrativo.

2.3.1 Mecanismos de implementação dos Serviços Diferenciados

Diversos elementos fazem parte de um roteador habilitado para Serviços Diferenciados. O modelo conceitual deste tipo de roteador inclui definições de alto-nível dos seguintes elementos:

- elementos de classificação de tráfego;
- funções de medição;
- elementos de condicionamento de tráfego, os quais realizam ações de marcação, descarte, contagem e multiplexação de pacotes;
- elementos de gerência de filas, incluindo capacidade de gerência ativa do processo de enfileiramento;

Estes elementos formam os blocos básicos desta arquitetura e a capacidade de combinação de um sub-conjunto desses elementos (ou de todos) determina o sucesso na obtenção da diferenciação de serviços. Esta combinação cria uma estrutura denominada Bloco de Condicionamento de Tráfego ou TCB (*Traffic Conditioning Block*). Um TCB é uma abstração de um elemento funcional que facilita a definição da funcionalidade de condicionamento de tráfego.

Diferentes implementações de cada um destes mecanismos são possíveis, desde que mantenham a funcionalidade desejada. Apresentamos uma definição da operação de alguns desses mecanismos nesta arquitetura:

Classificação

A classificação é realizada nos roteadores de borda considerando diversos campos do cabeçalho além do campo DS. Este processo se denomina **classificação por múltiplos campos**. Os roteadores internos classificam os fluxos baseando-se somente no valor do campo DS, preenchendo as filas de prioridade de acordo com a classe à qual pertence o pacote. Este processo se denomina **classificação agregada por comportamento** e significa que o tráfego é agregado de acordo com a classe dos pacotes. A escalabilidade nesta estratégia é maior, pois a manutenção de estado para os fluxos somente é necessária nos elementos de borda, diminuindo o esforço computacional dos roteadores de núcleo.

Filas de Prioridade

Os roteadores devem implementar pelo menos dois níveis de prioridade para atender as classes definidas nos serviços diferenciados, além do melhor-esforço. Um nível de prioridade, também conhecido por *Premium Service* [2, 10] (Seção 2.3.2), armazena os pacotes que contenham os bits do campo DS com esta codificação. Este nível possui uma prioridade absoluta no mecanismo de escalonamento. Um outro nível, também conhecido por *Assured Service* [2, 10] possui um valor intermediário de prioridade, sendo atendida logo após a classe anterior ter sido exaurida. Por fim, o tráfego de melhor-esforço pode ser encaminhado de acordo com a disponibilidade de transmissão.

Adequação de fluxo (*Shaping*)

Este mecanismo é utilizado nos roteadores que recebem o tráfego gerado pela aplicação (roteadores-folha). Um pacote ao ser recebido pela interface do roteador é encaminhado, considerando uma taxa máxima máxima de transmissão. O tráfego é mantido dentro dos critérios de policiamento. Todo tráfego excedente da taxa máxima permitida pode ser descartado ou ter diminuída sua prioridade de transmissão. A implementação deste processo possui certo esforço computacional, pois requer mecanismos de temporização, armazenagem de pacotes na memória (*buffering*) e manutenção de estado do fluxo em memória. Os algoritmos *leaky bucket* e *token bucket* geralmente são utilizados nesse mecanismo.

Escalonamento (*Scheduling*)

O mecanismo de escalonamento implementa a ação de transmitir os pacotes de acordo com a prioridade a que pertencem. A capacidade de garantir a alocação de recursos para cada uma das filas traduz a importância deste mecanismo. O escalonamento faz parte dos elementos de gerência de filas. Alguns algoritmos que exercem esta tarefa foram apresentados na Secão 2.1.

A Figura 2.2 esboça a estrutura a ser implementada nos roteadores.

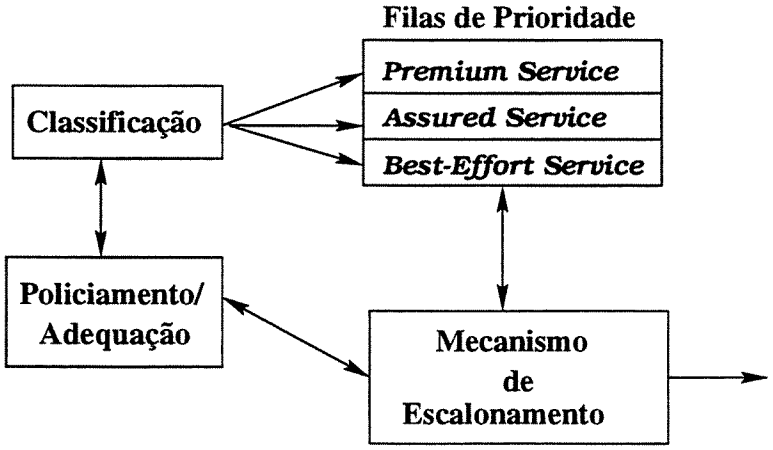

Figura 2.2: Relacionamento dos mecanismos implementando Serviços Diferenciados

2.3.2 Classes de Serviço

A definição das classes de serviço faz parte da tarefa de padronização do grupo de trabalho DiffServ. Atualmente, duas classes de serviço estão publicadas em RFCs (*Request For Comment*). A padronização consiste na definição do PHB que deve ser implementado pelos roteadores que declarem oferecer determinada classe de serviço.

O PHB *Expedited Forwarding*, ou EF [11], define o tratamento reservado a fluxos de dados com requisitos de baixo atraso e *jitter*. A vazão experimentada tem total prioridade e os pacotes desta classe não devem sofrer praticamente nenhum atraso de transmissão. Os pacotes fora dos parâmetros estabelecidos são descartados. Este PHB também é descrito como *Premium Service* [2]. Exemplos de mecanismos de implementação desse PHB são filas de prioridade e utilização do algoritmo CBQ. As filas de prioridade têm de ser controladas por mecanismos de policiamento para evitar esgotamento de recursos para outras classes de serviço.

O PHB *Assured Forwarding*, ou AF [12], define 4 classes de serviço. Ainda, cada classe possui 3 níveis de descarte. A qualidade de serviço percebida por um fluxo depende da quantidade de recursos reservada para cada classe e do congestionamento que a classe sendo utilizada estiver sofrendo. Ainda, para cada pacote, a qualidade de serviço depende do seu nível de precedência. Um roteador que implementa este PHB deve oferecer as 4 classes, sendo que pacotes de cada classe devem ser tratados de maneira independente. Essa política pode ser implementada utilizando-se, por exemplo, o algoritmo RED (*Random Early Detection*) [5];

A transmissão através de melhor-esforço é realizada para todos pacotes cujo campo DS não o classifique para nenhuma classe de serviço sendo considerada.

2.4 MPLS

A tecnologia MPLS (*MultiProtocol Label Switching*) [13] representa uma convergência entre as técnicas de transmissão orientadas à conexão e os protocolos de roteamentos da Internet. A abordagem do MPLS está em simplificar o processamento de pacotes nos roteadores de núcleo, sendo que a escolha do próximo *hop* de um pacote é feita através de uma simples consulta em uma tabela indexada de **etiquetas** ou **rótulos**, evitando todo o processo tradicional de classificação de cabeçalhos e consulta de tabelas de roteamento.

A definição do papel dos roteadores na tecnologia MPLS também é realizada de acordo com o posicionamento dos roteadores no domínio. Um roteador de núcleo é denominado LSR (*Label Switching Router*). O papel de um LSR é realizar a análise da etiqueta de um pacote ingressante, verificar qual interface de saída está mapeada para aquela etiqueta, remarcar o pacote com uma nova etiqueta representando a interface de saída e encaminhar o pacote. Um roteador de borda é denominado um LER (*Label Edge Router*) e sua função é mapear as tabelas de roteamento do domínio não-MPLS para os rótulos configurados em sua tabela de encaminhamento.

A informação presente nas tabelas de rótulos MPLS pode ser distribuída de duas formas principais: através da seleção manual nos LERs e LSRs e através de um protocolo de distribuição de rótulos, que cuida de realizar o mapeamento fim-a-fim entre LERs, através dos LSRs. A seleção de um caminho dentro de um domínio MPLS é denominada um LSP (*Label Switching Path*), que é uma seleção de diversos LSRs do domínio MPLS, no qual é conhecido todo o trajeto a ser seguido por um pacote que o atravesse. A inserção de um pacote no domínio MPLS é realizada nos LERs, utilizando-se de informações contidas em uma estrutura de dados denominada FEC (*Forwarding Equivalence Class*), que identifica o mapeamento entre fluxos de dados e LSPs.

2.5 Utilização de políticas no gerenciamento de recursos

A utilização de políticas permite definir regras que determinam a utilização dos recursos de QoS e sua aplicação, limitando quando, como e onde esses recursos estão disponíveis. Uma política, através de uma ou mais regras, determina a **ação** a ser realizada, dada a ocorrência de uma ou mais condições. Uma regra pode conter outras regras e, portanto, uma política pode ser composta de outras políticas.

Este modelo hierárquico simplifica o gerenciamento, permitindo que políticas complexas sejam criadas a partir de um conjunto de políticas simples. Uma restrição importante é de que o conjunto de condições e ações que compõem uma regra deve ser **verificável** e

não-ambíguo, ou seja, deve haver somente uma regra para um conjunto de determinadas condições.

Um sistema de gerência de tráfego é desejável quando se trata de uma infra-estrutura de rede de dimensões consideráveis, como por exemplo, um backbone. A quantidade de informações de configuração e de elementos de rede a serem freqüentemente atualizadas torna a tarefa de gerência de difícil execução. A estratégia é desenvolver de maneira prática e eficiente, uma solução que traduza políticas corporativas de uma empresa para o comportamento da infra-estrutura de rede, automatizando os processos de configuração dos equipamentos.

Este tipo de sistema determina um **nível estratégico**, no qual a análise e reavaliação do sistema é constante, permitindo os gerentes de informação tomar decisões quanto a infra-estrutura de rede e na elaboração das políticas. O nível estratégico compreende garantir que os novos requisitos de desempenho, por exemplo, suporte a tráfego de tempo-real e baixo atraso fim-a-fim possam ser mapeados para os novos mecanismos de diferenciação de tráfego já discutidos.

Ainda, existe o **nível de atuação**, no qual regras são traduzidas em parâmetros de configuração e depois propagadas para pontos estratégicos da rede. Este nível, portanto, reúne todos os mecanismos de diferenciação e ferramentas de gerência para obter o comportamento desejado de controle do tráfego e deve refletir o contexto corporativo criado a partir do nível estratégico.

Um sistema de classificação e tratamento de tráfego aplicável em múltiplos dispositivos se denomina uma **arquitetura de políticas**. Esta arquitetura define as estratégias e os elementos utilizados para traduzir decisões complexas de gerência em políticas de configuração e para valores dos parâmetros de rede.

Uma arquitetura de políticas tem o objetivo de prover uma infra-estrutura de controle de alto-nível no oferecimento de serviços, possibilitando o controle e monitoramento no uso dos recursos de rede por parte do administrador e dos clientes. A descrição das políticas pode incluir, por exemplo, a identificação do cliente, a identificação da aplicação, um contrato de tráfego, requisitos de banda, variáveis de atraso, *jitter* e taxas de perda, além de considerações sobre segurança e tarifação.

Este contexto sugere um problema de controle, no qual o controle das políticas é mapeado para o comportamento da rede. A tarefa de controle de admissão é uma das mais importantes desta estratégia, tendo em vista que está presente no nível de atuação, permitindo avaliar as requisições dos clientes por recursos na configuração dos parâmetros, e também presente no nível estratégico, realimentando o sistema de controle com informações que irão ser decisivas para os gerentes de informação lidarem com a previsão de demanda da rede.

O modelo de informações apresentado em [14] foi desenvolvido pelo grupo de trabalho

Policy Framework, do IETF, para representação de informações de políticas. Este modelo propõe extensões ao Modelo Comum de Informações [15] desenvolvido pelo *Distributed Management Task Force* [16]. A origem deste trabalho está no esforço realizado para a especificação DEN (*Directory Enabled-Networks*), através do grupo de trabalho DEN Ad-Hoc. A padronização do modelo de políticas é responsabilidade de um grupo de trabalho interno ao DMTF, denominado *SLA Policy Working Group*. O modelo de informações define duas hierarquias de classes:

- **classes estruturais**, que contêm as informações de policiamento e de controle das políticas e,
- **classes associativas**, utilizadas para representar como as classes estruturais estão relacionadas entre si.

As oito classes principais estruturais do modelo compreendem: *Policy*, *PolicyGroup*, *PolicyRule*, *PolicyCondition*, *PolicyAction*, *PolicyTimePeriodCondition*, *vendorPolicyCondition* e *vendorPolicyAction*. Estas são classes gerais, as quais modelos mais específicos de implementação podem herdar. Ainda, são definidas duas classes menos gerais, para suporte a informações específicas de produtos: *vendorPolicyConditionAuxClass* e *vendorPolicyActionAuxClass*. A Figura 2.3 mostra a hierarquia das principais classes deste modelo de informações:

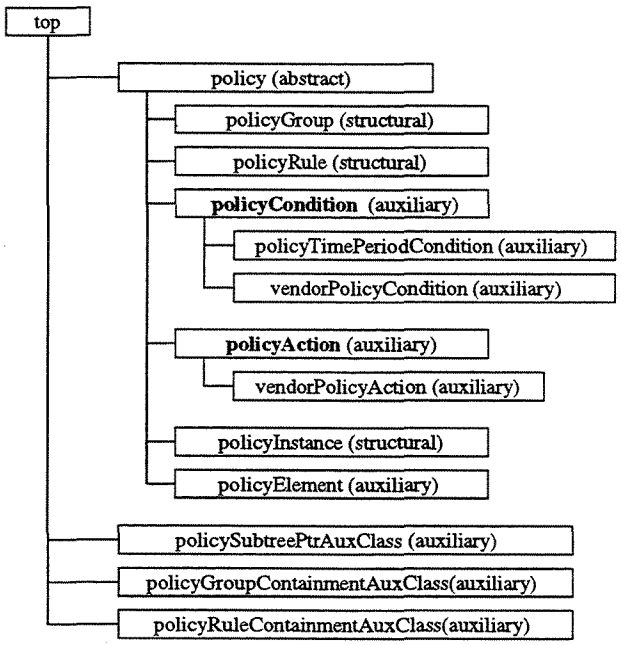

Figura 2.3: Hierarquia de Herança das Classes de Políticas

2.5.1 Camadas de Atuação

Um modelo genérico de uma arquitetura de políticas pode ser classificado em camadas [17], que refletem cada nível de abstração, até a infra-estrutura de rede. Um modelo simplificado consiste de uma camada de tratamento dos pacotes, discutida na Seção 2.1 e Seção 2.3.1, uma camada de decisão e uma outra de gerência de informação. Neste modelo, os componentes responsáveis pelo tratamento de pacotes aplicam regras instaladas pelos componentes de decisão, que por sua vez obtêm informações dos repositórios de dados. As políticas contidas nos repositórios de dados são traduzidas em regras e distribuídas para todos os dispositivos relevantes. A Figura 2.4 apresenta este modelo em camadas.

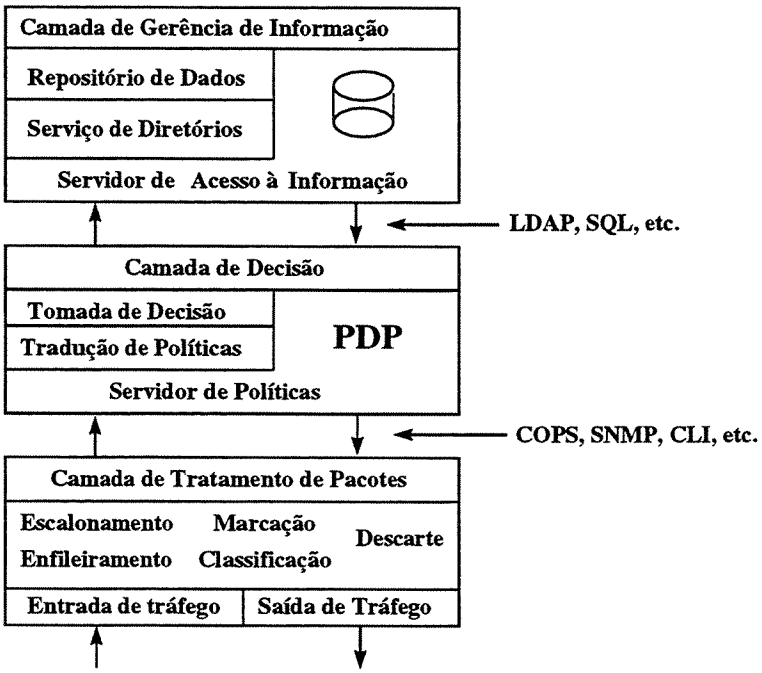

Figura 2.4: Modelo de Sistema de Políticas

Camada de Tratamento de Pacotes

A camada de tratamento de pacotes consiste nos componentes relacionados ao nível físico, os quais lidam diretamente com a classificação, condicionamento e adequação de fluxos de dados na infra-estrutura de rede. As tarefas desempenhadas nesta camada compreendem o enfileiramento e escalonamento de pacotes, descarte e adequação.

