

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE CIÊNCIAS DA COMPUTAÇÃO

**UMA PROPOSTA PARA PROVER QUALIDADE DE
SERVIÇO EM REDES IP**

**ÂNGELA MARIA ERDTMANN
WALTER FÉLIX CARDOSO NETO**

ORIENTADOR:

Prof. Dr. Carlos Becker Westphall

BANCA EXAMINADORA:

Prof. Fernando Cerruti

Prof^a Dr. Carla Merkle Westphall

Fevereiro de 2003

ÍNDICE

RESUMO	2
ABSTRATCT	3
CAPÍTULO 1 – INTRODUÇÃO, OBJETIVOS E ORGANIZAÇÃO	1
1.1 INTRODUÇÃO	1
1.1.1 UM ROTEADOR GENÉRICO.....	4
1.2 OBJETIVOS	8
1.2.1 - OBJETIVO ESPECÍFICO	8
1.3 ORGANIZAÇÃO	9
CAPÍTULO 2 - CONCEITOS E ARQUITETURAS DE QUALIDADE DE SERVIÇO .	10
2.1 QUALIDADE DE SERVIÇO	10
2.2 DEFINIÇÕES DE USO COMUM	15
2.3 MÉTRICAS DE QUALIDADE DE SERVIÇO	15
2.3.1 VAZÃO (THROUGHPUT).....	16
2.3.2 LATÊNCIA(ATRASO FIM-A-FIM)	17
2.3.3 JITTER(VARIAÇÃO DE ATRASO)	18
2.3.4 PERDA DE PACOTES	18
2.4 DISCIPLINAS DE FILA	19
2.4.1 FIFO	21
2.4.2 PRR	22
2.4.3 WRR.....	23
2.4.4 WFQ.....	26
2.5 MECANISMOS DE CONTROLE DE CONGESTIONAMENTO	29
2.5.1 CONTROLE DE CONGESTIONAMENTO DO TCP	29
2.5.2 GERENCIAMENTO ATIVO DE FILAS E O RED	35
2.5.3 VARIAÇÕES DO RED.....	37

2.6 CONTROLE DE ADMISSÃO, ADEQUAÇÃO E POLÍTICAS DE TRÁFEGO	39
2.6.1 TOKEN BUCKET	39
2.6.2 ADEQUAÇÃO DE TRÁFEGO	42
2.6.3 POLÍTICA DE TRÁFEGO	43
2.6.4 CONTROLE DE ADMISSÃO	44
2.7 SERVIÇOS INTEGRADOS	45
2.7.1 CONCEITOS DA ARQUITETURA DE SERVIÇOS INTEGRADOS	46
2.7.2 COMPONENTES DOS ELEMENTOS DE REDE	47
2.7.3 O PROTOCOLO DE RESERVA DE RECURSOS (RSVP)	50
2.7.4 AS MENSAGENS RSVP	50
2.7.5 OBJETOS DO PROTOCOLO RSVP ESPECIFICADOS PARA O USO COM A ARQUITETURA DE SERVIÇOS INTEGRADOS	51
2.7.6 O PROCESSO DE RESERVA DE RECURSOS	52
2.7.7 SERVIÇO GARANTIDO	54
2.7.8 SERVIÇO DE CARGA CONTROLADA	56
2.7.9 CONSIDERAÇÕES SOBRE A ARQUITETURA DE SERVIÇOS INTEGRADOS	57
2.8 SERVIÇOS DIFERENCIADOS	57
2.8.1 CONCEITOS DA ARQUITETURA DE SERVIÇOS DIFERENCIADOS	59
2.8.2 CLASSIFICAÇÃO E CONDICIONAMENTO DO TRÁFEGO	62
2.8.3 COMPORTAMENTOS-POR-SALTO - PHB	66
2.8.4 COMPORTAMENTO-POR-SALTO DEFAULT	67
2.8.5 GRUPO DE COMPORTAMENTOS-POR-SALTO CLASS SELECTOR	67
2.8.6 GRUPO DE COMPORTAMENTOS-POR-SALTO DE ENCAMINHAMENTO ASSEGURADO (AF PHB)	68
2.8.7 COMPORTAMENTO-POR-SALTO DE ENCAMINHAMENTO EXPRESSO (EF PHB)	70
2.8.8 CONSIDERAÇÕES SOBRE A ARQUITETURA DE SERVIÇOS DIFERENCIADOS	71
2.9 INTEROPERAÇÃO ENTRE OS SERVIÇOS DIFERENCIADOS E OS SERVIÇOS INTEGRADOS	72
2.9.1 DESCRIÇÃO DA ARQUITETURA HÍBRIDA	73
2.9.2 SUPORTE AO RSVP	77
2.9.3 EXEMPLO DE FUNCIONAMENTO DA ARQUITETURA HÍBRIDA	78

2.9.4 IMPLEMENTAÇÃO DO SERVIÇO DE CARGA CONTROLADA NA ARQUITETURA HÍBRIDA.....	81
2.9.5 IMPLEMENTAÇÃO DO SERVIÇO GARANTIDO	84
2.9.6 CONSIDERAÇÕES SOBRE A ARQUITETURA HÍBRIDA	86
CAPÍTULO 3 – MODELO DE SIMULAÇÃO	88
3.1 DESCRIÇÃO DO MODELO DE SIMULAÇÃO.....	88
3.2 IMPLEMENTAÇÃO DO MODELO DE SIMULAÇÃO	92
3.3 TRÁFEGO SIMULADO	94
3.4 VERIFICAÇÃO DO MODELO DE SIMULAÇÃO	95
3.5 VALIDAÇÃO DO MODELO DE SIMULAÇÃO.....	96
CAPÍTULO 4 – RESULTADOS DO MODELO DE SIMULAÇÃO	97
4.1. SUPORTE A FLUXOS DE CARGA CONTROLADA EM UMA CLASSE DE ENCAMINHAMENTO ASSEGURADO	97
4.1.1. INFLUÊNCIA DO NÚMERO DE MICROFLUXOS TCP E DA TAXA DE REPOSIÇÃO DE PACOTES.....	97
4.1.2 GRANULOSIDADE DE POLÍTICAS DE TRÁFEGO E TRÁFEGO TCP	100
4.1.3 GRANULOSIDADE DAS POLÍTICAS DE TRÁFEGO E TRÁFEGO UDP	102
4.2 DIFERENCIAÇÃO UTILIZANDO O GERENCIAMENTO ATIVO DE FILAS.....	105
4.3 ESCALONAMENTO ENTRE CLASSES DE TRÁFEGO	107
4.3.1. TRÁFEGO TCP	108
4.3.2. TRÁFEGO UDP.....	111
CAPÍTULO 5 - TRABALHOS FUTUROS E CONCLUSÕES	114
5.1. TRABALHOS FUTUROS	114
5.2. CONCLUSÕES	115
BIBLIOGRAFIA.....	126

Índice de Figuras

<i>Figura 1: Comparação entre as granulosidades de tratamento de tráfego das arquiteturas de qualidade de serviço</i>	4
<i>Figura 2: Funções do roteador</i>	5
<i>Figura 3: Fases do roteador</i>	7
<i>Figura 4: Critérios para QoS</i>	11
<i>Figura 5: Disciplina de fila FIFO</i>	22
<i>Figura 6: Disciplina de fila PRRJ</i>	23
<i>Figura 7: Disciplina de fila WRR</i>	25
<i>Figura 8: Disciplina de fila WFQ</i>	27
<i>Figura 9: Algoritmo de janelas deslizantes do TCP para o transmissor</i>	30
<i>Figura 10: Comportamento do throughput de um microfluxo TCP com Slow Start e Congestion Avoidance</i>	34
<i>Figura 11: Variação da probabilidade de Descarte de Pacotes e tamanho médio da fila no RED</i>	37
<i>Figura 12: O regulador de Tráfego Token Bucket</i>	41
<i>Figura 13: Modelo de Referência de implementação de um Elemento QoS capaz</i>	49
<i>Figura 14: Reserva de Recursos Unicast em uma rede com serviços diferenciados</i>	53
<i>Figura 15: Sintaxes do Campo DS e do campo de tipo de serviço do protocolo IPv4</i>	59
<i>Figura 16: A arquitetura de serviços diferenciados</i>	61
<i>Figura 17: Representação das funções de Classificação e Condicionamento de Tráfego</i>	63
<i>Figura 18: WRED configurado para suportar as classes de serviço AF</i>	70
<i>Figura 19: Ilustração de funcionamento da arquitetura híbrida</i>	73
<i>Figura 20: Modelo simplificado da arquitetura híbrida IntServ/DiffServ</i>	75
<i>Figura 21: Reserva de Recursos Unicast na arquitetura híbrida</i>	80

<i>Figura 22: Topologia de Simulação</i>	90
<i>Figura 23 (a) e (b): Frequência Relativa da Latência de pacotes verdes para políticas de tráfego com taxa de reposição de pacotes de 500000 bps</i>	100
<i>Figura 24(a) e (b): Frequência Relativa da Latência de pacotes verdes para simulações com 32 microfluxos, taxas de reposição de pacotes variadas e políticas de tráfego</i>	100
<i>Figura 25: Throughput Médio por fluxo versus Latência Física para 90% de banda passante reservada e política de tráfego com perfil por fluxo</i>	102
<i>Figura 26(a) e (b): Throughput Médio obtido por microfluxos UDP de acordo com variações na quantidade de microfluxos em um mesmo perfil de tráfego</i>	103
<i>Figura 27: Frequência Relativa Acumulada da Latência para simulações com e sem tráfego de melhor esforço</i>	107
<i>Figura 28: Distribuição Relativa da Latência para 90% de Banda Passante Reservada</i>	110
<i>Figura 29: Gráficos da Latência Média para a disciplina de filas</i>	112

Índice de tabelas

<i>Tabela 1: Throughput típico de algumas aplicações</i>	<i>17</i>
<i>Tabela 2: Requisito de QoS das aplicações.....</i>	<i>19</i>
<i>Tabela 3: Throughput médio de microfluxos TCP simulados.....</i>	<i>106</i>
<i>Tabela 4: Throughput Médio obtido pelos Microfluxos nas simulações utilizando PRR e WRR com tráfego TCPTabela.....</i>	<i>109</i>
<i>Tabela 5: Latência Média obtida pelos microfluxos nas simulações utilizando PRR e WRR com tráfego TCP</i>	<i>110</i>

Lista de abreviaturas

AF	Assured Forwarding
ATM	Asynchronous Transfer Mode
BA	Behaviour Aggregate
BE	Best-Effort
CAR	Committed Access Rate
CBR	Constant Bit Rate
Diffserv	Differentiated Services
DS	Differentiated Services
DSCP	Differentiated Services Code-Point
EF	Expedited Forwarding
FIFO	First In First Out
IETF	Internet Engineering Task Force
Intserv	Integrated Services
IP	Internet Protocol
LAN	Local Area Network
PPP	Point-to-Point Protocol
MF	Multi-field
PHB	Per-Hop Behaviour
PQ	Priority Queuing
QoS	Quality of Service
RTT	Round Trip Time
RSVP	Resource ReSerVation Protocol
TCP	Transmission Control Protocol
TOS	Type of Service
UDP	User Datagram Protocol
VoIP	Voice over IP
WAN	Wide Area Network
WFQ	Weighted Fair Queuing

RESUMO

Esse trabalho apresenta um estudo e avaliação sobre os mecanismos de implementação de Qualidade de Serviço em redes IP através de medições em um ambiente simulado. O trabalho foi dividido em três etapas: Na primeira etapa, são descritos os diversos componentes necessários ao controle de tráfego, incluindo-se métricas, políticas de tráfego e disciplinas de fila. Na segunda etapa são descritas as arquiteturas, utilizadas na implementação de políticas de Qualidade de Serviço em uma rede, que utilizam os componentes do estudo feito na primeira etapa. Primeiramente, é apresentado o modelo de serviços integrados e sua política de reserva de recursos, ao longo dos nós da rede, utilizando o protocolo RSVP, com controle de admissão e políticas de controle inerentes a ele. Em seguida, é feita a apresentação do modelo de serviços diferenciados, com a utilização do tratamento diferenciado para diversos fluxos de tráfego, previamente classificados ou marcados, para serem avaliados nos nós ao longo da rede e direcionados de acordo com o seu perfil (PHB's). Finalmente é realizado o estudo da inter-operação entre ambas arquiteturas que é o objetivo da proposta da implementação do trabalho. Na terceira parte do trabalho, foram realizados experimentos conclusivos, com base nas simulações que visam o estudo do suporte à arquitetura de serviços integrados em domínios de serviços diferenciados. As simulações focaram-se no serviço de carga controlada da arquitetura de serviços integrados e no encaminhamento assegurado da arquitetura de serviços diferenciados.

Abstract

This work presents a study and evaluation about implementation mechanisms of Quality of Service on IP networks by means of measures in a simulated environment. The study was divided in three parts. The first one describes several needed components by the traffic control, namely metrics, traffic policing, queue management. In the second part are described the main architectures used during the implementation mechanisms of Quality of Service on IP networks, which use the components already described in the first part. Firstly, is presented the integrated services model and its resource reservation policies, through the network hops, using the RSVP protocol, with admission control and policies control, intrinsic to the RSVP. On the sequence, is presented the differentiated services model, by using the differentiated treatment to several traffic flux, previously classified or marked, to be availed on the hops through the network, and driven according to its profile. Finally is studied the interoperation between the two architectures, which is the main purpose of this work. In the third part, was done conclusive experiments, based on the simulations which aim the study of the support to the integrated services architectures over differentiated services domains. The simulations focused on the controlled load service of the integrated services architecture and the assured forwarding of the differentiated services architecture.

Capítulo 1 – Introdução, Objetivos e Organização

1.1 Introdução

A arquitetura atual da Internet baseia-se em um modelo de serviço o qual tem se mantido inalterado desde seus primórdios[1]. Este modelo, conhecido como *melhor esforço* caracteriza-se pelo não fornecimento de garantias com relação à forma como a entrega de pacotes é realizada: cada pacote é entregue de forma independente do anterior ou posterior, sendo todos tratados sem distinção ao longo de sua rota pelo interior da rede. O modelo de melhor esforço possui características de simplicidade e escalabilidade, tendo se adequado ao grande crescimento da Internet.

Todavia, a crescente popularização e aumento do caráter comercial da Internet têm criado demandas por novos mecanismos e aplicações as quais não podem ser atendidas satisfatoriamente pelo modelo de melhor esforço. Dentre estes fatores podem ser citados[1][2]:

- a crescente demanda por aplicações multimídia em redes tais como voz, videoconferência e telemedicina, as quais freqüentemente possuem requisitos de qualidade que não podem ser atendidos pelo modelo de melhor esforço;
- a demanda por parte de provedores comerciais pela existência de mecanismos que permitam a diferenciação do serviço oferecido a cada cliente; e
- o aumento do uso das redes de longa de longa distância, as quais por serem mais caras, criam uma demanda por mecanismos que permitam uma melhor forma de gerenciamento do tráfego.

A fim de atender a esta demanda por qualidade de serviço (QoS)¹, modificações nos mecanismos utilizados em redes IP e, em particular, no modelo de melhor esforço estão sendo propostas. Atualmente há duas arquiteturas em estudo que especificam

¹ Veja tópico 2.1

estas modificações: a arquitetura de *serviços integrados* e a arquitetura de *serviços diferenciados* [1]. Apesar de utilizarem diferentes abordagens e mecanismos para a implementação de qualidade de serviço, ambas propõe extensões ao modelo de melhor esforço de modo a permitir o oferecimento de tratamento diferenciado com garantias de qualidade para o tráfego de aplicações mais sensíveis em detrimento de aplicações de menor importância.

A arquitetura de serviços integrados (*Integrated Services Architecture* ou *IntServ*) propõe a reserva prévia de recursos em uma rede a fim de prover suporte a serviços customizados para diferentes tipos de aplicações. Por exemplo, caso uma aplicação utilizando os serviços integrados necessite de um serviço que forneça algum controle sobre a latência e a perda de pacotes podem ser reservados dinamicamente recursos ao longo da rede que permitam o suporte aos requisitos da aplicação. Para isso os serviços integrados propõem um conjunto de extensões ao modelo de melhor esforço, as quais podem ser classificadas em dois componentes principais. O primeiro refere-se à comunicação dos requisitos de qualidade de serviço da aplicação aos elementos de rede e de informações de gerenciamento entre os elementos de rede e a aplicação, sendo em geral implementado utilizando-se um protocolo de reserva de recursos tal como o RSVP [3]. O segundo refere-se aos mecanismos que devem ser implementados nos elementos de rede para o suporte aos requisitos de tráfego reservados pelas aplicações, conhecidos como *serviços de controle QoS*. Atualmente há dois serviços de controle QoS especificados pelo IETF: o serviço garantido e o serviço de carga controlada.

O serviço garantido oferece uma garantia determinística e matematicamente comprovada com relação ao suporte aos requisitos de qualidade de serviço. A reserva de recursos pode ser realizada especificando-se quantitativamente os parâmetros QoS desejados. Já o serviço de carga controlada fornece ao tráfego um tratamento o qual se aproxima do serviço de melhor esforço tradicional quando em condições de pouco tráfego. Tipicamente o serviço garantido é utilizado por aplicações com requisitos rígidos e latência, tais como telefonia e videoconferência, enquanto o serviço de carga

controlada é utilizado por aplicações que podem tolerar pequenas variações na qualidade de serviço oferecida.

A arquitetura de serviços diferenciados propõe o uso de um número limitado de classes de serviços distintas identificadas utilizando-se um campo no cabeçalho dos pacotes IP. Este campo é utilizado para indicar o tratamento a ser recebido por cada pacote em cada nó no interior da rede, o qual é chamado de *comportamento-por-salto*. Nesta arquitetura, cada classe de serviço deve ser utilizada pelo tráfego de várias aplicações; não havendo distinção dentro de cada classe entre os recursos alocados ao tráfego de cada aplicação. Os serviços diferenciados permitem a obtenção de escalabilidade na implementação de qualidade de serviço, sendo indicados para a implementação em *backbones*.

Os principais comportamentos-por-salto especificados pelo IETF são o encaminhamento expresso e o encaminhamento assegurado. O comportamento-por-salto de encaminhamento expresso procura assemelhar-se a uma conexão ponto a ponto ou a uma “linha dedicada virtual” através de uma rede IP heterogênea, minimizando a latência. O grupo de comportamentos-por-salto de encaminhamento assegurado corresponde a um conjunto de 12 comportamentos-por-salto agrupados três a três em quatro diferentes classes de serviço. Cada classe de serviço possui uma quantidade de recursos (processador e buffer de memória) alocados em separado. Dentro de cada classe de serviço, há 3 diferentes probabilidades de descarte associadas aos comportamentos-por-salto. Tipicamente, o encaminhamento expresso é utilizado por aplicações com requisitos rígidos de latência, enquanto o serviço assegurado é voltado para aplicações com necessidade de confiabilidade em redes com congestionamento [4].

A figura 1 compara as arquiteturas de melhor esforço, serviços diferenciados e serviços integrados em relação à granulosidade com que cada um trata o tráfego das aplicações (fluxos). A comparação parte do modelo de melhor esforço, onde a diferenciação de tráfego é nula, até a arquitetura de serviços integrados, onde os requisitos de qualidade de serviço de cada fluxo são tratados separadamente. A

arquitetura de serviços diferenciados encontra-se em um ponto intermediário, onde os requisitos de qualidade de serviço são especificados para agregações de fluxos.

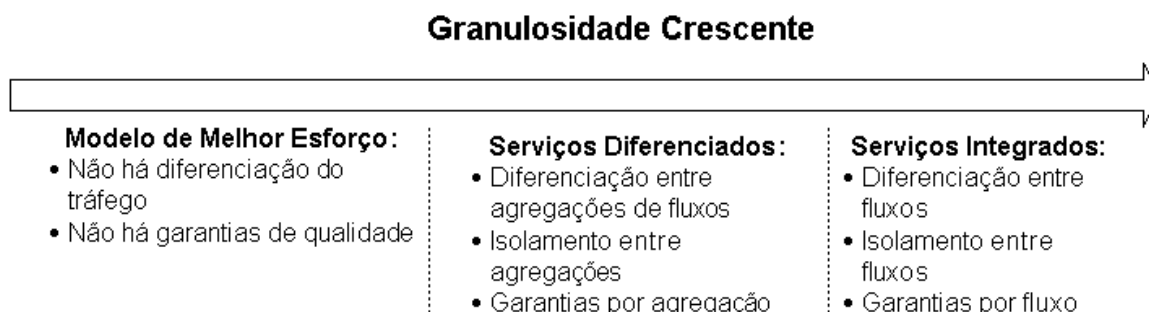


Figura 1: Comparação entre as granulosidades de tratamento de tráfego das arquiteturas de qualidade de serviço [5]

A interoperabilidade entre os serviços integrados e os serviços diferenciados, permite utilizar ambas as arquiteturas e obter as melhores características de cada uma visando a permitir a implementação de QoS fim-a-fim. A arquitetura proposta neste trabalho foi concebida como um conjunto de redes utilizando a arquitetura de serviços integrados nas bordas, interconectadas por uma ou mais redes implementando a arquitetura de serviços diferenciados no centro (*backbone*). Neste modelo, sob a perspectiva dos serviços integrados, as regiões com serviços diferenciados são tratadas como enlaces virtuais conectando roteadores suportando IntServ ou estações. Tais regiões podem ou não participar na sinalização fim-a-fim RSVP, com o propósito de otimizar a alocação de recursos e suportar o controle de admissão[6].

1.1.1 Um Roteador Genérico

Para ilustrar melhor a nossa pesquisa, será descrito como é o funcionamento de um roteador comum usado atualmente.

Um roteador genérico pode ser representado por blocos de funções básicas como mostra a figura abaixo:

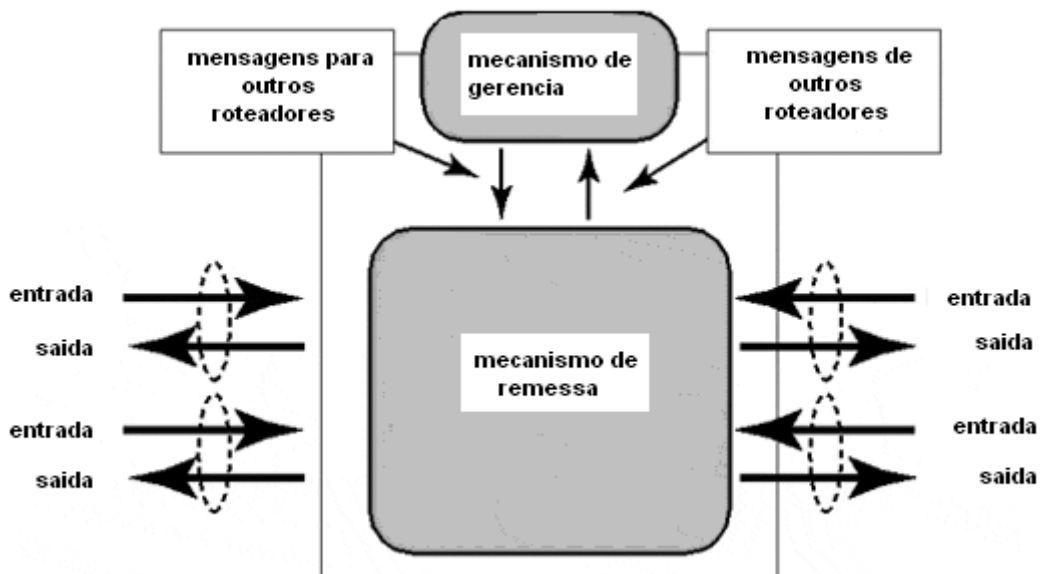


Figura 2: Funções do roteador

Onde:

- Interfaces de entrada aceitam pacotes de outros roteadores, e (baseado no endereço IP destino de cada pacote) o mecanismo de remessa passa os pacotes às interfaces de saída apropriadas. Cada interface usa mecanismos específicos ao tipo de link para transmitir o pacote ao próximo roteador (ou host) ao longo do caminho. Quando o roteador acredita que o congestionamento local está crescendo de uma maneira inapropriada, pacotes podem ser descartados ou marcados como uma maneira de indicar este estado ao redor da rede. O comportamento do mecanismo de remessa (escolha de interface de saída e resposta ao congestionamento) é no final das contas controlado pelo mecanismo de gerência.

Uma tabela conhecida como base de informação de remessa (FIB - forwarding information base) controla a decisão de remessa de um roteador (para onde enviar pacotes). Para todo possível endereço IP destino, uma pesquisa de prefixo longo é

executada pela FIB. Se um endereço é achado, o mecanismo de gerencia avisa qual interface de saída deve receber o pacote, se nenhum endereço é achado, o pacote é descartado. Os conteúdos da FIB refletem o estado atual da topologia IP que cerca o roteador, como determinado pelos protocolos IP de roteamento, por exemplo, Open Shortest Path First (OSPF) ou Protocolo de Gateway de Borda versão 4 (BGP4) rodando no mecanismo de gerencia do roteador.

Roteadores mais antigos tiveram uma única CPU central controlando toda a administração e funções de remessa de pacotes. Os Roteadores evoluíram desde então para arquiteturas mais distribuídas, todos projetados para remover ou reduzir gargalos de desempenho. Em roteadores de alto-desempenho dos backbones, o mecanismo de remessa é distribuído entre o conjunto de placas de interface interconectada por um comutador de alta velocidade ou back plane [27] Não obstante, todos os tais roteadores têm uma sucessão comum de passos pelos quais um pacote deve passar enquanto sendo processado.

Agora que QoS está se tornando importante, o processo de remessa está sendo redesenhado para prestar mais atenção em "quando" os pacotes devem ser enviados e não somente para "onde". A figura 3 abaixo mostra uma visão abstrata do processamento que acontece dentro do mecanismo de remessa da figura acima. Em geral, um pacote atravessa três fases principais:

1. Classificação e comparação feita na FIB (para estabelecer a identidade do pacote e onde é a interface de saída que ele utilizará)
2. Policiamento e marcação (para um eventual mecanismo de reação se o pacote não chegar em um prazo apropriado)
3. Enfileiramento e escalonamento (para enviar o pacote de acordo com o compartilhamento do link ou regras de formatação de trafego ou descartá-lo de acordo com regras de controle de congestionamento)

A fase de classificação de pacote estabelece o contexto para a manipulação subsequente do pacote pelo roteador. Embora na maioria dos casos esse contexto é usado para estabelecer características de manipulação temporais (policiamento, marcação, enfileiramento, e escalonamento), algum contexto adicional pode ser usado

para modificar a decisão de remessa. Por exemplo, um roteador avançado poderia manter múltiplas FIBs (representando arvores de menor caminho baseadas em métricas que diferem umas das outras) e escolher uma delas usando outra informação no cabeçalho do pacote (por exemplo, o endereço de origem do pacote). Uma equivalência, ou pelo menos uma relação íntima, freqüentemente existe entre o contexto de um pacote (estabelecido por classificação) e sua "classe" (percebido em uma base fim-a-fim).

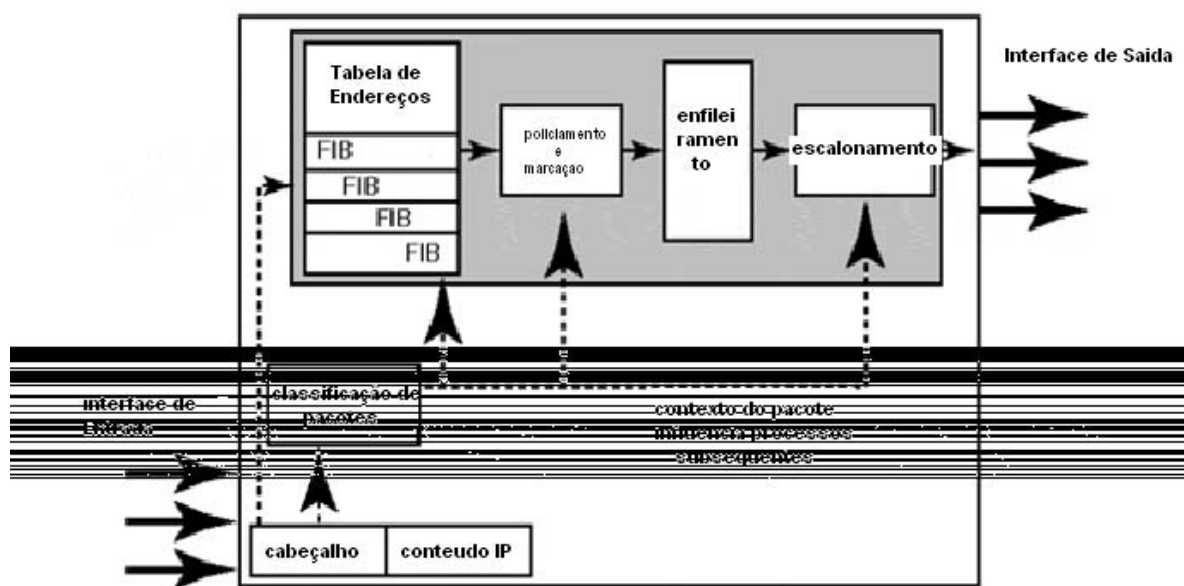


Figura 3: Fases do roteador

A figura 3 reflete uma suposição de que congestionamento só acontece nas interfaces de saída. Uma arquitetura CQS é necessária em todos os pontos de congestionamento, dentro de uma rede ou dentro de um roteador. Certas arquiteturas de roteadores podem ter pontos de congestionamento interno devendo também prover enfileiramento diferenciado e escalonamento em todos estes pontos.

Fabricantes de roteadores diferenciam seus produtos baseando-se em custo versus desempenho. Antigos roteadores que usavam melhor-esforço como modo de operação tinham o estagio de comparação da FIB como sendo o maior gargalo. Porém, nos

últimos anos vários algoritmos extremamente rápidos foram desenvolvidos para fazer milhões de comparações por segundo na FIB de roteadores de alto-desempenho, então esta preocupação não é mais o Mario problema. O fator que está fazendo os vendedores pararem e repensarem as suas arquiteturas é o adicional de processamento por pacote requerido em uma arquitetura de CQS. Assim que são introduzidas capacidades de QoS no mercado, muitos roteadores serão diferenciados por como (ou até mesmo se) eles implementam os vários componentes funcionais.

1.2 Objetivos

O objetivo deste trabalho é o de estudar aspectos relacionados com a implementação do suporte aos serviços integrados em um domínio de serviços diferenciados utilizando-se um ambiente de simulação. Os estudos focaram-se no suporte ao serviço de carga controlada em um domínio de serviços diferenciados utilizando comportamentos-por-salto de encaminhamento assegurado.