Os dispositivos de rede aplicam diversos métodos e algoritmos para realizar estas tarefas, por exemplo, estratégias de gerência ativa das filas para controle de situações de congestionamento. A Seção 2.1 apresentou alguns algoritmos tradicionais utilizados pelos

roteadores nesta camada. As Seções 2.2 e 2.3 apresentaram duas arquiteturas que atuam nesta camada e suas estratégias de operação.

Camada de Decisão

O modelo de gerenciamento e implementação de políticas depende de uma estratégia que permita encaminhar as decisões de configuração para os dispositivos. O protocolo COPS (*Common Open Policy Service*) [18, 19], estabelece uma solução cliente/servidor simples. Este protocolo foi idealizado para oferecer suporte a contextos de sinalização no controle de políticas e provisão de gerência de recursos, sendo seu modelo extensível para futuras definições de clientes deste protocolo.

Duas entidades são definidas neste protocolo. Um PDP (*Policy Decision Point*) é responsável por tomar decisões de gerência de acordo com as políticas definidas, sendo que informações de autenticação e de segurança na comunicação também devem ser consideradas. As decisões são encaminhadas para a segunda entidade, denominada PEP (*Policy Enforcement Point*), que cuida de implementar a configuração necessária para cumprir a decisão tomada. Estas duas entidades são logicamente separadas, apesar de poderem ser implementadas em um mesmo dispositivo.

Diversas instâncias PEP podem ser esperadas em um contexto administrativo, uma vez que a quantidade de dispositivos a serem gerenciados geralmente é grande. Por outro lado, um número limitado de PDPs é esperado, pois a tarefa de decisão possui um caráter mais centralizado. Um motivo para existir mais de uma instância está na tolerância a falhas, apesar de tornar a administração mais complexa.

A comunicação entre estas entidades tem de ser realizada por um mecanismo confiável, por exemplo, uma conexão TCP. O COPS é um protoclo que possui armazenamento de estado, mantendo durante o tempo de vida da comunicação, um histórico das decisões e requisições realizadas. As mensagens de requisição ou de decisão podem ser originadas assincronamente, embora mensagens de confirmação sejam previstas no protocolo.

As funções a serem desempenhadas por uma estratégia de policiamento envolvem **decisão**, **implementação** e **auditoria**. O processo de decisão é realizado pelo PDP, cuja tarefa é interpretar as políticas definidas e processar as requisições de decisão originadas pelos PEPs, verificando sempre a presença de inconsistências. A tarefa de implementação é exercida pelo PEP, que aplica as ações de acordo com as decisões tomadas pelo PDP, podendo ainda considerar informações locais das condições da infra-estrutura de rede (banda disponível, informações temporais, etc). O PEP ainda exerce a auditoria, verificando se as ações implementadas resultaram no estado que satisfaz a(s) política(s) sendo considerada(s).

A arquitetura de Serviços Diferenciados determina o controle de admissão de tráfego somente nos roteadores de borda. Neste contexto, a utilização do protocolo COPS é

proposta como um mecanismo de provisão [20], que possui a capacidade de levar as informações de configuração dos mecanismos de tratamento de tráfego para os roteadores de núcleo, onde estão localizados os PEPs. O servidor de decisões utiliza o repositório de dados nesta tarefa.

Este gerenciamento se assemelha ao gerenciamento com SNMP. A definição de uma PIB (*Policy Information Base*) é proposta em [21], que contém uma estrutura similiar às tabelas de uma MIB (*Management Information Base*), para representação de políticas em sistemas de armazenagem de informação. A arquitetura de policiamento é apresentada na Figura 2.5, relacionando as ferramentas de gerenciamento com os componentes do framework apresentado.

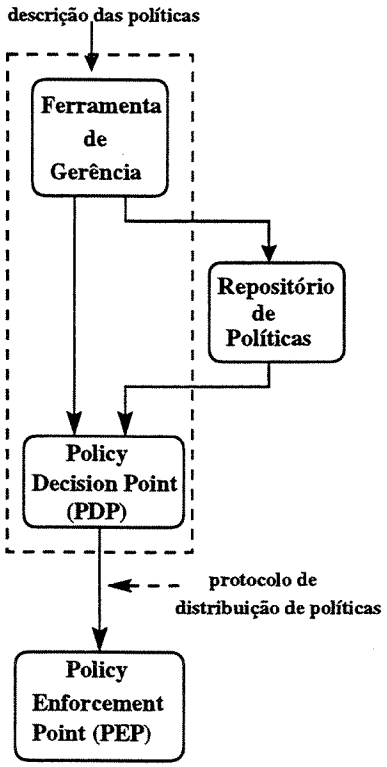

Figura 2.5: Arquitetura de Policiamento

Camada de Gerência de Informação

A utilização dos recursos de um domínio administrativo é determinada por acordos de utilização. O acordos são **contratos** e geralmente denominados de *SLAs* (*Service Level Agreements*). Através deste documento, um provedor e um cliente formalizam os serviços que serão disponibilizados e os detalhes de utilização, tais como a tarifação aplicada e o

tempo de validade. A criação deste tipo de contrato está vinculado a negociações entre agentes humanos, tais como administradores de rede, advogados, clientes, representantes, etc. Algumas informações que podem estar presentes nesse tipo de documento:

- conjunto de PHB aplicáveis no domínio;
- objetos condicionadores de tráfego, objetos de policiamento e os parâmetros de controle;
- políticas administrativas aplicáveis no domínio;
- critérios de auditoria da disponibilização da QoS.

A descrição dos detalhes técnicos relevantes para implementar o SLA está contida em uma especificação denominada *Service Level Specification* ou *SLS*. Esta especificação permite a identificação do tráfego a ser priorizado e a configuração dos dispositivos físicos para implementar esta tarefa. Estão presentes informações de identificação do tráfego relativo ao SLS, parâmetros de desempenho, escopo, ações de classificação e condicionamento do tráfego, tratamento de tráfego excedente (que viole o contrato) e período de aplicação. O desenvolvimento de um modelo de informações para este tipo de especificação tem sido discutido em [22, 23].

A descrição de fluxo fornece os valores dos parâmetros de classificação. A identificação é realizada através da origem e destino do pacote, da informação de Serviço Diferenciado (campo DS) e de informações de aplicação. A classificação do tipo BA (*behaviour aggregate*) determina que somente o campo DS seja fornecido. A classificação do tipo MF (*multi-field*) prevê a utilização de diversos campos de um pacote, sendo útil na identificação de micro-fluxos.

Os *parâmetros de desempenho* representam as garantias de serviço que um ou mais fluxos irão receber. Os quatros principais parâmetros a serem determinados são o atraso, o *jitter*, a probabilidade de descarte de pacotes e o *throughput*. Alguns algoritmos de condicionamento, de característica binária, somente permitem um nível de classificação. Todo tráafego excedente é considerado como sendo uma violação do contrato. No caso de algoritmos de característica multi-nível, os valores dos parâmetros podem ser determinados para diferentes níveis de prioridade sendo considerados.

As informações de escopo em um SLS delimitam a aplicação da política de QoS, identificando os limites (topológicos ou geográficos) aos quais o SLS corresponde. A aplicação de um SLS é associada de maneira uni-direcional com os fluxos de tráfego, sendo que SLS para diferentes direções podem ser combinados. Os limites do escopo devem especificar unicamente os limites envolvidos. Exemplos são números de IP válidos para a origem e destino do tráfego, dentro do domínio.

A classificação sobre o tipo de tráfego considerado é importante, por estar relacionado aos valores fornecidos para os parâmetros de desempenho. Considera-se **tráfego qualitativo** todo tráfego em que os parâmetros de desempenho tenham valores relativos, sendo que sua interpretação pode ser subjetiva. Um exemplo de tráfego qualitativo seria dado por um SLS que determinasse que todo tráfego de um determinado domínio tivesse **baixo** atraso, ou **taxas de perda** médias. O **tráfego quantitativo** possui valores específicos em um SLS, que especificam o valor exato a ser alcançado por um determinado parâmetro. Um exemplo deste tipo de tráfego seria determinar uma banda de 2 Mbps para todo tráfego UDP de um determinado link. Os valores fornecidos para tráfego quantitativo podem ser para o pior caso, por exemplo, atraso e descarte máximo de pacotes.

2.6 Oferta de QoS através da Engenharia de Tráfego

A utilização eficiente dos recursos de rede é fator determinante para o oferecimento de qualidade de serviço. O conceito de **engenharia de tráfego** envolve todo um contexto de modelagem, análise, avaliação, implementação e otimização dos processos e mecanismos que controlam o tráfego de uma rede, com a motivação de fazer o melhor uso dos recursos de rede, mantendo as garantias de QoS. O aspecto principal está no desempenho, mas analisando todos os componentes envolvidos na transmissão de dados, desde a fonte geradora, passando pela infra-estrutura de rede, até o destino.

A engenharia de tráfego é definida como uma estratégia de engenharia de redes que tem por objetivo a avaliação e otimização do desempenho da infra-estrutura de rede, envolvendo a aplicação de tecnologias e estratégias na avaliação, caracterização, modelagem e controle do tráfego e dos recursos de rede. Este objetivo é alcançado através do controle orientado aos requisitos de desempenho, utilizando os recursos de maneira eficiente, confiável e econômica. Exemplos de medidas orientadas ao desempenho incluem atraso, variação no atraso, perda de pacotes e *goodput* [24].

A proposta de otimização na engenharia de tráfego é alcançada através do gerenciamento da capacidade e do tráfego. O gerenciamento da capacidade está relacionado à infra-estrutura, envolvendo planejamento, controle de roteamento e gerenciamento de recursos. Por exemplo, recursos de rede tais como banda passante, *buffers* dos roteadores, etc. O gerenciamento do tráfego se relaciona com mecanismos que regulam os fluxos de dados ou que restringem o acesso aos recursos de rede. Os mecanismos de condicionamento e escalonamento são exemplos deste tipo de gerenciamento.

A proposta de otimização é um processo contínuo de análise, aplicação e revisão de resultados obtidos. A evolução dos recursos computacionais influi nesse processo, bem como as metas a serem alcançadas. O controle da infra-estrutura é a estratégia a ser utilizada para se implementar a engenharia de tráfego. Este tipo de controle envolve

dimensionamento, controle sobre o roteamento, controle sobre o tráfego e controle de acesso aos recursos. Obtem-se, dessa maneira, um sistema de controle responsável pela otimização do desempenho, que faz uso de informações tais como variáveis de estado da rede, políticas de controle e de decisão.

Diversos contextos são definidos na metodologia de aplicação da engenharia de tráfego:

- **Contexto de rede**: envolve as situações nas quais problemas de engenharia de tráfego ocorrem. Estão relacionados a esse contexto a estrutura de rede, as políticas e características da rede, bem como atributos de qualidade, regras e otimização da rede;
- **Contexto de problema**: define questões gerais e específicas a serem atacadas pela engenharia de tráfego. Implica na identificação, abstração de características relevantes e representação dos requisitos necessários e desejáveis de soluções;
- **Contexto de solução**: sugere como solucionar problemas de engenharia de tráfego;
- **Contexto de implementação e operação**: determina a aplicação das soluções aos problemas identificados. Envolve planejamento, organização e execução.

Um modelo de processos descreve as características práticas em alto-nível de um contexto operacional de engenharia de tráfego. Esta descrição é apresentada como uma seqüência de ações a serem realizadas, consistindo em uma metodologia que pode ser aplicada por elementos autônomos (automatização dos processos) ou por um gerente humano. A definição das políticas de gerencimento da operação da rede é realizada como estágio inicial, considerando diversos fatores tais como o modelo geral, o custo de operação, regras de operação e critérios de otimização. Os outros componentes desta metodologia compreendem: avaliação (medida), validação da corretude do modelo criado e uma fase de otimização do sistema.

A avaliação é um componente crucial na engenharia de tráfego. As medidas de desempenho da rede são essenciais para determinar o sucesso de uma determinada abordagem, além de fornecer informações para reavaliação e adaptação dos sistemas de engenharia de tráfego.

As soluções para o oferecimento de QoS têm sido incorporadas na discussão da engenharia de tráfego. A metodologia de diferenciação de serviços proposta na arquitetura de Serviços Diferenciados oferece a escalabilidade necessária para oferecer um comportamento mais previsível da rede. Entretanto, a granularidade no gerenciamento de fluxos individuais não é alcançada de maneira satisfatória. Nesse caso, a integração com a arquitetura de Serviços Integrados é uma alternativa a ser considerada.

Alguns requisitos para a implementação da engenharia de tráfego na Internet são apresentados e discutidos em alto-nível em [24]. Estes requisitos compreendem: requisitos gerais, requisitos de roteamento e medição, além de tolerância a falhas do sistema.

Um sistema de engenharia de tráfego deve oferecer **usabilidade**, ou seja, estar disponível para operação e ser de fácil manutenção. A **automação** deve estar presente o máximo possível, minimizando as intervenções manuais por parte do operador no controle da rede. Um fator determinante consiste na **escalabilidade** que este sistema deve possuir.

Qualquer modelagem deve ser escalável, adaptando-se facilmente às mudanças na infra-estrutura de rede e no volume de tráfego. A adaptação ao novo contexto não deve impactar outros processos na infra-estrutura de rede. A **estabilidade** é determinante neste tipo de sistema, principalmente na resposta a eventos gerados por mudanças ocorridas no estado da rede. Um compromisso entre a resposta e a estabilidade tem de ser determinado para garantir, sempre, um maior grau de controle.

A **flexibilidade** permite mudanças no sistema de engenharia de tráfego, com o principal objetivo de otimizar funções de controle, sem afetar a estabilidade da rede. A **observabilidade** consiste na capacidade de se obter estatísticas do funcionamento da rede, úteis na análise do funcionamento da infra-estrutura de rede. A facilidade de gerencimento desse tipo de sistema implica em se buscar ao máximo a **simplicidade** na modelagem das interfaces disponíveis. Duas características importantes são o **controle de congestionamento** e a **tolerância a falhas**.

O controle de congestionamento tem por objetivo evitar ou minimizar a perda de desempenho devido a uma situação em que os recursos de rede se esgotam e o atraso ou descarte de pacotes impedem de se manter as garantias de QoS. A tolerância a falhas é crítica para uma rede se manter operacional em situações de falhas de qualquer natureza na infra-estrutura de rede. A utilização de redundância dos recursos de rede e o desenvolvimento de mecanismos mais confiáveis fazem parte da implantação de um sistema tolerante a falhas.

O controle de roteamento é um dos aspectos mais importantes de um sistema de engenharia de tráfego. Este tipo de controle possibilita contornar diversas limitações dos protocolos de roteamento tradicionais que não atendem à expectativa de otimização dos recursos de rede.

O roteamento baseado em restrição (*constraint based routing*) é de grande utilidade para o desenvolvimento da engenharia de tráfego na Internet. Sua grande vantagem está em decidir por rotas de encaminhamento considerando diversas restrições que podem expressar as expectativas de QoS que se deseja obter, através de diversos novos parâmetros. Exemplos deste parâmetros são: banda disponível, contagem de *hops*, atraso e, ainda, políticas definidas para gerenciamento.

Outra grande vantagem do controle de roteamento está na capacidade de oferecer

suporte ao balanceamento de carga na rede, redistribuindo tráfego na infra-estrutura buscando otimizar sua utilização. Uma vez definidos os valores dos parâmetros de QoS, a seleção dos caminhos de roteamento são determinados com base nesses valores. Existe, nesse caso, a possibilidade de caminhos com um maior número de *hops* serem escolhidos, uma vez que ofereçam o suporte às exigências de transmissão ou que se apresentem como a melhor alternativa para uma utilização otimizada dos recursos.

Capítulo 3

Bandwidth Brokers

A função de um *Bandwidth Broker* é de gerenciar os recursos da rede, monitorando a utilização de recursos[25, 26, 27]. Ainda, deve atender às políticas estabelecidas para o domínio, implementado-as através da configuração dos roteadores-folha e de borda. Esta etapa irá determinar o tratamento destinado aos pacotes, de acordo com a classe de serviço a qual pertencem. Os parâmetros estabelecidos durante o controle de admissão devem ser usados para controlar os mecanismos de adequação, classificação e escalonamento dos pacotes. O papel desempenhado pelo Bandwidth Broker também pode ser visto como um gerenciador de políticas, que coordena a aplicação de políticas administrativas no acesso aos recursos de rede do domínio administrativo.

A Figura 3.1 mostra uma ilustração desta arquitetura. Os roteadores de núcleo são representados por *Ri* (roteador interno). A interação é realizada entre o Bandwidth Broker e os roteadores-folha e de borda, representados por *Rf* e *Rb*, respectivamente. A aplicação *Apl* se comunica com o Bandwidth Broker (através de uma API), realizando seus pedidos de reserva.

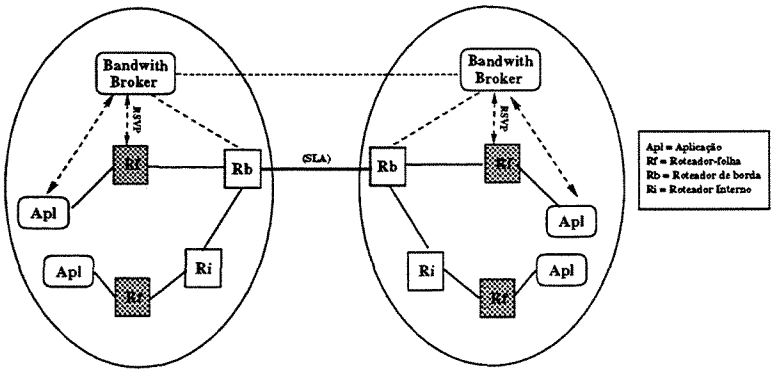

Figura 3.1: Gerenciamento de recursos de um domínio através de um Bandwidth Broker

O tráfego de *Apl* é enviado para o roteador-folha, que executa os mecanismos de tráfego de acordo com a configuração realizada. Todo tráfego com destino em outro domínio administrativo será controlado pelo roteador de borda, que aplicará os mecanismos de acordo com a configuração recebida do Bandwidth Broker. A comunicação entre Bandwidth Brokers vizinhos permite a troca de informações úteis na configuração dos roteadores de borda e também para atualização, estabelecimento ou remoção de um SLS.

Os roteadores-folha têm a função de controlar o tráfego no momento em que começa a ser transmitido. A Figura 3.2 mostra o processo de configuração do roteador que recebe o tráfego de uma aplicação. A aplicação interage com o Bandwidth Broker através de sua interface (API). As mensagens trocadas especificam os valores dos parâmetros a serem utilizados. O Bandwidth Broker, por sua vez, interage com o roteador-folha da aplicação (*Rf*), configurando seus mecanismos de tráfego. *Rf* irá aplicar estes mecanismos nos pacotes recebidos do fluxo de Apl. Eventualmente, o Bandwidth Broker irá atualizar um ou mais roteadores de borda para acomodar o novo fluxo inserido no domínio. Este exemplo utiliza o RSVP como protocolo para configuração dos roteadores.

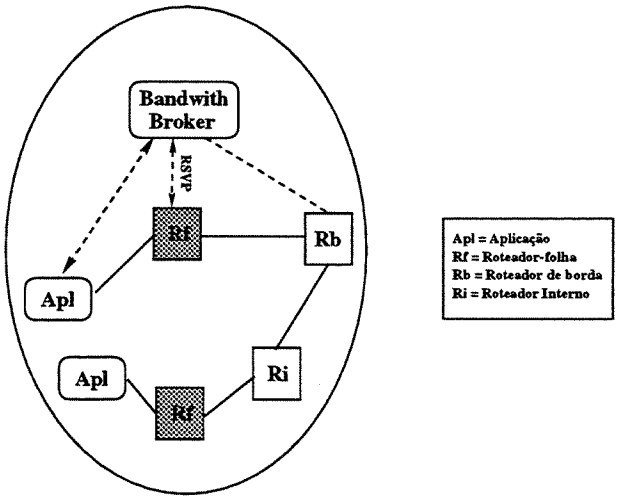

Figura 3.2: Configuração de um roteador-folha utilizando RSVP

O recebimento de um pedido de reserva inicia o processo de controle de admissão, no qual se avaliam os valores dos parâmetros sendo exigidos. O *Bandwidth Broker* mantém uma base de dados de todos os fluxos que entram e saem do domínio que ele controla. Esta base de dados deverá ser atualizada toda vez que um novo fluxo for admitido, ou um já existente seja alterado ou retirado. Uma vez considerado o pedido, caso não haja espaço para uma nova admissão, uma resposta negativa deve ser enviada para a aplicação. Caso contrário, a aplicação é notificada e pode iniciar a transmissão de acordo com o contrato estabelecido, sendo que o roteador-folha que recebe este tráfego cuida de garantir a manutenção do contrato.

Percebe-se que o policiamento e a adequação de tráfego realizados dessa forma contribuem bastante para a escalabilidade do sistema. A justificativa está justamente no fato de que a carga computacional é transferida para os roteadores-folha e de borda, até para o próprio Bandwidth Broker, fazendo com que os roteadores internos da rede executem os mecanismos mais simples da arquitetura.

Considerando redes de grande porte, os roteadores internos representam um *backbone* de alta velocidade, enquanto os outros roteadores cuidam de alimentar as linhas de transmissão com os pacotes gerados internamente e também aqueles provenientes de tráfego externo ao domínio.

A transmissão de dados fim-a-fim pode atravessar um ou mais domínios administrativos, e geralmente atravessam. Cada domínio possui sua própria política de gerenciamento de recursos. Sendo assim, a consulta prévia da disponibilidade de recursos para este tráfego, que tem origem em organizações vizinhas, tem de ser feita para que se possa garantir a utilização das classes de serviço propostas. Todo tráfego que não pode contar com garantias da infra-estrutura de rede deverá ser transmitido através de melhor-esforço.

3.1 Funcionalidades de um Bandwidth Broker

O desenvolvimento de um Bandwidth Broker depende da especificação de suas funcionalidades, que possibilita utilizá-lo na interação entre hosts e usuários da rede (além do próprio operador da rede). Algumas destas funcionalidades são apresentadas em [26], através das seguintes interfaces:

Interface com o operador de rede

Este tipo de interface é uma GUI que possibilita o operador da rede preencher o repositório de dados com informações de topologia e configuração, inserir as regras de políticas do domínio, inserir requisições de serviço no sistema, gerar consultas (relatórios) e mensagens de erro, entre outras tarefas administrativas.

A existência desta interface oferece facilidades de inserção de novas classes de serviço, gerência dos recursos utilizados, estabelecimento e atualização de acordos inter-domínios (SLAs) e configuração dos roteadores de núcleo e de borda no uso dos mecanismos para oferecer a prioridade de tráfego.

Interface* Host/*Usuário

Possibilita que aplicações e usuários direcionem requisições e consultas para o sistema na obtenção de serviços. Características de segurança na disponibilização destas informações são desejáveis. A interface com o usuário deve ser uma GUI, assim como a

fornecida para o operador de rede. A interface para aplicações exige um protocolo com a API correspondente.

A natureza da negociação dos contratos está relacionada às estratégias de oferecimento de QoS. Os contratos de natureza estática são geralmente criados a partir de acordos entre agentes humanos (administradores de sistema e usuários). A renegociação geralmente ocorre em um intervalo considerável de tempo (dias, meses, anos) e os serviços continuam a ser implementados mesmo que o padrão de tráfego se modifique, até que sejam renegociados.

Os contratos de natureza dinâmica estão relacionados à capacidade de adaptação ao padrão de tráfego. Estes contratos podem mudar com maior freqüência em um intervalo de tempo menor, até mesmo em reação a eventos da rede (queda de enlaces, aumento do tráfego em determinado período do dia, sinalização, etc). Uma observação importante é de que geralmente os SLS se referem a agregacões de tráfego, não sendo desejável a alteração dinâmica de um SLS toda vez que um micro-fluxo pertencente ao tráfego agregado sofra modificações ou alterações.

A obtenção da qualidade de serviço fim-a-fim depende da capacidade de se decidir e configurar os elementos de rede. O fornecimento destas informações geralmente é necessário em diversos pontos da rede. Decisões de caráter estático geralmente estão relacionadas a alterações na infra-estrutura de rede, como aumentar a capacidade de uma conexão física. O processo de decisão em um período curto de tempo possui uma característica dinâmica, estando relacionado às configurações lógicas dos mecanismos de diferenciação de serviços. Uma vez tomada a decisão, a tarefa de configuração consiste em distribuir essa informação, na forma dos parâmetros apropriados.

Os roteadores de borda têm um papel importante nesta etapa de configuração, pois são responsáveis pela implementação dos acordos de serviço. Estes representam pontos de interligação entre diferentes domínios e sua administração é mais complexa por envolver a utilização de mecanismos de classificação, adequação e de policiamento. Estes nós de rede possuem, portanto, um papel fundamental na utilização de um sistema gerenciador de QoS. Eles são responsáveis pela implementação dos mecanismos definidos na Seção 2.3.1.

Exemplos de informações são taxa de transmissão, valor máximo de rajada e tipo de classe de serviço. Diversas soluções podem ser usadas nesta etapa de configuração, por exemplo, através de protocolos tais como SNMP, RSVP e CLI (utilização de scripts), podendo ainda ser configuradas manualmente pelo administrador da rede. Um mecanismo de autenticação é necessário para realização de configuração remota (automática ou manual) para oferecer garantias de segurança no trânsito dessas informações.

Entretanto, o gerenciamento de uma grande quantidade de fluxos recebidos por esses nós de borda pode não ser uma tarefa fácil de se administrar. A configuração dos nós internos aumenta a complexidade do gerenciamento, pois além da grande quantidade

de dispositivos, vários deles podem exigir diferentes configurações ao longo do tempo e de acordo com o padrão de tráfego. A manutenção de uma base de dados com estas informações não é uma tarefa simples, sendo que a tarefa de gerenciar esses estados e manter a integridade das classes de serviço se torna mais complicada de acordo com a dimensão da infra-estrutura de rede.

3.2 Discussão sobre as estratégias atuais

O oferecimento de QoS através de classes de serviços além do melhor esforço tem cada vez mais sido discutido como um caminho natural para solucionar problemas característicos do serviço de melhor-esforço, por exemplo, imprevisibilidade da rede em situações de congestionamento. A utilização de gerenciadores de recursos tem sido vista como uma solução bastante adequada para o gerenciamento da oferta de qualidade de serviço em redes de computadores.

Atualmente, este tipo de gerenciamento ainda é feito através dos administradores de rede, que configuram manualmente os roteadores. Os valores escolhidos para configuração são definidos através do histórico de utilização da rede e estas configurações podem durar dias, semanas ou meses. O tráfego entre domínios vizinhos também é controlado desta forma, sendo que os valores são negociados por meios tais como e-mail, telefone, ou outro método que implique em intervenção manual.

A vantagem que pode ser observada na utilização de um *Bandwidth Broker* é que se permite uma melhor utilização dos recursos da rede, uma vez que somente são reservados quando necessário. Diversas abordagens são possíveis. Um Bandwidth Broker pode coordenar os pedidos de reserva dentro de uma quantidade de recursos definida através de um SLA, configurado manualmente. Ainda, o SLA pode simplesmente definir algumas políticas básicas a serem obedecidas (tarifação, horários de utilização, etc), sendo que o gerenciamento de recursos (banda passante, por exemplo) pode ser completamente gerenciado pelo Bandwidth Broker dinamicamente, através dos pedidos de reserva. Uma outra opção é utilizar esta entidade para gerenciar os próprios SLSs, definindo políticas para alteração dos SLS e atualizado-os de acordo com as estatísticas de uso da rede.

A estratégia adotada pode variar para cada domínio administrativo. A principal questão em aberto é a definição de um protocolo de interoperabilidade entre Bandwidth Brokers, de forma a permitir que a negociação dos contratos garanta qualidade de serviço fim-a-fim. Outra questão em aberto é justamente a definição de um modelo para este tipo de entidade, as interfaces para sua utilização e seu posicionamento dentro do domínio administrativo (centralizado ou distribuído).

3.3 Trabalhos Relacionados

Um grande esforço tem sido realizado no desenvolvimento da Internet2 [28], com o objetivo de disponibilizar um nova infra-estrutura, possibilitando o desenvolvimento de novas aplicações (laboratórios virtuais, tele-medicina, ensino à distância) com características de qualidade de serviço (banda larga, baixo atraso, etc) e de tempo-real, além de oferecer transferência de tecnologia para a comunidade da Internet. A Internet2 é formada por um consórcio de 180 universidades, que trabalham em parceria com a indústria e o governo no desenvolvimento das novas aplicações de rede e de tecnologia.

Um dos focos de desenvolvimento está na iniciativa do **QBone** [29], um ambiente de experimentação do controle dos recursos de rede utilizando as novas tecnologias de qualidade de serviço sobre IP. O QBone baseia-se na arquitetura de Serviços Diferenciados. Uma das iniciativas dos participantes do QBone está no desenvolvimento de protótipos de Bandwidth Brokers para automatização da configuração dos roteadores de borda. O estágio da documentação sobre Bandwidth Brokers do grupo QBone pode ser encontrada em [30]. Uma lista de referência de trabalhos relacionados ao desenvolvimento do QBone pode ser encontrada em [31].

Um trabalho neste contexto tem sido desenvolvido em [32]. A implementação de Bandwidth Broker realizada pela Siemens também utiliza CORBA para a comunicação das informações de gerenciamento com os dispositivos de rede e como interface de requisição por parte das aplicações. As interfaces de controle das filas e dos classificadores são especificadas em IDL. Os mecanismos de QoS se baseiam na implementação do algoritmo CBQ da Sun.

O gerenciamento é realizado por um objeto denominado *QoS Manager*. A interface IDL deste objeto é mapeada para a linguagem Java, oferecendo também uma interface gráfica Web, utilizando-se de servlets.

O trabalho apresentado em [33] divide as tarefas do Bandwidth Broker em dois planos: o Plano de Gerenciamento e o Plano de Encaminhamento. O Plano de Gerenciamento contém os mecanismos de armazenagem e manipulação das informações dos parâmetros de rede, acessíveis via interface gráfica com o administrador. O protocolo COPS é utilizado para acesso dos roteadores e configuração dos mecanismos no Plano de Encaminhamento. O protótipo desenvolvido utiliza o sistema operacional FreeBSD, com o *kernel* modificado através do pacote ALTQ.

O trabalho aqui apresentado oferece um protótipo para validar a utilização de estratégias de políticas. A utilização de CORBA foi realizada para oferecer uma independência não somente do ponto de vista das aplicações, mas também do ponto de vista do elementos de rede responsáveis pelas tarefas do nível de rede. A diferença principal em relação ao trabalho desenvolvido pela Siemens é propor uma modelagem para a distri-

buição dos componentes, também através da utilização de CORBA. Ainda, é proposto a capacidade de migração de um contexto de simulação para um contexto real, reutilizando as interfaces validadas em simulação.

A principal diferença em relação ao trabalho apresentado em [33] está em explorar a discussão atual do uso de políticas de alto-nível, através da adoção de um modelo de 3 camadas de atuação para a gerência dos dispositivos. A camada adicional busca integrar as propostas de controle baseadas em políticas de alto-nível, que determinam o comportamento da rede.

Um trabalho relacionado ao aqui apresentado, principalmente por fazer parte de uma das propostas de trabalhos futuro é o ALTQ. O desenvolvimento do pacote ALTQ se identifica com a Camada de Tratamento de Pacotes (Seção 2.5.1). Este projeto é uma arquitetura de qualidade de serviço desenvolvida com o objetivo de prover controle do processo de escalonamento de pacotes.

A implementação desta arquitetura está baseada no sistema operacional FreeBSD, inserindo alterações no *kernel* que modificam o sistema tradicional de gerenciamento de filas baseado em FIFO. Ainda, pequenas alterações nos drivers dos dispositivos foram desenvolvidas para adaptar o sistema operacional, oferecendo a capacidade de diferenciação de serviços no encaminhamento de pacotes.

A abordagem do ALTQ concentra-se em roteadores baseados em PCs. Esta estratégia é justificada considerando o crescimento da capacidade computacional dos computadores, aliado ao surgimento de placas de redes de alta velocidade e os decrescentes custos associados a estes equipamentos. Uma vantagem adicional está na flexibilidade de implementação e atualização de software, de acordo com as necessidades que surgem no controle do roteamento.