1.2.1 - Objetivo Específico

- Descrever os problemas encontrados nos modelos de QoS mais estudados e suas possíveis soluções;
- Demonstrar a viabilidade de implantação de QoS em uma rede IP;
- Verificar o comportamento dos parâmetros de QoS do modelo proposto através de simulações;
- Compreender os diferentes componentes envolvidos no tratamento de QoS (os requisitos, componentes afetados, protocolos, mecanismos e técnicas de tratamento);
- Conhecer os mecanismos IP que tratam a QoS;
- Estudar a concorrência e a integração de diferentes abordagens para QoS; Apresentar uma proposta para que possa garantir a QoS integrando os modelos IntServ e Diffserv.

1.3 Organização

Este trabalho está organizado da seguinte forma. O capítulo 2 apresenta os principais conceitos e mecanismos relacionados com qualidade de serviço, incluindo-se disciplinas de fila, algoritmos de controle de congestionamento e mecanismos de adequação e política de tráfego, além das arquiteturas de serviços integrados e Diferenciados e da proposta de interoperação entre ambas. O capítulo 3 descreve as características do ambiente de simulação. O capítulo 4 descreve os resultados obtidos e as análises realizadas. O capítulo 5 apresenta as conclusões deste trabalho.

Capítulo 2 - Conceitos e Arquiteturas de Qualidade de Serviço

2.1 Qualidade de serviço

Qualidade de serviço é uma expressão difícil de ser definida atualmente. O termo tem sido utilizado tanto por setores da indústria de comunicações quanto pela imprensa e mesmo pela própria comunidade científica para designar diferentes conceitos, por vezes com significados incompatíveis ou limitados a uma determinada área de aplicação.

Antes de definir o que é qualidade de serviço, devemos antes procurar definir os termos que compõe esta expressão, ou seja, definir o que é *qualidade* e o que é um *serviço* dentro do contexto de redes. Deve-se observar que mesmo estes dois termos apresentam significados que podem ser bastante variados.

A palavra *qualidade* está associada à idéia de distinção de uma ou mais características usuais relacionadas com o contexto. Desta maneira, quando nos referimos à qualidade de uma rede geralmente estamos nos referindo a uma ou mais propriedades desta, as quais se distinguem para melhor ou pior. Por exemplo, podemos afirmar que a qualidade de uma rede é ruim se observarmos que ela possui características como congestionamentos freqüentes e grandes taxas de perda de pacotes. O objetivo de qualidade está associado ao serviço fornecido e os critérios estão associados à empresa (operacional) ou à rede (específicos ao serviço). Portanto, existe uma combinação destes critérios, com a finalidade de formar um objetivo de qualidade que está associado a um serviço[7].

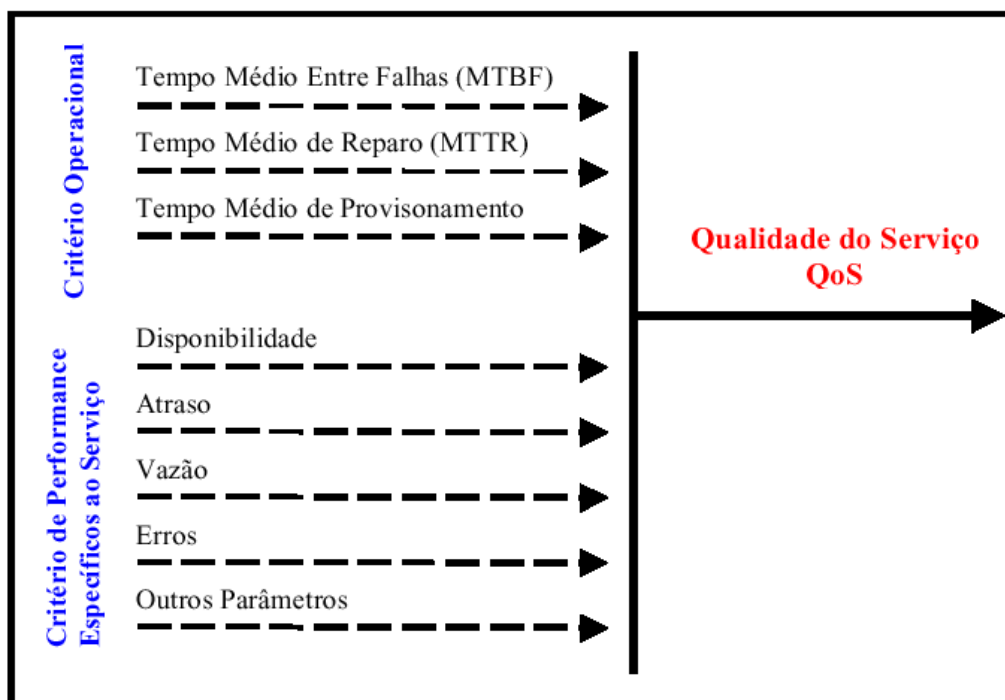


Figura 4: Critérios para QoS [7]

Muitas vezes propriedades relacionadas com a qualidade são especificadas com o intuito de serem quantificadas em valores. Neste caso, passam a receber a denominação de *métricas*. Métricas de qualidade em redes serão discutidas no item 2.3. Já o termo *serviço* pode abranger vários diferentes conceitos, muitas vezes gerando ambigüidades. Dependendo do modo como o termo é utilizado, um *serviço* pode significar desde as aplicações oferecidas aos usuários finais, tais como o correio eletrônico ou navegadores, a protocolos utilizados pelo sistema operacional de rede. Mesmo dentro do contexto em que este trabalho se enquadra, o termo não possui uma definição única podendo possuir vários significados, por vezes conflitantes, como pode ser observado a partir das definições utilizadas por diferentes grupos de trabalho do IETF [todas as RFCs]. Uma boa definição deste termo deve ser abstrata, a fim de abranger os diferentes significados possíveis para o termo. Consideraremos então um *serviço* como uma descrição de uma utilidade que é oferecida ao usuário.

A partir das definições anteriores, poderemos então definir *qualidade de serviço* como o conjunto de características de uma parte definida do tráfego criado por

funcionalidades existentes nos nós componentes de um subconjunto de uma rede. Esta definição abrange a existência de características a serem observadas (qualidade) e a existência de funcionalidades em uma parte da rede que permite o controle destas características (serviços).

Abaixo está listado algumas tentativas de fornecer às redes IP algumas funcionalidades de qualidade de serviço.

- **CAR (*Committed Access Rate*)**: esta técnica implementada por *roteador* de determinados fabricantes, limita a largura de banda consumida numa ligação por determinadas aplicações. Por exemplo, pode ser definido que o tráfego HTTP, SMTP ou telnet, ocupe apenas 50% da largura de banda de uma determinada ligação. O restante é para aplicações de VoIP;
- **CBQ (*Class-Based Queuing*)**: CBQ classifica como vária o tráfego de uma rede em categorias e atribui-lhes uma determinada percentagem de largura de banda disponível. As classes podem ser fluxos individuais de pacotes ou representar uma categoria inteira de aplicações. Podem ser definidos com base em grupos de endereços IP, protocolos, portas TCP ou UDP que representam as aplicações;
- **CoS (*Class of Service*)**: CoS está definido na especificação IEEE 802.1p. Usa 3 bits da trama *Ethernet* para atribuir sete níveis de prioridade a essas tramas. Estes níveis podem ser mapeados em níveis do campo ToS (*type of service*) do pacote IP;
- **DiffServ (*Differentiated Services*)**: redefine 6 dos 8 bits do campo ToS do pacote IP para permitir que este campo seja utilizado para a diferenciação de serviços. Estes 6 bits podem ser combinados de forma a constituírem 64 classes de serviço, que representam várias categorias de aplicações. Esta funcionalidade é interessante, mas necessita que todos os *roteador* entendam as categorias do DiffServ. O DiffServ não apresenta garantias absolutas de QoS, como por exemplo, no caso da VoIP, o melhor que o DiffServ pode fazer é garantir que os pacotes são colocados primeiro nas filas de espera;

- **IP Precedence:** este método compete com o anterior uma vez que também recorre à alteração do campo ToS do pacote IP. Desta vez, este campo é alterado com valores de 0 a 7, sendo o 7 o mais prioritário;
- **MPLS (Multiprotocol Label Switching):** este método é um *standard* do IETF. O DiffServ fornece um mecanismo de identificar classes de serviço mas deixa a implementação dessas classes a cargo das aplicações. MPLS fornece um possível mecanismo exigindo que os *roteador* passem a ser comutadores (*switches*). Uma das formas de fazer isso é de juntar um *router* e um *switch* ATM. O MPLS requer que exista uma infraestrutura que processe “etiquetas” usadas no protocolo;
- **Filas QoS:** ou também filas de classes de serviço. Neste método, os *roteador* ou *switches* de uma rede têm um número de filas para cada porta de saída de tráfego. Os pacotes são identificados com as prioridades dos campos ToS e colocados nas filas conforme a sua prioridade. As filas com maior prioridade são as que são mais rapidamente são atendidas;
- **RED (Random Early Discard):** este método baseia-se em regras definidas para que o *router* possa descartar pacotes de uma fila a partir de um determinado nível de ocupação das filas. Por exemplo, um *router* pode começar a descartar pacotes de uma fila a partir de um valor de 80% de ocupação da mesma. O objetivo é evitar que a fila fique cheia e comece a dispensar pacotes de maior prioridade como os de VoIP. Desta forma, é preferível perder pacotes com prioridade inferior. Este método pode ser combinado com outras técnicas de QoS e não precisa de ser implementado em todos os *roteador* para ser eficiente;
- **RSVP (Resource Reservation Protocol):** há uns tempos atrás, este protocolo liderava as hipóteses de se tornar uma norma para acrescentar funcionalidades de QoS às redes IP. Um equipamento terminal suportando RSVP poderia fazer pedidos muito específicos de QoS à rede e os *roteador* com RSVP poderiam garantir esses pedidos. Desta forma, o RSVP precisa que para além dos *routers*, também os terminais implementem o protocolo. Atualmente, muitas das

potencialidades esperadas pelo RSVP passaram para implementações de DiffServ;

- **ToS (*Type of Service*):** o cabeçalho IP contém um campo de 8 bits designado por *type of service* que era supostamente para ser usado para indicar a prioridade de pacotes. A maioria dos fabricantes de *roteador* ignora este campo porque a maioria das aplicações não o usa. Este campo é reutilizado no DiffServ;
- **Traffic Shaping:** utilizando este método, o tráfego é formatado, isto é, ao tráfego que entra numa rede é retirada a componente de rajada que este possa ter. Este processo é feito recorrendo a *buffers*. Desta forma a componente de rajada do tráfego é dispersa ao longo do tempo, garantindo que não existem picos de tráfego em determinadas alturas;
- **Weighted Fair Queuing (WFQ):** este método aplica-se à largura de banda que uma aplicação recebe nas filas de saída. A cada fluxo de pacotes a que o WFQ é aplicado é colocado em filas separadas e recebe largura de banda de uma forma pesada e variável;
- **WRED (*Weighted Random Early Discard*):** é uma variante pesada do RED. Num *router* RED, os pacotes que são descartados são escolhidos aleatoriamente. Neste caso, essa escolha não é arbitrária, tentando-se escolher os pacotes com mais baixa prioridade [11].

Alguns destes conceitos serão melhores descritos nos tópicos seguintes.

O termo qualidade de serviço freqüentemente é mal empregado, existindo uma certa confusão em torno de seu significado e dos termos classe de serviço (CoS) e tipo de serviço (ToS). Apesar destes termos serem muitas vezes tratados como sinônimos, eles descrevem funcionalidades diferentes. Enquanto QoS possui um significado mais amplo e genérico, os termos CoS e ToS estão ligados a funcionalidades mais específicas dentro do que representa qualidade de serviço. *Classe de serviço* ou CoS está relacionado com a capacidade de diferenciar o tráfego segundo um critério único, e classificá-lo em classes, de modo que as mesmas possam ser reconhecidas de forma distinta ao longo da topologia de uma rede. Já o *ToS* representa o uso dos bits de tipo de serviço do protocolo IPv4 para a diferenciação do tráfego[12], o qual será abordado

com mais detalhes na seção 3.2 (serviços diferenciados).

2.2 Definições de Uso Comum

No decorrer desta dissertação são utilizadas várias expressões as quais ou foram adaptadas de termos de uso comum na língua inglesa ou cujo significado é restrito e diferente. A seguir serão fornecidas definições para as expressões de uso mais importante, com o objetivo apenas de clarificar seu emprego:

- *microfluxo*: um *microfluxo* é definido como uma instância unidirecional do tráfego gerado pela comunicação de uma aplicação para outra, identificado pelo endereço IP de origem, porta de origem, endereço IP destino e porta destino;
- *tratamento*: conjunto de funcionalidades aplicadas a um subconjunto do tráfego por um nó ou uma rede com o fim de alterar algumas de suas características;
- *estação*: computador capaz de se comunicar utilizando os protocolos de Internet;
- *enlace*: um *enlace* ou *link* é uma única conexão física entre nós de uma rede;
- *caminho*: seqüência na forma $\langle h_0, l_1, h_1, \dots, l_n, h_n \rangle$, onde $n \geq 0$ e h_1 e h_0 são estações, $l_1 \dots l_n$ enlaces e $h_1 \dots h_{n-1}$ roteadores;
- *nuvem*: grafo cujos vértices são roteadores e cujas bordas são enlaces que conectam pares de roteadores;
- *banda passante*: capacidade nominal de transmissão de um enlace em bits/s [9]. Esta expressão foi empregada de uma forma diferente do termo *throughput*, o qual é utilizado para referenciar uma métrica (vide item 2.3.1).

2.3 Métricas de Qualidade de Serviço

Não poderia haver QoS se não houvesse métricas de qualidade em redes. Afinal, o próprio conceito de qualidade está ligado à existência de características que possam ser determinadas e comparadas. A seguir serão apresentadas as métricas utilizadas nos experimentos de qualidade de serviço realizados neste trabalho, e

especificados conceitos e terminologias relacionadas. Sempre que possível, as definições utilizadas serão as definidas pelo IETF no grupo de trabalho de Métricas de Performance IP (*IP Performance Metrics*).

2.3.1 Vazão (*Throughput*)

O *throughput* para pacotes do tipo *P* entre uma estação transmissora e uma estação receptora no intervalo de tempo *T* é definido como a métrica obtida pelo quociente do somatório do tamanho dos pacotes de tipo *P* enviados pela estação transmissora e recebido com sucesso pela estação receptora pelo intervalo de tempo *T*. O termo *pacotes de tipo P* refere-se ao subconjunto do tráfego a ser analisado[13].

Geralmente nos experimentos realizados é utilizada uma estatística baseada na métrica acima definida nomeada de *throughput médio*. O *throughput médio* para um conjunto de amostras de *throughput* para pacotes do tipo *P* é definido como a média aritmética das amostras. O *desvio padrão do throughput médio* é definido como o erro médio quadrático das amostras. Um exemplo de aplicação desta estatística seria em um experimento que fosse repetido para aumentar a confiabilidade dos resultados: a estatística seria aplicada às amostras de cada experimento.

Em algumas situações, o *throughput* pode alterar-se drasticamente em decorrência às falhas nos nós da rede ou linhas devido ao congestionamento da rede. A tabela 1 apresenta o *throughput* típico de algumas aplicações.

Aplicação	Vazão (típica)
Aplicações transacionais	1 kbps a 50 kbps
Quadro branco (<i>Whiteboard</i>)	10 kbps a 100 kbps
Voz	10 kbps a 120 kbps
Aplicações web	10 kbps a 500 kbps
Transferências arquivos grandes	10 kbps a 1 Mbps
Vídeo (streaming)	100 kbps a 1 Mbps
Aplicação conferência	500 kbps a 1 Mbps
Vídeo MPEG	1 Mbps a 10 Mbps
Aplicação imagens médicas	10 Mbps a 100 Mbps
Aplicação realidade virtual	80Mbps a 150 Mbps

Tabela 1: Throughput típico de algumas aplicações[7]

2.3.2 Latência (atraso fim-a-fim)

O conceito de atraso, também chamado de retardo ou latência, está ligado ao intervalo de tempo levado por um pacote para percorrer um caminho determinado em uma rede. Há dois diferentes conceitos relacionados com a latência. A *latência real* corresponde ao tempo levado na transmissão dos sinais de dados no meio físico do caminho entre o transmissor e o receptor. Já a *latência induzida* corresponde aos intervalos de tempo introduzidos pela ação de elementos ativos de rede tais como roteadores e *switches*. A diferenciação entre os dois tipos de latência é importante por separar o atraso introduzido pelo meio físico, e que, portanto, não pode ser alterado sem modificações na topologia da rede, do atraso relacionado com os diversos algoritmos utilizados nos softwares dos elementos ativos de rede e que se altera de forma freqüente.

Existem vários tipos de métricas relacionadas com atraso, as quais dependem da metodologia de mensuração. Neste trabalho será utilizada apenas uma métrica para

esta grandeza: atraso em um único sentido.

2.3.3 Jitter (Variação de Atraso)

O termo *jitter* (ou *tremulação*) refere-se às métricas relacionadas com variações na latência de pacotes em um mesmo caminho de uma rede [8]. Se as variações nos atrasos são devido às imperfeições do sistema na rede, ou devido às condições de tráfego dentro da rede, estas variações são normalmente chamadas de *jitter*. A ocorrência de *jitter* é de grande influência na qualidade observada de aplicações de tempo real tais como videoconferência ou telefonia, estando diretamente relacionada com a latência induzida introduzida pelas filas dos elementos ativos de rede.

2.3.4 Perda de Pacotes

Em certas aplicações de tempo real a *perda de pacotes* é um parâmetro de importância fundamental na determinação da performance observada pelos usuários. Em alguns casos mesmo a ocorrência de diferentes distribuições de perdas de pacotes, mantida a mesma taxa de perda pode potencialmente produzir percepções de performance com grande diferença. O impacto da perda de pacotes também é extremamente importante mesmo para aplicações que utilizam protocolos adaptativos como o TCP.

Neste trabalho, será utilizada a métrica de perda de pacotes especificada como: o valor 0 para a perda de um pacote do tipo P entre duas estações O e D significa que O enviou o primeiro bit de um pacote do tipo P a D no wire-time T e que D recebeu o pacote (ou seja, o valor 0 representa a transmissão com sucesso de um pacote). Do mesmo modo, o valor 1 significa que O enviou o primeiro bit de um pacote do tipo P no wire-time T e que D não recebeu o pacote (ou seja, o valor 1 representa a perda de um pacote). Note-se que, do mesmo modo como as métricas utilizadas na latência, esta definição associa a métrica ao tipo de pacote utilizado na mensuração.

Nos experimentos realizados será utilizada uma estatística com base na métrica acima definida referente à contagem de pacotes perdidos no intervalo de tempo de simulação. A estatística utilizada será a de *pacotes do tipo P perdidos no intervalo de tempo T*, a qual se refere à soma dos valores de perda de pacotes (soma de pacotes

perdidos) de determinado tipo em um intervalo de tempo definido. O tipo de pacote refere-se ao subconjunto do tráfego utilizado na mensuração e deve ser definido para o uso da estatística. Por exemplo, pode-se aplicar a estatística para mensurar a perda de pacotes de uma aplicação de voz no intervalo de tempo de 30s.

Algumas aplicações são mais sensíveis ao atraso que outras. A tabela 2 apresenta os requisitos de QoS das aplicações.

Requisitos de QoS	Voz	FTP	E-mail	Vídeo-Broadcast	Vídeo Interativo
Largura de Banda	Baixa a Média	Baixa	Baixa	Alta	Alta
Descarte de Pacotes	Média	Média	Média	Média	Média
Atraso	Alta	Baixa	Baixa	Baixa	Alta
Jitter	Alta	Baixa	Baixa	Média	Alta

Tabela 2: Requisito de QoS das aplicações

2.4 Disciplinas de Fila

Em redes IP, roteadores são elementos centrais para a obtenção de qualidade de serviço: pacotes em geral trafegam por vários antes de chegar ao seu destino final. Cada roteador pode ser considerado como um recurso compartilhado, onde cada aplicação gera requisições por uma parcela dos recursos disponíveis através do envio de pacotes a serem processados e encaminhados. Às vezes a quantidade de pacotes a serem processados é compatível com os recursos disponíveis do roteador, permitindo que sejam processados imediatamente. Outras vezes ocorre disputa pelos recursos no roteador (contenção): o número de pacotes é maior que a quantidade de recursos disponíveis para processá-los. Um aspecto crítico do gerenciamento de performance é a forma como cada roteador resolve a contenção de seus recursos, estando grande parte deste fator relacionada com a *disciplina de fila* utilizada [9].

A função de uma *disciplina de fila* é a de definir qual pacote deve ser processado após o final do processamento do pacote anterior. Durante congestionamentos em roteadores, à medida que a quantidade de recursos necessários para o processamento de novos pacotes é maior do que os disponíveis, os pacotes em excesso são enfileirados. Caso o congestionamento seja longo, a fila eventualmente atingirá um

tamanho máximo a partir do qual pacotes serão descartados. Disciplinas de fila permitem o gerenciamento da contenção de recursos no roteador, adquirindo uma grande importância no tratamento ofertado ao tráfego. Há cinco características principais as quais devem ser levadas em conta na implementação e escolha de uma disciplina de fila: a latência induzida e o jitter a serem obtidos pelo tráfego, a escolha dos pacotes a serem descartados e o isolamento e a justiça entre diferentes classes de tráfego [10].

Tanto a latência induzida quanto o jitter obtidos por uma determinada classe de tráfego estão diretamente relacionados ao tempo de espera dos pacotes nas filas dos roteadores. Em geral, quanto maior o tamanho das filas dos roteadores (configuradas a partir da disciplina de fila), maior serão a latência e o jitter médios dos pacotes encaminhados [10]. Disciplinas de fila também podem ser configuradas para privilegiar o encaminhamento de determinadas classes de tráfego em detrimento de outras. Por exemplo, pacotes de uma aplicação B podem ser configurados para serem encaminhados apenas se não houver pacotes de uma aplicação A na fila; o que possibilita que o tráfego de A obtenha um nível de serviço melhor que B.

Disciplinas de fila também podem incluir gerenciamento das funções de descarte de pacotes. Quando a fila do roteador está cheia e um novo pacote chega, a disciplina de fila deve possuir um algoritmo que escolha se o novo pacote deve ser descartado ou se algum outro pacote já enfileirado deve ser escolhido para descarte para que o novo pacote possa ser processado. Em geral, quanto menor o tamanho das disciplinas de fila de roteadores, maior será o descarte de pacotes em caso de congestionamentos.

Disciplinas de fila podem possuir algoritmos que garantam uma alocação mínima de recursos (espaço na fila, vazão) a determinadas classes de tráfego. Em geral, a alocação de recursos é garantida apenas quando o tráfego adequa-se a um perfil de tráfego previamente definido. O *perfil de tráfego* representa as características que o tráfego deve possuir para que as garantias das disciplinas de filas possam ser atendidas. Perfis de tráfego serão melhor caracterizados durante a definição de políticas de tráfego no item 2.6. O *isolamento* é definido como a proteção fornecida a uma classe de tráfego com relação ao tráfego fora do perfil (ou tráfego em excesso) de

outras classes de modo que o tráfego em excesso não degrade o tratamento oferecido pela disciplina de filas ao tráfego dentro do perfil [15].

Uma outra característica importante é como a *justiça (fairness)* é implementada pela disciplina de filas. O termo justiça pode ser definido como a forma como a disciplina de filas divide os recursos entre requisições concorrentes e de tamanho variado, de modo a obter-se um *compartilhamento por igual (equal sharing)* dos recursos disponíveis entre as várias requisições [4]. A forma como o *compartilhamento por igual* é obtido pode variar levando a diferentes interpretações no uso do termo, sendo dependente do algoritmo utilizado pela disciplina de fila.

Uma forma de obtenção de compartilhamento por igual bastante popular é o *critério max-min*, o qual é utilizado por diversas disciplinas de fila de importância como o *Weighted Fair Queuing*.

2.4.1 FIFO

O FIFO é o tipo de disciplina de fila mais antigo e comum utilizado em redes IP. Com efeito, pode-se afirmar que grande parte das características do modelo de melhor esforço utilizado na Internet atual se deve à larga implementação do FIFO. O FIFO consiste simplesmente em processar os pacotes segundo sua ordem de chegada, de modo que os pacotes que sejam recebidos primeiro sejam encaminhados primeiro (*First-In-First-Out*). Sua maior vantagem é a simplicidade, que permite que sua implementação seja pequena e rápida. De fato, o algoritmo utilizado pelo FIFO possui uma complexidade de $O(1)$, o que o torna bastante escalável. Entretanto, o FIFO não permite nenhum tipo de diferenciação no tratamento oferecido a pacotes, isolamento entre classes de tráfego ou a existência de justiça (*fairness*) em seu encaminhamento, o que não o torna adequado para a implementação de qualidade de serviço.

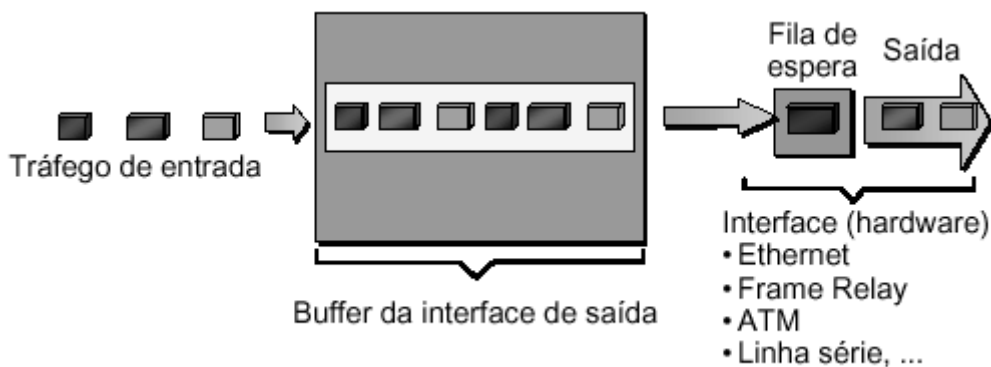


Figura 5: Disciplina de fila FIFO[2]

2.4.2 PRR

Uma das primeiras variações do FIFO a ter popularidade foi o *Priority Round Robin* (PRR). Este tipo de enfileiramento é baseado no conceito de que certos tipos de tráfego podem ser identificados e conceitualmente movidos para o início da fila de saída, de modo que classes de tráfego de maior prioridade sejam sempre transmitidas antes de outros tipos de tráfego.

O funcionamento do algoritmo do PRR é relativamente fácil de ser compreendido. Inicialmente, os pacotes são classificados de acordos com regras preestabelecidas e colocados em filas de serviço associadas. Para cada fila é definida uma prioridade, de modo que haja uma ordem de prioridade entre as filas existentes. Em cada fila, o processamento de pacotes só ocorre caso as filas de maior prioridade estiverem vazias [4].

Como exemplo o PRR pode ser visualizado como uma coleção de filas de saída FIFO. Digamos que sejam definidas quatro filas com quatro prioridades diferentes: alta, média, normal e baixa. À medida que os pacotes são recebidos passam a ser classificados em uma das filas por uma função de classificação. Neste exemplo, os pacotes classificados na fila de alta prioridade são encaminhados antes de qualquer pacote nas filas de média, normal ou baixa prioridade; enquanto os pacotes na fila de média prioridade são encaminhados antes de qualquer pacote da fila de normal prioridade, porém apenas depois dos pacotes da fila de alta prioridade.

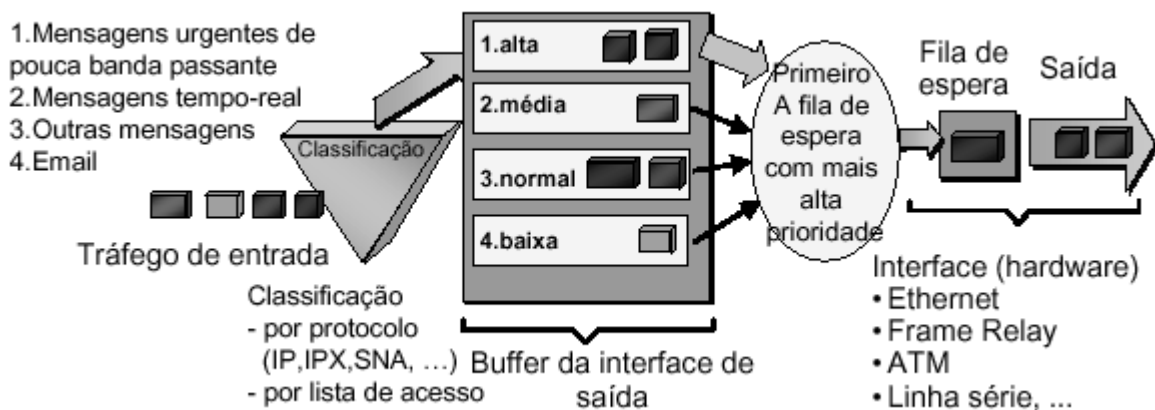


Figura 6: Disciplina de fila PRR[4]

O PRR é de grande utilidade quando se deseja minimizar a latência ou o jitter para determinadas classes de serviço. Com este tipo de enfileiramento, o tráfego de alta prioridade apresenta latência induzida mínima, o que o torna adequado para oferecer tratamento especial ao tráfego de aplicações sensíveis ao atraso. Com efeito, o PRR é utilizado em algumas implementações de arquiteturas de qualidade de serviço (notadamente nos serviços diferenciados) com este propósito.

Entretanto, o PRR também possui várias desvantagens. Não há isolamento entre as classes de serviço ou justiça na alocação de recursos: caso o tráfego de alta prioridade seja muito grande, os recursos disponíveis para o tráfego de menor prioridade serão escassos, levando a congestionamentos nas filas associadas e conseqüente degradação no tratamento ofertado ao tráfego das mesmas. Outro problema refere-se a escalabilidade deste tipo de enfileiramento: quanto maior o número de prioridades utilizadas, maiores serão os recursos necessários para o processamento de pacotes.

2.4.3 WRR

A disciplina de fila *Weighted Round Robin* (WRR) também conhecida como *Class Based Queuing* (CBQ) é uma disciplina de fila de grande popularidade. O WRR é uma variação do PRR, tendo sido proposta a fim de permitir a priorização de certas classes de serviço e, ao mesmo tempo, evitar a completa negação de recursos para as

classes de menor prioridade. O WRR permite o isolamento entre as classes de serviço com uma garantia de recursos de processamento de pacotes mínima para cada classe de serviço definida [4].

Do mesmo modo como o PRR, o algoritmo de escalonamento de pacotes do WRR permite a definição de diversas classes de serviço através da criação de várias filas associadas com cada classe. À medida que cada pacote é recebido, passa a ser classificado e encaminhado para uma das filas existentes. Entretanto, diferentemente do que ocorre no PRR, no WRR o escalonador realiza iterações processando pacotes de todas as filas. A quantidade de pacotes a serem processados em cada fila por iteração e o tamanho de cada fila são configuráveis, permitindo a distinção entre as filas de maior e menor prioridade. Para as filas de menor prioridade há sempre a garantia de que a cada iteração um determinado número de pacotes será processado, evitando a completa negação de serviço em favor das filas de maior prioridade a qual pode ocorrer no PRR.