A funcionalidade do ALTQ consiste em:

- estratégias de escalonamento;
- estratégias de descarte de pacotes;
- estratégias de alocação de *buffers*;
- múltiplos níveis de prioridade e,
- filas *non-work conserving.*

Diferentes disciplinas podem compartilhar os seguintes blocos básicos:

- classificação de pacotes;
- gerência de pacotes;

- suporte de *drivers* dos dispositivos.

Através do compartilhamento destes blocos básicos, novas disciplinas podem ser desenvolvidas sem necessidade de se preocupar com a implementação contida no kernel.

Uma característica interessante da arquitetura ALTQ está em oferecer suporte para pesquisa, oferecendo uma plataforma de pesquisa em gerenciamento de filas, possibilitando que novos algoritmos sejam experimentados em situações reais de utilização, mas também permitindo sua utilização como infra-estrutura prática de suporte a redes com roteamento baseado em PCs. Este objetivo foi alcançado através da implementação dos algoritmos CBQ e RED, possibilitando a utilização imediata da arquitetura em um contexto real de roteamento.

Capítulo 4

Modelagem de um Bandwidth Broker

4.1 Requisitos

Alguns requisitos podem ser definidos na discussão das funcionalidades de um Bandwidth Broker. A definição de diversos componentes desta entidade e das características básicas é discutida em [26]. Alguns dos requisitos estão relacionados às facilidades operacionais básicas que um Bandwidth Broker tem de oferecer:

Facilidade de administração

Esta interface possibilita o administrador da rede gerenciar as informações de configuração. Isto consiste em preencher as bases de dados com informações de topologia e configuração, inserir as regras de políticas do domínio, inserir requisições de serviço no sistema, gerar consultas (relatórios) e avaliar mensagens de erro, entre outras tarefas administrativas.

A existência desta interface deve oferecer ainda facilidades de inserção de novas classes de serviço, gerência dos recursos utilizados, estabelecimento e atualização de acordos inter-domínios (SLAs) e configuração dos roteadores de núcleo e de borda no uso dos mecanismos para oferecer a prioridade de tráfego.

Facilidade de requisição

Possibilita que aplicações e usuários direcionem requisições e consultas para o sistema na obtenção de serviços. Características de segurança na disponibilização destas informações são necessárias. A interface com o usuário deve ser preferencialmente gráfica, assim como a fornecida para o administrador de rede. A interface para aplicações pode oferecer um protocolo para envio das requisições ou uma API específica.

Facilidade de operação

A capacidade operacional constitui a colagem entre as camadas apresentadas na Seção 2.5. A operação de um sistema baseado em uma arquitetura de políticas implementa uma série de traduções de instruções de alto-nível em valores ou estruturas computacionais de baixo-nível. A importância deste requisito é de integrar diversas soluções, características de cada camada.

4.2 Modelo de Desenvolvimento

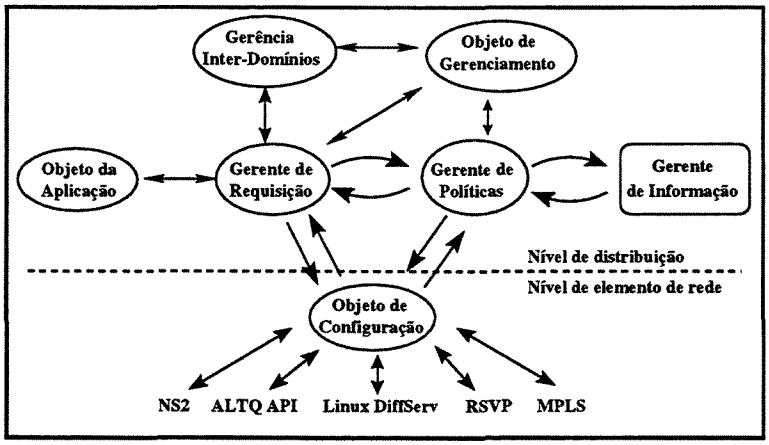

Figura 4.1: Modelo de Desenvolvimento para o Bandwidth Broker

O modelo de desenvolvimento apresentada na Figura 4.1 mostra o posicionamento dos objetos em dois níveis. O nível de distribuição contém os componentes pertencentes às camadas de armazenagem e de decisão. O nível de elemento de rede contém o componente que oferece a interface entre os componentes do nível de distribuição e o nível físico de encaminhamento de pacotes.

A separação dos componentes da arquitetura nestes dois níveis se mostra interessante para viabilizar nosso objetivo: oferecer uma implementação de Bandwidth Broker, buscando o máximo de abstração da camada de tratamento de pacotes. Os componentes compreendem:

Objeto da Aplicação

A obtenção de recursos por parte de uma aplicação consiste em sinalizar uma requisição pelos recursos desejados e se submeter ao controle de admissão. O Objeto da Aplicação deve oferecer a interface para realizar requisições, mapeando as exigências da aplicação para os métodos da interface do Gerente de Requisição.

Gerente de Requisição

Este objeto é intermediário nas requisições por recursos de rede. O processo de controle de admissão implica na consulta às políticas estabelecidas para o domínio administrativo, assim como na avaliação da disponibilidade de recursos para fornecer novas reservas a usuários autorizados.

As requisições principais são:

- registro de um novo elemento de rede, informando ao sistema quais as características de configuração de dispositivos específicos;
- manutenção das políticas administrativas de gerenciamento do domínio (acesso a recursos, tarifação, etc);
- requisição por parte de Objetos da Aplicação por recursos de rede para tratamento diferenciado do seu tráfego.

Outras tarefas que podem ser atribuídas a este componente estão relacionadas à autenticação do usuário (ou da aplicação) e tarifação do uso dos recursos de rede.

Objeto de Gerenciamento

A funcionalidade exercida pelo Objeto de Gerenciamento permite ao administrador da rede a execução de instruções explícitas no controle sobre a configuração dos recursos de rede, bem como controle sobre a manutenção das políticas implementadas pelo objeto **Gerenciador de Políticas**, oferecendo a interface a ser utilizada por ferramentas de controle e configuração;

Gerente de Políticas

Uma vez aprovada uma solicitação por recursos, esta tem de ser traduzida para os parâmetros específicos do dispositivo configurado. Esta tarefa de tradução é importante para possibilitar um gerenciamento de alto-nível de dispositivos heterogêneos.

O componente Gerente de Políticas cuida da interpretação e do mapeamento das políticas corporativas em parâmetros de configuração. Este componente pertence à Camada de Decisão (Seção 2.5.1), exercendo a função de um PDP.

Objeto de Configuração

O elemento de rede necessita de informações de configuração, de acordo com a solução adotada para oferecer qualidade de serviço no nível físico, o que irá determinar o comportamento deste objeto. Por exemplo, a utilização da API do pacote ALTQ para configuração dos mecanismos de encaminhamento presentes no kernel do FreeBSD.

Este componente pertence à Camada de Tratamento de Pacotes (Seção 2.5.1) e aplica os mecanismos de gerência de tráfego. A instância deste objeto representa um PEP, que realiza a tarefa de implementar as configurações em um roteador. A utilização do protocolo COPS [19] é uma opção de comunicação com os componentes Gerente de Requisição e Gerente de Políticas.

Gerência Inter-Domínios

Uma das principais funcionalidades que uma implementação de Bandwidth Broker tem de possuir é oferecer a capacidade de negociação de qualidade de serviço com domínios administrativos vizinhos. O componente Gerência Inter-Domínios tem de poder negociar e instanciar acordos inter-domínios que viabilizem que tráfegos originados localmente tenham sua qualidade de serviço garantida até o destino, no caso do destino estar em outro domínio. Ainda, esta gerência é determinante para provedores de serviço, uma vez que seu tráfego é quase que totalmente de trânsito.

Um aspecto interessante deste componente está em prover a tarifação do uso de recursos. Particularmente para os provedores de acesso, o controle de admissão está vinculado a cobrança realizada e no ajuste dos acordos de tráfego entre os domínios vizinhos.

Gerente de Informação

O componente Gerente de Informação controla o acesso as informações sendo utilizadas para a operação do Bandwidth Broker. Estas informações incluem desde parâmetros de configuração, determinados por políticas de uso, até parâmetros relativos aos acordos inter-domínios. Este componente cuida da manutenção das bases de dados relevantes ao domínio administrativo de atuação do Bandwidth Broker.

4.2.1 Modelagem da Arquitetura

A arquitetura apresentada na Figura 4.1 coloca em alto-nível as tarefas a serem realizadas pelos componentes para obter o gerenciamento da qualidade de serviço.

A abordagem de desenvolvimento deste trabalho considera a arquitetura CORBA (Seção 5.1) como suporte de comunicação entre objetos. As interfaces que serão discutidas mostram alguns métodos que exercem funcionalidades básicas, as quais devem estar

presentes em uma implementação. Estes métodos realizam tarefas básicas de registro, atualização e sinalização.

Interface Gerente de Políticas

A interface do **Gerente de Políticas** possui métodos para registro dos Objetos de Configuração, receber **alertas** de eventos e requisições por tomada de decisão. Esta interface possui atributos que armazenam as referência do Objeto de Gerenciamento e do Gerente de Requisição. A descrição dos métodos consiste em:

- register_pep(identificador, domínio)

  O método *register_pep()* armazena o identificador do Objeto de Configuração. Esta informação é mantida no contexto do Gerente de Políticas, sendo utilizada nos eventos de distribuição das informações de configuração. O atributo **dominio** especifica o escopo do Objeto de Configuração.

- request(codigo_requisicao)

  o método *request()* possibilita um ou mais Objetos de Configuração realizarem requisições por configuração. Estas requisições podem estar relacionadas com as informações sobre as políticas a serem instaladas. O atributo **codigo_requisicao** determina o tipo de requisicao. As especificações do protocolo COPS [18, 19] definem possíveis valores para este atributo.

- alert_evento(codigo_evento, identificador)

  Este método é utilizado para sinalizar o Gerente de Políticas sobre a ocorrência de um evento. Baseado neste evento, uma tomada decisão por parte do Gerente de Políticas pode ser necessária. A definição do atributo **codigo_evento** e os instantes da utilização deste método estão relacionados às decisões de implementação.

- valida_requisicao(codigo_requisicao, identificador)

  Um Objeto de Configuração pode receber requisições de reserva de recursos na interação com outras soluções de sinalização de qualidade de serviço, como o protocolo RSVP [7]. Este método é definido para demonstrar a possibilidade deste tipo de integração com outros protocolos.

Interface Gerente de Requisição

A interface do componente Gerente de Requisição oferece métodos para obtenção de informações da utilização dos recursos de rede. Os atributos desta interface armazena as referências dos Objetos de Configuração registrados. Os métodos oferecem as seguintes funcionalidades:

- obtem_mapeamento_phb(identificador)

  Este método fornece para os Objetos de Configuração os mapeamentos das filas de encaminhamento em função de um ou mais PHBs implementados.

- obtem_info_fila(identificador)

  Este método deve fornecer as informações específicas de configuração para uma ou mais filas de encaminhamento. Um exemplo da utilização deste método é a configuração dos parâmetros do algoritmo de gerência da fila (RED, RIO, DropTail, etc).

- requisição_qos(identificador)

  O método requisição_qos() sinaliza a necessidade de um Objeto de Configuração de iniciar ou atualizar uma reserva de recursos. A utilização deste método pode implicar na verificação das políticas para Objeto de Configuração especificado pelo parâmetro **identificador**.

- registrar_pep(identificador)

  Este método registra o Objeto de Configuração no Gerente de Requisição. Este registro pode ser compartilhado com o registro realizado através da interface do Gerente de Políticas. Esta é uma decisão de implementação, pois a capacidade de se registrar com as interfaces de controle pode ser levada em consideração em restrições de segurança.

interface Gerencia_Inter_Dominios

Esta interface possibilita a troca de informações entre domínios administrativos, permitindo a negociação de acordos de utilização de recursos para fluxos externos ao domínio de origem.

- registra_fluxo(origem, destino, protocolo)

  O método de registro de fluxo fornece informações para um determinado fluxo ou um agregado. Este método tem a função de sinalizar qual tráfego externo deverá ser mapeado para um determinado contrato de utilização de recursos.

- cria_sls(), modifica_sls() e atualiza_sls()

  Estes métodos têm de estar presentes na manipulação dos acordos de tráfego. Diversos atributos podem ser definidos para utilização, sendo bastante dependente da implementação. O desenvolvimento da arquitetura QBone [30] prevê a definição de um protocolo inter-domínios para ser utilizado na Internet2.

interface Objeto de Configuração

A interface do Objeto de Configuração permite que o Gerente de Políticas e Gerente de Requisição atuem na configuração de seu comportamento. Uma implementação real do componente Objeto de Configuração deverá mapear esta interface na API de controle dos mecanismos de qualidade de serviço. Os seguintes métodos podem ser sugeridos:

- configura_politica()

  Este método determina que o Objeto de Configuração deve obedecer a uma determinada política. Esta política depende de especificação, mas pode, por exemplo, determinar a configuração de valores para um TCB instalado (Seção 2.3.1).

- configura_fila()

  Este método é sugerido para controle dos parâmetros de fila, através da configuração de algoritmos de gerência de fila, ou outros parâmetros que afetem o encaminhamento.

4.2.2 Casos de Uso

A modelagem do nosso trabalho é apresentada em alto-nível na Figura 4.2. Os casos de uso representam nosso entendimento como tarefas básicas a serem desempenhadas na utilização de uma arquitetura de políticas e de um sistema de gerenciamento de recursos. A notação foi realizada em UML (*Unified Modeling Language*).

Os seguintes atores são descritos:

Gerente

O ator Gerente representa a intervenção humana no sistema, por exemplo, administradores de rede e/ou gerentes de informação. Sua tarefa é manter a estabilidade do sistema, provendo as informações para sua operação. Entretanto, uma parte das tarefas deste ator em relação aos casos de uso pode ser realizada por agentes que ofereçam automatização de funções, tais como geração de relatórios e autenticação de usuário. Cabe ao administrador a configuração desta facilidade. No modelo de desenvolvimento apresentada, o mapeamento é realizado pelo Objeto de Gerenciamento.

Usuário

O ator Usuário pode representar um ser humano neste sistema. Entretanto, nosso enfoque está na representação de aplicações habilitadas para qualidade de serviço, ou seja, que possam interagir com um sistema que ofereça este tipo de controle. A abstração

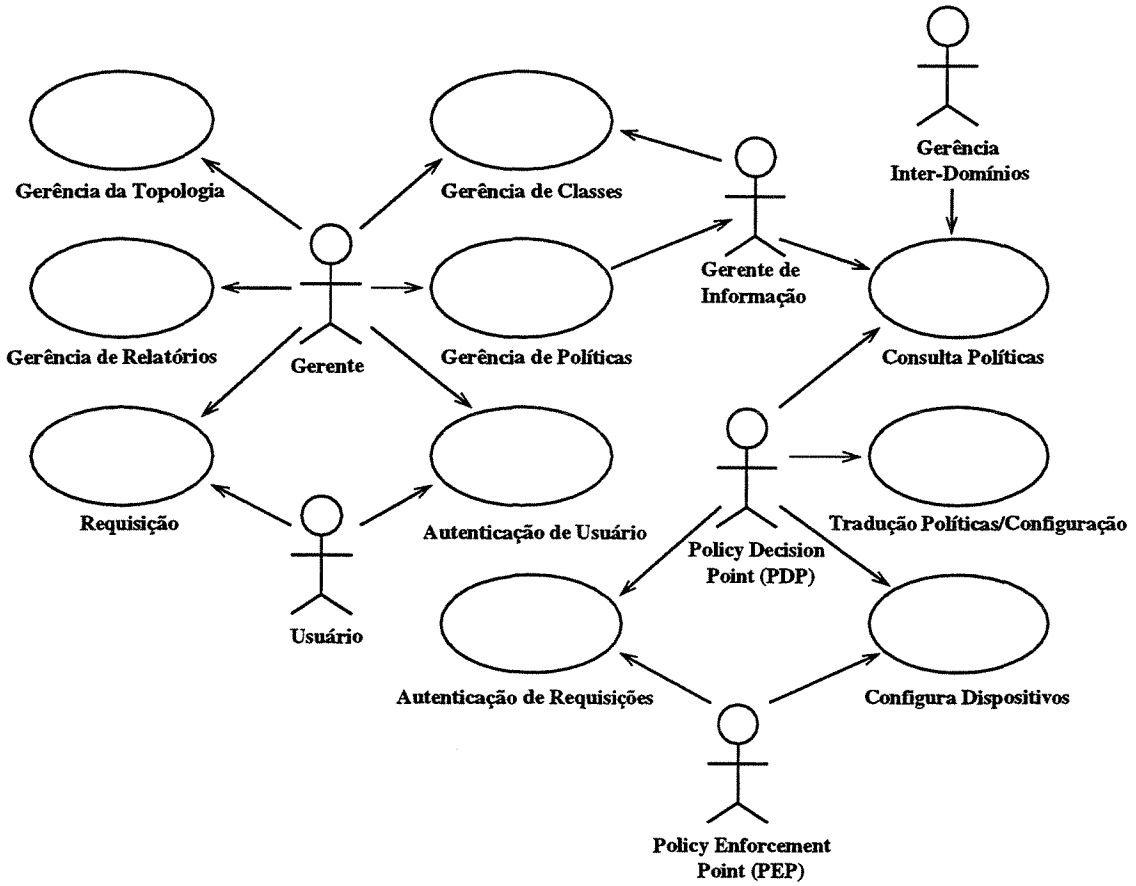

Figura 4.2: Casos de Uso para o Modelo de Desenvolvimento

do sistema para este ator tem de ser maximizada, portanto, suas interações se reduzem à requisição por serviços e autenticação no sistema. O mapeamento deste ator no modelo de desenvolvimento é feito com o Objeto da Aplicação.

Gerente de Informação

Este ator representa as soluções de implementação que atuam na camada de armazenagem de informação, apresentada na Seção 2.5.1. Um exemplo seria uma solução que utilize um banco de dados, acessível via SQL, interagindo com um serviço de diretórios com facilidade de acesso via protocolo LDAP (*Lightweight Directory Access Protocol*) [34]. Este ator é identificado no modelo de desenvolvimento como o componente Gerente de Informação.

Gerência Inter-Domínios

O suporte para tráfego inter-domínios é decisivo em qualquer implementação de Bandwidth Broker. Nosso trabalho insere esta facilidade na modelagem como um ator que

atua nos casos de uso relacionados à consulta de políticas e de tradução e configuração de dispositivos. A existência desta entidade é fundamental, pois o cumprimento dos acordos entre domínios depende, além da negociação dos contratos, da correta configuração dos elementos de borda entre os domínios vizinhos. Este ator está presente no modelo de desenvolvimento no componente Gerência Inter-Domínios.

PDP (*Policy Decision Point*)

O ator PDP representa a entidade de tomada de decisão discutida em [19] e apresentada na Seção 2.5.1. Este elemento é responsável pela tradução de políticas em parâmetros de configuração, tratamento e autenticação de requisições por recursos e manutenção do estado dos dispositivos de rede controlados. O mapeamento deste ator no modelo de desenvolvimento é realizado através do componente Gerente de Políticas.

PEP (*Policy Enforcement Point*)

O ator PEP representa o Objeto de Configuração do modelo de desenvolvimento. Sua tarefa compreende sinalizar o PDP em relação a eventos de requisição por recursos e de mudança de contexto no dispositivo de rede. Conforme descrito em [19], a decisão do PDP tem de ser acatada pelo PEP, que cuida de configurar o dispositivo de rede.

Os casos de uso apresentados são:

Gerência da Topologia

Este caso de uso envolve as tarefas de integração com ferramentas de gerência no controle da topologia da infra-estrutura de rede. Essas ferramentas podem obter estatísticas de uso da rede, oferecer alarmes para problemas nos enlaces, entre outras funções. Tradicionalmente, o protocolo SNMP é bastante utilizado. O surgimento da tecnologia MPLS insere uma nova tarefa para este caso de uso, que é a gerência de caminhos explicítos para os LSPs.

Gerência de Relatórios

Este caso de uso é incluído para oferecer a facilidade de se obter uma documentação da operação da rede. A geração e utilização dos relatórios é um problema a ser abordado pelo nível estratégico de um sistema de políticas (Seção 2.5).

Gerência de PHBs

O caso de uso de Gerência de PHBa deve oferecer a facilidade de configurar parâmetros relacionados ao comportamento dos mecanismos nos roteadores. Neste caso, as configurações a serem realizadas devem estar de acordo com as políticas relevantes para a classe, para um ou mais usuários envolvidos e para o roteador. Um exemplo dessa utilização seria a intervenção no sistema por um administrador para configurar uma reserva para um ou mais fluxos específicos, cujas aplicações não têm capacidade de realizar sinalização.

Gerência de Políticas

A gerência de políticas envolve todas as tarefas de configuração de políticas administrativas, tais como inserção, remoção e atualização. Esta tarefa envolve a administração da rede e de gerentes de informação. O resultado será refletido na instanciação de contratos de qualidade de serviço, na forma de SLSs (Seção 2.5.1).

Requisição e Autenticação de Usuário

Estes casos de uso se relacionam a todos eventos envolvendo o ator Usuário. Basicamente, os eventos são requisições por qualidade de serviço e autenticação para utilização do sistema. O ator Gerente participa destes casos de uso pois está envolvido no cadastramento dos usuários, bem como na configuração de reservas para aplicações que não possuem a capacidade de sinalizar uma requisição

Consulta de Políticas

O caso de uso Consulta de Políticas envolve a interação dos atores de Gerência Inter-Domínios, Gerente de Informação e PDP. As tarefas se relacionam à consulta por políticas administrativas que regem o sistema. A interação maior será realizada pelo ator PDP, para atender as requisições originadas pelos roteadores. O Gerente de Informação é consultado para atualizações nas políticas. Todo tráfego cujo destino esteja fora do domínio administrativo envolve uma interação com a consulta de políticas e com o ator de Gerência Inter-Domínios

Tradução Políticas/Configuração

Este caso de uso possui a tarefa de mapear as políticas administrativas obtidas pelo PDP na Consulta de Políticas em parâmetros a serem enviados para o ator PEP. O formato das mensagens é tarefa do PDP. Esta tradução pode envolver outras consultas ao Gerente de Informação, uma vez que políticas podem conter outras políticas.

Requisição de Recursos e Configuração de Dispositivos

Estes casos de uso envolvem os atores PEP e PDP e estão relacionados às tarefas de prover informações de configuração para os dispositivos de rede. Conforme discutido em [19], o envio de decisões por parte do PDP e a requisição por informações por parte do PEP são eventos assíncronos. Por isso, cada evento pertence a um caso de uso específico.

Capítulo 5

Implementação do Bandwidth Broker

5.1 Utilização de CORBA

Nosso objetivo é explorar o potencial de distribuição de diversas tarefas desempenhadas pelos objetos do modelo aqui proposto. Nosso ponto de vista é que a modelagem de um Bandwidth Broker deve possuir características de distribuição no desempenho de suas funcionalidades.

A motivação é oferecer flexibilidade ao se tratar de questões relevantes tais como tolerância a falhas e desempenho. A preocupação com tolerância a falhas está na importância que um Bandwidth Broker possui nesta abordagem de oferecimento de qualidade de serviço, uma vez que centraliza o gerenciamento dos recursos de rede. O desempenho, por sua vez, pode ser otimizado se a localização dos objetos for transparente ao exercerem sua funcionalidade.

Identificamos alguns pontos principais que, inicialmente, uma implementação distribuída de um Bandwidth Broker deve possuir e que motiva a utilização de CORBA:

Independência de localização

A transparência de localização do Bandwidth Broker é bastante desejável do ponto de vista de tolerância a falhas e flexibilidade de implementação. Uma vez que grande parte da responsabilidade de gerenciamento está nesta entidade, sua persistência tem de ser garantida, impedindo que qualquer tipo de falha impeça que o Bandwidth Broker continue a receber e estabelecer reservas de recursos.

A localização do componenente deve poder ser estabelecida de acordo com a disponibilidade de recursos para implementá-lo, sem que as aplicações que utilizam seus serviços dêem conta disso.

A localização do objeto servidor é transparente para o objeto cliente, que realiza as invocações de métodos como se fossem chamadas locais. Neste caso, o Bandwidth Broker faz o papel de objeto servidor que atende as requisições dos objetos cliente representados pelas aplicações.

Heterogeneidade

Existe uma diversidade de linguagens de programação nas quais as aplicações são desenvolvidas, mas todas têm em comum a necessidade do acesso à infra-estrutura de rede. O acesso ao Bandwidth Broker pode ser feito através de CORBA de maneira transparente em relação à linguagem, utilizando-se de um mapeamento em IDL. Dessa forma, podemos oferecer uma API acessível a todo tipo de aplicação, independente da linguagem em que é implementada. O objeto servidor representado pelo Bandwidth Broker pode também utilizar a linguagem mais conveniente, sem perder em funcionalidade. A Figura 5.1 mostra um modelo deste tipo de interação.

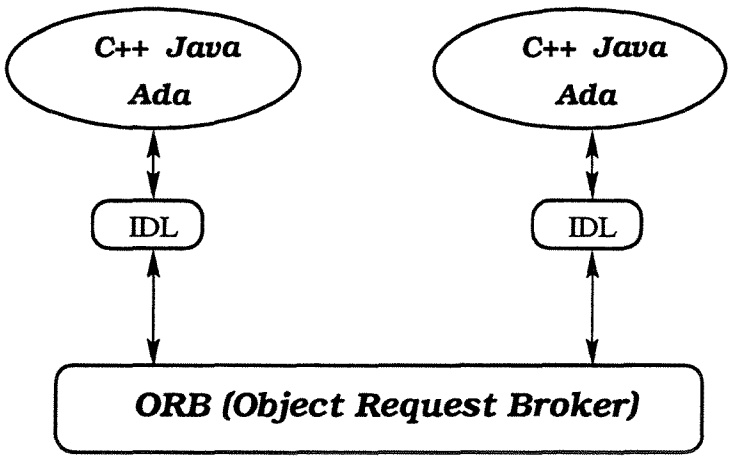

Figura 5.1: Heterogeneidade oferecida pela arquitetura CORBA

5.2 Descrição do Protótipo

5.2.1 Ambiente de Simulação

O simulador utilizado em nosso trabalho foi o NS (*Network Simulator*) [35, 36]. Trata-se de um simulador de eventos discretos direcionado para pesquisa em redes de computadores. Oferece suporte para simulação de protocolos TCP e UDP, roteamento, protocolos multicast e de redes de comunicação sem fio (redes locais e de satélites).

Este simulador surgiu como uma variação do simulador de redes REAL (*Realistic and Large*), em 1989. Seu desenvolvimento e aceitação no meio acadêmico tem sido crescente desde então. Atualmente, tem sido desenvolvido no LBNL (*Lawrence Berkeley National Laboratory*), da Universidade da Califórnia, em Berkeley (UCB), além de receber um grande número de contribuições de desenvolvimento da comunidade que o utiliza. O NS pode ser obtido livremente através da Internet, sendo compatível com uma série de sistemas operacionais.

Esta ferramenta foi desenvolvida para a simulação de uma diversidade de topologias e modelos de tráfego, permitindo ainda que novas funcionalidades sejam acrescentadas conforme a necessidade. Esta característica a torna bastante atrativa para a pesquisa acadêmica.

A máquina de simulação de eventos discretos é implementada em C++. A interface do simulador é oferecida em OTcl [37], uma versão orientada a objetos da linguagem Tcl [38]. Através desta interface, a simulação é criada através da elaboração de *scripts*. Uma parte da simulação pode ser realizada no nível OTcl, entretanto, por razões de otimização e desempenho, as funções básicas são executadas no nível C++.

O código original do NS não oferece suporte para Serviços Diferenciados. Utilizamos a extensão do NS para Serviços Diferenciados desenvolvida pela Nortel [39]. Esta extensão oferece módulos que inserem algumas facilidades no simulador. Um módulo implementa características básicas de um roteador habilitado para Serviços Diferenciados:

- capacidade de implementar múltiplas filas físicas RED em um único enlace;
- capacidade de implementar múltiplas filas virtuais dentro de cada fila física, cada qual com um conjunto independente de parâmetros;
- capacidade de selecionar o enfileiramento de um pacote entre as diversas filas físicas e virtuais, de acordo com as especificações do usuário

Os outros módulos implementam características específicas de roteadores de borda e de núcleo, incluindo um módulo de policiamento que permite a determinação de políticas de classificação e marcação de pacotes nos roteadores de borda.

Utilizamos também neste trabalho o pacote de desenvolvimento Combat [40], que oferece o mapeamento de CORBA para a linguagem Tcl. Através do uso da DII (*Dynamic Invocation Interface*) e da DSI (*Dynamic Skeleton Interface*), este pacote permite a criação de scripts em ambos os lados cliente e servidor de uma comunicação CORBA. O suporte do lado servidor faz uso da extensão iTcl [41], um pacote de orientação a objetos para a linguagem Tcl. Este pacote é disponível sob os termos da licença GNU LGPL (*Library General Public License*) e a versão utilizada suporta o mapeamento para a versão 2.3 da especificação CORBA do OMG [42].

5.2.2 Diagramas de Classes

As classes **dsREDQueue**, **EdgeQueue** e **CoreQueue** pertencem ao nível C++ do simulador, inseridas conforme descrito em [39]. Os métodos destas classes são apresentados na Figura 5.2 e são utilizados para manipular as filas de encaminhamento, aplicando mecanismos tais como classificação, adequação e escalonamento (Seção 2.3.1) nos pacotes de dados.

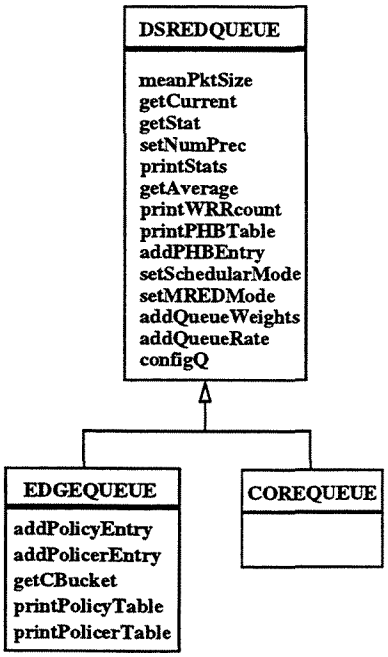

Figura 5.2: Diagrama de classes do simulador

A Figura 5.3 mostra o diagrama contendo as principais classes envolvidas na implementação. Algumas destas classes pertencem ao próprio simulador NS. As classes **dsRED(OTcl)**, **edge(Otcl)** e **core(Otcl)** também são definidas em [39] e oferecem as interfaces em nível de *script*, permitindo o controle dos mecanismos de Serviços Diferenciados implementados pelos métodos apresentados na Figura 5.2.

As próximas classes a serem descritas foram inseridas neste trabalho e são mostradas na Figura 5.4. As classes **PEP(OTcl)** e **CORE(OTcl)** proporcionam uma estratégia de controle das estruturas do simulador, permitindo a configuração de parâmetros a partir de agentes externos. Estes agentes realizam as tarefas definidas para os componentes Gerente de Políticas e Gerente de Requisição.

Esta modificação do simulador consistiu em herdar as funcionalidades das classes **edge(Otcl)** e **core(Otcl)**, oferecendo nossa própria interface de controle dentro do simulador. As classes **PEP(OTcl)** e **CORE(OTcl)** funcionam como um envoltório das classes

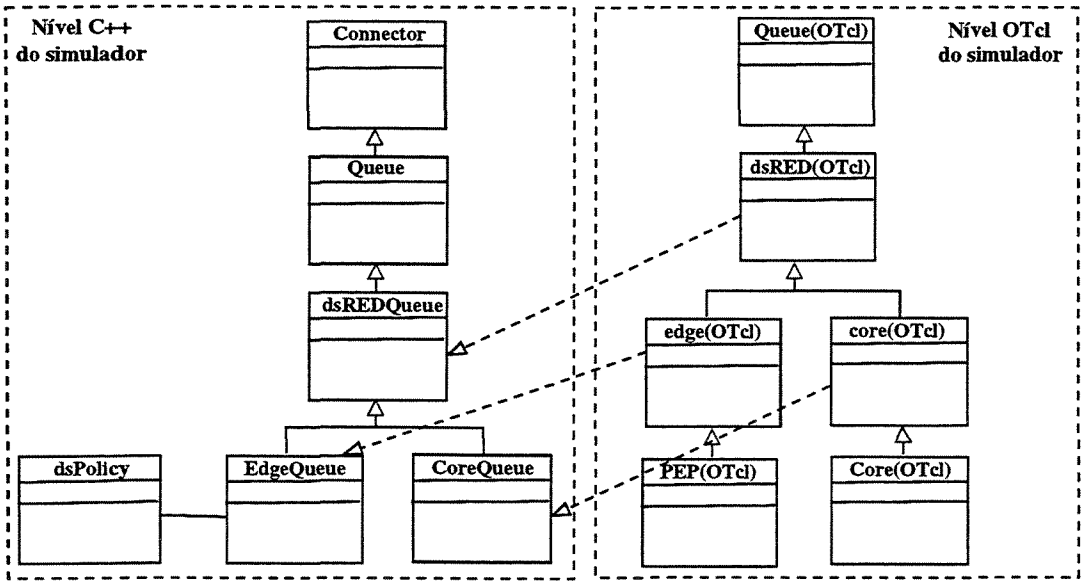

Figura 5.3: Diagrama de classes do simulador

herdadas e implementam algumas das funcionalidades discutidas na Seção 4.2.1. Sendo assim, permite-se que uma simulação instancie objetos que permitirão os agentes externos terem acesso internamente ao simulador. Esta característica é interessante, pois oferece a capacidade de gerenciar externamente o estado das filas, atendendo nosso objetivo de implementar o comportamento de um Bandwidth Broker.