Um exemplo de uma implementação do WRR seria a configuração de três diferentes *buffers*, correspondendo a filas de alta, média e baixa prioridade a serem processadas por um escalonador. À medida que os pacotes são recebidos, passam a ser classificados em um dos *buffers* por uma função de classificação. Para cada fila, é associada uma quantidade de bytes a serem processados por iteração (*gatilho*), para os quais poderiam ser atribuídos valores como 200 bytes para a fila de maior prioridade, 150 bytes para a média e 100 bytes para a de menor prioridade (correspondendo a pesos relativos de 4, 3 e 2 respectivamente). A cada iteração do escalonador, os pacotes são removidos de cada fila até que a quantidade de bytes de pacotes processados exceda o gatilho configurado para a fila, ou que a fila seja esvaziada. Desta maneira, cada fila receberá uma quantidade mínima de serviço a cada interação, garantindo que mesmo para as filas de menor prioridade sempre haverá processamento de pacotes.

Os gatilhos configurados para cada fila no WRR podem ser interpretados como garantias mínimas de banda passante para o tráfego associado. Em caso de congestionamento, os gatilhos configurados para cada fila garantem que sempre uma

determinada quantidade de bytes de pacotes serão servidos em cada fila por iteração. Assim, durante congestionamentos, os gatilhos determinam a proporção de uso da banda passante do enlace associado à disciplina de fila, o que se traduz na garantia de uma banda passante mínima associada a cada classe de serviço. Entretanto, devido ao fato dos pacotes em geral possuírem tamanhos variados, a exatidão da banda passante mínima oferecida a cada fila dependerá do tamanho máximo dos pacotes em cada classe de serviço.

O WRR também permite a oferta de garantias com relação à latência induzida máxima a ser ofertada a cada classe de serviço. Isto pode ser facilmente realizado a partir da configuração do gatilho associado a cada fila, e do tamanho máximo associado à fila. Como as iterações possuem uma duração máxima, a latência induzida para cada classe de serviço é limitada pelo tamanho máximo da fila, pelo gatilho e pelo tamanho máximo dos pacotes, os quais determinarão o número máximo de iterações possíveis para que um pacote seja encaminhado.

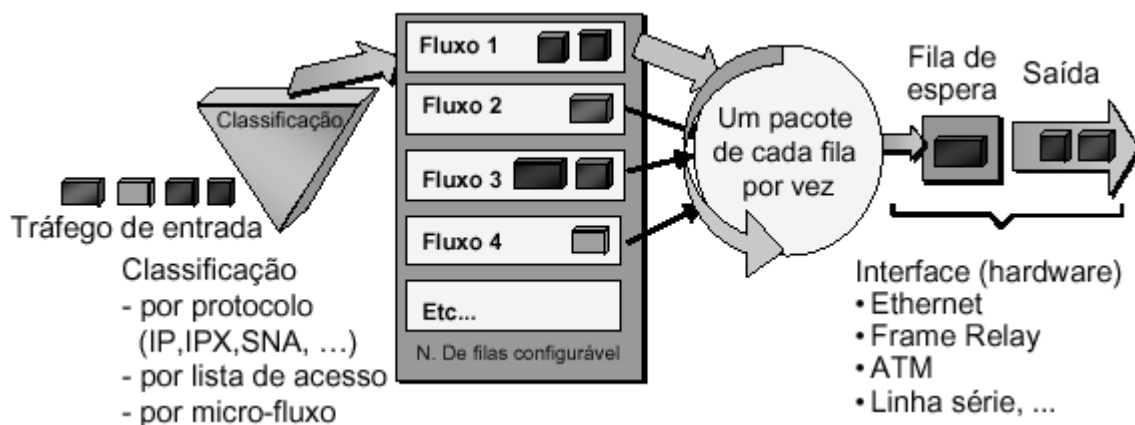


Figura 7: Disciplina de fila WRR[4]

O WRR é considerado como um método básico de diferenciar tráfego em várias classes de serviço com isolamento, sendo ainda um modo eficiente de gerenciar recursos relacionados com disciplinas de filas [4]. Entretanto, também possui desvantagens. O WRR não possui boas características de justiça na alocação de recursos entre as classes de tráfego. Isto ocorre devido aos efeitos da ocorrência de pacotes de tamanhos variados em filas, sendo mais acentuado em enlaces com

pequena banda passante e gatilhos pequenos. Como em enlaces com pouca banda passante os gatilhos em geral não podem possuir tamanhos muito maiores que o tamanho máximo dos pacotes em cada classe de serviço (gatilhos muito grandes causam um aumento na duração máxima de cada interação, levando a um aumento na latência induzida e no jitter), as garantias de banda passante oferecidas pelo WRR tornam-se dependentes do tamanho dos pacotes processados a cada interação. Um outro problema relaciona-se com o número de filas configuradas: quanto mais filas a serem processadas, maiores serão os recursos de processamento necessários.

2.4.4 WFQ

O WFQ ou *Weighted Fair Queuing* é uma disciplina de filas que tem se mostrado bastante popular, possuindo grande aceitação na indústria e servido como base para vários outros trabalhos. Para compreendê-la, faz-se necessário explicar o modelo teórico ótimo o qual o WFQ tenta aproximar. Este modelo é conhecido como *Generalized Processor Sharing* ou GPS [24].

O GPS baseia-se no critério de compartilhamento por igual *max-min*, com uma modificação a qual atribui pesos que refletem uma diferenciação entre requisições (weighted fair share). Considere o exemplo geral de n requisições de tamanhos $s_1, s_2, s_3, \dots, s_i, \dots, s_n$ e pesos relativos associados $w_1, w_2, w_3, \dots, w_i, \dots, w_n$. Seja W a soma dos pesos relativos ($W = \sum w_i$). Considere ainda que a quantidade de recursos disponíveis é R . Inicialmente, um conjunto de tamanhos ponderados das requisições é gerado a partir do quociente (s_i/w_i) , gerando a sequência $w_1/s_1, w_2/s_2, \dots, w_n/s_n$. Esta sequência é ordenada de forma crescente de acordo com o tamanho ponderado das requisições. Inicialmente, o servidor tentará alocar $(w_1 * R/W)$ recursos à primeira requisição. Se esta quantidade de recursos é maior que s_1 , então a demanda será completamente satisfeita e o balanço de recursos faltando será $(R - s_1)$. Se a quantidade for menor, então a demanda será apenas parcialmente e o balanço dos recursos remanescentes será $(R - (w_1 * R/W))$. Em ambos os casos há agora uma nova quantidade de recursos disponíveis a serem compartilhados entre as $(n-1)$ requisições remanescentes, e o processo é aplicado novamente [4].

O GPS pressupõe um modelo fluido onde o tráfego é infinitamente divisível e o escalonamento é feito com granulosidade ínfima entre os fluxos ativos a cada instante. Na realidade a implementação do modelo fluido não é viável, de modo que sua principal utilidade é como referência. O GPS é especialmente útil neste sentido porque possui características ótimas de isolamento entre classes de serviço, justiça e garantias de latência máxima [20].

A disciplina de fila *Packetized Generalized Processor Sharing* (PGPS) também conhecida como *Weighted Fair Queuing* (WFQ) utiliza um algoritmo que procura emular o GPS com uma granulosidade ao nível de pacotes. A implementação desta emulação é realizada a partir de uma função utilizada para calcular o instante em que cada pacote seria transmitido caso o GPS estivesse implementado. Esta função conhecida como função de tempo virtual atribui a cada pacote um instante de tempo de início e um instante de tempo de fim, os quais correspondem aos tempos no sistema de referência GPS. A ordem em que os pacotes são transmitidos corresponde à ordem crescente dos instantes de tempo virtuais de fim de cada um deles [4].

A figura 6 extraída de [1] representa melhor este algoritmo.

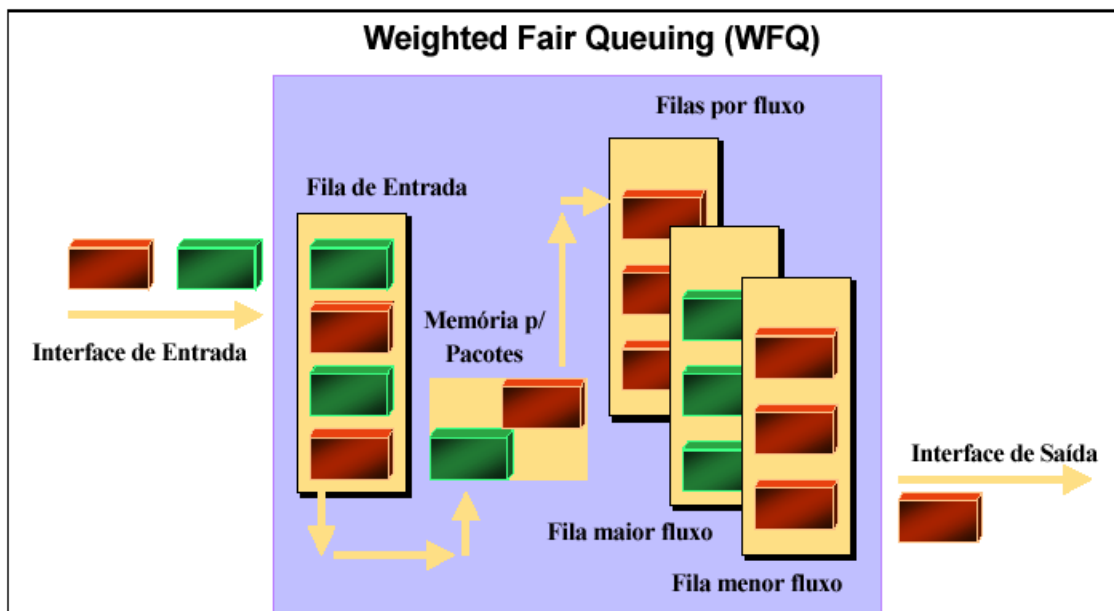


Figura 8: Disciplina de fila WFQ[1]

O WFQ possui características bastante interessantes de justiça e isolamento

entre classes de serviço. Como no GPS, o WFQ procura implementar a justiça entre as classes de serviço utilizando o compartilhamento por igual max-min ponderado: se duas classes de serviço possuem pacotes enfileirados durante qualquer período de tempo, elas são servidas em direta proporção aos seus pesos, independentemente da quantidade de tráfego em excesso que qualquer um deles possa ter gerado; quando uma classe de serviço requer uma quantidade menor de recursos que o efetivamente alocado, o excesso é compartilhado entre todas as outras classes em proporção direta aos pesos relativos de serviço. O compartilhamento por igual max-min também implementa o isolamento entre as diversas classes de serviço: o tráfego em excesso de uma das classes não prejudica o tratamento ofertado ao tráfego das restantes.

Deve-se frisar que o WFQ é apenas uma aproximação do GPS: enquanto o WFQ manipula pacotes de tamanho variado, o GPS pressupõe que o tráfego é infinitamente divisível. A aproximação implementada pelo WFQ é proporcional ao tamanho médio dos pacotes do tráfego servido. Mesmo assim, o WFQ pode ser considerado como uma grande melhora em precisão na justiça entre as classes de serviço e garantias de performance sobre outras disciplinas de fila como o WRR [4].

Uma outra vantagem do WFQ é que um limite de latência fim-a-fim pode ser computado baseada no peso atribuído a cada fluxo. Este limite pode ser calculado para todos os enlaces entre a origem e o destino do fluxo e é independente do número de fluxos que são multiplexados em cada enlace [26]. Esta propriedade faz com que este algoritmo seja bastante utilizado em implementações de arquiteturas de qualidade de serviço que requerem limites quantitativos de latência, como é o caso dos serviços integrados.

A principal desvantagem do WFQ é a complexidade de seu algoritmo, a qual para as operações de inserção e remoção de pacotes na fila é em média de $O(n)$, onde n é o número de classes de serviço ativas. Esta característica faz com que o WFQ não deva ser implementado quando a granulosidade das classes de serviço for muito fina (por microfluxo). Uma outra desvantagem é a de que a única limitação que o WFQ impõe ao *jitter* é a latência máxima fim-a-fim. Disciplinas de filas baseadas em prioridade (como o PRR) apresentam resultados melhores com relação ao jitter.

2.5 Mecanismos de Controle de Congestionamento

Em redes IP, roteadores não podem reservar memória ou recursos de comunicação em antecipação ao recebimento de pacotes. Como resultado, podem ser sobrecarregados com tráfego, em uma condição conhecida como congestionamento[2]. A ocorrência de congestionamentos é a principal causa de degradação no tratamento ofertado ao tráfego, sendo o conceito de qualidade de serviço muito ligado aos mecanismos de controle e reação aos mesmos. Afinal, congestionamentos só não ocorrem em redes onde a quantidade de recursos disponíveis é muito maior do que os recursos a serem alocados pelo tráfego; levando a uma subutilização dos mesmos. Antes de descrever as arquiteturas de qualidade de serviço, faz-se necessário apresentar os mecanismos de controle de congestionamento existentes em redes IP.

2.5.1 Controle de Congestionamento do TCP

O TCP possui dois tipos de controle de fluxo e congestionamento distintos. Um está ligado apenas à interação entre o transmissor e o receptor da conexão TCP, e é responsável por evitar que o transmissor gere mais tráfego do que o receptor possa tolerar. Este mecanismo de controle de fluxo fim-a-fim é implementado no algoritmo de janelas deslizantes do TCP. Um controle deste tipo é essencial em um ambiente como o da Internet, onde máquinas de várias capacidades e tamanhos diferentes se comunicam [16].

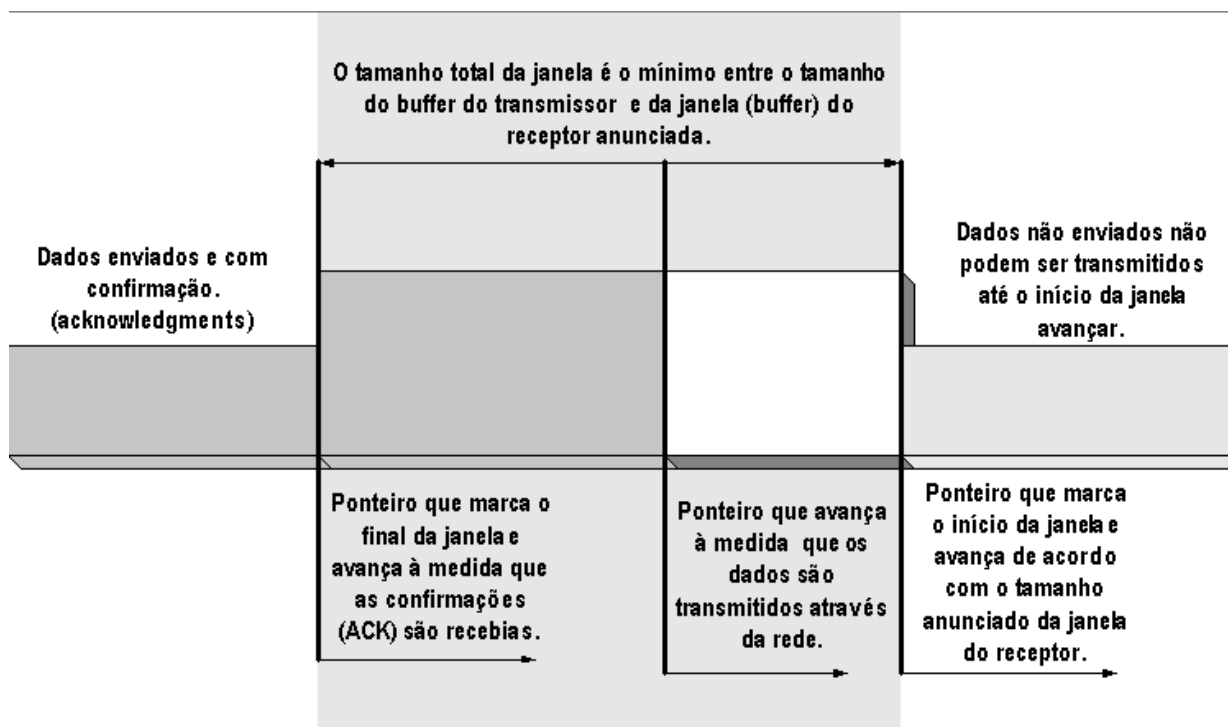


Figura 9: Algoritmo de janelas deslizantes do TCP para o transmissor [4]

O algoritmo de janelas do TCP pode ser resumido da seguinte maneira. Inicialmente, o receptor informa ao transmissor (durante o estabelecimento da conexão TCP), o tamanho do buffer (ou janela de recepção) disponível. O transmissor então calcula o mínimo entre o tamanho deste buffer e o tamanho do buffer local, e o utiliza para configurar o tamanho inicial da janela de transmissão. A partir deste ponto, o transmissor poderá enviar os dados disponíveis na janela de transmissão para o receptor, com a restrição de que deve ser armazenada uma cópia dos dados que ainda não houverem sido acusados como recebidos. A cada vez que uma parte dos dados na janela de transmissão tiver seu recebimento confirmado (utilizando-se pacotes de *acknowledgment* ou *ACK*), o lado esquerdo da janela de transmissão é avançado e o mínimo entre o buffer do transmissor e do receptor é utilizado para calcular um novo limite esquerdo. Se este limite é maior que a quantidade de dados já enviados e não acusados como recebidos; dados adicionais podem então ser enviados até o limite da janela do transmissor (vide figura).

Um segundo grupo de algoritmos permitem que o TCP reaja à ocorrência de

congestionamentos na rede. Estes algoritmos estão em constante modificação, podendo variar de acordo com a especificação do TCP utilizada. Atualmente há quatro versões do algoritmo de controle de congestionamento do TCP em uso na Internet conhecidas como *Tahoe*, *Reno*, *New Reno* e *Sack*. A descrição a seguir é válida para a versão do TCP conhecida como *TCP Reno* [20], a qual será a utilizada na parte experimental deste trabalho. Um fator importante na escolha foi o fato do TCP Reno ter sido utilizado em quase todos os trabalhos utilizados como referência.

O mecanismo de controle de congestionamento na rede do TCP é necessário devido à forma com que os congestionamentos são percebidos pelas estações (pontos finais) da rede. Para estas, a ocorrência de congestionamentos representa o aumento na latência e da perda de pacotes. Infelizmente, a reação a estes fatores por parte da maioria dos protocolos de transporte (incluindo o TCP) é a espera pela confirmação de entrega (*acknowledgement*) dos pacotes e a retransmissão, caso a confirmação demore mais que um determinado intervalo. As retransmissões aumentam o tráfego na rede agravando o congestionamento em vez de aliviá-lo. Caso não haja nenhuma outra forma de reação ao congestionamento, as retransmissões causarão aumentos ainda maiores na latência e na perda de pacotes, as quais por sua vez causarão novas retransmissões, mantendo um ciclo que continuará até que a rede torne-se inutilizável. Esta condição é chamada de colapso por congestionamento [16].

Para evitar esta situação, o TCP implementa mecanismos para a detecção e reação a congestionamentos na rede entre o transmissor e o receptor de fluxos TCP. Note-se que este não é um problema simples. Além de reagir a congestionamentos, o protocolo deve procurar maximizar a eficiência da transferência de dados. Para isso, deve ser encontrado um limiar de equilíbrio dinâmico, no qual a taxa de envio de pacotes é maximizada e a perda de pacotes é minimizada. Este limiar corresponde ao ponto onde a taxa de entrada de pacotes na rede é estabelecida logo abaixo da taxa correspondente ao início da ocorrência de congestionamentos.

Para atingir o ponto de melhor eficiência o TCP utiliza um algoritmo dinâmico de ajuste constante do tamanho da janela de transmissão de pacotes. Como o transmissor de um fluxo TCP pode realizar no máximo uma transferência dos dados disponíveis na

janela de transmissão por round-trip-time (RTT), o tamanho da janela é um fator crítico na determinação da taxa de envio de pacotes de um fluxo. O TCP utiliza uma combinação de algoritmos de gerenciamento da janela de transmissão, de detecção de perda de pacotes e de retransmissão, para o controle da taxa de envio de pacotes.

O gerenciamento da transmissão de pacotes utiliza uma modificação no algoritmo de janelas deslizantes do transmissor para o controle da janela de transmissão. Esta modificação consiste na criação de uma nova janela chamada de *janela de congestionamento* (*congestion window* ou *cwnd*), de modo que a janela de transmissão passe a ser calculada utilizando-se o mínimo entre o buffer de transmissão, o buffer de recepção e a *cwnd*. O objetivo deste algoritmo é o de inicializar a transmissão a uma taxa que possua uma probabilidade muito baixa de perda de pacotes, então aumentar a taxa (através do aumento da janela de congestionamento) até o transmissor receber uma indicação (perda de pacotes) que a taxa excedeu a capacidade disponível na rede.

O ajuste do tamanho da janela de transmissão é realizado utilizando-se algoritmos de incremento da taxa de transmissão de pacotes, de detecção e de controle de congestionamento. Existem dois algoritmos de incremento da taxa de transmissão do TCP conhecidos como *slow start* e *congestion avoidance*. O *slow start* permite que os fluxos TCP atinjam de forma rápida a capacidade disponível da rede, a qual corresponde à taxa de envio de pacotes de maior eficiência. Este algoritmo incrementa a janela de congestionamento do TCP pela quantidade de dados com acusações de recebimento (*acknowledgements*) a cada *round-trip-time*, o que se traduz na duplicação da janela de congestionamento a cada RTT. Isto permite que o TCP dobre a taxa de envio de pacotes a cada RTT, até que a *cwnd* se torne maior que o buffer de recebimento, o buffer de transmissão ou a capacidade disponível na rede. Para o último caso são utilizados os algoritmos de detecção de congestionamento e *congestion avoidance*.

O algoritmo de *congestion avoidance* permite que o TCP incremente a janela de congestionamento através do acréscimo de um valor constante a cada ACK recebido. Isto faz com que o TCP tenha um incremento na taxa de transmissão linear, em

contraste com o incremento exponencial do *slow-start*. A seleção entre os dois modos é realizada utilizando-se algoritmos de detecção de congestionamento do TCP.

Para a detecção de congestionamento, o TCP utiliza a perda de pacotes como indicador, o que pode ser realizado de duas maneiras. A primeira corresponde a apenas à perda de um pacote em uma seqüência. Neste caso o receptor (de acordo com a especificação do TCP) gera um ACK duplicado correspondendo ao pacote com maior seqüência recebido na ordem correta para cada pacote fora de ordem recebido. O recebimento de três ACK duplicados faz com que o TCP execute dois algoritmos: o *fast recovery* e o *fast retransmit* [16]. O *fast retransmit* consiste no envio imediato do segmento que aparenta ter sido perdido na seqüência sem esperar pela expiração do timer de retransmissão do mesmo. Já o *fast recovery* é utilizado para reagir a um congestionamento considerado como moderado. De acordo com este algoritmo, o TCP cessa o *slow start*, reduz a janela de congestionamento à metade e passa a incrementar o tamanho da janela de congestionamento de acordo com o algoritmo de *congestion avoidance*.

No segundo caso de detecção de congestionamentos, são perdidos pacotes correspondendo aos últimos em uma seqüência. Neste caso, não é gerado nenhum tipo de ACK para o TCP da estação transmissora, o que faz com que o TCP não possua nenhuma informação a respeito da capacidade disponível da rede. O TCP então aguarda o intervalo de tempo correspondente ao *timeout* dos pacotes perdidos, diminui a janela de congestionamento para um pacote por RTT e entra no modo *slow start*. O incremento da janela de congestionamento com o *slow start* continua até que um novo congestionamento seja detectado ou até que o tamanho da janela de congestionamento atinja a metade do tamanho original (antes do congestionamento inicial), quando então o TCP entra no modo *congestion avoidance*.

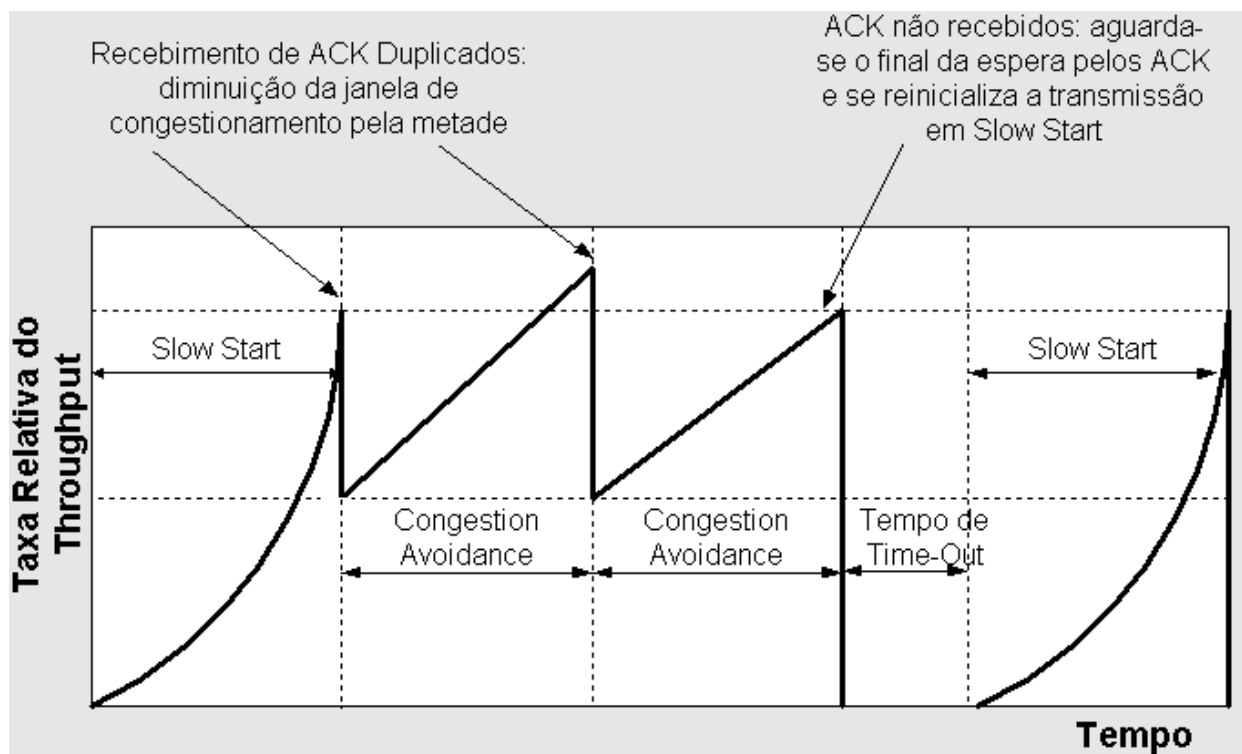


Figura 10: Comportamento do throughput de um microfluxo TCP com Slow Start e Congestion Avoidance [5]

Resumindo de uma forma genérica, podemos afirmar que os algoritmos de detecção e controle de congestionamento permitem que o TCP atinja a capacidade disponível na rede e então a mantenha (considerando-se que os buffers de transmissão e recepção de pacotes não são gargalos). Isto é realizado através do incremento constante da janela de congestionamento utilizando-se os algoritmos de slow start e congestion avoidance. O slow start corresponderia a um ajuste “grosso”, que permite que o TCP atinja rapidamente uma taxa de envio de pacotes correspondente à metade da utilizada antes da detecção de um congestionamento. Já o *congestion avoidance* corresponderia a um ajuste fino, o qual permite que o TCP atinja de uma forma mais aproximada a capacidade disponível. Em geral, em uma rede que não apresente congestionamentos freqüentes, o modo *slow start* é utilizado apenas no início de uma nova conexão, sendo o *congestion avoidance* utilizado continuamente para verificar o ponto que apresente a melhor eficiência para o envio de dados. Um resumo da operação dos algoritmos de controle de congestionamento do é

mostrado na figura 8.

2.5.2 Gerenciamento Ativo de Filas e o RED

Para pequenas quantidades de microfluxos, os algoritmos implementados no TCP são efetivos na detecção e no controle de congestionamentos. Entretanto, em redes IP de maior escala onde milhares de microfluxos TCP estão ativos ao mesmo tempo e uma situação de congestionamento acontece em um gargalo em particular, pode acontecer que vários microfluxos experimentem perda de pacotes ao mesmo tempo, gerando uma situação conhecida como *sincronização global* [23]. A *sincronização global* ocorre quando centenas ou milhares de microfluxos em uma mesma rede e que atravessam um mesmo nó congestionado, inicializam e terminam o algoritmo de slow start do TCP a aproximadamente o mesmo intervalo de tempo. Cada transmissor TCP detecta a perda de pacotes e reage de acordo: entra em slow start, diminui o tamanho de sua janela, espera e então procura retransmitir os dados. Se a situação de congestionamento persistir, cada transmissor TCP detecta a perda de pacotes novamente e o processo se repete, resultando em uma má utilização dos recursos da rede.

Um segundo problema relacionado com os algoritmos de controle de congestionamento do TCP envolve a ocorrência de picos de tráfego de curta duração. A ocorrência destes picos pode causar perda de pacotes que levem grande número de fluxos a reagir com o algoritmo de slow start; fazendo com que parte dos recursos disponível na rede não seja utilizado. Uma solução para este problema é o aumento do tamanho da fila dos roteadores de modo que congestionamentos de curta duração sejam absorvidos através do enfileiramento dos pacotes em excesso. Entretanto o aumento das filas dos roteadores também causa o aumento da latência induzida e do jitter, o que pode ser indesejável.

Uma solução para estes problemas envolve a criação de novos mecanismos de controle de congestionamento que permitam que a rede se mantenha com baixa latência induzida e elevado uso de banda passante, enquanto ao mesmo tempo evitem o problema da sincronização global. Para isso, tais mecanismos devem possuir a

capacidade de monitorar o tráfego da rede, de modo a que possam distinguir entre congestionamentos de curta e longa duração, e permitir a detecção e a reação aos mesmos. Em redes com tráfegos TCP com diversos RTT, requisitos de banda passante e sensibilidades à latência, a forma mais apropriada de implementar estes mecanismos é através da monitoração e do gerenciamento de filas de roteadores, o que é chamado de *gerenciamento ativo de filas*. Dentre os mecanismos de gerenciamento ativo de filas, o mais conhecido e de maior aceitação é o *Random Early Detection* ou RED [13].

O RED utiliza dois diferentes algoritmos para detectar e reagir a congestionamentos. A detecção é realizada a partir do monitoramento do tamanho médio da fila. A reação é realizada a partir do descarte aleatório de pacotes, configurado a partir de dois gatilhos correspondentes a tamanhos da fila, e que são comparados com o tamanho médio monitorado para a obtenção da probabilidade de descarte de pacotes. O descarte aleatório de pacotes e a monitoração do tamanho médio da fila evitam a ocorrência da sincronização global e, ao mesmo tempo, permitem que congestionamentos incipientes sejam sinalizados a aplicações TCP.