A classe **PEP(iTcl)** implementa a funcionalidade definida para o componente Objeto de Configuração. As instâncias desta classe oferecem uma interface CORBA sendo que, através desta interface, as operações são mapeadas para estruturas dentro do simulador, refletindo no comportamento desejado. Esta classe é instanciada internamente em uma instância **PEP(OTcl)** ou **CORE(OTcl)**, conforme será descrito na próxima seção.

A classe **PDP** implementa as funcionalidades definidas para os componentes Gerente de Políticas e Objeto de Gerenciamento, enquanto que a classe **ResourceManager** implementa as funcionalidades do componente Gerente de Requisição.

5.3 Descrição da Implementação

A abordagem deste trabalho consiste em oferecer extensões para o simulador de rede NS, com o objetivo de obter um ambiente para simular uma arquitetura de políticas (Seção 2.5.1). Através da utilização do pacote de serviços diferenciados oferecido pela Nortel, adicionamos o código necessário para nosso trabalho.

A linguagem Tcl utiliza de extensões que oferecem a facilidade de orientação a objetos.

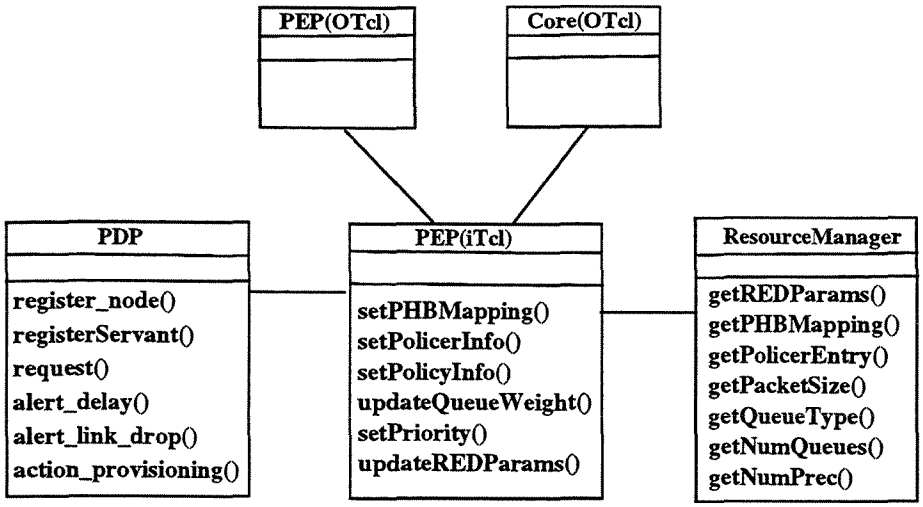

Figura 5.4: Classes inseridas no simulador

A extensão OTcl é utilizada pelo simulador na definição de objetos da simulação. A extensão iTcl foi utlizada neste trabalho para incluir a capacidade de criação de *servants* CORBA na linguagem Tcl. A instanciação de um objeto PEP conforme será descrita, realiza a conexão entre as extensões da linguagem Tcl utilizadas.

Basicamente, foi especificada uma interface que possibilitasse a configuração de parâmetros de QoS nos enlaces que interligam dois nós de rede. Neste trabalho, esta interface foi bastante simplificada, consistindo de métodos que possibilitassem obter o mapeamento dos PHBs, os parâmetros de configuração das filas, as políticas de condicionamento e a prioridade de escalonamento das filas.

As interfaces foram descritas em IDL, sendo que a comunicação é realizada utilizando-se de CORBA. A justificativa para esta escolha está em desenvolver um ambiente que possibilite a transição das interfaces de gerenciamento para uma rede de computadores, avaliando os resultados da simulação com um contexto de uso real.

A definição da classe PEP é realizada como uma extensão da classe Queue/dsRED. Esta classe oferece a ligação entre o script de simulação e os daemons que implementam o PDP e o gerenciador de recursos. A Figura 5.5 apresenta o posicionamento da classe PEP com as classes do simulador. A instanciação de um objeto no simulador implica em três passos:

- criação de um objeto no script de simulação do tipo Queue/dsRED/PEP. Este é um objeto OTcl, e fica disponível para utilização no script de simulação. Algumas interfaces são oferecidas para serem utilizadas de dentro do script;
- criação de um objeto em iTcl, realizando seu registro e ativação no POA. A criação deste objeto é realizada no construtor da classe OTcl. Este objeto é do tipo PEP

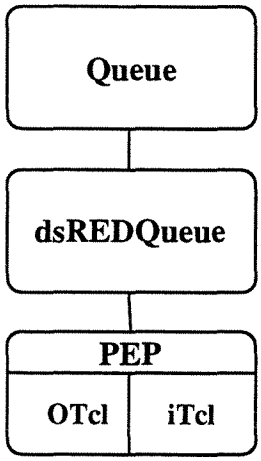

Figura 5.5: Diagrama representando classe inserida no simulador

e implementa as interfaces descritas em IDL para gerenciamento da camada de tratamento de pacotes;

- registro da IOR do objeto iTcl com o daemon PDP. A IOR é utilizada pelo PDP nas ações de configuração do objeto PEP;

Cada objeto PEP possui um identificador. Este identificador é uma *String* utilizada para ativar o objeto no POA. Os objetos PEP armazenam uma referência para um objeto da simulação que controla as filas do roteador. Esta referência é passada através de parâmetros no momento da instanciação do objeto e este tipo de objeto possui métodos que permitem o controle do estado e da operação dos mecanismos de diferenciação.

A estratégia consiste em traduzir as invocações recebidas pelo objeto PEP em invocações para o objeto desta referência, permitindo que o PDP (ou instruções dentro do script) controle o comportamento da camada de tratamento de pacotes. A Figura 5.6 mostra a distribuição dos objetos no ambiente de simulação.

As referências armazenadas nos PEPs são de objetos da classe Queue/dsRED/edge ou Queue/dsRED/core. A documentação completa destas classes está contida em [39]. As interfaces exportadas são utilizadas na configuração dos parâmetros das filas, na seleção dos algoritmos de policiamento e, no caso de objetos da classe Queue/dsRED/edge, na configuração dos classificadores e marcadores de tráfego. A Figura 5.2 contém a descrição destas classes.

O PDP tem o papel de um gerenciador de configuração e de tomada de decisão considerando políticas do domínio. Uma decisão resulta na invocação de métodos de um ou mais *servants* PEP para armazenar ou atualizar o estado nos roteadores. Um *servant* PEP pode fazer o papel de um LDP (*Local Decision Point*), mas isto não está sendo considerado nesta implementação.

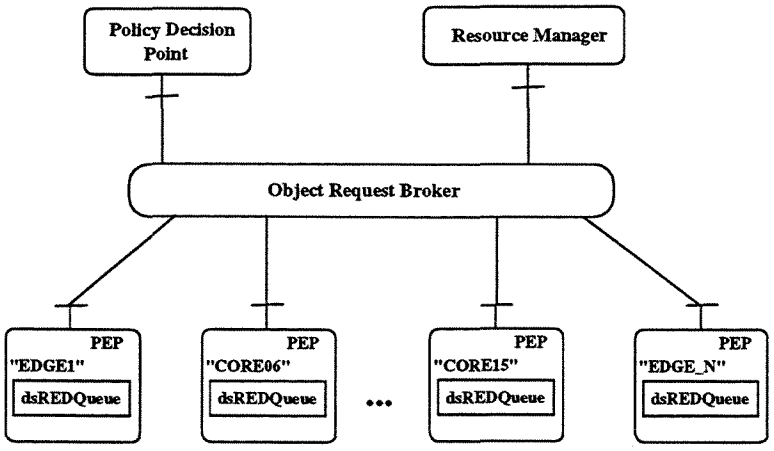

Figura 5.6: Distribuição dos objetos na implementação

As funções dos objetos da arquitetura de policiamento são realizadas por dois *daemons*:

- pdpserver: realiza o papel de um PDP, recebendo as requisições dos PEPs. Este daemon tem de estar instanciado antes da execução de um script de simulação;
- rmanager: tem a função de gerenciar as requisições por recursos de rede, monitorando o arquivo de configuração.

A instanciação destes dois daemons simula a operação de um gerenciador de políticas e de um gerenciador de recursos. Todos objetos PEP criados dentro da simulação se registram no PDP, fornecendo sua IOR. O PDP armazena esta IOR, associando-a com a identificação do PEP. A opção que a ação de registro seja efetuada pelo PEP reflete a definição de operação presente no protocolo COPS [19].

5.3.1 Cenários de utilização

Nesta seção apresentamos alguns cenários de utilização da nossa implementação. As etapas consistem no processo de inicialização da simulação, na qual as alterações inseridas integram o simulador com a arquitetura CORBA.

A Figura 5.7 mostra as etapas de inicialização da simulação. Neste cenário, os daemons são criados e os objetos PEP são instanciados e ativados no POA. Após a ativação, estes objetos registram sua IOR no daemon PDP.

1. daemon rmanager é inicializado:
    - registra um objeto ResourceManager no Serviço de Nomes;

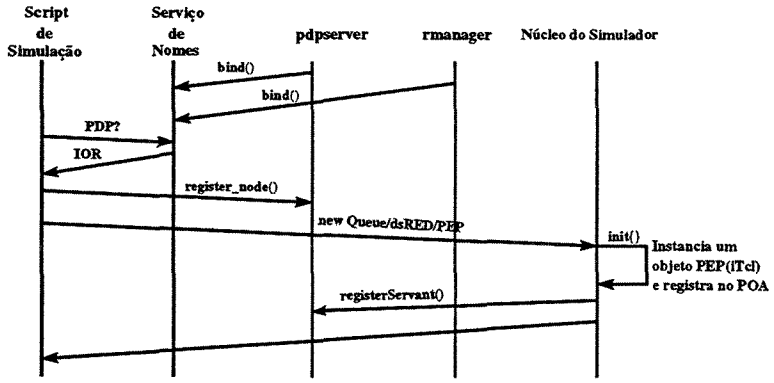

Figura 5.7: Cenário de inicialização da simulação

- percorre o arquivo de configuração, registrando as informações de recursos dos roteadores do domínio.

2. daemon pdpserver é inicializado:

   - registra um objeto PDP no Serviço de Nomes;
   - obtem e armazena referência para o *servant* de ResourceManager;

3. Ativação de objetos PEP (roteadores de borda e de núcleo):

   A ativação de objetos PEP ocorre internamente no simulador. Um objeto da classe PEP(iTcl) é instanciado, recebendo no construtor a referência do objeto de gerência de filas, da classe dsREDQueue.

   Após a instanciação, este objeto PEP(iTcl) é ativado no POA e registrado, através do método *registerServant()* no PDP. Este método tem como parâmetro o identificador do objeto PEP(iTcl) e sua IOR, o que permitirá acesso a sua interface.

O cenário apresentado na Figura 5.8 mostra a inicialização de um objeto PEP, na qual são feitas requisições à entidade externa de gerenciamento de recursos para obter os valores de configuração. A interface implementada permite controlar as informações de configuração das filas dos roteadores no núcleo do simulador. Este controle de configuração é realizado traduzindo os valores obtidos do gerenciador de recursos para os métodos da referência de fila contida em todo objeto PEP.

A configuração dos elementos de borda está representada na Figura 5.9. O script de simulação determina uma requisição por configuração de política (*request POLICY*). Esta invocação faz com que o objeto PEP obtenha a referência do PDP e realize a requisição para ser configurado. O PDP realiza a invocação da interface do objeto PEP(iTcl), o que resulta na configuração dos parâmetros de policiamento (classificadores e marcadores de tráfego e mapeamento dos objetos de policiamento com as classes de serviço).

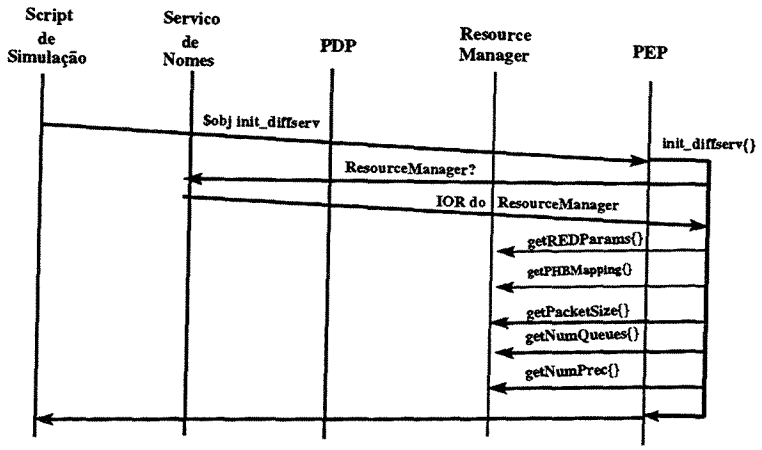

Figura 5.8: Cenário de configuração inicial dos objetos PEP

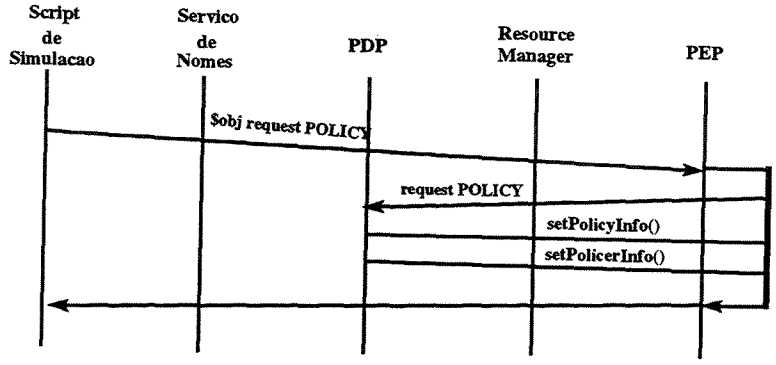

Figura 5.9: Cenário de utilização dos objetos PEP

5.4 Estudos de casos

A validação da implementação apresentada no início deste capítulo (Seção 5.2) foi realizada através de experimentos no simulador NS. Estes experimentos foram modelados para simular cenários com diferentes tipos de tráfego que possuem requisitos específicos de qualidade de serviço. Basicamente, foram modelados fluxos de tráfego de voz, que possuem exigências de baixo atraso, *jitter* e descarte. Ainda, inserimos diversos tráfegos CBR utilizando UDP/IP, para impor congestionamento na infra-estrutura de rede.

A topologia utilizada é apresentada na Figura 5.10. Os roteadores de núcleo estão numerados de 0 a 6. Todos os enlaces possuem banda de 8,64 Mbps, com atraso médio de propagação de 2 ms. O tamanho da fila foi definido para 1500 pacotes. No restante deste texto, a referência ao enlace entre dois roteadores será feita como *enlace(n1n2)*, representando o enlace entre os nós n1 e n2.

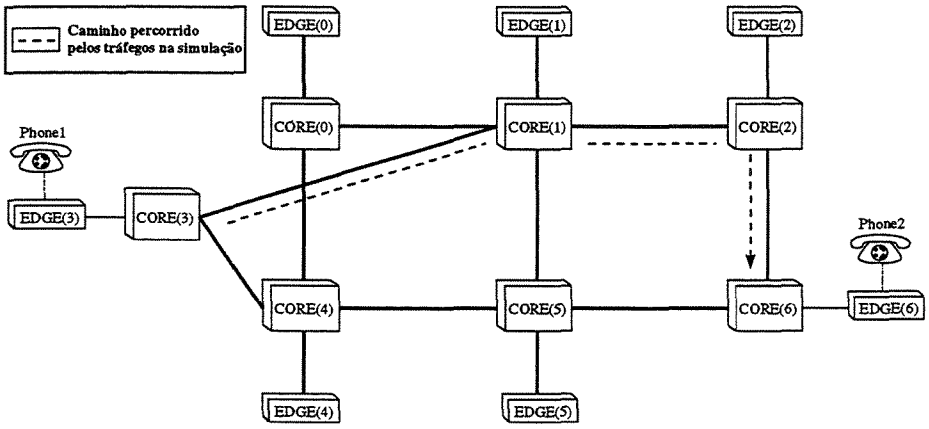

Figura 5.10: Topologia adotada na simulação

Modelagem das Fontes de Tráfego

A simulação das fontes de tráfego seguiu o seguinte critério:

- **Tráfego de voz**

Estas fontes utilizam UDP/IP. O tamanho dos pacotes se constitui de 66 bytes, *throughput* de 50 pacotes/seg. Foram criados 115 fluxos de chamadas, o que fornece uma carga de throughtput de $\simeq$3,03 Mbps. Este valor representa uma utilização de 35% dos enlaces de 8,64 Mbps.