O algoritmo de monitoração do tamanho médio da fila realiza o cálculo do tamanho médio de forma a evitar que congestionamentos transientes ou picos de tráfego de curta duração causem aumentos significantes no resultado. Isto é realizado a partir da utilização de uma média exponencial móvel com pesos, a qual filtra variações no tamanho da fila de curta duração. A influência dos eventos das variações de curta duração no cálculo do tamanho médio da fila pode ser configurado a partir de um parâmetro fornecido ao algoritmo: quanto maior estes parâmetros, maiores podem ser as durações dos picos de tráfego e dos congestionamentos transientes sem causar aumentos significantes nos valores do tamanho médio das filas.

Já o algoritmo de descarte de pacotes utiliza o tamanho médio da fila e dois gatilhos para o cálculo da probabilidade de descarte. Tais gatilhos são conhecidos como gatilho mínimo e gatilho máximo. Quando o tamanho médio da fila é menor que o gatilho mínimo, nenhum pacote é descartado. Quando o tamanho médio da fila é maior que o tamanho máximo da fila, todos os pacotes são descartados. Entretanto, quando o tamanho médio da fila está entre os gatilhos mínimo e máximo, cada pacote é

descartado com uma probabilidade a qual é um função linear do tamanho médio da fila, variando de 0 no gatilho mínimo a um valor máximo no gatilho máximo. Isto está representado na figura 9 (abaixo).

Os parâmetros utilizados no RED costumam ser representados no formato: *Probabilidade de Descarte/Gatilho Mínimo/Gatilho Máximo* [23]. Por exemplo, para uma probabilidade de descarte de pacotes de 2%, um gatilho mínimo de 10 pacotes e um máximo de 20 teríamos 0.02/10/20.

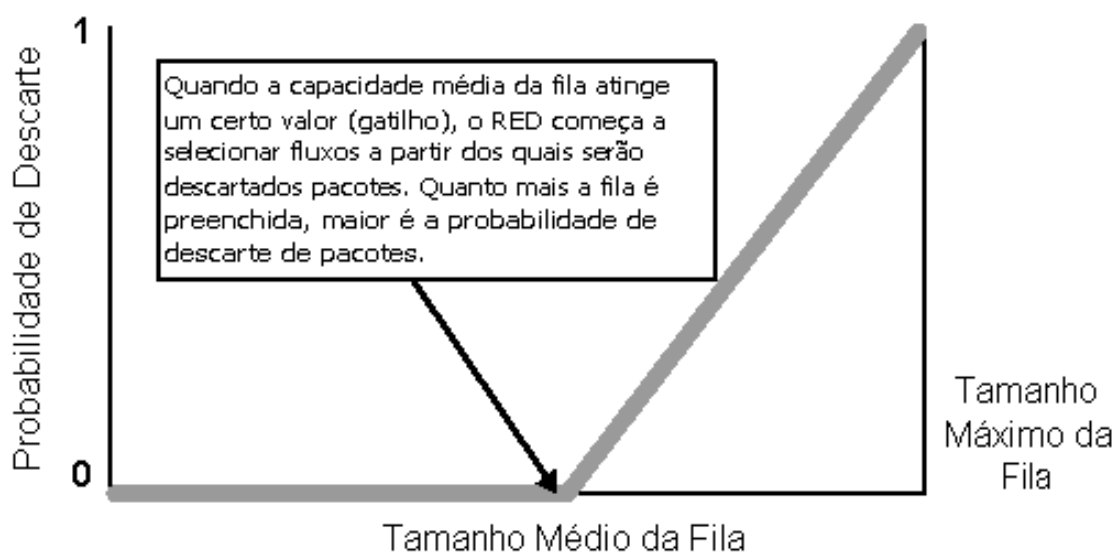


Figura 11: Variação da probabilidade de Descarte de Pacotes e tamanho médio da fila no RED[6]

2.5.3 Variações do RED

O RED pode ser estendido de modo a permitir a diferenciação do tráfego em classes de serviço com diferentes características de descarte de pacotes. Em uma mesma fila pode haver vários gatilhos e probabilidades de descarte associadas a intervalos entre gatilhos, permitindo a diferenciação entre as classes com o estabelecimento de diferentes parâmetros para cada uma (probabilidade de descarte, gatilho mínimo, gatilho máximo). Para cada classe, o tamanho médio da fila utilizado para o cálculo da probabilidade de descarte também pode ser calculado de duas maneiras diferentes. Uma envolve o cálculo de vários tamanhos médios baseados nos

pacotes pertencentes a cada uma das classes. A segunda envolve o cálculo de apenas um tamanho médio baseado nos pacotes existentes na fila.

Uma forma de classificar as possíveis implementações do RED utiliza quatro categorias genéricas baseadas nos parâmetros utilizados:

- Única Média Único Gatilho (Single Average Single Threshold - SAST);
- Única Média Múltiplos Gatilhos (Single Average Multiple Thresholds - SAMT);
- Múltiplas Médias Único Gatilho (Multiple Average Single Threshold - MAST); e
- Múltiplas Médias Múltiplos Gatilhos (Multiple Average Multiple Thresholds - MAMT) [27].

A partir da generalização proposta alguns exemplos podem ser citados. O RED em sua forma mais simples é um exemplo de um SAST, onde há apenas um gatilho e uma média utilizada para a diferenciação. Um SAMT onde há apenas uma média adotada para o tamanho médio da fila e vários gatilhos pode ser exemplificado pelo WRED ou *Weighted Red* proposto pela empresa *Cisco Systems*. O WRED permite uma distribuição de descarte de pacotes diferenciada, de acordo com probabilidades de descarte e gatilhos atribuídos a cada classe de serviço. Funções de classificação de tráfego locais a cada roteador são utilizadas para separar os pacotes em cada classe de serviço de acordo com regras customizáveis [24] [4]. Nas categorias MAMT e SAMT o cálculo do tamanho médio da fila é realizado de forma diferente para cada classe de serviço, podendo abranger no cálculo os pacotes pertencentes apenas à classe de serviço ou os pacotes pertencentes a outras classes.

Uma das variações do RED mais conhecidas, proporcionando diferenciação de tráfego é chamada de RIO, ou RED-with-In-and-Out. O RIO oferece tratamento diferenciado a duas diferentes classes de serviço, chamadas de *in* and *out*. A idéia é a de que os pacotes sejam classificados segundo a adequação de seus fluxos a um perfil esperado de tráfego: os pacotes pertencentes a fluxos dentro do perfil são classificados como *in*, e os outros como *out*. A classificação é realizada por um mecanismo externo ao RIO, de acordo com diferentes critérios, os quais podem variar de acordo com a implementação [2].

O RIO pode ser classificado como um MAMT ou SAMT, dependendo da forma

como o cálculo do tamanho da fila é realizado. O funcionamento do RIO é semelhante ao do RED, com a diferença que em lugar de dois gatilhos são utilizados três, oferecendo diferentes probabilidades de descarte de pacotes de acordo com o tamanho da fila. Quando o tamanho da fila está abaixo do primeiro gatilho nenhum pacote é descartado. Quando está entre o primeiro e o segundo apenas pacotes *out* são descartados. Quando está após o segundo, tanto pacotes *in* quanto *out* são descartados, entretanto os pacotes *out* o são de uma forma mais agressiva. Finalmente quando o tamanho da fila atinge o terceiro gatilho, todos os novos pacotes são descartados.

2.6 Controle de Admissão, Adequação e Políticas de Tráfego

O gerenciamento ativo de filas e os algoritmos implementados no TCP descritos anteriormente são apenas defensivos com relação a congestionamentos, reagindo aos mesmos sem, entretanto poder evitá-los. Para o fornecimento de garantias de qualidade é necessária a implementação de novos mecanismos, que permitam um maior controle sobre as características do tráfego admitido no interior da rede, sobre a quantidade de recursos disponíveis e sobre a alocação de recursos a novos tipos de tráfego. Estes mecanismos consistem nas funções de controle de admissão, adequação e política de tráfego [4].

2.6.1 Token Bucket

Tanto os mecanismos de controle de admissão, quanto os de adequação e política de tráfego podem utilizar um mesmo modelo conceitual para caracterizar o tráfego. O modelo mais comum utilizado é o do *Token Bucket*, ou *balde de símbolos* [2].

O *token* ou *símbolo* representa a admissão de uma determinada quantidade de dados na rede. A idéia é a de que os *tokens* são gerados a uma determinada taxa regular, representando uma taxa de admissão de dados constante a longo prazo. Em sua forma mais simples, o modelo com tokens utiliza apenas a *taxa de geração de tokens* (ou taxa de geração de símbolos) para caracterizar o tráfego: os pacotes

representados são apenas aqueles recebidos no momento em que um *token* está disponível. Na prática este modelo simples representa um tráfego com throughput constante. Este modelo pode ser utilizado por exemplo; em um algoritmo para regular a entrada de tráfego em uma rede: quando um novo pacote é recebido, verifica-se se há tokens disponíveis para a quantidade de dados do pacote; caso haja o pacote será admitido no interior da rede; caso não haja, o pacote será descartado. Como veremos, este exemplo corresponde a um mecanismo de adequação de tráfego.

Um modelo que permite uma melhor aproximação do tráfego real utiliza um acumulador de Tokens, ou *Token Bucket*. Para a caracterização do tráfego, utiliza-se além da taxa de geração de Tokens, uma quantidade de Tokens que pode ser armazenada denominada de *profundidade do balde (bucket depth)*. Os Tokens são armazenados no balde sempre que não há demanda imediata até um limite estabelecido pela profundidade do balde.

Na prática, o Token Bucket pode ser utilizado de duas maneiras para caracterizar o tráfego. Caso a profundidade do balde seja estabelecida em apenas um Token e cada Token represente no máximo apenas um pacote, então o tráfego caracterizado pelo Token Bucket será o de um throughput constante, semelhante ao do modelo simples. Caso a profundidade do balde possa armazenar Tokens equivalentes a uma quantidade de dados que represente mais de um pacote, então o Token Bucket caracteriza um tráfego com um throughput médio de longo prazo equivalente ao da taxa de geração de Tokens e variações de curta duração limitadas pela quantidade de Tokens que podem ser acumulados no balde.

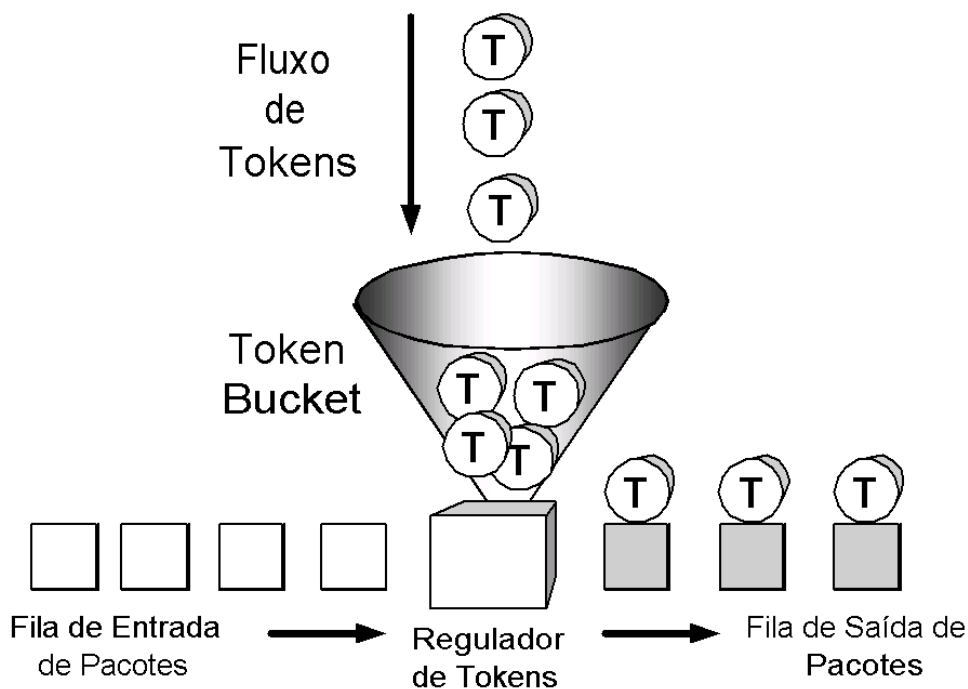


Figura 12: O regulador de Tráfego Token Bucket[2]

A fim de melhor compreender o modelo representado pelo Token Bucket será analisado um exemplo de sua utilização. A figura 2.4 representa a implementação de um mecanismo para regular a entrada do tráfego em uma rede. O mecanismo funciona da seguinte maneira. Um modelo Token Bucket é utilizado para caracterizar o tráfego que se deseja admitir: tokens são gerados a uma taxa constante e um acumulador armazena os tokens não utilizados até um limite configurado por sua profundidade. Quando um pacote é recebido, ele só é admitido na fila de saída se houver tokens suficientes no balde correspondentes à quantidade de dados existentes no pacote. Se não existirem, o pacote é descartado ou enfileirado, dependendo da configuração da função reguladora. Se o pacote é admitido, uma quantidade de tokens correspondente ao tamanho do pacote é removida do balde. Se a quantidade de tokens se acumular até o limite do balde, novos tokens serão descartados. O mecanismo exemplificado representa um adequador de tráfego, no qual o tráfego admitido é suavizado de modo a apresentar um throughput médio de longo prazo correspondente à taxa de geração de tokens e uma variação correspondente à profundidade do balde.

2.6.2 Adequação de Tráfego

O termo *adequação de tráfego* refere-se à implementação de mecanismos que permitam controlar a quantidade e o volume do tráfego recebido e/ou enviado por uma rede. A adequação de tráfego é geralmente realizada nos nós localizados nas fronteiras administrativas, possibilitando um controle sobre fontes geradoras de tráfego externas. Mecanismos de adequação de tráfego são em geral associados a mecanismos de classificação de pacotes que diferenciem o tráfego a ser adequado do restante.

Dois tipos predominantes de mecanismos de adequação de tráfego existem: um para suavizar o tráfego de modo a limitar o throughput a uma taxa máxima, e outro que suaviza o tráfego a uma taxa média por períodos maiores e permite picos de tráfego a taxas maiores que a taxa média, sendo os picos porém limitado por uma taxa máxima pré-definida. O mecanismo de adequação com suavização do throughput a uma taxa máxima faz com que o tráfego suavizado apresente um throughput máximo constante, em oposição ao tráfego comum com picos erráticos de vários microfluxos. Isto pode ser implementado utilizando-se um regulador com um Token Bucket com uma profundidade de balde equivalente à de um único pacote, conforme descrito no item anterior.

O segundo tipo de adequação de tráfego utiliza um regulador com um Token Bucket com uma profundidade equivalente a mais de um pacote. Mecanismos de adequação de tráfego com esta implementação permitem picos de tráfego de curta duração limitados pela profundidade do balde, enquanto é mantido um throughput constante a longo prazo.

Adequadores de tráfego podem ser utilizados em conjunto para implementar a adequação de tráfego nas fronteiras administrativas de uma rede. Quando utilizados em paralelo os adequadores de tráfego podem ser utilizados juntamente com classificadores de forma a permitir que classes de serviço possam ser suavizadas em separado. Quando utilizados em seqüência, os adequadores de tráfego podem ser utilizados para permitir uma maior customização das características de suavização do tráfego. Um exemplo de uso de adequadores em seqüência será fornecido no próximo

item.

2.6.3 Política de Tráfego

O termo *política de tráfego* pode ser definido como o conjunto de critérios utilizados para discriminar para cada classe de serviço o tráfego que pode utilizar os recursos reservados para a classe e o tráfego em excesso que deve ser descartado ou receber um tratamento pior sob pena de degradar a qualidade de serviço [4]. Este conceito é uma parte crucial de qualquer implementação de QoS: se não se pode diferenciar o tráfego em trânsito em uma rede, não se pode ter controle sobre congestionamentos, o que torna a qualidade de serviço dependente da quantidade de tráfego gerado.

Mecanismos de política de tráfego são utilizados para verificar se o tráfego conforma-se a um *perfil de tráfego (traffic profile)* previamente definido, e adotar ações caso não se conforme. O perfil de tráfego corresponde a uma caracterização do tráfego que se espera admitir no interior de uma rede e geralmente é negociado entre o administrador da rede (provedor) e as redes geradoras de tráfego externo (usuários). A forma mais comum de implementar-se perfis de tráfego é utilizando-se Token Buckets para descrever o tráfego esperado.

Além do perfil de tráfego, a política de tráfego também compreende as ações a serem tomadas caso o tráfego esteja fora do perfil. Estas ações podem compreender desde o mapeamento (através da marcação de pacotes) do tráfego em excesso para uma classe de serviço com pior tratamento, até o descarte de todos os pacotes em excesso.

Um exemplo de perfil de tráfego utilizando Token Bucket para uma implementação de uma política de tráfego é a especificação de tráfego utilizada pela arquitetura de serviços integrados (item 3.1) chamada de TSpec (de Traffic Specification). O TSpec é utilizado para caracterizar o tráfego que receberá um tratamento diferenciado e previamente reservado dos nós de uma rede. O TSpec consiste de três parâmetros informados ao mecanismo de controle de admissão: uma taxa de geração de tokens (parâmetro r), a profundidade do balde (parâmetro b) e uma

taxa de pico (parâmetro p). O mecanismo de controle de admissão é implementado utilizando-se dois Token Buckets em série, sendo o primeiro um Token Bucket com taxa de geração de tokens r e profundidade b e o segundo um Token Bucket com taxa de geração de tokens p e profundidade de 1 token. O tráfego caracterizado pelos dois Token Bucket possui um throughput médio por períodos longos r , o qual pode variar em períodos de curta duração a picos limitados por b , até uma taxa máxima limitada por p . Deve-se notar que apesar da profundidade do balde (b) limitar os picos a um máximo, este limite depende do intervalo de tempo de duração do pico: picos de curta duração podem atingir um throughput maior do que picos de duração maior. O parâmetro p permite um limite facilmente configurável para o throughput dos picos de tráfego [21][11].

Políticas de tráfego podem ser implementadas apenas nas fronteiras administrativas de uma rede, ou em cada nó da rede. O primeiro caso consiste na abordagem utilizada pela arquitetura de serviços diferenciados, e possibilita o controle sobre fontes de tráfego externas à rede. O segundo caso consiste na abordagem utilizada pela arquitetura de serviços integrados [2].

2.6.4 Controle de Admissão

O termo *controle de admissão* se refere à prática de determinar se há recursos disponíveis na rede suficientes para oferecer garantias de qualidade de serviço a um subconjunto do tráfego, e utilizar esta informação para controlar sua entrada [4]. Para isso, os mecanismos de controle de admissão devem possuir conhecimento sobre a alocação dos recursos disponíveis na rede.

O controle de admissão diferencia-se do conceito de política de tráfego apresentado por lidar com a alocação de recursos a novas fontes geradoras de tráfego: representa uma decisão prévia sobre se uma requisição de qualidade de serviço pode ser atendida ou não pelo quantitativo de recursos disponíveis na rede. O controle de admissão pode ser dinâmico ou estático, estando ligado diretamente à forma como o Acordo de Condicionamento de Tráfego (TCA) é negociado entre o usuário e o provedor de serviços.

Tipicamente na arquitetura de serviços integrados o controle de admissão é dinâmico, sendo realizado em cada nó da rede. Na arquitetura de serviços diferenciados devido à falta de um mecanismo que permita a obtenção dinâmica de informações sobre a alocação de recursos na rede, o controle de admissão é feito de forma estática, sendo realizado apenas em suas fronteiras administrativas da rede.

2.7 Serviços Integrados

A arquitetura de *serviços integrados* (*Integrated Services Architecture* ou *IntServ*) propõe a reserva prévia de recursos em uma rede a fim de prover suporte a serviços customizados para diferentes tipos de aplicações. Por exemplo, caso uma aplicação utilizando os serviços integrados necessite de um serviço que forneça algum controle sobre a latência e a perda de pacotes podem ser reservados recursos ao longo da rede que permitam sua implementação. Com esta finalidade, os serviços integrados propõem um conjunto de extensões que permitem a existência de serviços mais direcionados que o tradicional serviço de melhor esforço.

O modelo utilizado nesta arquitetura permite que as aplicações realizem requisições a fim de direcionar a rede a alocar recursos para a transmissão ponto-a-ponto. Estas requisições podem ser passadas aos roteadores por procedimentos de gerenciamento de redes, ou mais geralmente, utilizando um protocolo de reserva de recursos. A rede pode ou aceitar ou rejeitar a requisição de alocação de recursos, mas não pode negociar com as aplicações. Caso a requisição seja aceita, a alocação de recursos pode perdurar por tanto tempo quanto a aplicação permaneça ativa.

A arquitetura de serviços integrados também possui como característica uma alocação de classes de serviço fluxo-a-fluxo em cada nó de uma rede. Isto significa que os mecanismos de reserva de recursos, política e adequação de tráfego em uma rede com suporte aos serviços integrados são implementados em cada nó. Apesar de um fluxo não necessariamente corresponder a uma sessão de uma aplicação, esta característica faz com que a arquitetura de serviços integrados possua uma

granulosidade fina com relação ao tratamento do tráfego, o que tem implicações na escalabilidade e no gerenciamento de recursos[3].

2.7.1 Conceitos da Arquitetura de Serviços Integrados

No contexto desta arquitetura, alguns termos de uso comum são utilizados com significados diferentes [3]. Assim, o termo *elemento de rede* refere-se a qualquer componente da rede que manipule pacotes e seja capaz de impor um controle sobre a qualidade de serviço dos dados que fluem através dele. Já o termo *fluxo* denota um conjunto de pacotes percorrendo um único elemento de rede e possuindo a mesma requisição de controle de qualidade de serviço. Deve-se notar que um *fluxo* representa o subconjunto do tráfego que pode ser gerenciado pelos serviços integrados, podendo ter uma granulosidade que pode variar desde um conjunto de pacotes provenientes de sessões de várias aplicações diferentes até a uma única sessão de uma aplicação.

Do mesmo modo, o termo *serviço de controle QoS* se refere a um conjunto coordenado de capacidades de QoS as quais são mantidas por um único elemento de rede. Como veremos há apenas dois tipos de serviço de controle QoS atualmente especificados: o serviço garantido e o serviço de carga controlada. Um elemento de rede que é capaz de fornecer um ou mais dentre os serviços de controle QoS especificados na arquitetura de serviços integrados é chamado de elemento *QoS-capaz*. No mesmo contexto, o termo *comportamento* denota a performance de QoS fim-a-fim observada por uma sessão de uma aplicação. O *comportamento* é o resultado da composição dos serviços oferecidos por cada elemento de rede ao longo do caminho do fluxo de dados [3].

A arquitetura de serviços integrados foca-se principalmente em um tipo de aplicação conhecido como de *tempo real*. Estas aplicações são caracterizadas por um tempo de reprodução (o qual pode ser fixo ou adaptativo) no receptor, de tal modo que pacotes de tempo real chegando após o tempo de reprodução são descartados. Aplicações de tempo real podem ser classificadas em duas categorias: as que são *tolerantes* e as que são *intolerantes* ao jitter. Aplicações tolerantes, tais como várias implementações de áudio e vídeo, podem funcionar bem em caso de variações de

latência e mesmo assim possuir uma boa qualidade de sinal quando reproduzido. Já as aplicações intolerantes, como por exemplo a telefonia em IP, têm a qualidade de seu sinal distorcida em caso de jitter, levando a uma qualidade de sinal inaceitável ou mesmo ao não funcionamento da aplicação.

2.7.2 Componentes dos Elementos de Rede

Em cada elemento QoS-capaz existem quatro componentes principais, os quais constituem as funções de controle de tráfego: o mecanismo de controle de admissão de fluxos, o escalonador de pacotes, o classificador e a implementação de um protocolo de reserva de recursos. A ação destes componentes permite a diferenciação do tráfego e a implementação dos serviços de controle QoS.

O *mecanismo de controle de admissão* tem como função determinar se há recursos suficientes no elemento de rede para receber uma nova requisição de reserva de recursos para um fluxo, sem afetar os fluxos já estabelecidos. Dependendo dos recursos disponíveis, a requisição de reserva pode ser aceita ou negada pelo elemento de rede. A sinalização entre os diversos elementos de rede e entre os nós origem e destino necessária para a reserva de recursos e para o resultado da requisição é fornecida pelo protocolo de reserva de recursos.

O *classificador de pacotes* possui como principal função separar cada pacote em uma classe de serviço e encaminhá-los ao escalonador de pacotes. As classes de serviço em geral correspondem a cada fluxo com reserva de recursos aceita pelo mecanismo de controle de admissão. Entretanto pode haver agregação da reserva de diversos fluxos, o que permite a implementação de elementos de rede onde apenas as políticas de tráfego são realizadas em separado para cada fluxo. A implementação de classes de serviço com granulosidade fluxo-a-fluxo permite um melhor monitoramento do estado de cada fluxo e da reserva de recursos [13].

O *escalonador de pacotes* é responsável por gerenciar as funções de disciplina de fila do elemento de rede, de modo a alocar os recursos para cada classe de serviço de acordo com a reserva realizada. Assim, por exemplo, um pacote pertencente a um fluxo com reserva de recursos requerendo uma latência induzida máxima no elemento

será tratado de forma diferente de outro pertencente a um fluxo com requisitos de throughput mínimo. A disciplina de fila utilizada é dependente da implementação e do serviço a ser fornecido, sendo os mais comuns já discutidos no capítulo 2 .

Finalmente o *protocolo de reserva de recursos* é utilizado para configurar o estado de fluxo nos sistemas fim-a-fim requisitados e também para realizar a reserva de recursos de cada elemento de rede no percurso do fluxo. O protocolo de reserva de recursos mais utilizado é o RSVP [6].

O modelo mostrado na figura 11 representa uma implementação dos quatro componentes detalhados anteriormente em um roteador implementando os serviços integrados [1]. Pode-se notar que na figura há uma clara distinção entre as funções de gerenciamento e controle (mostrados na parte superior) e os mecanismos de encaminhamento de dados (mostrados na parte inferior). No modelo, o classificador de pacotes e o escalonador são implementados por mecanismos específicos. Já o controle de admissão é implementado utilizando-se duas funções de gerenciamento (base de dados de controle de tráfego e função de controle de admissão) que atuam junto ao escalonador de pacotes e ao encaminhador. Finalmente o protocolo de reserva de recursos é implementado utilizando-se um agente de inicialização da reserva que atua juntamente com outras funções de gerenciamento.

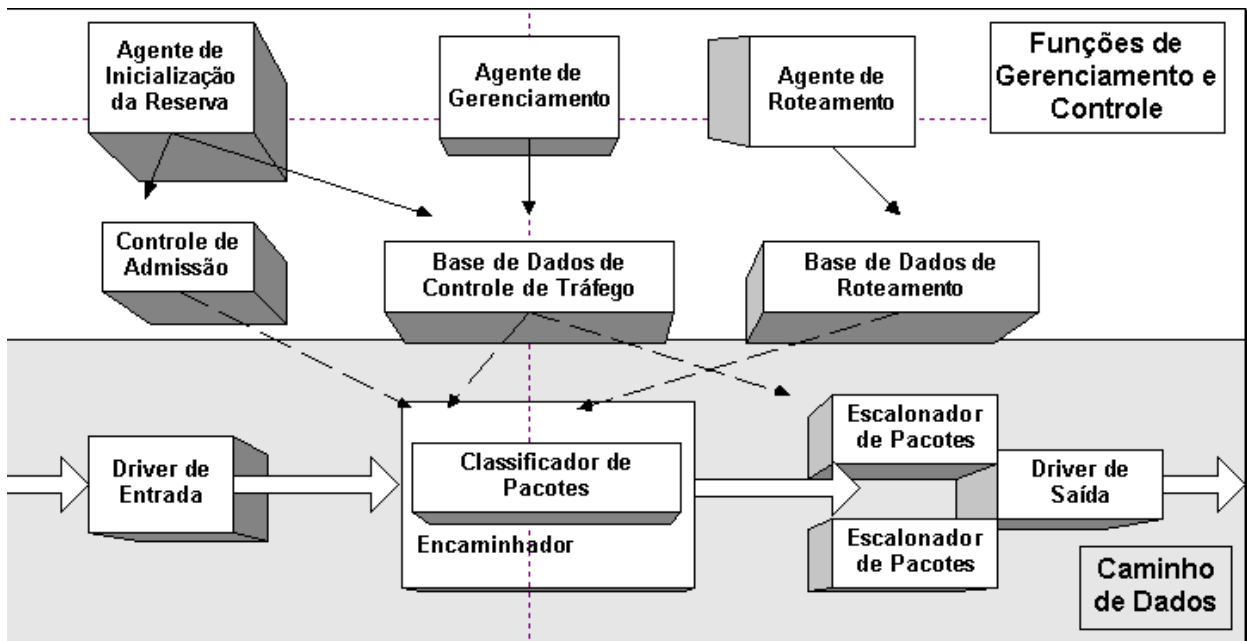


Figura 13: Modelo de Referência de implementação de um Elemento QoS capaz [1]

Os mecanismos de encaminhamento de dados representam as funções básicas do roteador executadas para todos os pacotes. O driver de entrada recebe os pacotes do meio físico (interface) e os repassa ao mecanismo encaminhador de pacotes. O encaminhador classifica os pacotes e utiliza informações dentre as funções de gerenciamento de tráfego para encaminhá-los a um dos escalonadores de pacotes nas interfaces (drivers) de saída. Dentre estas informações destacam-se o roteamento do pacote, e o controle de tráfego do fluxo ao qual o pacote pertence. Finalmente o driver de saída representa a interface por onde o pacote é transmitido.

As funções de gerenciamento e controle possuem a incumbência de criar estruturas de dados que controlem o encaminhamento de pacotes. O agente de roteamento implementa um protocolo de roteamento particular e constrói uma base de dados de roteamento. O agente de inicialização da reserva implementa o protocolo de reserva de recursos em uso. A função de controle de admissão gerencia as requisições de reserva de recursos admitindo-as ou não a partir da base de dados de controle de tráfego. Finalmente, o agente de gerenciamento possibilita a modificação das bases de dados de classificação de pacotes e de escalonamento para configurar o compartilhamento de recursos do roteador e as políticas de controle de admissão[1][2].

2.7.3 O Protocolo de Reserva de Recursos (RSVP)

O *protocolo de reserva de recursos* (RSVP) permite que as aplicações e os elementos de rede se comuniquem, a fim de configurar os mecanismos necessários para suportar os serviços de controle QoS descritos anteriormente. O RSVP não é um protocolo de roteamento sendo utilizado apenas para reservar recursos ao longo de rotas já existentes e configuradas utilizando o protocolo de roteamento em uso na rede.