O tráfego das chamadas telefônicas é simulado através da utilização de fontes CBR (*Constant Bit Rate*), disponíveis no simulador NS. A inserção do tráfego de voz é realizada

em diferentes instantes do tempo de simulação. As fontes obedecem uma distribuição exponencial média de 20 $ms/115$. A duração das chamadas é uma distribuição exponencial de média igual a 50s.

A inserção das fontes é iniciada no instante 20s. Cada uma dessas fontes obedece uma distribuição exponencial de média 0.1, com tempo de reativação seguindo uma distribuição exponencial de média 5s, caso a fonte seja desativada antes do instante 120s.

- **Tráfego de fundo**

O trafégo de fundo foi inserido para simular a utilização normal de recursos de rede por tráfego de melhor-esforço, sendo constituído de fontes UDP/IP. A escolha de UDP se justifica para evitar o controle de congestionamento característico do protocolo TCP.

O tamanho destes pacotes é de 512 bytes. *Duas fontes* oferecem um *throughput* médio de 2.16Mbps, o que representa uma utilização de 25% dos enlaces, além de uma *terceira fonte* com *throughput* médio de 1.728 Mpbs. Este valor representa uma utilização de 20% dos enlaces. Estes tráfegos são gerados através da utilização de fontes CBR. Os três fluxos de tráfego de fundo competem por recursos com o tráfego de voz.

- **Tráfego de interferência**

A presença do tráfego de interferência teve por objetivo inserir uma perturbação no acesso à rede, para simular situações de congestionamento. A fonte também faz uso do protocolo UDP, com comportamento CBR. O tamanho dos pacotes utilizado é de 512 bytes, com *throughput* de 1,296 Mbps (15% de ocupação do enlace).

Foram selecionados dois fluxos do tráfego de fundo para **tratamento preferencial**, sendo atendidos pelos mecanismos de qualidade de serviço nos momentos de congestionamento. O terceiro fluxo de tráfego de fundo e todo o tráfego de interferência continuam mapeados para o melhor-esforço. O objetivo foi simular a presença de tráfego com algum nível de exigência de qualidade de serviço, característico do PHB AF [12].

A obtenção dos gráficos dos resultados somente leva em consideração os tráfegos de fundo.

Cenário dos experimentos

Todo tráfego gerado segue o trajeto entre os enlaces {**enlace31**, **enlace12**, **enlace26**}. A simulação foi realizada para um tempo total de 150s. As seguintes etapas ocorrem na inicialização dos experimentos:

- instanciação dos objetos de simulação relacionados à construção da topologia e das fontes de tráfego;

- instanciação dos objetos de controle dos enlaces. Estes objetos correspondem à implementação descrita na Seção 5.2;
- inicialização da simulação. A partir deste momento, os objetos de simulação passam a monitorar o estado dos enlaces, reagindo de acordo com a política adotada.

Os enlaces monitoram, a cada ciclo de 1s, o descarte de pacotes e o atraso percebido pelo fluxo. Os parâmetros de atraso e de descarte possuem valores máximos que não devem ser ultrapassados. A sinalização do PDP ocorre nos momentos em que os valores são excedidos.

A sinalização é realizada através dos métodos *alert_delay* e *alertLinkDrop* da interface do PDP, respectivamente para o atraso e o descarte. A cada evento de sinalização, o PDP registra qual PEP efetuou o evento, armazenando um histórico de quais enlaces estão sofrendo congestionamento. Ao final de cada ciclo, os parâmetros de controle são verificados. Caso estejam acima do valor permitido, o PDP atua nas configurações dos PEPs que sinalizaram.

A ação realizada consiste em analisar o histórico armazenado e reconfigurar os parâmetros de escalonamento das filas dos roteadores. As informações do histórico contêm a classe de serviço, o identificador do PEP e o valor do descarte ou do atraso sofrido.

Apenas duas classes de serviço foram consideradas neste trabalho. O tráfego de voz, por ter requisitos de qualidade de serviço rígidos em relação ao atraso dos pacotes (**prioridade quantitativa**), possui tratamento de alta prioridade nos roteadores. Uma taxa alta de atraso pode inviabilizar a comunicação. Apesar de não estarem sendo implementados os mecanismos do PHB EF [11] neste trabalho, as características do tráfego de voz justificariam o mapeamento para a classe EF.

Os dois fluxos do tráfego de fundo selecionados como preferenciais, possuem requisitos de **prioridade qualitativa**, exigindo um nível relativo de qualidade de serviço (baixas taxas de atraso e de descarte de pacotes). O PHB AF seria a representação da classe deste tráfego, caso estes mecanismos estivessem sendo implementados seguindo a especificação contida em [12]. O terceiro fluxo do tráfego de fundo e todo o tráfego de interferência continuam sendo atendidos pelo **melhor-esforço** da infra-estrutura de rede.

Os experimentos utilizaram diferentes configurações da topologia. Uma configuração explora um modelo tradicional, sem utilização de nenhum mecanismo de qualidade de serviço (utilizando filas *FIFO*), caracterizando serviço de melhor-esforço. Outra configuração adota o algoritmo de escalonamento WRR (*Weighted Round Robin*) para todos os roteadores, com o peso inicial de cada fila sendo 1. A última configuração é uma variação da anterior, porém adotando as alterações apresentadas na Seção 5. O mapeamento de cada tipo de tráfego na segunda e na terceira configuração é feito em três filas de encaminhamento do algoritmo WRR. Estas filas são do tipo *DROP* [39].

Obtenção e análise dos resultados

A **política de atuação** consiste em incrementar o peso de escalonamento das filas, de acordo com a prioridade de PHB no qual o tráfego se identifica. A Figura 5.11 mostra o descarte ocorrido no tráfego de voz nos três experimentos realizados. O experimento em melhor-esforço impõe um descarte de até 40% no tráfego de voz, enquanto que a aplicação do algoritmo WRR, com configuração inicial (peso 1 para todas as filas) impõe até 60% de perda.

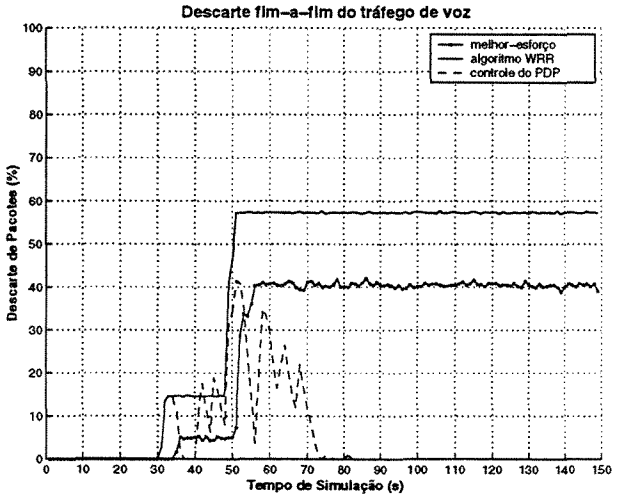

Figura 5.11: Descarte experimentado por tráfego de voz

Percebe-se o alto índice de descarte sofrido pelo tráfego de voz na seleção do mecanismo WRR. Isto resulta da operação deste algoritmo, que seleciona uma quantidade de pacotes de cada fila igual ao peso atribuído.

O gráfico da Figura 5.12 mostra o comportamento das taxas de atraso em cada um dos experimentos realizados. O experimento de melhor-esforço obteve um atraso fim-a-fim de 800ms, mantendo uma taxa de descarte de 40% (Figura 5.11). A aplicação do algoritmo WRR reduz o atraso para 500ms. Entretanto, esta redução possui bastante influência da alta taxa de descarte de 60% do tráfego de voz (Figura 5.11).

A política de controle adotada estabeleceu que o atraso do tráfego de voz não deveria ser superior a 50ms. Ainda, o descarte máximo de pacotes não deveria ser superior a 1%. O descarte máximo de pacotes para o tráfego preferencial foi configurado para 5%. Estas restrições estão consistentes com o fato de que o atraso em comunicação de voz acarreta maior prejuízo do que o próprio descarte e que o tráfego preferencial espera um serviço de qualidade superior do que o melhor-esforço.

Considerando o tipo de tráfego, foi possível manter o atraso sob controle e ainda assim minimizar o descarte de pacotes, para o tráfego de voz. A aplicação do controle

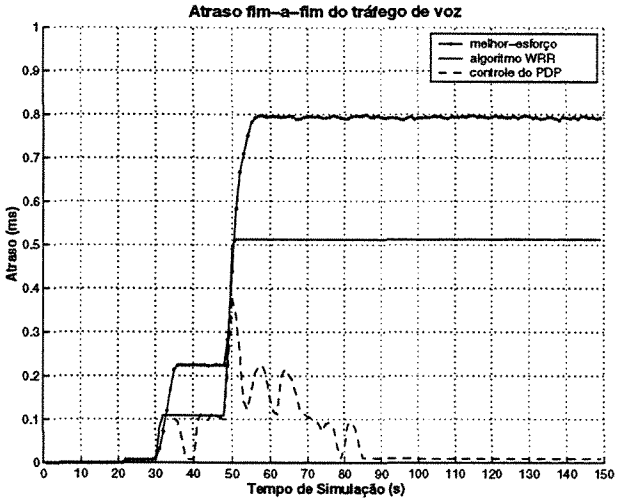

Figura 5.12: Atraso experimentado por tráfego de voz

ofereceu um excelente ganho de desempenho, através do ajuste dos pesos das filas. A curva **controle do PDP** na Figura 5.11 e na Figura 5.12 mostra o ganho em relação ao descarte e ao atraso, respectivamente.

A Figura 5.13 mostra o valor de descarte para o tráfego preferencial em um contexto de melhor-esforço. Este experimento obteve uma taxa de descarte dentro do parâmetro máximo permitido para este tipo de tráfego, mantendo uma média de 7%. O experimento com a aplicação do algoritmo WRR não registrou nenhum descarte de pacotes.

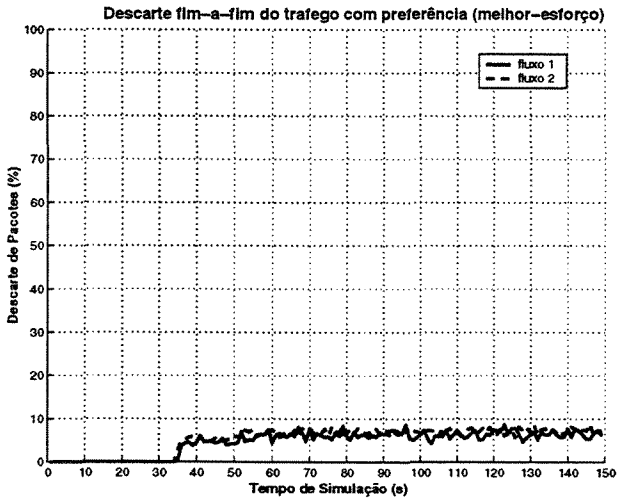

Figura 5.13: Descarte do tráfego preferencial (sem qualidade de serviço)

As políticas de controle estão inter-relacionadas, sendo que o ajuste em relação ao descarte ou ao atraso influenciam entre si. As sinalizações ocorrem na medida em que

os valores máximos são alcançados. Estes eventos são assíncronos, podendo ocorrer em qualquer instante da simulação. Entretanto, a atuação do Bandwidth Broker somente é realizada ao final de cada ciclo de avaliação, momento no qual é conhecida a situação de congestionamento de cada enlace.

A atuação por uma sinalização de descarte ou por atraso pode influenciar o congestionamento percebido por um outro tráfego, que por sua vez sinaliza em resposta ao novo contexto. Esta interação ocorre até que os pesos das filas estejam ajustados para atender os requisitos dos tráfegos sendo priorizados, estabilizando o sistema. A visualização da interação pode ser percebida na análise do gráfico apresentado para o experimento de controle com incremento para o tráfego preferencial, na Figura 5.14.

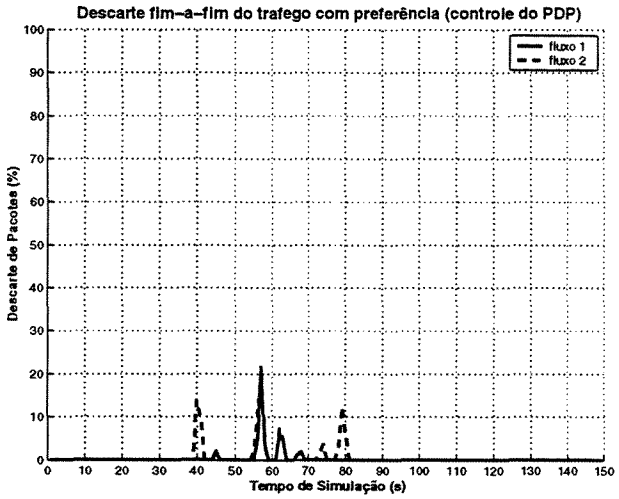

Figura 5.14: Descarte do tráfego preferencial (controle do PDP)

Neste gráfico, os descartes ocorrem na medida em que é oferecida maior banda para o tráfego de voz. Uma vez que o incremento do peso da fila de voz implica na diminuição da banda do tráfego preferencial, este começa a experimentar descarte de pacotes. Esta situação faz com que ocorra uma nova sinalização por parte dos tráfegos preferenciais, aumentando sua largura de banda. O sistema se torna estável a partir do instante 80 do tempo de simulação.

O valor final do peso de escalonamento de cada fila determina a alocação de banda destinada para o tráfego. Entretanto, uma consideração tem de ser realizada em relação à ocupação dos *buffers* dos enlaces, que justifica o prejuízo no desempenho do descarte de pacotes de voz na aplicação do mecanismo WRR (Figura 5.11).

Conforme apresentado, os pacotes de tráfego de voz possuem um tamanho de 66 bytes, enquanto que o tráfego preferencial possui pacotes de tamanho 512 bytes. A proporção de tamanho destes pacotes é aproximadamente 1/8. Esta relação implica que, para o mesmo

peso de escalonamento inicial (peso 1), o tráfego preferencial ocupa 8 vezes mais recurso de *buffer* nas filas do que o tráfego de voz. Esta proporção tem de ser considerada no aumento de prioridade para os pacotes de voz.

A utilização do controle desativado reflete o cenário em que o escalonamento entre as filas garante uma prioridade mínima, uma vez que o serviço não é mais melhor-esforço. A desativação significa que todas as filas têm a mesma prioridade, operando como um escalonamento *round-robin*. Esta configuração garante 1/3 da banda para cada fila, dada a utilização de 3 filas.

Considerando a proporção do tamanho dos pacotes se verifica que a banda efetiva para o tráfego de voz é de 1/17 ou $\simeq$5,88%, enquanto cada tráfego preferencial obtém uma parcela de 8/17 ou $\simeq$47,05%. Uma vez que foram inseridos dois tráfegos preferenciais, sua soma ocupa uma parcela de $\simeq$94,11%.

Esta alocação indevida representa um cenário interessante, no qual a seleção de um mecanismo de priorização não garante um ganho efetivo de desempenho. A injustiça introduzida pelo algoritmo WRR pode ser resolvida com nossa abordagem de gerência, no qual a presença do Bandwidth Broker detecta e corrige esta situação.

Uma vez que o Bandwidth Broker possui as informações necessárias sobre o tipo de tráfego a ser controlado, as decisões de gerência são tomadas levando ainda em consideração as prioridades de cada fila e as políticas de atuação definidas para cada tráfego.

Conforme citado anteriormente, a atuação foi realizada através do incremento dos pesos de cada fila mapeada pelo algoritmo WRR. A curva denominada **controle do PDP** mostra o comportamento no ajuste da prioridade de cada fila, para os tráfegos de voz e preferencial.

Os gráficos apresentados nas Figuras 5.15 e 5.16 mostram a relação existente entre o tráfego de voz e preferencial. Conforme citado anteriormente, o tráfego preferencial se identifica com o PHB AF. As curvas são denominadas AFx, sendo x o peso de escalonamento, variando de 1 a 5.