Existe uma separação lógica entre os serviços integrados e o RSVP. Serviços de controle QoS, tais como o serviço garantido e o de carga controlada, podem ser utilizados com outros protocolos de sinalização e controle. Por exemplo, outros protocolos de reserva de recursos em IP tal como o ST-II podem ser utilizados sem prejuízo algum para a funcionalidade dos serviços. Além disso, o RSVP não define o formato interno dos objetos relacionado à caracterização dos serviços de controle QoS: estes objetos são opacos ao protocolo. Isto permite que o RSVP seja utilizado fora da arquitetura de serviços integrados, como atualmente está sendo proposto para a arquitetura de serviços diferenciados e para o MPLS. Em outras palavras, o RSVP é simplesmente o protocolo de sinalização e os serviços de controle de QoS são o conteúdo. O RSVP é atualmente o mecanismo de configuração de reserva de recursos o qual possui o maior suporte para o uso com os serviços integrados[3].

2.7.4 As mensagens RSVP

O RSVP utiliza duas mensagens primárias no processo de reserva de recursos: a mensagem *PATH*, a qual se origina do transmissor e a mensagem *RESV*, a qual se origina do receptor. As mensagens *PATH* estabelecem informações de estado nos elementos de rede entre o transmissor e o receptor, de modo que cada elemento de rede possa armazenar informações que o permitam enviar pacotes para o próximo elemento do caminho no sentido do transmissor ou do receptor, criando o que é chamado de estado de caminho (*path state*). O estado de caminho pode ser estabelecido entre o transmissor e um ou vários receptores (a reserva de recursos pode ser unicast ou multicast) permitindo que haja um único caminho entre o transmissor e cada receptor durante uma sessão. Além disso, esta mensagem fornece

informações sobre o tráfego a ser gerado pelo transmissor aos receptores e sobre o próprio caminho, a fim de que possam ser realizadas as reservas apropriadas.

Já a principal função da mensagem RESV é a de transportar requisições de reservas aos elementos QoS-capazes ao longo da árvore de distribuição entre os transmissores e os receptores. A mensagem RESV faz com que os elementos de rede no caminho reservem recursos e armazenem informações sobre a reserva, mantendo o que é chamado de estado de reserva (reservation state).

Tanto o estado de caminho quanto o de reserva são conhecidos como estados do tipo *soft*. Um estado *hard* requer a configuração de circuitos virtuais durante toda a duração da sessão de transferência de dados. Por contraste, um estado *soft* só requer a configuração de circuitos virtuais, utilizando mensagens periódicas para manter a reserva de recursos em cada elemento de rede.

Devido ao fato do RSVP ter sido projetado como um protocolo de reserva de recursos em separado da arquitetura de serviços integrados, a especificação do RSVP não define o formato interno dos campos relacionados com a invocação dos serviços QoS. Em lugar disso, a especificação do RSVP define objetos para a sinalização da reserva de recursos os quais são opacos com relação ao protocolo: o conteúdo de cada objeto pode variar conforme a arquitetura e o serviço em que o RSVP será utilizado[12].

2.7.5 Objetos do Protocolo RSVP especificados para o uso com a arquitetura de Serviços Integrados

O RSVP, quando utilizado na arquitetura de serviços integrados, utiliza três objetos para a sinalização da reserva de recursos: o objeto *AdSpec*, o *FlowSpec* e o *Sender_TSpec*. Os objetos *AdSpec* e *Sender_TSpec* são utilizados na mensagem PATH, enquanto o *FlowSpec* é utilizado na mensagem RESV [6].

O objeto *Sender_Tspec* é utilizado pelas aplicações transmissoras para descrever os parâmetros do tráfego que será gerado e pelos serviços de controle QoS para descrever os parâmetros para o qual a reserva deve ser aplicada. Esta especificação é realizada modelando-se o tráfego através de um Token Bucket . O

Token Bucket é descrito por uma taxa média de reposição de símbolos [r] e pela profundidade do balde [b]. Opcionalmente também pode ser utilizada uma taxa de pico [p], representando a taxa máxima a qual o transmissor e quaisquer pontos de readequação de tráfego podem transmitir na rede. Outros parâmetros também são utilizados no *Sender_TSpec*: o tamanho mínimo do pacote a ser controlado pelas políticas de controle de tráfego [m], e um tamanho máximo de pacote [M].

O objeto *AdSpec* é utilizado para transportar informações as quais (ao contrário do *Sender_TSpec*) podem ser alteradas no caminho entre o transmissor e o receptor. Estas informações incluem parâmetros descrevendo as características do caminho, tais como a quantidade de nós e uma estimativa do throughput disponível, e parâmetros específicos de cada serviço de controle QoS.

Finalmente, o objeto *FlowSpec* transporta requisições de reserva de recursos gerados pelo receptor, além de uma descrição do tráfego esperado. O *FlowSpec* é composto pelo serviço de controle QoS desejado e uma especificação de tráfego (*Receiver_TSpec*). O *Receiver_TSpec* é composto pelos mesmos parâmetros do *Sender_TSpec* tendo como função descrever o perfil de tráfego para o qual os recursos devem ser reservados. Opcionalmente, de acordo com o serviço de controle QoS requisitado uma especificação do nível de serviço desejado também é incluída (*RSpec*). Estas informações são transmitidas no sentido receptor-transmissor através do caminho configurado na mensagem PATH inicial, e podem ser utilizadas ou atualizadas em elementos de rede intermediários antes de chegarem ao receptor.

2.7.6 O processo de Reserva de Recursos

Um resumo da operação do RSVP dentro da arquitetura de serviços integrados em uma reserva de recursos unicast é ilustrado a seguir, a partir da figura 12 :

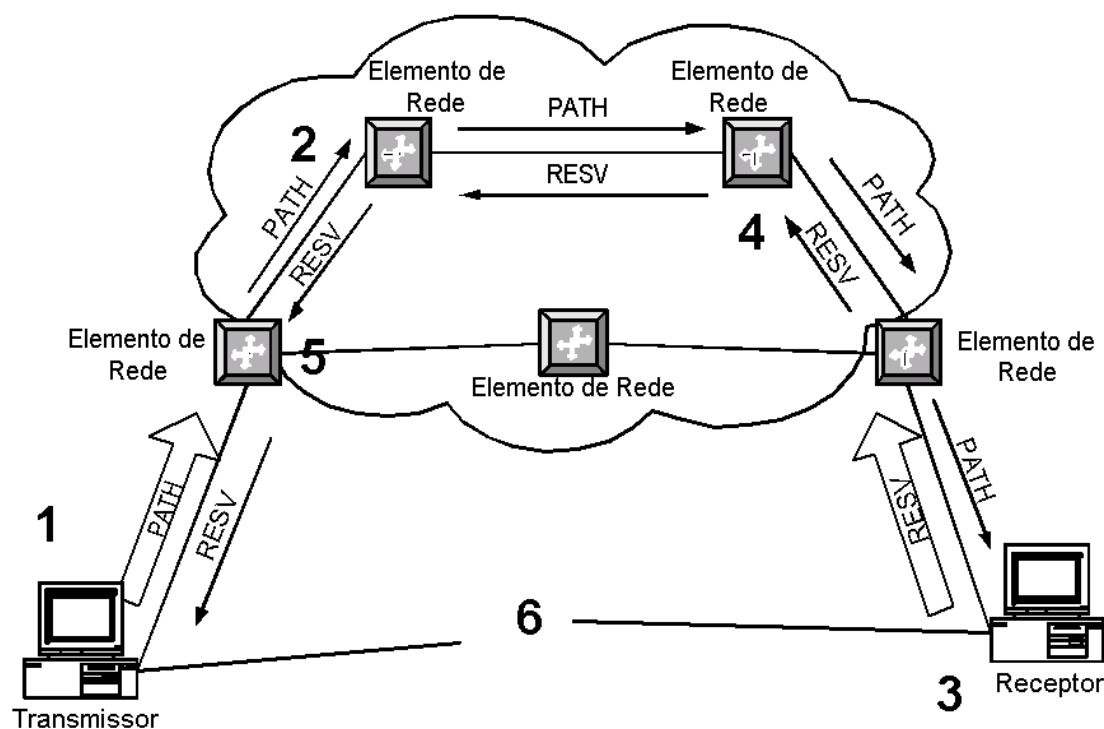


Figura 14: Reserva de Recursos Unicast em uma rede com serviços diferenciados[12]

1. os transmissores geram mensagens PATH com informações sobre o tráfego esperado. O objeto Sender_TSpec é utilizado para descrever o tráfego que o transmissor espera gerar, enquanto o parâmetro Sender_Template é utilizado para identificar o transmissor. Além disso, um parâmetro AdSpec pode ser criado a fim de anunciar aos receptores as características do caminho de comunicação fim a fim. A mensagem PATH é enviada na direção do receptor e encaminhada de acordo com o protocolo de roteamento utilizado na rede;
2. em cada elemento de rede QoS-capaz no caminho entre o transmissor e o receptor, o TSpec e o parâmetro Sender Template são utilizados para requisitar uma reserva apropriada de recursos do serviço de controle QoS desejado. Além disso, o parâmetro AdSpec é atualizado com informações descrevendo o caminho de rede entre o transmissor e o receptor e parâmetros relacionados com os serviços de controle QoS. O TSpec, o AdSpec, o Sender Template e um parâmetro Phop identificando o elemento de rede anterior QoS-capaz são

utilizados para criar uma caracterização do estado de caminho para o fluxo do transmissor em questão;

3. após a chegada da mensagem PATH no receptor, os dados dos parâmetros TSpec e AdSpec são utilizados para guiar na seleção da reserva de recursos. Então, uma mensagem RESV com os parâmetros da reserva é gerada e enviada de volta ao transmissor. A mensagem RESV é composta pelo objeto FlowSpec, contendo parâmetros de especificação do tráfego a ter recursos reservados, o serviço de controle QoS requisitado e uma especificação do nível de serviço (RSpec – gerada apenas para o serviço garantido) ;
4. em cada elemento de rede mantendo o estado do caminho criado com a mensagem PATH original, a mensagem RESV é utilizada para reservar os recursos apropriados para o serviço de controle QoS desejado. O caminho é novamente percorrido no sentido do receptor para o transmissor utilizando-se as informações de estado do caminho em cada elemento de rede, e particularmente as informações do parâmetro Phop da mensagem PATH original;
5. quando o último elemento de rede recebe o RESV e aceita a requisição, ele envia uma confirmação de volta ao receptor;
6. finalmente, o fluxo é transmitido do transmissor para o receptor, recebendo os parâmetros de qualidade requisitados. Mensagens periódicas são enviadas através do caminho entre o transmissor e o receptor para renovar o estado do caminho em cada elemento de rede. Caso a mensagem periódica não seja enviada em um tempo preestabelecido o estado do caminho e o estado de reserva são rompidos nos elementos de rede. Isto também pode ocorrer explicitamente, a partir de uma mensagem de *TEAR-DOWN* originária do transmissor ou do receptor.

2.7.7 Serviço Garantido

O serviço de Controle QoS garantido (*Guaranteed Service*) permite o fornecimento de um throughput assegurado, uma latência limitada, e perda de pacotes

mínima em filas de elementos da rede para o tráfego de fluxos de dados que se adequem à política de admissão. Este serviço computa apenas a latência introduzida pelas filas dos elementos de rede no caminho fim-a-fim – a latência fixa gerada por fatores tais como a velocidade de propagação da luz não é controlada. Do mesmo modo a variação da latência não é controlada ou minimizada diretamente. Entretanto o serviço fornece meios para limitar os valores máximos que a latência pode atingir.

O serviço garantido é invocado através da especificação do tráfego (Receiver_TSpec) e do serviço desejado (RSpec) a cada elemento de rede. Tanto o RSpec quanto o Receiver_TSpec fazem parte do objeto FlowSpec da mensagem RESV. O RSpec ou requisição de especificação de serviço representa a especificação da qualidade de serviço que um fluxo deseja requisitar de um elemento da rede. O RSpec consiste de uma taxa de dados (R) e de um termo solto (S) ou *slack term*. A taxa de dados representa o throughput previsto para o fluxo. O termo solto representa a diferença entre a latência desejada e a latência obtida utilizando-se um nível de reserva de taxa de dados R. Ele pode ser utilizado pela rede para reduzir a reserva de recursos para o fluxo. Já o Receiver_TSpec representa uma descrição do tráfego para os quais os recursos serão reservados. O RSpec é utilizado apenas pelo serviço garantido, não fazendo parte da sinalização do serviço de carga controlada.

Dois tipos de políticas de tráfego estão associados com o serviço garantido: políticas simples (*simple policing*) e de readequação (*reshaping*). As políticas simples são implementadas tipicamente nas fronteiras administrativas das redes e são utilizadas para assegurar a observância do Receiver_TSpec. A readequação consiste em uma tentativa de modelar as características do tráfego de modo que o mesmo esteja de acordo com a especificação da reserva de recursos, consistindo na implementação de um adequadador de tráfego utilizando os parâmetros do Receiver_TSpec. Em geral, a readequação cria um pouco mais de latência, entretanto é capaz de reduzir a variação de latência total do fluxo. Quaisquer pacotes que não sejam aceitos pela readequação ou pelas políticas simples são tratados como tráfego de melhor esforço[12].

2.7.8 Serviço de Carga Controlada

O serviço de controle QoS de carga controlada (*Controlled-Load Service* ou *CL*) fornece aos fluxos de dados dos clientes um comportamento de tráfego fim-a-fim o qual se aproxima ao obtido pelo serviço de melhor esforço tradicional quando em condições de pouco tráfego. Em outras palavras, a latência e a perda de pacotes para os datagramas em um fluxo de carga controlada não se deterioram perceptivelmente à medida que o tráfego da rede aumenta. Isto ocorre independentemente do aumento de tráfego observado. Em contraste, um fluxo de melhor esforço experimentaria progressivamente um pior serviço (maior latência e perda de pacotes) à medida em que a carga da rede aumentasse.

O serviço de carga controlada é invocado através de uma especificação do tráfego para o qual deverá ser realizada a reserva de recursos em cada elemento de rede. Caso o protocolo de reserva de recursos seja o RSVP, isto é especificado através do *Receiver_TSpec*, o qual faz parte do objeto *FlowSpec* da mensagem *RESV*. Há algumas diferenças no uso dos objetos RSVP para o serviço de carga controlada em relação ao serviço garantido. O uso do parâmetro de taxa de pico (*p*) do *Receiver_TSpec* é opcional, sendo seu uso não previsto pela especificação atual do serviço de carga controlada. Ao mesmo tempo, como o serviço de carga controlada não apresenta nenhuma garantia quantitativa, não há nenhuma requisição de especificação de serviço (*Request Specification* ou *RSpec*).

Cada elemento de rede deve checar se os fluxos de carga controlada estão de acordo com seus *TSpec*. Isto deve ser feito para evitar que fluxos de dados que não estejam nesta condição afetem o QoS oferecido a outros fluxos de carga controlada ou mesmo ao tráfego de melhor esforço. Além disso, cada elemento de rede deve tentar encaminhar o tráfego em excesso como tráfego de melhor esforço, caso existam recursos suficientes para isso.

O serviço de carga controlada pode ser implementado facilmente utilizando vários algoritmos de escalonamento que permitam a diferenciação de tráfego, tais como o PRR ou o WFQ. Para as políticas de tráfego há duas abordagens possíveis [12]. Uma é a de diminuir a prioridade do tráfego em excesso, sendo a forma como esta

abordagem é implementada dependente da implementação. Uma segunda é a de diminuir a prioridade do tráfego de todo um fluxo de carga controlada caso o mesmo possua tráfego em excesso. Esta segunda abordagem possui a vantagem de apresentar respostas mais rápidas para a diminuição do tráfego em excesso.

2.7.9 Considerações sobre a Arquitetura de Serviços Integrados

Há certo consenso de que a arquitetura de serviços integrados e o RSVP são excessivamente complexos e possuem problemas de escalabilidade. Com relação ao RSVP, a maior dificuldade reside no fato de que os requisitos de recursos (processamento computacional e consumo de memória) para sua execução em roteadores crescem em proporção direta em relação ao número de reservas (sessões) acomodadas. Assim, o suporte a um grande número de reservas RSVP envolveria um impacto negativo significativo no roteador. A performance de encaminhamento dos roteadores também pode ser impactada negativamente pelos mecanismos de classificação de pacotes e de escalonamento utilizados para fornecer a diferenciação de serviços. Em especial, o serviço garantido apresenta alguns desafios para os mecanismos de escalonamento, devido aos requisitos de alocar níveis absolutos de throughput para fluxos individuais, e ao mesmo tempo, suportar uma classe de serviços de melhor esforço[1].

Atualmente há várias propostas interessantes no IETF para estudos futuros que sugerem métodos para agrupar fluxos de serviços garantidos com o fim de reduzir os requisitos de throughput que cada fluxo pode consumir individualmente. Ao mesmo tempo, há trabalhos envolvendo simplificações do RSVP, a fim de procurar solucionar alguns destes problemas [4].

2.8 Serviços Diferenciados

Os *serviços diferenciados* ou *DiffServ* (de *Differentiated Services*) surgiram como uma resposta às preocupações sobre a escalabilidade do uso dos serviços integrados em uma rede de grande porte como a Internet. Esta abordagem propõe a implementação de uma arquitetura de qualidade de serviço baseada na identificação

do tratamento a ser aplicado em pacotes IP a partir do conteúdo de um campo no cabeçalho de cada pacote (campo de serviços diferenciados ou *campo DS*) [4]. Não há a necessidade de diferenciação no tratamento ofertado em separado para o tráfego de cada aplicação com requisitos de qualidade de serviço: tanto os recursos reservados no interior da rede como os mecanismos de políticas e adequação de tráfego são diferenciados para cada pacote apenas pelo conteúdo do campo DS [4]. Isto possibilita que a arquitetura de serviços diferenciados seja simples e escalável quando comparada com o modelo proposto pelos serviços integrados.

Deve-se notar que a idéia de uso de um campo do cabeçalho IP para indicar os requisitos de qualidade de serviço de cada pacote não é nova, tendo já sido proposta anteriormente com a especificação do *campo de tipo de serviço* ou *TOS* (de *Type-Of-Service*) definidos no cabeçalho do protocolo IP. O campo TOS possui 8 bits, sendo que dentre estes apenas quatro são utilizados para especificar os requisitos de qualidade de serviço. Cada um dos quatro bits é utilizado para indicar um requisito de qualidade a ser atingido pela rede: maximizar a confiabilidade, minimizar a latência, maximizar o throughput ou minimizar o custo monetário. Apenas um dentre os requisitos pode ser indicado por pacote. Os bits não utilizados para indicar os requisitos de qualidade são utilizados para sinalizar a precedência a ser aplicada aos pacotes [1][4].

O campo TOS nunca foi largamente utilizado conforme sua idéia original. Falhas na definição, a qual tornou-se excessivamente genérica, tornaram seu uso muitas vezes inadequado para especificar o tipo de serviço desejado. Além disso, neste modelo são as próprias aplicações as responsáveis por marcar o tipo de serviço desejado em cada pacote IP. Como não há nenhuma política de admissão de pacotes definida, nada impede que as aplicações gerem tanto tráfego com uma determinada especificação de tipo de serviço que a mesma não possa ser atingida.

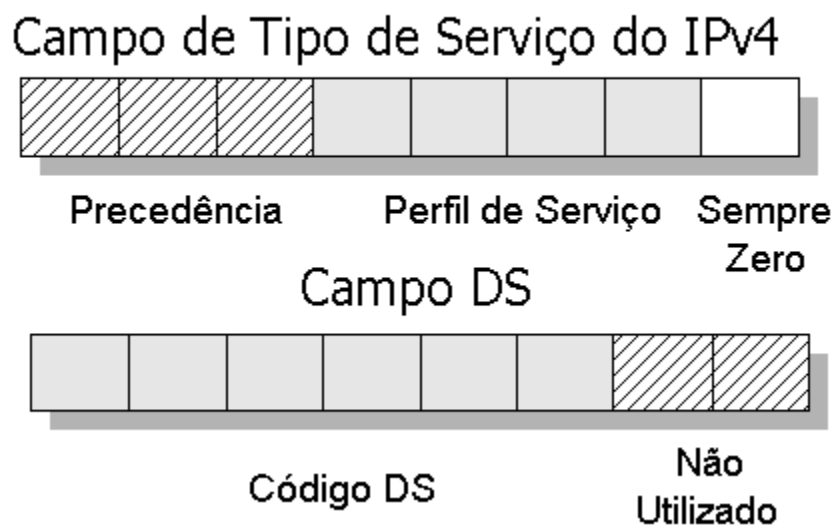


Figura 15: Sintaxes do Campo DS e do campo de tipo de serviço do protocolo IPv4[2]

A arquitetura de serviços diferenciados apresenta considerável diferença quando comparada com o modelo de tipo de serviço. Em uma rede de serviços diferenciados, cada roteador possui um conjunto de tratamentos pré-definidos, sendo o campo DS utilizado meramente para identificar qual dentre os tratamentos será aplicado a cada pacote. Assim os tratamentos a serem ofertados e o campo DS são independentes: um é utilizado apenas para identificar o outro, não para especificá-lo. Os serviços diferenciados são compostos de uma série de elementos funcionais implementados nos nós da rede, incluindo funções de classificação de pacotes, mecanismos de condicionamento de tráfego e de controle de admissão. Os conceitos e mecanismos relacionados com esta arquitetura serão discutidos a seguir.

2.8.1 Conceitos da Arquitetura de Serviços Diferenciados

Os conceitos e funcionalidades relacionados com a arquitetura de serviços diferenciados podem ser separados em quatro partes distintas: os serviços ofertados aos clientes pela rede; o campo DS; as funções de classificação e condicionamento de tráfego; e a implementação utilizada em cada nó do interior da rede dos tratamentos ofertados aos pacotes. As duas primeiras partes serão discutidas a seguir; sendo as restantes apresentadas nos itens 2.8.2 e 2.8.3.

Os serviços representam os tipos de tratamento que podem ser oferecidos ao tráfego dos clientes em uma rede implementando os serviços diferenciados [5]. Em geral, os serviços devem ser especificados por provedores de acesso (ISP) com redes implementando os serviços diferenciados através de *acordos de níveis de serviços* com os clientes ou SLA (de *Service Layer Agreements*). Um SLA basicamente define os serviços suportados e a quantidade de tráfego permitida em cada classe de serviço. Em particular, a arquitetura de serviços diferenciados define um subconjunto de um SLA padrão, o qual descreve os componentes relacionados com os serviços diferenciados. Tal subconjunto é chamado de *acordo de condicionamento de tráfego* ou **TCA** (de *Traffic Conditioning Agreement*). O TCA especifica parâmetros de serviço detalhados para cada classe de serviço desejada, tais como probabilidade de descarte de pacotes, latência esperada, throughput esperado, e disposição do tráfego em excesso. Estas classes de serviço podem ser definidas pelo provedor de serviços, pelo usuário, ou geralmente via negociação do SLA/TCA entre as partes [4].

Os códigos DS são associados a tratamentos definidos para os pacotes em cada nó intermediário, à medida em que eles cruzam a rede. Para a versão 4 do protocolo IP o campo utilizado é o mesmo especificado para uso com o modelo de tipo de serviço (byte TOS), tendo entretanto uma sintaxe diferente, conforme mostrado na figura 13. Já na versão 6 do protocolo IP, o campo DS corresponde ao campo de classes de serviço previsto na especificação do próprio protocolo.

Cada conjunto de pacotes com o mesmo código DS cruzando a rede em determinada direção é chamado de *agregação de comportamentos* (*Behavior Aggregate* ou *BA*) . Por sua vez, o tratamento recebido por uma agregação em um determinado nó da rede suportando serviços diferenciados é chamado de *comportamento-por-salto* (*Per-Hop-Behavior* ou *PHB*). Deve-se notar que a associação entre o código DS e o comportamento-por-salto correspondente é local para cada rede, havendo apenas alguns códigos previamente definidos e associados a comportamentos-por-salto. Assim um provedor de serviços pode associar um determinado código DS com um comportamento-por-salto que especifique baixa

latência, enquanto outro pode associar o mesmo código com um comportamento-por-salto que maximize o throughput [3][5].

Um conjunto contíguo de nós que operam com os mesmos comportamentos-por-salto e as mesmas políticas de provisionamento de tráfego é chamado de *domínio de serviços diferenciados* [5]. Tipicamente, um domínio de serviços diferenciados compreende uma ou várias redes locais sob a mesma administração, como por exemplo a Intranet de uma organização ou um provedor de acesso .

Deve-se notar que os comportamentos-por-salto representam **apenas** o tratamento a ser ofertado aos pacotes de determinada agregação: a forma como são implementados pode variar, desde que atenda às especificações de cada comportamento-por-salto. Isto significa que um mesmo comportamento-por-salto pode ser implementado de várias formas diferentes em cada nó, mesmo para nós em um mesmo domínio de serviços diferenciados.

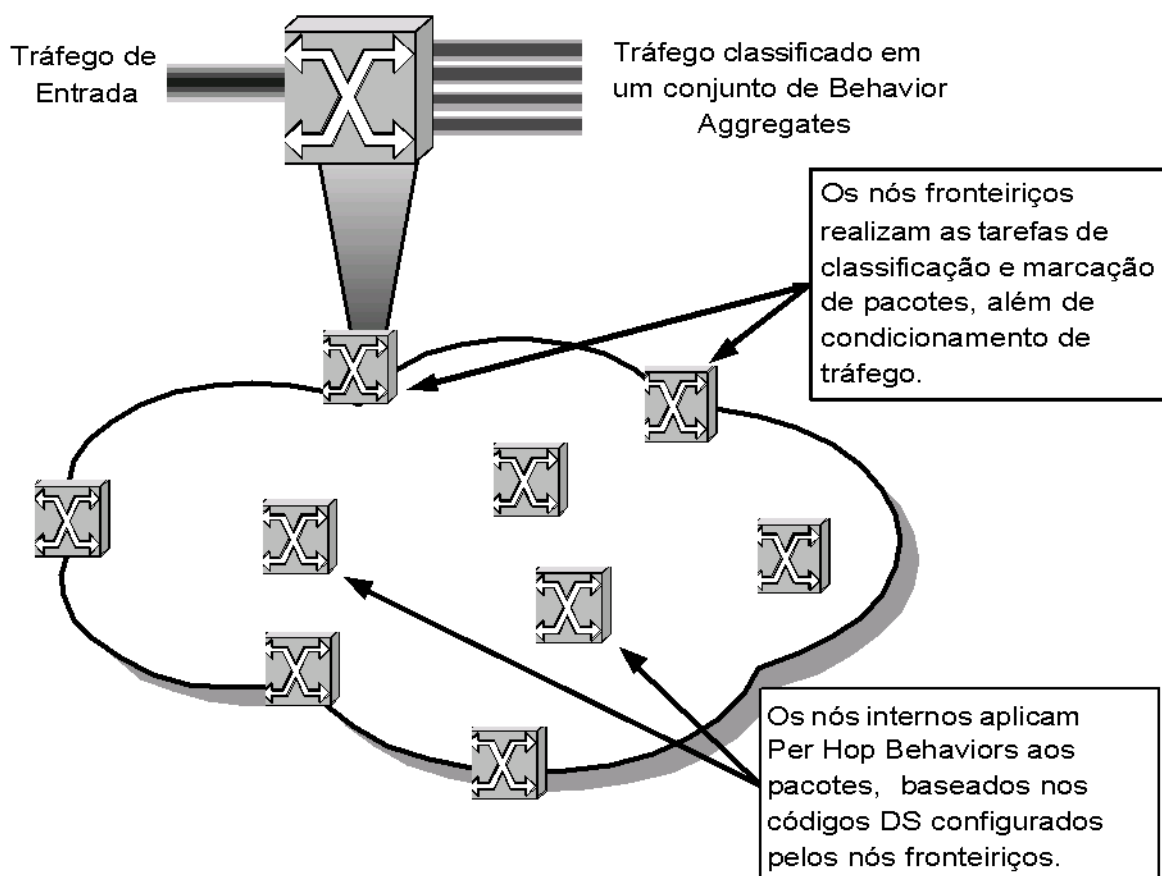


Figura 16:A arquitetura de serviços diferenciados

Cada domínio de serviços diferenciados possui uma região conhecida como fronteira, a qual é delimitada pelos nós que conectam o domínio de serviços diferenciados a outro domínio o qual pode ou não suportar serviços diferenciados. Tais nós são chamados de nós fronteirizos (border nodes) em contraste com os nós restantes do domínio, chamados de nós internos.

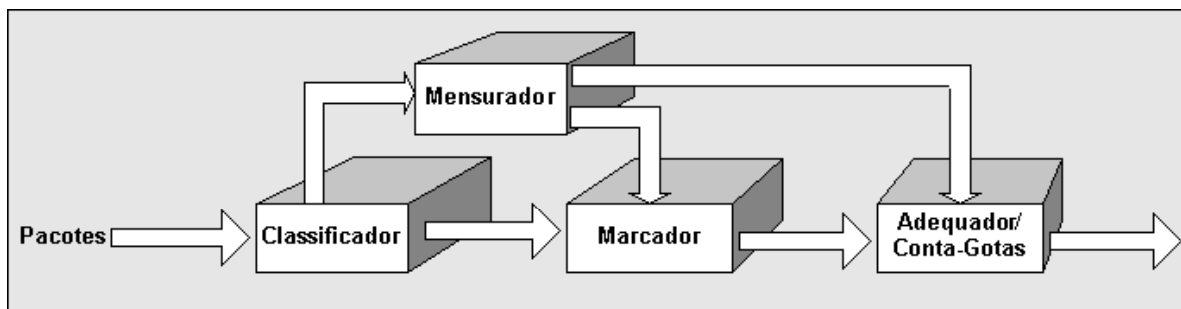
2.8.2 Classificação e Condicionamento do Tráfego

As funções de classificação e condicionamento do tráfego geralmente são realizadas por nós fronteirizos para o tráfego que entra e potencialmente para o que sai do domínio de serviços diferenciados. Com isso, as funções mais complexas são mantidas nas bordas da rede, de modo a preservar a escalabilidade e a performance da mesma. Entretanto, apesar de não ser recomendável, nada impede que nós internos também realizem estas tarefas [4]. A figura 14 mostra a Arquitetura de serviços diferenciados com relação às funcionalidades dos nós internos e fronteirizos.

A *função de classificação de tráfego (classificador)* seleciona pacotes baseado no conteúdo de alguma porção do cabeçalho do mesmo. Para isso, é necessário que sejam configuradas regras de admissão, a serem utilizadas pelo nó classificador e definidas de acordo com o SLA e a política administrativa do domínio. O conjunto do tráfego compreendendo um ou mais fluxos de pacotes fim-a-fim entre aplicações e selecionado pelo classificador de pacotes é chamado de *corrente de tráfego (traffic stream)*. O tráfego pode ser dividido em uma ou mais correntes, segundo o desejado administrativamente. Há dois tipos de classificadores possíveis em um domínio de serviços diferenciados. Um é o classificador que se baseia apenas no campo DS, chamado de classificador de *agregação de comportamentos* ou classificador *BA* (de *Behavior Aggregate*). Um exemplo do uso desta classificação poderia ser em um nó fronteiro, na conexão entre dois domínios de serviços diferenciados. O outro tipo de classificador seleciona pacotes baseado em uma combinação de um ou mais cabeçalhos, tais como endereço de origem, endereço destino, campo DS e protocolo, sendo chamado de classificador *multi-campo* ou *MF* (de *Multi-Field*) [4].

O condicionamento de tráfego geralmente está associado a uma parte específica do TCA a qual descreve o perfil de tráfego a ser utilizado para modelar o tráfego que irá receber um tratamento diferenciado. Perfis de tráfego são componentes opcionais de Acordos de Condicionamento de Tráfego e seu uso depende dos detalhes do serviço oferecido e da política adotada. Condicionadores de tráfego podem possuir até quatro elementos diferentes dependendo da funcionalidade desejada: mensurador, marcador de pacotes, adequador e conta-gotas (figura 15).

Os *mensuradores* (meters) são responsáveis por verificar se determinada corrente de tráfego selecionada pelo classificador está dentro ou fora de seu perfil de tráfego. O mensurador também informa sobre o status de cada corrente de tráfego a outros mecanismos, de modo a acionar uma ação particular caso determinada corrente de tráfego esteja fora de seu perfil. Já os Marcadores de Pacotes (Packet Markers) modificam ou verificam o campo DS de cada pacote em uma corrente de tráfego, de



modo a associar o mesmo a uma agregação de comportamentos.

Figura 17: Representação das funções de Classificação e Condicionamento de Tráfego[5]

Os *marcadores de pacotes* dependem das informações do mensurador sobre cada corrente de tráfego. Dependendo da adequação de cada corrente de tráfego ao seu perfil, os pacotes podem ser marcados para uma agregação associada a um comportamento-por-salto com maior prioridade ou para outro com menor prioridade.

O marcador e o mensurador de pacotes são componentes que necessitam ser especificados em conjunto, já que o marcador depende das informações do mensurador para associar pacotes a agregações. A utilização de diferentes algoritmos no mensurador e no marcador possibilita tanto a criação de Acordos de Condicionamento de Tráfego (TCA) com parâmetros customizáveis quanto a criação

de políticas de tráfego específicas para os comportamentos-por-salto em uso. As principais implementações de mensuradores e marcadores de pacotes em estudo ou especificadas pelo IETF são:

- **Token Bucket simples ou Single Rate Two Color Marker:** o marcador/mensurador Token Bucket utiliza dois parâmetros para marcar os pacotes de uma corrente de tráfego: *Comitted Information Rate* ou CIR, e *Comitted Burst Size* (CBS). O mensurador é implementado utilizando-se um algoritmo de Token Bucket para classificar os pacotes de cada corrente de tráfego como adequados ao perfil de tráfego ou não: o parâmetro CIR corresponde à taxa de reposição de símbolos do algoritmo enquanto o CBS corresponde ao tamanho do balde. Os pacotes podem ser marcados com dois códigos DS apelidados de “verde” e “vermelho”: pacotes são marcados como verdes se o CBS não é excedido e como vermelhos em caso contrário [35] ;
- **Single Rate Three Color Marker:** o SRTCM utiliza três parâmetros para marcar os pacotes de uma corrente de tráfego: *Committed Information Rate* (CIR), *Committed Burst Size* (CBS), e *Excess Burst Size* (EBS). De acordo com estes parâmetros os pacotes podem ser marcados com três diferentes códigos DS, conhecidos como verde, amarelo ou vermelho. O SRTCM é implementado utilizando-se um mensurador que utiliza dois Token Bucket conhecidos como C e E para verificar se determinada corrente de tráfego está ou não dentro de um perfil. O parâmetro CIR corresponde à taxa de reposição de símbolos de ambos os Token Bucket, o CBS como o tamanho do balde do Token Bucket C e o EBS como o tamanho do balde do E. Enquanto o balde C possuir símbolos, os pacotes são marcados como verdes; quando o balde C não possuir símbolos e o balde E possuir, como amarelos; e quando nenhum possuir como vermelhos. A idéia é a de permitir que os pacotes de uma corrente de tráfego sejam mapeados para três comportamentos-por-salto diferentes, permitindo uma maior customização dos throughput máximos e da duração dos mesmos [56] ;
- **Two Rate Three Color Marker (TRTCM):** o TRTCM utiliza quatro parâmetros para determinar como os pacotes de uma corrente de tráfego serão marcados:

Peak Information Rate (PIR), *Peak Bucket Size (PBS)*, *Comitted Information Rate (CIR)* e *Comitted Burst Size (CBS)*. Do mesmo modo como no SRTCM, o mensurador é implementado a partir de dois Token Bucket (conhecidos como P e C); tendo entretanto parâmetros diferentes: os parâmetros CBS e CIR correspondem ao tamanho do balde e à taxa de reposição de símbolos do Token Bucket C, enquanto os parâmetros PBS e PIR corresponde ao tamanho do balde e à taxa de reposição de símbolos do Token Bucket P. Enquanto o balde C possuir símbolos os pacotes são marcados como verdes, quando o balde C não possuir símbolos e o balde E possuir, como amarelos; e quando nenhum possuir como vermelhos. Um exemplo de uso do TRTCM seria na implementação de um TCA onde os pacotes verdes teriam um throughput garantido com taxa CIR e uma variação máxima do throughput determinada pelo CBS ; os pacotes em excesso marcados como amarelos seriam descartados com determinada probabilidade enquanto se adequassem a um throughput PIR com uma variação PBS; e os pacotes marcados como vermelhos fossem descartados. O SRTCM é um caso especial do TRTCM onde a PIR e a CIR são iguais;

- *Time Sliding Window Marker (TSWTCM)*: O TSWTCM consiste de um mensurador que estima o throughput de correntes de tráfego dentro de um intervalo de tempo pré-determinado e de um marcador de pacotes capaz de utilizar as informações do mensurador para marcar o campo DS com três diferentes valores (vermelho, amarelo e verde). A marcação dos pacotes é realizada baseada na comparação entre o throughput estimado de cada fluxo e um perfil de tráfego composto de dois parâmetros chamados de Comitted Target Rate (CTR) e Peak Target Rate (PTR). Fluxos de tráfego com throughput menor que a CTR possuem todos seus pacotes marcados como verdes. Pacotes pertencendo a fluxos com throughput entre a CTR e a PTR podem ser marcados como verdes ou amarelos, de acordo com uma função probabilística proporcional à fração do throughput mensurado que excede a CTR. Do mesmo modo, pacotes pertencendo a fluxos de tráfego com throughput maior que a PTR

podem ser marcados como verdes, amarelos ou vermelhos, de acordo com uma função probabilística calculada a partir da fração de pacotes que excede a PTR e da fração de pacotes entre a PTR e a CTR [4].

O *adequador* (shaper) atrasa alguns ou todos os pacotes em uma corrente de tráfego, a fim de tentar adaptar a corrente ao seu perfil de tráfego associado. Para isso, o adequador recebe informações do mensurador. Um adequador geralmente possui um buffer de tamanho finito, e utiliza o algoritmo de Token Bucket para atrasar ou descartar pacotes.

Um outro componente é o mecanismo de conta-gotas (*dropper*), o qual descarta alguns ou todos os pacotes em uma corrente de tráfego, a fim de tentar adaptá-la ao perfil de tráfego associado a partir de informações recebidas do mensurador. Este processo é chamado de “política de tráfego” da corrente. Note que o conta-gotas pode ser implementado como um caso especial de um adequador, através da configuração do tamanho do buffer do adequador para zero.

2.8.3 Comportamentos-por-salto - PHB

O *comportamento-por-salto* ou *PHB* (de *Per-Hop-Behavior*) é uma descrição do tratamento externamente observável aplicado em um nó implementando a arquitetura de serviços diferenciados a uma agregação de comportamentos [1][5]. Cada comportamento-por-salto pode ser especificado em termos da prioridade de algum recurso da rede particular (por exemplo buffer, descarte durante congestionamentos) em relação a outros PHB ou em termos de suas características de tráfego observáveis (por exemplo, latência, perda de pacotes). Quando existe um requisito comum que faz com que comportamentos-por-salto só possam ser implementados e especificados simultaneamente o conjunto de comportamentos-por-salto é chamado de *grupo de comportamentos-por-salto*. Tipicamente isto acontece quando os comportamentos-por-salto possuem características em comum as quais se aplicam a todos os comportamentos-por-salto dentro do grupo, tais como políticas de gerenciamento de buffers ou de escalonamento de pacotes. Um único PHB pode ser considerado como um caso especial de um grupo de comportamentos-por-salto com apenas um elemento

[5].

Atualmente há apenas quatro tipos de comportamento-por-salto especificados pelo IETF: os grupos de comportamentos-por-salto de encaminhamento assegurado, de encaminhamento expresso, de Seletor de Classes e o PHB Default[4].

2.8.4 Comportamento-por-salto Default

O comportamento-por-salto Default foi especificado de modo a representar o encaminhamento de pacotes por “melhor esforço” atualmente existente nas redes IP em um domínio DS [58]. A implementação deste PHB é obrigatória em qualquer nó que suporte serviços diferenciados. Ao mesmo tempo, caso existam pacotes que não possuam nenhum TCA associado deve-se assumir que os mesmos pertencem a este PHB. Recomenda-se ainda o uso do código DS com o valor binário “000000”, para os comportamentos-por-salto que irão receber o tratamento relacionado com o PHB Default.

2.8.5 Grupo de Comportamentos-por-salto Class Selector

O grupo de comportamentos-por-salto Class Selector foi especificado a fim de permitir que a arquitetura de serviços diferenciados preserve parcialmente a compatibilidade com o uso do campo de precedência do cabeçalho IP [2]. Este campo utiliza três bits do byte de tipo de serviço (TOS), a fim de definir uma prioridade de entrega entre pacotes, de modo que um pacote com prioridade maior não chegue ao seu destino após um pacote com prioridade menor. Ao mesmo tempo, define uma ordem de descarte em caso de congestionamento da rede, a qual é preferencial para os pacotes com menor precedência.

Para isso, a definição dos códigos DS deste grupo procura manter a mesma semântica estabelecida para o campo de precedência. Assim, o PHB Class Selector é formado por um grupo de PHB que correspondem a um conjunto de prioridades relativas para o tráfego. Da mesma maneira, os códigos DS associados a cada PHB mantém a semântica definida para o campo de precedência, ocupando os primeiros 3 bits do campo DS. Deste modo, os códigos DS no formato “xxx000” são reservados para o uso com este grupo de PHB definindo 8 diferentes códigos e mantendo a

compatibilidade com o código “000000” utilizado pelo Default PHB. Ao mesmo tempo, os PHB mapeados para os 8 códigos DS devem gerar pelo menos duas diferentes classes de encaminhamento de modo que pacotes com valores maiores no código DS sejam interpretados como possuindo maior prioridade relativa em relação aos que possuem valores menores [2]. Os PHB pertencentes a este grupo podem ser facilmente implementados utilizando alguns dos mecanismos de enfileiramento vistos anteriormente na seção 2, incluindo-se WFQ, WRR e PQ.

2.8.6 Grupo de Comportamentos-por-salto de Encaminhamento Assegurado (AF PHB)

O grupo de comportamentos-por-salto de *encaminhamento assegurado* (*Assured Forwarding PHB* ou *AF PHB*) é voltado para aplicações que necessitam de serviços confiáveis, mesmo em caso de congestionamentos [1]. Ele permite que um provedor de serviços ofereça vários níveis de confiabilidade de entrega para pacotes IP, de acordo com Acordos de Condicionamento de Tráfego (*Traffic Conditioning Agreement* ou *TCA*) com os usuários.

Neste grupo de comportamentos-por-salto, pacotes em excesso com relação ao TCA são considerados como possuindo maior probabilidade de serem descartados. Uma segunda característica é a de que todos os pacotes são entregues na mesma ordem em que foram transmitidos, mesmo quando excedentes. Estas qualidades fazem com que o encaminhamento assegurado seja adequado para aplicações baseadas em TCP, nas quais pacotes individuais em uma conexão podem ser descartados e o reordenamento de pacotes é indesejável por apresentar sinais de congestionamento falsos à pilha TCP.

Os diferentes níveis de confiabilidade são criados a partir de quatro classes e de três valores de descarte de pacote para cada classe, totalizando doze comportamentos-por-salto. A cada classe são alocadas determinadas quantidades de recursos (throughput e espaço em buffer) em cada nó do domínio DS. Ao mesmo tempo, pacotes IP utilizando os serviços possibilitados pelo AF PHB, são alocados pelo usuário ou pelo provedor a uma ou mais destas classes, de acordo com o

especificado no TCA.

Em cada classe, os pacotes IP são marcados (novamente pelo usuário ou pelo provedor) com um dentre três possíveis valores de precedência de descarte. Em caso de congestionamento, a precedência de descarte de um pacote determina sua importância relativa dentro da classe. Assim, um nó localmente congestionado tentará proteger os pacotes com menores valores de precedência de descarte, descartando preferencialmente os pacotes com valores maiores. Cada nó implementando o encaminhamento assegurado deve suportar todos os três valores de precedência de descarte para cada classe.

Os acordos de condicionamento de tráfego para o uso deste grupo de PHB especificam o throughput alocado por classe e por usuário. Cada classe deve obter a taxa de serviço estabelecida no TCA (throughput) tanto em escalas de tempo pequenas quanto em longas. A fim de atingir este objetivo, todo nó DS deve ser configurado para permitir um mínimo de espaço em buffer local e throughput para cada uma das quatro classes de encaminhamento assegurado. Isto pode ser feito através de uma configuração fixa, ou os nós podem ser configurados para oferecer uma quantidade maior de recursos a uma determinada classe, caso haja capacidade disponível de outras classes ou de outros grupos de PHB.

As classes e os valores de precedência de descarte podem ser utilizados por um provedor para customizar os serviços oferecidos. Por exemplo, três classes de serviço poderiam ser criadas: bronze, prata e ouro. Os pacotes poderiam ser alocados para estas três classes de modo que os pacotes na classe ouro obtivessem maior quantidade de recursos (throughput e espaço no buffer) do que os da classe prata, possuindo assim maior confiabilidade de que serão entregues. Se desejado, os pacotes em cada classe poderiam ser separados ainda por valores de precedência de descarte, os quais poderiam ser baixo, médio ou alto.

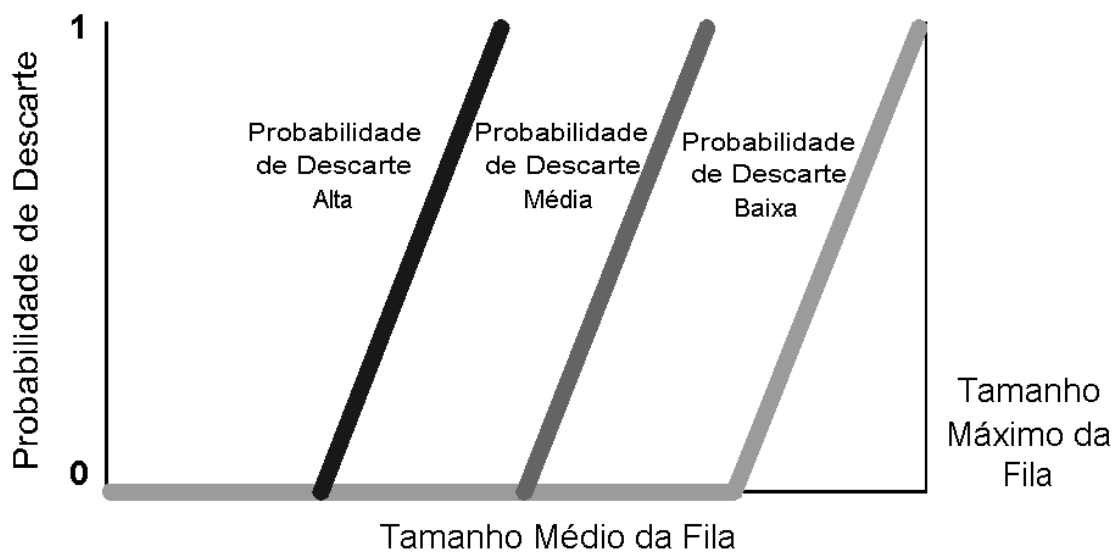


Figura 18: WRED configurado para suportar as classes de serviço AF[4]

Para implementar as classes de encaminhamento assegurado, cada nó precisa implementar algum tipo de alocação de throughput para cada classe, com um tamanho de buffer associado. Uma abordagem seria utilizar WRR para implementar cada classe, com os pesos ajustados para refletir as alocações de throughput feitas a cada classe de serviço. Para implementar os valores de precedência de descarte poderia ser utilizado o algoritmo WRED, configurado de modo a que o peso associado a cada pacote corresponda aos valores de precedência de descarte existentes no código DS, conforme mostrado na figura 16, para três diferentes precedências de descarte.

2.8.7 Comportamento-por-salto de Encaminhamento Expresso (EF PHB)

O comportamento-por-salto de *encaminhamento expresso* (*Expedited Forwarding PHB* ou EF PHB) garante um serviço fim-a-fim com throughput assegurado, baixa latência e variação de latência para clientes que geram tráfego com taxas máximas de transmissão fixas. Tal serviço assemelha-se a uma conexão ponto a ponto ou a uma linha dedicada virtual através de uma rede IP heterogênea, sendo também conhecido como serviço *Premium*. Estas características permitem que este PHB suporte aplicações de tempo real inelástica tais como videoconferência ou telefonia [1].

O EF PHB baseia-se na observação de que a perda de pacotes, a latência e o jitter são todos resultados dos efeitos resultantes das filas de pacotes existentes nos nós da rede. Assim para prover qualidade de serviço para uma agregação de comportamentos de encaminhamento expresso é necessário assegurar que os pacotes enfrentem filas mínimas.

Como as filas são geradas quando a taxa de recebimento de tráfego excede a taxa de envio do mesmo, para implementar o encaminhamento expresso em um domínio de serviços diferenciados duas condições devem ser satisfeitas. Primeiro cada nó deve ser configurado de modo que a agregação de comportamentos possua uma taxa de transmissão assegurada a qual é independente de qualquer outro tráfego existente na rede. Tal taxa deve permanecer válida quando medida durante qualquer intervalo de tempo igual ou superior ao tempo de transmissão de um pacote com o tamanho máximo permitido (MTU) e deve ser passível de configuração por parte do administrador.

Segundo, a agregação de comportamentos deve ser condicionada de modo que a taxa com a qual os pacotes são entregues em qualquer nó seja sempre menor que a taxa de transmissão configurada. Este condicionamento é geralmente implementado nos nós fronteiriços.

O TCA para um EF PHB especifica uma taxa de transmissão de pico para um fluxo específico, ou um conjunto de fluxos. O cliente é responsável por não exceder a taxa de pico, caso contrário o tráfego excedente será descartado. Este PHB pode ser implementado utilizando um algoritmo de fila de prioridade (PQ) simples, desde que não haja filas com prioridades maiores que possam alterar a taxa configurada para o EF por intervalos maiores que o de transmissão de um pacote.

2.8.8 Considerações sobre a Arquitetura de Serviços Diferenciados

A arquitetura de serviços diferenciados possui duas desvantagens principais. Uma é a fraca abordagem que esta arquitetura possui com relação ao gerenciamento de recursos de rede, o que é um resultado direto da granulosidade em que o tráfego é diferenciado (geralmente como um agregado de fluxos de diferentes aplicações). A

outra é a falta de um modelo de serviços fim-a-fim como o existente nos serviços integrados: não existem mecanismos que permitam que as aplicações negociem com a rede o perfil de serviço a ser mantido, ou mesmo a alocação dinâmica de recursos no interior de um domínio de serviços diferenciados. Isto poderá ser implementado ou através de alguma forma de descoberta dinâmica, ou através de alguma forma de negociação tal como o RSVP é utilizado na arquitetura de serviços integrados. Invariavelmente este modelo não será tão detalhado quanto o dos serviços integrados, entretanto faz pouco sentido a uma aplicação proceder com um perfil de serviço premium (encaminhamento expresso) se a rede não possui recursos suficientes para atendê-lo [1].

Os serviços diferenciados apresentam características que o tornam de maior utilidade no backbone de redes de grande e médio porte. O tratamento do tráfego por agregação e a concentração dos mecanismos de políticas, marcação e adequação de tráfego nas bordas da rede fazem com que esta arquitetura sejam simples e escalável[4].

2.9 Interoperação entre os Serviços Diferenciados e os Serviços Integrados

As diferentes características dos serviços integrados e dos serviços diferenciados fazem com que nenhuma destas arquiteturas possa ser vista como uma solução completa para a implementação de qualidade de serviço na Internet. Enquanto os serviços integrados apresentam problemas de escalabilidade, dificultando sua implementação em redes de grande porte, os serviços diferenciados apresentam mecanismos simples e voltados para a agregação de microfluxos, o que dificulta o gerenciamento e a alocação de recursos. Uma arquitetura que permita a interoperação entre os serviços integrados e os serviços diferenciados possibilitaria a adoção das arquiteturas nos locais onde seu uso é mais vantajoso e a obtenção das melhores características de cada uma para a implementação de QoS.

A figura abaixo extraída de [2] ilustra o uso de uma arquitetura híbrida.

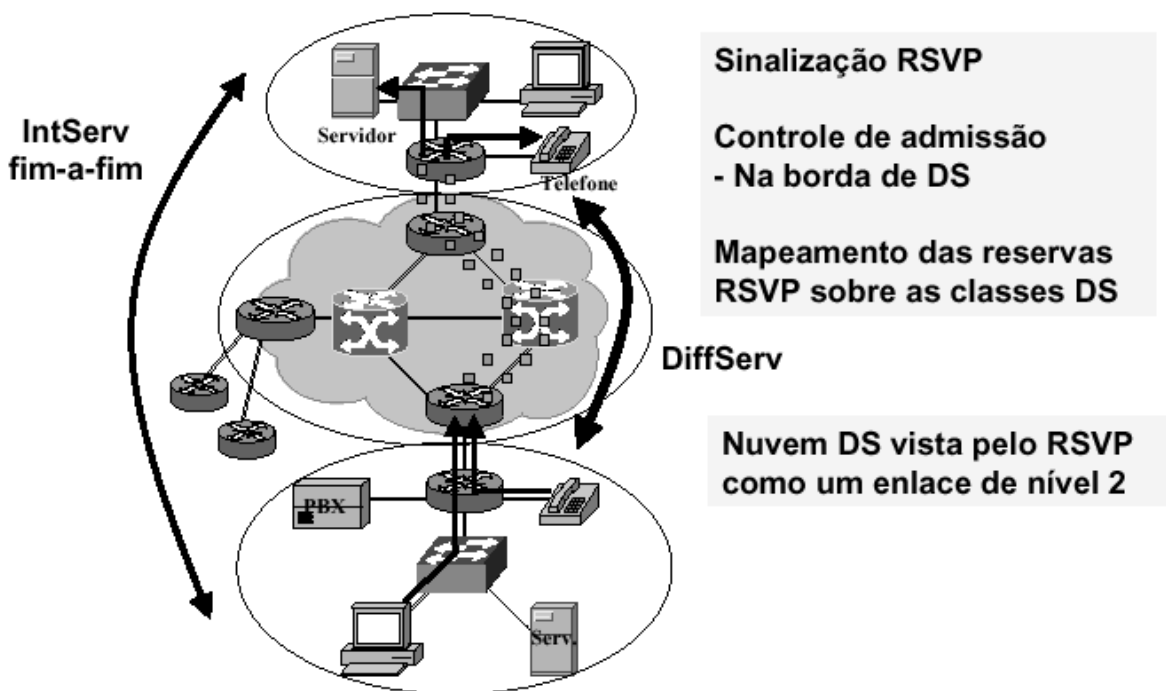


Figura 19: Ilustração de funcionamento da arquitetura híbrida

2.9.1 Descrição da Arquitetura Híbrida

A arquitetura para a interoperação entre os serviços diferenciados e os serviços integrados especificada pelo IETF [8] envolve a adoção do modelo de reserva de recursos com RSVP utilizado pelos serviços integrados através de uma rede contendo um ou mais domínios de serviços diferenciados. Os domínios DiffServ podem mas não são obrigados a participar da sinalização e da reserva de recursos RSVP: as regiões com serviços diferenciados podem tanto suportar o RSVP e serem capazes de efetuar sinalização e controle de admissão por fluxos, quanto serem capazes apenas de encaminhar mensagens RSVP de forma transparente.

Nesta arquitetura, os domínios DiffServ atuam como componentes de uma rede fim-a-fim implementando QoS utilizando os padrões de sinalização, reserva de recursos e de classes de serviço do modelo de serviços integrados. Em outras palavras, para as aplicações que requisitam qualidade de serviço uma rede utilizando a arquitetura híbrida deve ser semelhante a redes implementando apenas os serviços integrados, mesmo que haja uma ou mais regiões que não suportem seus

mecanismos. Com este fim, os diferentes componentes existentes em nós QoS capazes devem ser mapeados para mecanismos implementáveis em regiões com serviços diferenciados [8]. A figura 20 mostra de forma simplificada a utilização de um domínio de serviços diferenciados como um elemento de rede de serviços integrados.

As características de tráfego obtidas em cada nó de serviços integrados são implementadas através do mapeamento de fluxos para agregações de tráfego utilizando comportamentos-por-salto apropriadamente selecionados, configurados e com recursos alocados. Estes comportamentos-por-salto quando concatenados ao longo dos nós no interior do domínio DiffServ devem prover características de tráfego que adequadamente se aproximem dos resultados esperados para o serviço QoS de serviços integrados reservado para cada fluxo. A alocação de recursos para os comportamentos-por-salto tanto pode ser feita dinamicamente utilizando-se um protocolo de sinalização como o RSVP como estaticamente, conforme será visto mais tarde.

Em termos de mecanismos de classificação e marcação de tráfego, a necessidade de identificação do fluxo a que cada pacote pertence e a grande variação de granulosidade possível para fluxos de serviços integrados requer que estas funções sejam realizadas por classificadores MF. A localização dos classificadores e marcadores pode estar tanto em roteadores localizados nas bordas dos domínios de serviços diferenciados, quanto em roteadores externos aos mesmos. Os roteadores externos compreendem aqueles adjacentes a roteadores fronteira de domínios DiffServ, sendo conhecidos como *roteadores limítrofes (border routers)*[7].

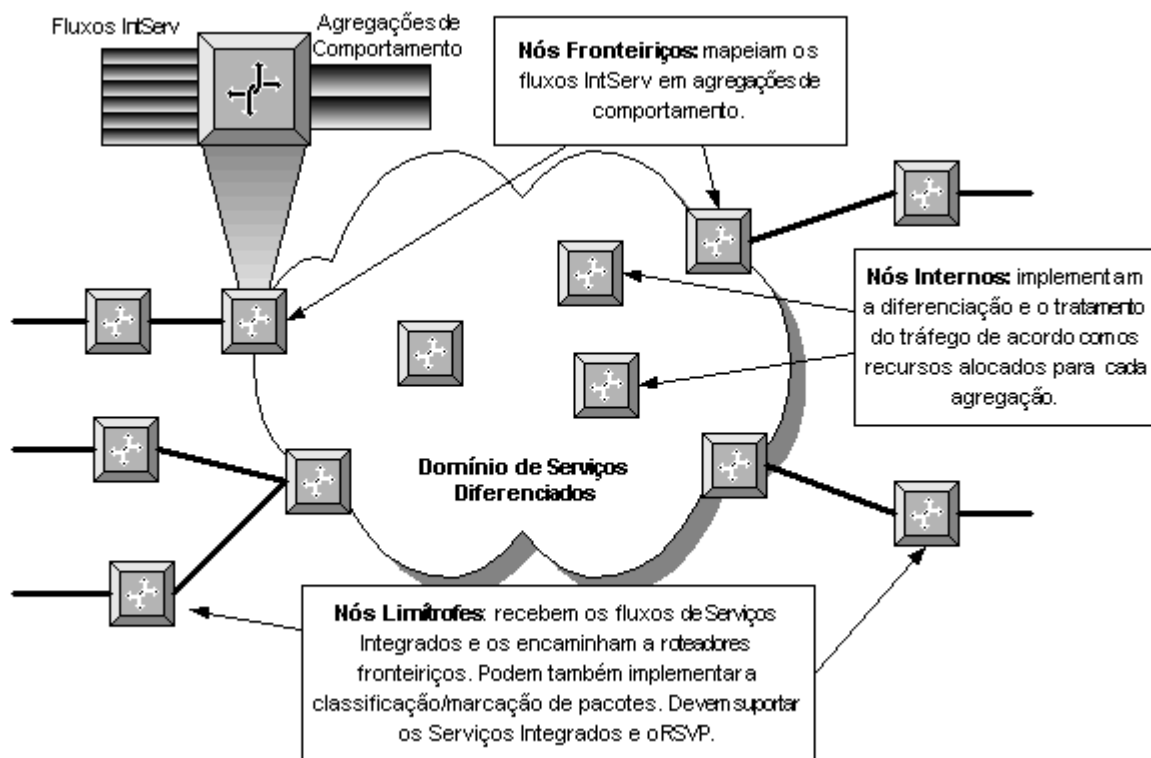


Figura 20: Modelo simplificado da arquitetura híbrida IntServ/DiffServ[4]

O uso de roteadores limítrofes possui a vantagem de possibilitar uma maior escalabilidade ao empurrar os mecanismos para nós fora do domínio e que ordinariamente são em maior número que os roteadores fronteira. A superioridade numérica dos roteadores limítrofes se deve ao fato da arquitetura de serviços diferenciados possuir características que fazem com que os domínios DiffServ sejam implementados em áreas com maior tráfego, centrais à rede e que recebem o tráfego de várias regiões adjacentes. Uma outra vantagem é a de possibilitar que toda a sinalização relacionada com a alocação de recursos necessária para a configuração dinâmica dos mecanismos seja feita externamente ao domínio. Como veremos, isto permite implementações em que domínios DiffServ participando em redes com arquitetura híbrida não tenham necessariamente de suportar o protocolo de sinalização (RSVP). Entretanto, o uso de roteadores fronteira também possui desvantagens as quais são mais evidentes quando as regiões externas aos domínios DiffServ são administradas em separado. Neste caso, são necessários acordos de prestação de

serviços que devem ser estritamente observados para garantir a manutenção das características de qualidade no interior dos domínios.

Os mecanismos de política e de adequação de tráfego também podem ser implementados tanto em roteadores limítrofes como em fronteiriços. Do mesmo modo como para os mecanismos de classificação e marcação de pacotes, o uso de roteadores fronteiriços apresenta um problema de escalabilidade, o qual é agravado pela maior quantidade de recursos exigida quando é realizada a diferenciação do tráfego por fluxo IntServ. A diferenciação por fluxo possibilita que os mecanismos sejam implementados de forma semelhante à realizada nos nós IntServ. Entretanto, a quantidade de recursos exigida em sua implementação faz com que deva ser disponibilizada preferencialmente em roteadores limítrofes.