O gráfico da Figura 5.15 representa o ganho de largura de banda para o tráfego de voz, em função do peso de escalonamento da fila para o tráfego preferencial. A proporção de pacotes é de 1 para 8 e esta relação vale para a utilização do algoritmo WRR neste experimento. Sendo $p(\kappa)$ o peso do tráfego do tipo κ, com $\kappa \in K = \{$voz, preferencial, melhor-esforco$\}$ e sendo $\mathcal{B}_{voz}$ a largura de banda obtida pelo tráfego de voz, então: $\mathcal{B}_{voz} = \frac{p(voz)}{\sum_{\kappa \in K} p(\kappa)}$

Esta relação entre os pesos é válida para a obtenção de largura de banda para os dois tipos de tráfego. A Figura 5.16 mostra a largura de banda obtida para os tráfegos em função da variação do peso de escalonamento do PHB AF. São apresentadas as curvas para os pesos de 1 a 5. No exemplo, a obtenção de 35% de largura de banda para o tráfego de voz e de 50% para o tráfego preferencial implica em estabilizar o peso do PHB AF em

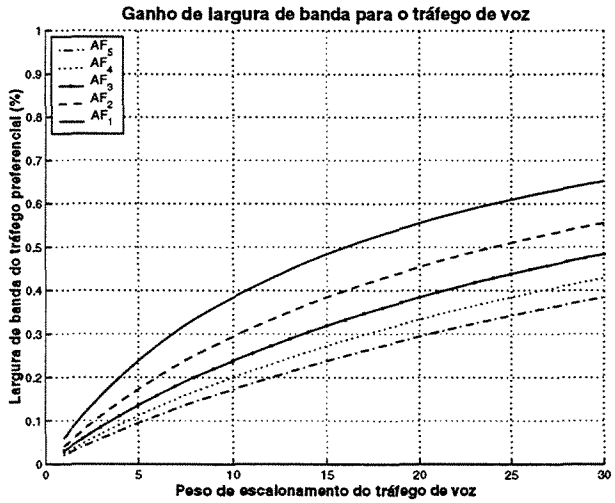

Figura 5.15: Ganho de largura de banda para o tráfego de voz

4, considerando a proporção de 1 para 8 no tamanho dos pacotes.

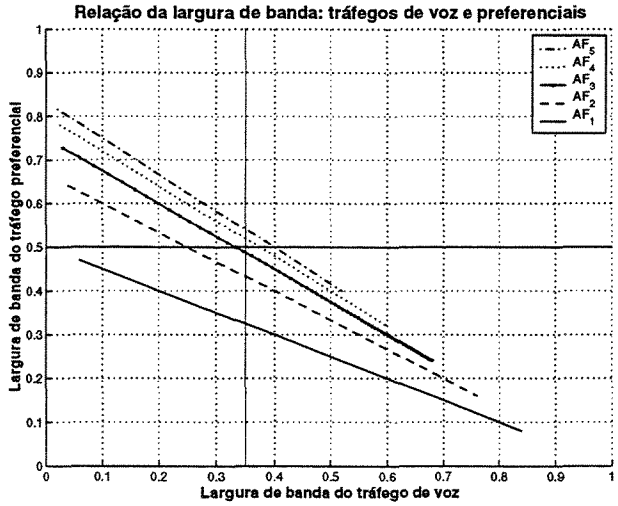

Figura 5.16: Ganho de largura de banda em função do peso da fila do tráfego preferencial

A estabilidade do sistema em relação aos parâmetros de descarte e atraso implica na diminuição da largura de banda para o tráfego de melhor-esforço. O gráfico da Figura 5.17 mostra o prejuízo de descarte imposto ao tráfego de melhor-esforço em função dos ajustes dos pesos das filas. Uma vez que o peso da fila de melhor-esforço não é incrementado, este perde uma grande parcela da banda, obtendo uma alta taxa de descarte. Este cenário é aceitável, uma vez que para o serviço de melhor-esforço nenhuma garantia de qualidade de serviço é realizada.

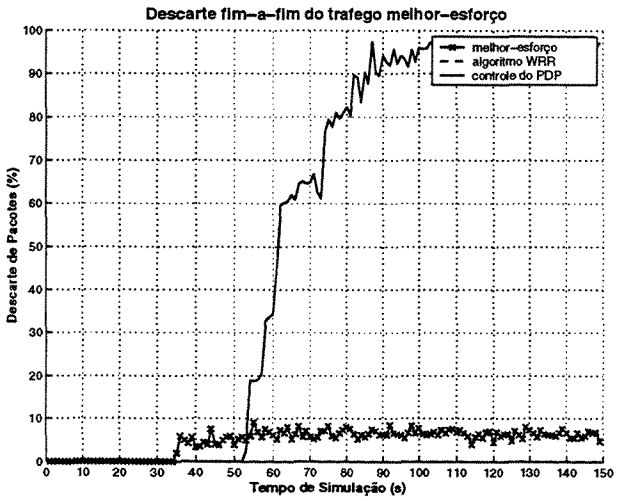

Figura 5.17: Descarte do tráfego de melhor-esforço para todos os experimentos

Capítulo 6

Conclusão

A separação do controle de qualidade de serviço em camadas de atuação modulariza a tarefa geral de obter diferenciação de serviços. As camadas oferecem a possibilidade de atuação com diversas ferramentas já existentes, o que permite combinar estratégias já consolidadas.

As arquiteturas de qualidade de serviço discutidas no âmbito do IETF (Serviços Integrados e Serviços Diferenciados) oferecem um contexto interessante para um oferecimento mais sistemático de QoS. Através da padronização de várias abordagens e mecanismos para o tratamento e gerência do tráfego, a interoperabilidade de diferentes implementações tende a ter maiores chances de sucesso, algo imprescindível para obter garantia do comportamento da rede.

A arquitetura de Serviços Diferenciados tem merecido uma maior atenção devido a maior escalabilidade, apresentando-se como um potencial componente básico de qualquer equipamento que ofereça capacidade de priorizar tráfego.

A utilização de políticas no controle dos recursos é um grande passo rumo a automatização das tarefas de gerenciamento. A capacidade de expressar em um nível mais alto o comportamento da infra-estrutura de rede sugere a possibilidade do desenvolvimento e da utilização de estratégias de controle nesta área de gerência de qualidade de serviço.

Esta visão é substancialmente diferente da gerência tradicional, pois busca mapear os modelos de negócios, relacionando-os ao uso da rede através de uma abordagem direta, que procura minimizar a intervenção humana no mapeamento destes modelos para a configuração dos dispositivos de rede.

Outras estratégias de qualidade de serviço também podem tirar proveito da abordagem de políticas. A engenharia de tráfego é o próximo componente a ser integrado, pois oferece um nível ainda maior de controle sobre a utilização de recursos da rede.

O modelo de desenvolvimento apresentado neste trabalho contribui na identificação dos componentes básicos de uma hierarquia de gerenciamento. Estes componentes realizam

as funções mínimas para implementar o modelo de gerência em camadas. Através da definição da interface destes componentes, foi possível implementar um protótipo que valida a estratégia de utilização de políticas no controle de utilização dos recursos de rede.

A distribuição dos componentes do modelo de desenvolvimento permitiu modelar separadamente a funcionalidade de alto-nível (controle) e a funcionalidade de encaminhamento, relacionada ao nível físico. A contribuição está no desenvolvimento de um protótipo em um ambiente de simulação que valida esta proposta de uso de políticas.

Os experimentos realizados neste trabalho mostraram as vantagens de se combinar estratégias de controle, no encaminhamento de pacotes, com um controle centralizado, que distribui as informações de configuração de acordo com o desempenho em diferentes pontos da rede.

O esforço de desenvolvimento esteve direcionado para validar a utilização de um Bandwidth Broker, mas com a preocupação de manter o máximo de independência do nível de elemento de rede. A decisão por utilizar CORBA na comunicação entre os componentes permite a migração deste trabalho para uma estrutura física real, na qual são reaproveitadas as políticas e estratégias avaliadas no simulador.

A continuidade deste trabalho está relacionada justamente à realização desta migração. Uma possibilidade é incrementar a interface dos componentes, permitindo integrar o controle com soluções de meio físico como a arquitetura ALTQ. Esta é uma proposta bastante viável, devido a presença de uma API de gerenciamento no ALTQ, oferecida através de bibliotecas em nível de usuário.

As estratégias de gerência de filas, utilizando algoritmos como o RED e RIO não foram abordadas neste trabalho. Estes algoritmos permitem um controle mais refinado da ocupação dos *buffers*. As informações de configuração destes algoritmos (gatilhos de ativação, valores das probabilidades de descarte) podem ser manipuladas pela estratégia de políticas. Esta abordagem poderia permitir que os parâmetros de configuração considerassem não somente o estado atual da fila sendo gerenciada, mas talvez o comportamento das filas anteriores no caminho de transmissão. A avaliação deste cenário pode vir a ser um trabalho futuro.

Existe ainda um vasto campo de exploração de novas políticas de controle, que podem ser incorporadas na simulação, para validar novos cenários. Foram utilizadas neste trabalho estratégias simples de monitoração e atuação, que atenderam ao objetivo de validação do protótipo. Algumas propostas de formalização da representação de políticas têm sido apresentadas no âmbito do IETF [14, 43], o que demonstra o interesse neste tipo de controle. O estudo da aplicação de novas políticas é uma das linhas mais interessantes de continuidade deste trabalho.

A integração da tecnologia MPLS com o Bandwidth Broker, utilizando técnicas de

engenharia de tráfego, constitui outra linha de trabalho. As técnicas de engenharia de tráfego oferecem um nível a mais de gerência. A pesquisa da integração destas técnicas, aliadas ao gerencimento de priorização de tráfego, sugerem que o controle pode ser realizado em vários níveis de granularidade. Adicionando a capacidade de controle sobre a gerência das filas de encaminhamento, tem-se um contexto bastante interessante, que comporta todos os níveis onde a qualidade de serviço pode atuar.

Referências

[1] S. Blake e D. Black e M. Carlson e E. Davies e Z. Wang e W. Weiss. An Architecture for Differentiated Services. RFC 2475, Dezembro 1998.

[2] K. Nichols e V. Jacobson e L. Zhang. A Two-Bit Differentiated Services Architecture for the Internet. internet-draft, Dezembro 1997.

[3] Vern Paxson e Sally Floyd. Why We Don't Know How To Simulate The Internet. Proceedings of the 1997 Winter Simulation Conference.

[4] Internet Engineering Task Force. http://www.ietf.org.

[5] Sally Floyd e V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking 1, n. 4*, 1993.

[6] S. Shenker R. Braden, D. Clark. Integrated Services in the Internet Architecture: An Overview. RFC 1633, Junho 1994.

[7] R. Braden e L. Zhang e S. Berson e S. Herzog e S. Jamin. Resource reSerVation Protocol (RSVP) - Version 1 Functional Specification. RFC 2205, Setembro 1997.

[8] J. Wroclawski. The Use of RSVP With Integrated Services. RFC 2210, Setembro 1997.

[9] P. Almquist. Type of Service in the Internet Protocol Suite. RFC 1349, Julho 1992.

[10] Xipeng Xiao e Lionel Ni. Internet Qos: A Big Picture. IEEE Network Magazine, Março 1999.

[11] V. Jacobson e K. Nichols e K. Poduri. An Expedited Forwarding PHB. RFC 2598, Junho 1999.

[12] J. Heinanen e F. Baker e W. Weiss e J. Wroclawski. Assured Forwarding PHB Group, Junho 1999.

[13] Rosen E. e A. Viswanathan e R. Callon. A Proposed Architecture for MPLS. draft-ietf-mpls-arch-06.txt, Agosto 1999.

[14] B. Moore e E. Ellensson e J. Strassner e A. Westerinen. Policy Core Information Model - Version 1 Specification. draft-ietf-policy-core-info-model-08.txt, Julho 2000.

[15] Inc. Distributed Management Task Force. Common Information Model (CIM) Specification, version 2.2. http://www.dmtf.org/spec/cims.html, Junho 1999.

[16] Inc. Distributed Management Task Force. DMTF Technologies: CIM Standards. http://www.dmtf.org.

[17] Alistair Crooll e Eric Packman. *Managing Bandwidth. Deploying QoS in Enterprise Networks.* Prentice Hall, 2000.

[18] Boyle J. e Cohen R. e Durham D. e Herzog S. e Raja R. e Sastry A. The COPS (Common Open Policy Service) Protocol. RFC 2748, Janeiro 2000.

[19] Kwok Ho Chan e David Durham e Silvano Gai e Shai Herzog e Keith McCloghrie e Francis Reichmeyer e John Seligson e Andrew Smith e Raj Yavatkar. Cops Usage for Policy Provisioning (COPS-PR). draft-ietf-rap-pr-04.txt, Agosto 2000.

[20] F. Reichmeyer e S. Herzog e K. Chan e D. Durham e R.Yavatar e S. Gai e K. McCloghrie e A. Smith. Cops Usage for Policy Provisioning. draft-ietf-rap-pr-00.txt, Junho 1999.

[21] M. Fine e K. McCloghrie e J. Seligson e K. Chan e S. Hahn e A. Smith. Quality of Service Policy Information Base. draft-ietf-mfine-cops-pib-01.txt, Junho 1999.

[22] Service Level Specification Semantics and Parameters. draft-tequila-sls-00.txt, Novembro 2000.

[23] Service Level Specification Semantics, Parameters e Negotiation Requirements. draft-tequila-diffserv-sls-00.txt, Julho 2000.

[24] Daniel O. Awduche e Angela Chiu e Anwar Elwalid e Indra Widjaja e Xipeng Xiao. A Framework for Internet Traffic Engineering. internet-draft, Janeiro 2000.

[25] Technology Center. Group for Advanced Information Technology. *Bandwidth Broker Sytem Specification.* British Columbia Institute of Technology, Outubro 1998. Revisão 1.1.

[26] Rob Neilson e Jeff Wheeler e Francis Reichmeyer e Susas Hares. A Discussion of Bandwidth Broker Requirements for Internet2 Qbone Deployment. Technical report, Internet2 Qbone BB Advisory Council, Agosto 1999.

[27] F. Reichmeyer e L. Ong e A. Terzis e L. Zhang. A Two-Tier Resource Management Model for Differentiated Services Networks. internet-draft, Novembro 1998. draft-rotzy-2-tier-management-00.txt.

[28] Internet2 website. http://www.internet2.edu.

[29] The Internet2 QBone Bandwidth Broker Advisory Council. Qbone architecture (v1.0). Internet2 QoS Working Group Draft, Agosto 1999. http://www.internet2.edu/qos/wg/papers/qbArch/.

[30] QBone Bandwidth Broker Work Group. Qbone Bandwidth Broker Architecture, Junho 2000. http://qbone.internet2.edu/bb/bboutline2.html.

[31] QBBAC. The Internet2 Qbone Bandwidth Broker Advisory Council. http://www.internet2.edu/qos/qbone/QBBAC-0107.shtml.

[32] Munich Siemens Ag. The Siemens Bandwidth Broker. Proceedings of the First Joint Internet2 / DOE QoS Workshop, Fevereiro 2000.

[33] Andreas Terzis. The Role of BB in Traffic Engineering. Proceedings of the First Joint Internet2 / DOE QoS Workshop.

[34] J. Sermersheim. Lightweight Directory Access Protocol (v3). draft-ietf-ldapbis-protocol-00.txt, Janeiro 2001.

[35] The VINT Project. http://www.isi.edu/nsnam/ns.

[36] Lee Breslau e Deborah Estrin e Kevin Fall e Sally Floyd e John Heidemann e Ahmed Helmy e Polly Huang e Steven McCanne e Kannan Varadhan e Ya Xu e Haobo Yu. Advances in network simulation. IEEE Computer, Maio 2000.

[37] D. Wetherall e C. J. Linblad. Proc. Usenix Tcl/TK Workshop, Usenix, Berkeley, Califórnia, 1995.

[38] J. K. Ousterhout. *Tcl and Tk Toolkit.* Addison-Wesley, 1994.

[39] Mandeep Baines Farhan Shallwani Peter Pieda, Jeremy Ethridge. A Network Simulator Differentiated Services Implementation. Open IP, Nortel Networks, Julho 2000.

[40] Frank Pilhofer. Combat. http://www.informatik.uni-frankfurt.de/ fp/Tcl/, Maio 2000.

[41] Michael J. McLennan. [incr tcl] - version 3.0 for Tcl/Tk 8.0.3. http://www.tcltk.com/itcl/.

[42] Object Management Group. http://www.omg.org.

[43] Y. Snir e Y. Ramberg e J. Strassner e R. Cohen. Policy Framework QoS Information Model. draft-ietf-policy-qos-info-model-06.txt, Abril 2000.