Uma alternativa ao uso de roteadores limítrofes para propiciar escalabilidade é a implementação de políticas de adequação de tráfego por agregação de fluxos nas bordas dos domínios DiffServ. Esta abordagem é considerada preferencial quando os mecanismos de política e adequação de tráfego são localizados nos roteadores fronteiriços [7], possuindo como vantagens melhores escalabilidade e facilidade de gerenciamento em relação à abordagem por fluxo. Entretanto, também possibilita que fluxos mal-comportados obtenham mais recursos da rede do que os reservados para estes dentro da agregação, degradando desta maneira o serviço provisionado a outros fluxos. Deve-se notar que apesar das políticas de adequação de tráfego por agregação em roteadores limítrofes também serem especificados para a arquitetura de serviços diferenciados [4], só são considerados preferenciais na arquitetura híbrida [8].

A arquitetura especificada em [8] não descreve detalhes específicos do suporte dos serviços diferenciados aos requisitos dos serviços QoS IntServ. Este suporte inclui vários aspectos dentre os quais podem ser citados a política de controle de admissão, a seleção do PHB ou do grupo de PHB para o serviço requisitado e a atualização de parâmetros utilizados na sinalização da reserva de recursos dos serviços integrados a partir de regiões DiffServ.

Há diversas características que dificultam a implementação deste suporte. As principais são o tratamento de tráfego por agregação e a reserva e alocação dinâmica

de recursos. O tratamento do tráfego por agregação é uma característica da arquitetura de serviços diferenciados que se opõe ao tratamento por fluxo dos serviços integrados: em cada nó interior de uma rede DiffServ os recursos são alocados para toda a agregação de tráfego, sem diferenciação entre os diversos fluxos que a compõe. Dependendo da implementação, as políticas e a adequação de tráfego também podem ser realizadas por agregação. Isto possibilita que o tráfego em excesso de fluxos mal comportados possa disputar recursos dentro da agregação com o tráfego de fluxos bem comportados, degradando a qualidade de serviço obtida por estes[4].

A reserva e a alocação dinâmica de recursos é uma característica da arquitetura de serviços integrados que foi especificada para a arquitetura híbrida com o uso do RSVP [49]. Contudo, não há implementações semelhantes especificadas para a arquitetura de serviços diferenciados: inexistem mecanismos que permitam quantificar dinamicamente os recursos alocados no interior de um domínio DiffServ ou sinalizar sua disponibilidade; o que faz com que o controle de admissão e a consequente reserva de recursos tenham de ser realizados a partir de informações estáticas. Existem diversas propostas em desenvolvimento com relação a este problema, as quais deverão permitir sua solução no futuro.

2.9.2 Suporte ao RSVP

Na arquitetura híbrida os domínios DiffServ podem suportar ou não o protocolo de sinalização de reserva de recursos utilizado pelas regiões implementando os serviços integrados.

O RSVP é utilizado para comunicar os requisitos quantitativos de qualidade de serviço e de sinalização da reserva de recursos. Assume-se para a arquitetura híbrida que as mensagens RSVP trafegam fim-a-fim e que são pelo menos encaminhadas através de toda a rede [8]. Entretanto não se assume que todos os elementos de rede tenham de ser capazes de processar mensagens RSVP. Mesmo as regiões externas aos domínios DiffServ podem compreender vários tipos de elementos de rede os quais não necessariamente devem suportar os serviços integrados e a sinalização RSVP. Nas regiões em que a arquitetura IntServ não é suportada, a rede deve ser super-

dimensionada em relação ao tráfego esperado de modo a evitar congestionamentos e a degradação da qualidade de serviço oferecida [8].

Para domínios DiffServ em uma rede com a arquitetura híbrida o processo de sinalização e reserva de recursos varia conforme há o suporte ou não ao RSVP. Quando este protocolo não é suportado, o domínio DiffServ é incapaz de realizar reserva de recursos dinamicamente, de acordo com requisições dos clientes. O controle de admissão é realizado pelos roteadores limítrofes, de acordo com critérios definidos estaticamente para cada classe de serviço [7]. Nesta situação, estes roteadores processam mensagens de sinalização RSVP do transmissor (Tx) e do receptor (Rx). Esta abordagem torna difícil o suporte a classes de serviço que mudam com frequência, já que as alterações envolvem reconfigurações dos roteadores fronteiros. Um outro problema é a alocação ineficiente dos recursos da rede, sem levar em conta a disponibilidade de recursos ao longo do caminho específico a ser impactado por cada requisição de reserva[4].

Nos casos em que a região com serviços diferenciados suporta o RSVP, os roteadores fronteiros e alguns ou mesmo todos os roteadores internos são capazes de processar mensagens RSVP. Neste caso, como os agentes de controle admissão são parte do domínio, alterações no quantitativo de recursos disponíveis na rede DiffServ podem ser indicadas aos nós IntServ externos utilizando RSVP. Esta abordagem possui como vantagens o aumento da eficiência na alocação de recursos e da confiabilidade na disponibilidade dos recursos alocados. O aumento da eficiência está relacionado com a possibilidade de se alterar a alocação de recursos baseado nas requisições dos clientes. Já o aumento da confiabilidade está ligado à participação dos nós interiores ao domínio DiffServ na sinalização RSVP, o que possibilita que o controle de admissão leve em conta os recursos disponíveis ao longo dos caminhos impactados por cada reserva.

2.9.3 Exemplo de Funcionamento da Arquitetura Híbrida

Um resumo da operação do RSVP em um rede implementando a arquitetura híbrida é mostrado a seguir. Neste exemplo, é realizada uma reserva *RSVP unicast*

para um fluxo gerado por uma aplicação RSVP/IntServ entre uma estação transmissora e uma estação receptora :

1. O transmissor (Tx) gera uma mensagem RSVP PATH com a descrição do tráfego que se espera ser gerado pela aplicação transmissora. A mensagem PATH é transmitida na direção da estação receptora (Rx). Em cada elemento de rede no caminho entre Rx e Tx, o processamento padrão RSVP/IntServ é realizado nos elementos de rede QoS capazes;
2. no roteador limítrofe L1 a mensagem PATH é submetida ao processamento RSVP padrão, e o estado PATH é estabelecido no roteador. A mensagem PATH é enviada para a região DiffServ;
3. caso os roteadores internos ao domínio DiffServ não suportem o processamento de mensagens RSVP, a mensagem será ignorada pelos roteadores no domínio sendo processada apenas pelos roteadores limítrofes L1 e L2. Se o domínio suportar o processamento RSVP, então a mensagem PATH será processada pelos roteadores fronteira sendo estabelecido o estado PATH. A mensagem também poderá ser processada por um ou mais dentre os roteadores internos;
4. quando a mensagem PATH atinge a estação receptora Rx, o sistema operacional gera uma mensagem RSVP RESV, a qual é enviada de volta para a estação transmissora. Esta mensagem é processada pelos elementos de rede de acordo com a especificação RSVP/IntServ, o que significa que pode ser rejeitada por qualquer nó QoS-capaz no caminho caso não haja recursos suficientes para atender à requisição de reserva;

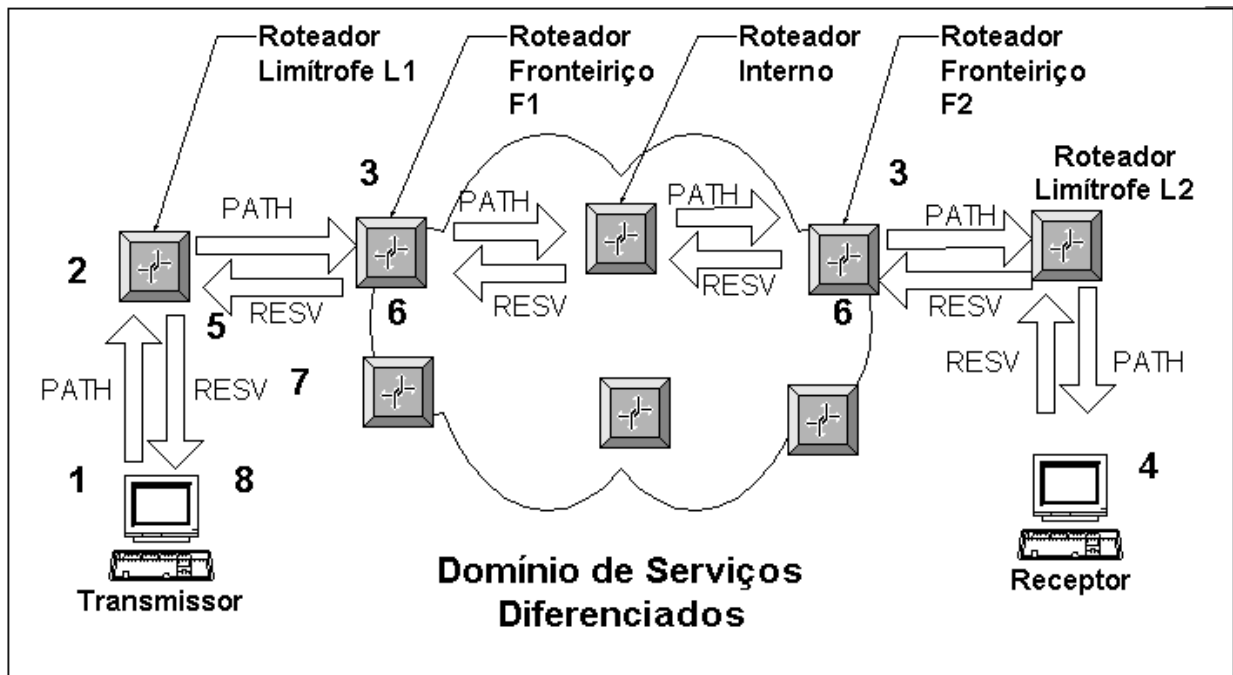


Figura 21: Reserva de Recursos Unicast na arquitetura híbrida [8]

5. caso o domínio DiffServ não suporte a sinalização RSVP/IntServ, então a mensagem RESV será processada pelo roteador limítrofe L2 e encaminhada para o domínio de serviços diferenciados, sendo ignorada em seu interior e nos roteadores fronteiros F1 e F2. No roteador limítrofe L1, os recursos requisitados na mensagem RESV para a classe de serviço correspondente serão comparados com os recursos disponíveis para o domínio DiffServ; informados ao roteador através de configuração estática prévia. L1 pode tanto aceitar a reserva de recursos, quando então a mensagem RESV continuará seu caminho para o transmissor, quando negá-la, quando então as mensagens de erro RSVP apropriadas serão geradas e a mensagem RESV não será encaminhada ;
6. caso o domínio suporte a sinalização RSVP/IntServ então a mensagem será processada pelos roteadores limítrofes L1 e L2 e por alguns ou todos os roteadores no interior do domínio. Dependendo da implementação, o controle de admissão poderá ser realizado pelos roteadores limítrofes baseados em um banco de dados de disponibilidade de recursos no domínio, ou mesmo de acordo com os recursos disponíveis apenas nos caminhos a serem impactados pela reserva. A reserva pode

ser processada apenas para o fluxo, ou acrescentada à reserva de uma agregação. A sinalização para verificar dinamicamente a disponibilidade de recursos pode ser realizada utilizando-se o próprio RSVP;

7. a mensagem RESV procede através da rede na direção da estação transmissora. Os nós RSVP/IntServ no caminho podem rejeitar a requisição de reserva devido a recursos inadequados ou políticas de tráfego existentes. Caso a mensagem não seja rejeitada ela chegará à estação transmissora Tx;
8. em Tx, a aplicação QoS recebe a mensagem RSVP e a interpreta como uma indicação de que o fluxo de tráfego especificado foi aceito para o tipo de serviço IntServ requisitado. A partir daí, a estação pode gerar os pacotes referentes ao fluxo reservado com o campo DSCP já marcados com um valor padrão para serviço de controle QoS requisitado, ou fornecido pela mensagem RSVP.

2.9.4 Implementação do Serviço de Carga Controlada na Arquitetura Híbrida

O serviço de carga controlada (CL) apresenta diversas características que simplificam seu suporte em redes implementando a arquitetura híbrida. A principal é que a definição deste serviço é qualitativa e não quantitativa. Não há garantias explícitas de latência ou mesmo de perda de pacotes definidas quantitativamente, apenas o objetivo de aproximar a qualidade observada à de uma rede com pouca carga, sem congestionamentos. Como os PHB e grupos de PHB da arquitetura de serviços diferenciados também possuem apenas especificações qualitativas, o suporte ao serviço de carga controlada por uma rede com serviços diferenciados é facilitado[8].

O serviço de carga controlada possui também a vantagem de ser voltado para aplicações que requerem um tipo de qualidade o qual pode ser provido pelo serviço de melhor esforço em redes sem congestionamentos [4]. Isto abrange aplicações de tempo real elásticas e outros tipos de aplicações que podem ser implementadas em redes com o serviço de melhor esforço.

O serviço de carga controlada prevê uma regra de soma de especificações de tráfego TSpec para uma reserva compartilhada de fluxos [4]. Esta regra possibilita a

agregação das reservas de vários fluxos de modo a implementar-se políticas de tráfego por agregação conforme sugerido em [8], sendo utilizada nas simulações realizadas neste trabalho. Um TSpec agregando as reservas de recursos de vários fluxos pode ser obtido da seguinte maneira:

- o parâmetro r (taxa de geração de símbolos) da reserva agregada é obtido pela soma do mesmo parâmetro (r) do TSpec de cada fluxo;
- o parâmetro b (profundidade do balde) da reserva agregada é obtido pela soma do mesmo parâmetro (b) do TSpec de cada fluxo;
- o parâmetro p (taxa de pico) da reserva agregada é obtido pela soma do mesmo parâmetro (p) do TSpec de cada fluxo;
- o parâmetro m (tamanho mínimo do pacote utilizado nas políticas de tráfego) é obtido pelo menor valor de m dentre todos os TSpec dos fluxos a serem agregados;
- o parâmetro M (tamanho máximo dos pacotes) é obtido pelo maior valor de M dentre todos os TSpec dos fluxos a serem agregados.

Há duas possíveis implementações do serviço de carga controlada em uma rede com serviços diferenciados. Uma envolve o mapeamento dos fluxos CL em uma agregação que utiliza o encaminhamento expresso. Uma segunda possibilidade é o mapeamento utilizando comportamentos-por-salto de encaminhamento assegurado. Dentre ambas, a implementação utilizando o encaminhamento assegurado oferece um maior número de vantagens; sendo considerada preferível.

O uso do encaminhamento assegurado permite a existência de várias agregações de fluxos de carga controlada, independentes entre si e com diferentes recursos reservados em cada nó. Isto é possível devido ao mapeamento dos fluxos de carga controlada para diferentes classes de encaminhamento assegurado: como cada classe deve possuir recursos alocados em separado é possível dividir os fluxos em agregações com quantidade de recursos reservados diferentes no interior do domínio. Assim, os fluxos podem ser separados em diferentes agregações de acordo com os seus requisitos de qualidade; permitindo uma melhor gerência e distribuição dos recursos existentes no interior do domínio de serviços diferenciados. Uma forma

simples de realizar a separação dos fluxos em classes e sugerida em [8] é utilizando-se a razão entre o *tamanho do balde* (B) e a *taxa de transmissão de pacotes* (R) da especificação de tráfego de cada fluxo (TSpec). Esta razão é utilizada para definir a latência e a perda de pacotes esperados para cada fluxo de carga controlada [4].

Uma segunda vantagem do uso do encaminhamento assegurado é a existência de dois ou mais comportamentos-por-salto com diferentes probabilidade de perda de pacotes dentro de cada classe, o que permite uma forma elegante de tratar o tráfego em excesso: os pacotes excedentes pertencentes a fluxos de carga controlada podem ser marcados com o DSCP correspondente a um comportamento-por-salto que pertença à mesma classe do restante dos pacotes e que possua maior probabilidade de descarte. Esta abordagem é semelhante à já utilizada para o tráfego em excesso com o encaminhamento assegurado e atende aos requisitos de tratamento de tráfego do serviço de carga controlada (o tráfego em excesso deve ser encaminhado, desde que haja recursos disponíveis e que o restante do tráfego não seja afetado) .

A abordagem de utilização do comportamento-por-salto de encaminhamento expresso também permite a implementação de uma aproximação do serviço de carga controlada suportando as características de tráfego exigidas por este serviço de controle QoS. Entretanto, o uso deste comportamento-por-salto apresenta duas sérias limitações quando comparado à implementação utilizando o encaminhamento assegurado. Como há apenas um comportamento-por-salto, é impossível dividir os fluxos em várias agregações de acordo com os seus requisitos de qualidade; logo todo o tráfego de carga controlada deve ser mapeado em uma única agregação e os recursos devem ser alocados de modo a satisfazer aos requisitos de qualidade dos fluxos com maior demanda dentre todos os da agregação. Uma segunda limitação é a de que como o encaminhamento expresso impõe um limite máximo para o tráfego na agregação, o excesso não pode ser tratado de uma forma tão adequada como na implementação utilizando o encaminhamento assegurado, onde é mapeado para uma agregação com maior probabilidade de descarte dentro da mesma classe. Em vez disso, o tráfego em excesso deve ser descartado na entrada de cada domínio ou remarcado para outro comportamento-por-salto, o que é indesejável pois causa

reordenação dos pacotes em trânsito.

Apesar destas desvantagens, o mapeamento do serviço de carga controlada para o encaminhamento expresso pode ser vantajoso em alguns casos como, por exemplo, quando o domínio não suportar o encaminhamento assegurado. A implementação envolve a marcação de todo o tráfego de carga controlada para o DSCP do comportamento-por-salto EF, em roteadores limítrofes e fronteiriços. As políticas de tráfego são realizadas por fluxo ou para toda a agregação, sendo os pacotes em excesso remarcados para outro comportamento-por-salto ou descartados. Em cada nó interior, deve ser realizada uma alocação de recursos suficiente para atender aos requisitos de cada fluxo.

2.9.5 Implementação do Serviço Garantido

O serviço garantido oferece ao tráfego das aplicações clientes um tratamento confiável e com limites firmes (matematicamente comprovados) para a latência induzida fim-a-fim e para o throughput, assumindo apenas que a rede funcione corretamente. Uma segunda característica importante deste serviço de controle QoS são os dois termos de erro (chamados de C e D na especificação do serviço [8]), os quais são fornecidos pelos elementos de rede e enviados para a estação cliente a fim de que possa ser calculado o throughput que deve ser reservado para que a latência máxima desejada seja atingida. Assim, para que o serviço garantido seja suportado em uma região implementando os serviços diferenciados, é necessário que ambas as características sejam suportadas: as implementações de escalonamento, adequação e políticas de tráfego devem suportar limites firmes de latência e throughput e os termos de erro devem ser calculados com os valores correspondentes ao caminho percorrido pelo tráfego no interior da região DiffServ.

Os requisitos do serviço garantido fazem com que a melhor forma de implementar seu suporte seja com o uso do comportamento-por-salto de encaminhamento expresso, em conjunto com mecanismos de adequação e políticas de tráfego. O encaminhamento expresso possui características que minimizam a latência

no interior de domínios de serviços diferenciados, o que não é possível se o encaminhamento assegurado for utilizado.

A latência em regiões com serviços diferenciados pode ser classificada em três componentes: latência física, latência de adequação/readequação de tráfego e latência de escalonamento [8]. A fim de que os termos de erro C e D e os limites superiores de latência possam ser determinados é necessário que cada um destes componentes seja corretamente caracterizado. A maior dificuldade é que em regiões com serviços diferenciados estes componentes podem depender não só da topologia da rede, mas também das características de potencialmente todo o tráfego relacionado com os fluxos de serviços garantido.

Os limites superiores para a latência física e os termos de erro são facilmente obtidos a partir das informações sobre os tempos de propagação e serialização dos pacotes em cada enlace no interior do domínio de serviços diferenciados. Do mesmo modo, para a latência de adequação/readequação de tráfego, estes valores podem ser calculados sem dificuldades para o suporte ao serviço garantido. Neste último caso, a obtenção dos valores é facilitada pelo fato dos mecanismos de adequação/readequação de tráfego em domínios de serviços diferenciados estarem localizados em roteadores fronteiriços ou limítrofes; o que permite que o cálculo da latência seja realizado de forma independente para cada fluxo de serviço garantido (e não para toda a agregação de encaminhamento expresso) [4].

Entretanto, para a latência relacionada com o escalonamento de pacotes o cálculo dos valores de latência torna-se complexo. A principal dificuldade se deve à agregação dos fluxos no interior dos domínios de serviços diferenciados. Quando um pacote de um fluxo f percorre uma seqüência de nós onde os escalonadores mantêm uma mesma fila para toda a agregação a que f pertence; a latência de pior caso para os pacotes em f depende do tráfego de outros fluxos na mesma agregação. Além disso, a latência de um pacote p de um fluxo f no momento t pode ser afetada mesmo por fluxos cujos últimos pacotes deixaram a rede antes dos primeiros pacotes de f começarem a percorrê-la [4].

Atualmente, o único limite teórico comprovado de latência que é válido para uma topologia e uma distribuição de rota arbitrária mantém-se apenas para uma taxa de utilização da fila (quociente entre a quantidade de dados em bytes processada na fila e a quantidade máxima que pode ser processada) de $u < 1/h-1$; onde h é o número máximo de saltos a ser percorrido por qualquer fluxo na rede [61]. Este limite mantém-se apenas para o caso onde a capacidade de qualquer enlace é significativamente menor que a capacidade total de todas as interfaces de qualquer roteador. Podem ser obtidos limites teóricos melhores para determinadas classes de redes com topologias ou rotas específicas [4].

Diante destas dificuldades, o suporte ao serviço garantido em regiões com serviços diferenciados requer que a quantidade de tráfego referente aos fluxos de serviço garantido seja sempre bem menor do que o disponível. Uma outra consequência é a de que se houver tráfego de encaminhamento expresso não referente a fluxos de serviço garantido em um mesmo domínio de serviços diferenciados onde estes fluxos trafegam, então deve haver mecanismos de escalonamento que assegurem a separação entre os dois tipos de tráfego.

2.9.6 Considerações sobre a Arquitetura Híbrida

Tem havido um grande esforço por parte do IETF e do grupo de trabalho de serviços integrados sobre Camadas Específicas (Integrated Services over Specific Layers ou ISSL) em criar um modelo de qualidade de serviço fim-a-fim para a Internet baseado nos mecanismos de reserva de recursos e tratamento de tráfego dos serviços integrados. Fazem parte deste esforço especificações de suporte aos serviços de controle QoS sobre ATM [2], Frame-Relay ou redes Ethernet [4]. Neste contexto, o modelo híbrido para a interoperação entre os serviços integrados e os serviços diferenciados se encaixa como uma peça fundamental: à medida em que o uso dos serviços diferenciados for se popularizando o suporte aos serviços de controle QoS se tornará fundamental para a obtenção de qualidade de serviço fim-a-fim. Este modelo possui também a vantagem de permitir a implementação de qualidade de serviço em redes com grande escala baseadas apenas em IP.

As principais dificuldades relacionadas com a implementação da arquitetura híbrida estão diretamente relacionadas com a relativa juventude das arquiteturas de qualidade de serviço em IP, e principalmente dos serviços diferenciados. A falta de mecanismos de obtenção dinâmica de informações sobre o uso dos recursos no interior de domínios de diferenciação de serviços e de reserva de recursos tornam difícil o suporte aos requisitos quantitativos dos serviços de controle QoS.

Um segundo problema envolve a interoperação entre redes com diferentes administrações como as que constituem a Internet: para a garantia de suporte da reserva de recursos é necessário que a reserva seja adequadamente provisionada em todo caminho entre cada transmissor e o receptor. Há várias questões envolvendo este tema e que ainda não foram adequadamente discutidas. Como será cobrada dos usuários finais a reserva de recursos? Como serão remunerados os provedores com redes intermediárias implementando QoS? Como deverão ser feitos acordos entre os diversos provedores de serviços para o suporte a reservas de recursos dos usuários finais? Antes que a implementação de uma arquitetura de qualidade de serviço fim-a-fim em toda a Internet seja possível, estas perguntas terão de ser respondidas. Este problema limita o uso da arquitetura híbrida a curto prazo apenas a redes com administração centralizada.

Capítulo 3 – Modelo de Simulação

3.1 Descrição do Modelo de Simulação

A idéia de elaborar um projeto de simulação para analisar a interoperação das arquiteturas de serviços diferenciados e integrados surgiu com base na leitura das especificações relacionadas a este tópico no IETF [8][7]. Foi observado que alguns dentre os tópicos abordados nas especificações não possuíam uma análise detalhada, e que ainda não havia outros trabalhos que a realizassem. A dificuldade de acesso a equipamentos e o fato de que as especificações do IETF, devido a serem relativamente, novas ainda não possuíam suporte por parte de fabricantes tornaram a simulação a escolha mais prática como ferramenta de análise.

O objetivo do projeto de simulação foi definido como o de analisar tópicos relacionados com o suporte aos requisitos dos serviços de controle QoS de carga controlada de serviços por parte de comportamentos-por-salto de encaminhamento assegurado em domínios de serviços diferenciados de acordo com a arquitetura especificada. O suporte ao serviço garantido foi intencionalmente não estudado por considerar-se sua implementação difícil e de pouca utilidade devido a características já discutidas no capítulo 2. O suporte ao serviço de carga controlada utilizando-se o encaminhamento expresso também não foi analisado devido a sua complexidade para implementação na simulação.

Para facilitar a tarefa de elaboração do modelo de simulação, a seleção de suas características foi dividida em duas partes: na primeira foram definidos os requisitos a serem atendidos pelo modelo, e na segunda foram estabelecidas as características do modelo. Os requisitos foram estabelecidos com base no objetivo do projeto de simulação e foram:

- o modelo deveria ser tão simples quanto possível, desde que não afetasse os resultados das simulações;

- o modelo deveria ser o mais próximo possível do utilizado em outros trabalhos tomados como referência envolvendo os serviços diferenciados, a fim de facilitar sua validação;
- o modelo deveria representar um domínio de serviços diferenciados atuando como elemento de controle QoS de serviços integrados de acordo com a arquitetura descrita e detalhada no item 2.9.

Com base nestes requisitos foram definidas as características do modelo a ser simulado. Estas características podem ser divididas em duas partes: a primeira refere-se à topologia do modelo e a segunda à definição dos mecanismos específicos da arquitetura de interoperação.

A topologia de simulação escolhida foi a representada na figura 22. Nesta topologia há dezesseis nós geradores (representados pela letra Fn) e receptores de tráfego (Rn) conectados quatro a quatro a nós representando roteadores limítrofes ao domínio de serviços diferenciados por enlaces de 10Mbps e latência física de 1ms. A seleção da quantidade de nós foi feita com base no uso de recursos da estação por parte do simulador: o número de nós escolhido revelou-se como sendo o maior possível válido para todas as simulações sem ultrapassar os limites impostos pela memória da estação.

Os roteadores limítrofes estão conectados dois a dois a outros dois nós representando roteadores fronteira por enlaces com banda passante de 100 Mbps e latência física de 1 ms. Os roteadores fronteira estão conectados a um único nó, o qual representa um roteador interno ao domínio de serviços diferenciados. A topologia é simétrica em relação a um eixo que separe os dois roteadores internos, possuindo uma estrutura semelhante para os nós destino. O “gargalo” da topologia é um enlace com banda passante de 2,048 Mbps e latência física de 40 ms entre os dois nós internos ao domínio DS, representando um enlace de longa distância E1.

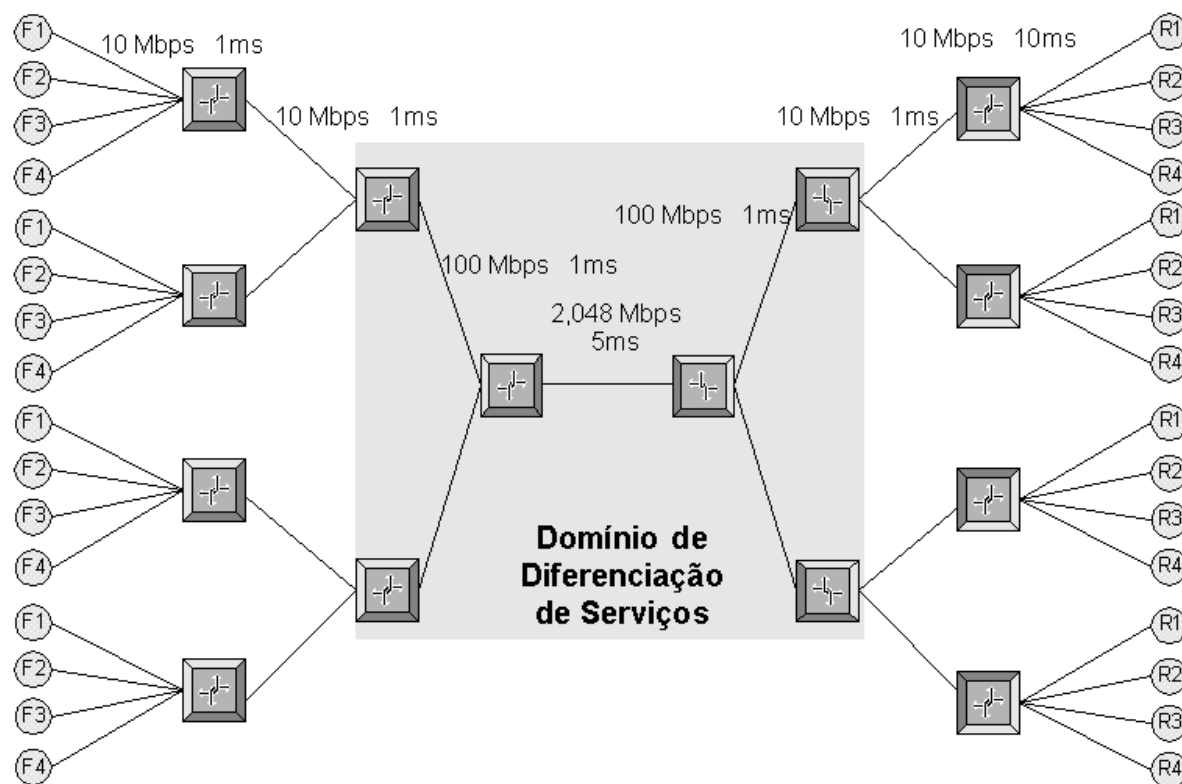


Figura 22: Topologia de Simulação[4]

Esta topologia foi escolhida por permitir a representação dos principais nós componentes da arquitetura híbrida (roteadores fronteira, roteadores internos aos domínios de serviços diferenciados e roteadores limítrofes). Ao mesmo tempo permite a representação de fontes de tráfego pertencentes a diferentes redes externas ao domínio de serviços diferenciados: cada roteador limítrofe pode ser utilizado para representar um nó externo de uma rede administrativamente independente. Isto permitiu o estudo de políticas de agregação de tráfego, onde cada política é aplicada a uma agregação de tráfego pertencente a uma rede externa[4].

Uma outra vantagem da topologia escolhida é o fato de atender a dois dos requisitos especificados para o modelo. O primeiro é a simplicidade, a qual facilita a obtenção e posterior análise dos resultados. O segundo é o fato de ser bastante próxima à topologia utilizada entre os trabalhos referenciados.

Com relação às características específicas dos mecanismos da arquitetura de interoperação, decidiu-se pela não representação da sinalização RSVP ou de

elementos QoS capazes de serviços integrados, com o fim de simplificar o ambiente de simulação e de aproximá-lo do utilizado em diversos trabalhos tomados como referência. Apesar de aparentemente esta escolha ir de encontro ao esperado para uma simulação da interoperação entre os serviços integrados e os diferenciados, ela pôde ser realizada sem prejuízo da representatividade dos resultados. Em lugar de elementos QoS capazes com reserva de recursos utilizando RSVP nas regiões externas ao domínio de serviços diferenciados (roteadores fronteiriços e nós fonte e destino de tráfego), foram simulados enlaces de melhor esforço em que o throughput gerado pelos microfluxos simulados é sempre muito menor que a banda passante, garantindo a inexistência de congestionamentos e a necessidade de simulação de mecanismos de qualidade de serviço nestes enlaces. Como as simulações realizadas buscam estudar o suporte dos serviços diferenciados aos serviços integrados, os congestionamentos simulados e as alterações das características do tráfego oriundas dos mesmos se localizaram sempre no interior do domínio de diferenciação de serviços.

O tipo de política de tráfego selecionado foi o de Token Bucket simples (*Single Rate Two Color Marker ou SRTCM*), onde o perfil de tráfego de cada microfluxo é formado pelos parâmetros de taxa de reposição de pacotes (r) e de profundidade do balde (b). Quando perfis de tráfego são utilizados para agregações de microfluxos foi utilizada a regra de soma de reserva de recursos especificada em [29], onde os parâmetros r e b do TSpec de cada microfluxo são somados e utilizados para compor uma política de tráfego única para toda a agregação utilizando-se o *Token Bucket* simples. O uso desta política de tráfego possibilita a utilização dos mesmos parâmetros de descrição de tráfego (TSpec) especificados para o RSVP na arquitetura de serviços integrados. Uma exceção é o parâmetro de pico de tráfego (p), o qual por simplificação não foi utilizado nas simulações. O uso do parâmetro (p) requeriria a implementação de uma política de tráfego do tipo *Single Rate Three Color Marker*.

Em todas as simulações foi utilizado o gerenciamento ativo de filas do tipo RIO Coupled (RIO-C) [4]. O RIO-C é uma variação do RED do tipo MAMT, onde a probabilidade de descarte dos pacotes marcados como dentro do perfil de tráfego

(verdes) é calculada com base apenas em sua quantidade, enquanto a probabilidade de descarte dos pacotes fora do perfil de tráfego (vermelhos) é calculada tendo como base todos os pacotes na fila (verdes e vermelhos). Seu uso permitiu que em todas as simulações os pacotes verdes só fossem descartados quando os pacotes vermelhos já o tivessem sido: o gatilho máximo do RIO configurado para os pacotes vermelhos foi sempre menor que o gatilho mínimo dos pacotes verdes. Tal configuração é necessária com o fim de atender aos requisitos de descarte de pacotes do serviço de carga controlada, permitindo que os mesmos sejam suportados em agregações de comportamento de encaminhamento assegurado.

3.2 Implementação do Modelo de Simulação

A plataforma de simulação escolhida foi o simulador NS (“Network Simulator”) [7]. O NS é um simulador orientado a eventos discretos (*discrete-event simulator*) desenvolvido pela Universidade da Califórnia (UCLA) com base no simulador REAL [7]. A escolha foi feita com base na grande aceitação deste software na comunidade científica e no seu uso em vários trabalhos em simulações envolvendo os serviços diferenciados. O simulador também possibilita o suporte aos requisitos descritos no modelo de simulação. E uma das grandes vantagens de utilizar o NS é a possibilidade de alterar o código fonte do programa para refletir a pesquisa que está sendo realizada. O códigos usados nesta simulação foram obtidos com base em [4] e estão descritos no anexo I.

A versão utilizada foi a 2.8b. O simulador foi executado em uma estação com processador Duron 900 Mhz, 448 MB de memória RAM. O sistema operacional escolhido para a utilização do simulador foi o Linux na distribuição Red Hat.

Os serviços diferenciados foram implementados no simulador utilizando uma versão modificada de um módulo do NS. O módulo possibilitou a implementação dos diversos mecanismos existentes em um domínio com serviços Diferenciados tais como marcação de pacotes, disciplinas de fila com diferenciação entre classes de tráfego (CBQ e PRR), gerenciamento ativo de filas, dentre outros. Entretanto, para a obtenção

dos resultados desejados foram necessárias algumas modificações. As principais foram a implementação de mecanismos de política de tráfego para agregações de fluxos com reservas RSVP [4] e a criação de novas funções de monitoramento que possibilitaram a obtenção de estatísticas sobre a latência e o descarte de pacotes.

As simulações foram criadas utilizando-se *scripts* na linguagem OTCL [4], a qual possibilita o acesso aos objetos de simulação do NS. Para a análise dos resultados foram utilizados os dados armazenados em arquivos *trace* gerados pelo simulador. Cada arquivo possui como conteúdo todos os eventos associados com cada pacote gerado durante a simulação. Para facilitar a análise foram exportados os arquivos de *trace* para o Microsoft Excel[7] e utilizando-se *scripts* na linguagem PERL [4], os quais geraram relatórios relativos às estatísticas de interesse.

Foram utilizados dois tipos de parâmetros de entrada comuns a todas as simulações: *duração da simulação* e *duração do transiente*. A *duração da simulação* corresponde à condição de parada das simulações sendo especificada por um intervalo de tempo virtual utilizado pelo simulador e diferente do tempo real. O NS utiliza um processo o qual atua como um *relógio de simulação* fornecendo uma variável global (chamada de *tempo de simulação*) representando o tempo simulado a qual é utilizada para escalonar a execução de eventos tais como a criação de pacotes. Já a *duração do transiente* refere-se ao intervalo de tempo de simulação no qual o modelo simulado está em processo de inicialização e ainda não atingiu um estado de estabilidade. Para o aumento da confiabilidade dos resultados, os eventos relacionados com o transiente foram removidos dos dados analisados.

Para a escolha do melhor valor para estes parâmetros foram utilizados dois métodos: comparação com implementações realizadas em outros trabalhos e *scripts* de simulação simplificados. As implementações realizadas em outros trabalhos [25][30] forneceram os valores iniciais. Partindo-se destes valores foram executados *scripts* de simulação simplificados e analisados seus resultados. Para a escolha da duração das simulações foi observada a variância do throughput de microfluxos TCP entre simulações similares. Para a escolha da duração do transiente, foi comparado o

throughput médio dos microfluxos no início de cada simulação, com o throughput médio no restante da mesma.

A duração de transiente escolhida foi de 10 segundos do tempo de simulação, nos quais as fontes de tráfego são inicializadas de acordo com uma variável aleatória uniformemente distribuída, conforme sugerido em trabalhos já realizados e citados na referência bibliográfica [25]. A duração da simulação foi estabelecida em 100 segundos do tempo de simulação, dos quais os 10 primeiros correspondem ao transiente.

3. 3 Tráfego Simulado

O tráfego simulado foi selecionado de modo a representar os principais tipos de aplicações a serem utilizados com o serviço de carga controlada da arquitetura de serviços integrados. O serviço de carga controlada é mais adequado para aplicações que podem caracterizar seus requisitos de tráfego, tais como aplicações baseadas no transporte de mídia contínua dentre as quais são bom exemplos aplicações de áudio e vídeo. Entretanto este serviço pode ser útil a qualquer aplicação sensível a congestionamentos. Optou-se assim por utilizar-se fontes CBR para a modelagem de aplicações multimídia de voz e vídeo, e de fontes FTP para a modelagem de conexões TCP de longa duração e sensíveis à carga da rede.

Fontes CBR podem representar de forma realista vários tipos de tráfego de voz e vídeo. As fontes CBR utilizadas possuem uma variação de +10% ou -10% da sua taxa de envio de pacotes. Fontes deste tipo são conhecidas como fontes CBR com variação de retardo (jittered CBR sources), e representam de forma realista implementações de fontes CBR por software onde devido ao escalonamento de processos na máquina executando a aplicação que origina o tráfego há pequenas variações na taxa de transmissão nominal.

As fontes FTP foram utilizadas para simular aplicações com grande quantidade de transferência de dados “bulk data transfer applications”. Considera-se que este tipo de tráfego pode ser utilizado para modelar um grande número de aplicações baseadas em conexões TCP de longa duração[4].

3.4 Verificação do Modelo de Simulação

A verificação do modelo de simulação refere-se à averiguação da implementação do modelo na plataforma de simulação. A verificação está diretamente ligada à correção de erros de análise e codificação, os quais podem afetar os resultados mesmo se os pressupostos utilizados no modelo forem válidos [30].

Houve três tipos de codificação: alteração de código do simulador, criação dos *scripts* de simulação e criação de *scripts* de análise dos resultados. A alteração de código do simulador foi realizada para permitir a criação de simulações com políticas de tráfego por agregação e para alterar os arquivos *trace* gerados pelo simulador de modo a facilitar a obtenção de informações sobre a latência induzida dos pacotes simulados; este código foi obtido com base nas pesquisas já realizadas utilizando este modelo. A verificação desta parte do código foi realizada utilizando-se a execução de *scripts* de simulação simplificados e da análise de seus resultados. Um outro método utilizado foi a criação de mensagens de log indicando os procedimentos em execução pelo simulador.

A verificação dos *scripts* de simulação foi realizada de várias maneiras. Do mesmo modo já citado anteriormente, foram criados *scripts* simplificados e utilizadas mensagens de log para a descoberta de erros. Também foram utilizados os próprios arquivos *trace* do simulador para a verificação da execução dos eventos codificados.

Finalmente para os *scripts* de análise dos resultados, a verificação foi realizada através da comparação entre resultados obtidos de formas diferentes. O próprio simulador possui facilidades que permitem a obtenção de estatísticas sobre a simulação utilizando-se linhas de código no próprio *script*. Apesar destas facilidades serem limitadas, elas permitiram a verificação do código dos *scripts* de análise dos arquivos *trace*, os quais por possibilitarem a obtenção de informações detalhadas foram os efetivamente utilizados na análise dos resultados. Em cada simulação realizada foram comparados os resultados referentes ao throughput médio e à sua variância, à latência média e à sua variância e ao descarte de pacotes dentro do perfil de tráfego (verdes). Valores diferentes indicaram erros nos *scripts*.

3.5 Validação do Modelo de Simulação

A validação do modelo de simulação refere-se às técnicas utilizadas para assegurar que as suposições utilizadas no desenvolvimento do modelo foram razoáveis, de modo a assegurar que se estiverem corretamente implementadas o modelo produzirá resultados próximos aos existentes em sistemas reais. A validação foi facilitada pela opção de não serem utilizados mecanismos de serviços integrados no modelo. Conforme já citado, devido ao fato do objetivo do projeto de simulação estar relacionado apenas ao estudo de requisitos a serem suportados por comportamentos-por-salto de serviços diferenciados, a existência de mecanismos de serviços integrados faz-se desnecessária.

Para a validação foram realizadas simulações com o objetivo de comparar resultados com os obtidos em outros trabalhos dados o mesmo contexto, os mesmos parâmetros de entrada e o mesmo tipo de tráfego. As comparações foram realizadas principalmente com os resultados obtidos em [25] e [29] referentes a agregações de encaminhamento assegurado. Foram comparados resultados relacionados com estatísticas de throughput quando variada a taxa de reposição de símbolos de fluxos TCP. A comparação foi satisfatória, sendo obtidos os resultados esperados.

Capítulo 4 – Resultados do Modelo de Simulação

4.1. Suporte a Fluxos de Carga Controlada em uma Classe de Encaminhamento Assegurado

As simulações apresentadas neste item procuram estudar o suporte a requisitos de qualidade de fluxos de carga controlada utilizando uma única classe de encaminhamento assegurado. O objetivo é o de investigar sob que condições a diferenciação provida por comportamentos-por-salto de encaminhamento assegurado é capaz de suportar os requisitos de qualidade reservados por fluxos de carga controlada. Com esta finalidade foram realizadas simulações com diferentes tipos de tráfego, onde a diferenciação em cada nó é realizada por gerenciamento ativo de filas.

As simulações deste item baseiam-se no modelo proposto em [4], o qual sugere a utilização de uma classe de encaminhamento assegurado unicamente para prover suporte a fluxos de carga controlada. Pressupõe-se ainda a utilização de uma disciplina de fila que garanta uma banda passante mínima alocada para a classe de encaminhamento assegurado em situações de congestionamento, tais como o WRR ou o WFQ. Situações em que a disciplina de fila não suporta o isolamento entre as classes de serviço, ou onde uma mesma classe de serviço é utilizada para tráfegos diferente do de carga controlada serão analisadas em itens posteriores.

4.1.1. Influência do número de microfluxos TCP e da Taxa de Reposição de Pacotes

As simulações neste item procuram investigar a influência do número de microfluxos TCP e da taxa de reposição de pacotes na qualidade de serviço obtida por fluxos de carga controlada. Estes parâmetros foram escolhidos por representarem formas de alocação de recursos no interior de domínios de serviço diferenciados. Em especial, a taxa de reposição de pacotes controla a quantidade de pacotes marcados

como dentro do perfil de tráfego e que portanto deve receber um tratamento condizente com as especificações do serviço de carga controlada. As simulações também procuraram levar em consideração a influência da política de tráfego utilizada nos resultados obtidos, através da execução de simulações implementando políticas com perfis de tráfego para agregações de microfluxos e políticas com perfis de tráfego por microfluxos.

Foram realizadas 16 simulações com políticas de tráfego implementadas utilizando-se um marcador/mensurador de pacotes Token Bucket simples e 16 com políticas de tráfego Token Bucket simples com perfis de tráfego para agregações de 8 microfluxos, onde o número de fluxos e a taxa de reposição de pacotes na política de tráfego implementada foram variadas no interior do domínio de serviços diferenciados. A taxa de reposição de pacotes para cada microfluxo foi variada entre 15625 bps, 31250 bps, 46875 bps e 62500 bps; correspondendo a frações do enlace gargalo de 2 Mbps. Para cada um destes valores foram realizados quatro grupos de simulações com 8, 16, 24 e 32 microfluxos TCP em cada grupo. As agregações de fluxos utilizadas na política de tráfego foram implementadas sempre agrupando as reservas de oito microfluxos; nos casos em que o número de microfluxos simulados foi maior que 8 foi utilizada mais de uma agregação de 8 microfluxos. O tamanho do balde reservado para cada microfluxo foi de 10 pacotes. As probabilidades de descarte foram escolhidas de acordo com [24], sendo utilizados os parâmetros 0.1/10/20 para o comportamento-por-salto com maior probabilidade de descarte e 0.02/20/40 para o comportamento-por-salto com menor probabilidade de descarte. O tamanho das filas no interior do domínio de serviços diferenciados foi escolhido de modo a limitar a latência induzida no enlace gargalo para valores próximos de 100ms, sendo utilizado um tamanho máximo de para as filas de 40 pacotes e de pacotes de 576 bytes.

O gráfico 4.1 (a) e (b) mostra a frequência relativa acumulada da latência para os pacotes das simulações onde a taxa de reposição de pacotes somada dos perfis de tráfego de todos os microfluxos foi configurada para 500000 bps ou 25 % do enlace gargalo para política de tráfego Token Bucket simples (a) e por agregação (b). Como pode ser observado, a distribuição da latência varia de forma bastante acentuada à

medida que a quantidade de microfluxos TCP é aumentada, tanto no gráfico (a) quanto em (b). Ao mesmo tempo, a influência da taxa de reposição de pacotes é pequena na latência média obtida. Isto é demonstrado nos gráficos 4.2 (a) e (b) onde são mostradas as freqüências relativas acumuladas da latência para taxas de reposição de pacotes totais (soma dos perfis dos microfluxos) de 500 kbps, 1000 kbps, 1500 kbps, 2000 kbps; e 32 microfluxos.

Os resultados sugerem que quando o tráfego de uma agregação de encaminhamento assegurado com fluxos de carga controlada for composto por microfluxos TCP, a diminuição na quantidade de microfluxos pode ser utilizada para diminuir a latência média dos pacotes da agregação, mesmo se for mantida a mesma taxa de reposição de pacotes para a agregação de microfluxos. Ao mesmo tempo, a variação na taxa de reposição de pacotes não possui influência relevante na latência obtida. Estes resultados podem ser explicados pelos algoritmos de Slow Start e Congestion Avoidance do TCP: os microfluxos incrementam a janela de transmissão até que um congestionamento seja detectado pela perda de pacotes. Um aumento do número de microfluxos faz com que haja um maior número de pacotes dentro do perfil de tráfego devido ao fato destes pacotes terem uma probabilidade de descarte menor, não gerando indicação de congestionamento ao TCP. A perda de pacotes ocorre em média apenas para pacotes marcados como fora do perfil de tráfego (vermelhos). Isto ocasiona um aumento no tamanho médio na fila de pacotes dentro do perfil de tráfego (verdes), aumentando o tamanho da fila e como consequência a latência induzida.

Com relação ao throughput, os resultados são condizentes com o esperado para fluxos TCP em agregações de encaminhamento assegurado de acordo com o critério proposto em [25] e [26] no qual o throughput médio justo para microfluxos em agregações de encaminhamento assegurado corresponde a:

$$ThroughputM\u00e9dio = r + \frac{(B - r)}{n}$$

Nesta equação B representa a banda passante disponível no enlace com menor capacidade e r a taxa de reposição de pacotes. A tabela 1 (apêndice 1) apresenta um

resumo dos resultados obtidos. Os resultados não mostraram nenhuma diferença observável com relação à granulosidade dos perfis de tráfego utilizados nas políticas.

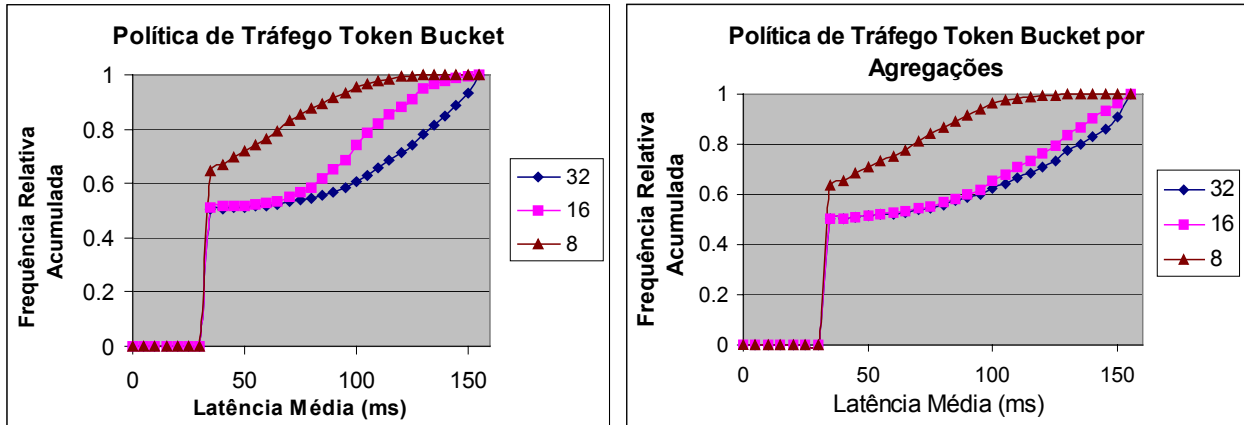


Figura 23 (a) e (b): Frequência Relativa da Latência de pacotes verdes para políticas de tráfego com taxa de reposição de pacotes de 500000 bps Token Bucket simples (a) e Token Bucket por agregação (b)

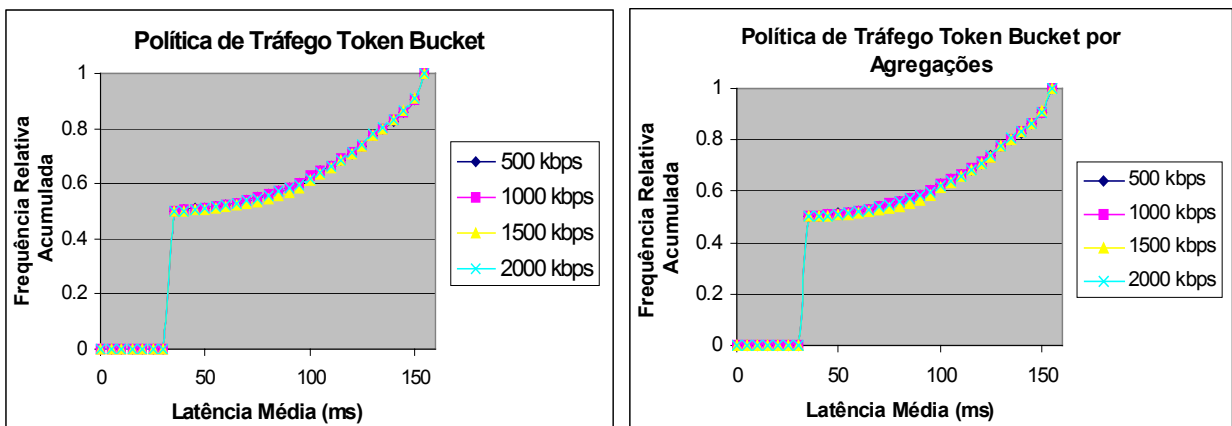


Figura 24(a) e (b): Frequência Relativa da Latência de pacotes verdes para simulações com 32 microfluxos, taxas de reposição de pacotes variadas e políticas de tráfego Token Bucket simples (a) e Token Bucket por agregação (b)

4.1.2 Granulosidade de Políticas de Tráfego e Tráfego TCP

Uma implementação do serviço de carga controlada em um domínio de serviços diferenciados pode utilizar políticas por agregação de perfis de tráfego nas bordas da rede em lugar de mecanismos com perfis por fluxo. Neste caso, os parâmetros para as

políticas por agregação são fornecidos a partir das regras de soma de reservas RSVP para fluxos de carga controlada [4]. Esta abordagem possui grandes vantagens em termos de escalabilidade dos mecanismos de política. Entretanto, devido aos mecanismos de política por agregação não diferenciarem os perfis de tráfego reservados e associados aos diversos fluxos, fluxos com tráfego em excesso têm a possibilidade de obter mais recursos que os reservados para o restante da agregação. Nesta seção, serão comparados os mecanismos de políticas por agregação e por fluxo com relação à justiça na alocação de recursos para tráfego TCP. O objetivo é o de estudar em que situações o uso de políticas por fluxo faz-se necessário para se evitar injustiças no tratamento ofertado a fluxos de carga controlada em redes implementando o encaminhamento assegurado.

A seqüência de simulações teve por objetivo comparar o impacto do uso de políticas de tráfego por agregação e por fluxo nos recursos obtidos por microfluxos TCP com latências físicas diferentes. As simulações foram executadas com 32 fontes de tráfego FTP, todas com a mesma taxa de reposição de pacotes e tamanho do balde de 10 pacotes. Foram utilizadas políticas de tráfego Token Bucket simples e por agregação de 32 microfluxos TCP. Os parâmetros do RED utilizados foram 10/20/0.1 para pacotes vermelhos e 20/40/0.05 para pacotes verdes. Os fluxos simulados possuíam latências físicas variando entre 20 ms e 620 ms com acréscimos de 40 ms, totalizando 16 grupos com dois microfluxos TCP em cada. Na seqüência de simulações foi variada ainda a taxa de reposição de pacotes (CIR) para os microfluxos TCP, com valores para a agregação de fluxos correspondendo a 30%, 60% e 90% da banda passante total. As simulações foram repetidas 15 vezes de modo a minimizar a dispersão.

Os resultados mostraram que o uso da política por agregação de perfis de tráfego favorece os microfluxos com menor RTT. O gráfico 23 mostra o throughput médio de microfluxos TCP para uma taxa de reposição de pacotes da agregação de microfluxos correspondente a 90% do enlace gargalo. Como pode ser observado, para políticas de tráfego Token Bucket simples o throughput médio mantém-se dentro de uma faixa de valores com pouca variação, enquanto para políticas de tráfego por

agregação de perfis de tráfego o throughput médio decai à medida que o RTT aumenta.

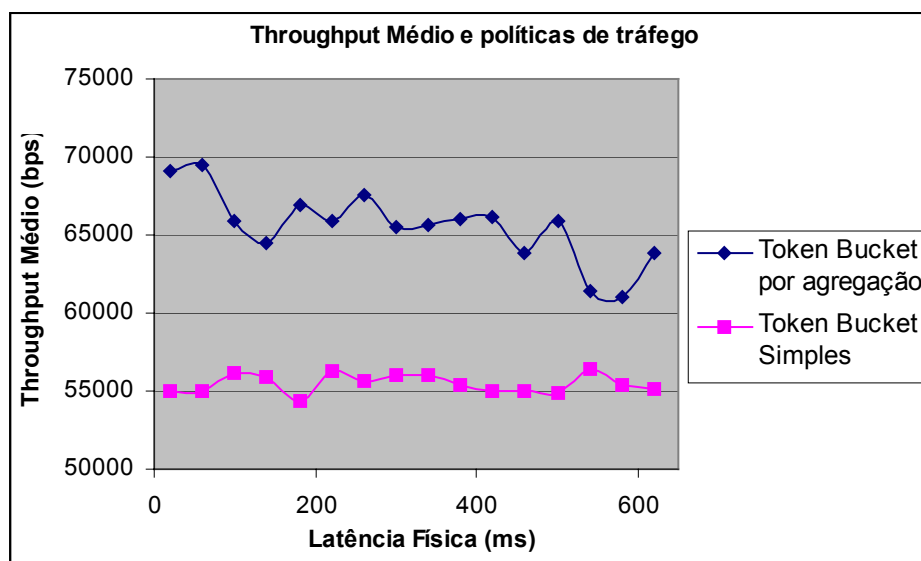


Figura 25: Throughput Médio por fluxo versus Latência Física para 90% de banda passante reservada e política de tráfego com perfil por fluxo

4.1.3 Granulosidade das Políticas de Tráfego e Tráfego UDP

As simulações neste item têm como objetivo investigar como a granulosidade da política de tráfego afeta microfluxos TCP quando competem por recursos com microfluxos UDP em uma mesma classe de encaminhamento assegurado. Os microfluxos UDP e TCP representam fluxos de carga controlada com requisitos de qualidade (informados pelo perfil de tráfego) a serem suportados por um domínio de serviços diferenciados. Microfluxos UDP são comuns em aplicações multimídia e, por não possuírem mecanismos de controle de congestionamentos como os existentes em fluxos TCP podem gerar tráfego em excesso que degrade o tratamento obtido por outros microfluxos não diferenciados.

O tráfego UDP simulado consistiu em fluxos CBR com ruído, os quais simulam tráfego de aplicações de transmissão de voz. Foram realizadas 30 simulações com 16 fontes de tráfego TCP e 16 fontes UDP, onde foi variada a taxa de transmissão de cada fonte UDP e a quantidade de microfluxos em um mesmo perfil de tráfego. Os

microfluxos TCP e UDP foram configurados com os mesmos parâmetros de reserva de recursos (representando a reserva RSVP): taxa de reposição de pacotes (r) de 37500 bps e profundidade do balde (b) de 5760 bytes. A variação da granulosidade das políticas de tráfego foi implementada utilizando-se um marcador/mensurador de tráfego Token Bucket simples, onde a quantidade de microfluxos por perfil de tráfego foi variada entre 16, 8, 4, 2 e 1. A cada perfil de tráfego com granulosidade maior que 1 foi sempre associado o mesmo número de microfluxos UDP e TCP. Os parâmetros dos perfis de tráfego corresponderam à soma dos parâmetros de reserva de recursos (r e b) dos microfluxos associados, conforme a regra de soma de reservas RSVP para fluxos de carga controlada descrita em [29].

Para cada granulosidade de perfil de tráfego, foi variada a taxa de geração de tráfego das fontes UDP. A taxa variou para cada microfluxo entre 37500 bps, 50000 bps, 62500 bps, 75000 bps, 87500 bps e 100000 bps. Estes valores foram escolhidos de modo a permitir que o tráfego UDP total correspondesse a 30%,40%,50%,60% ,70% e 80% da banda passante do enlace gargalo (de 2 Mbps). O gerenciamento ativo de filas, tempo de simulação e a topologia utilizada foram mantidos conforme as simulações anteriores.

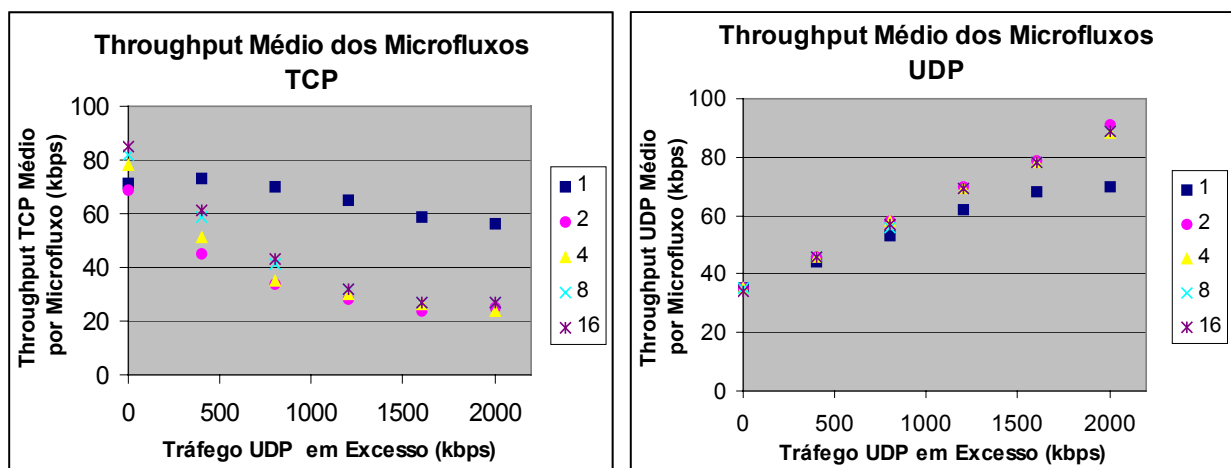


Figura 26(a) e (b): Throughput Médio obtido por microfluxos UDP de acordo com variações na quantidade de microfluxos em um mesmo perfil de tráfego

As figuras 24 (a) e (b) mostram o throughput médio dos microfluxos TCP (a) e UDP (b) para vários valores de granulosidade dos perfis de tráfego e do tráfego UDP

em excesso. Como pode ser observado, há uma grande diferença nos resultados mostrados entre as simulações onde o perfil de tráfego é utilizado para apenas um microfluxo e o restante: o throughput obtido pelos microfluxos TCP é maior e o throughput obtido pelos microfluxos UDP é menor o que se acentua à medida que o tráfego UDP em excesso aumenta. Os resultados comprovam que os microfluxos UDP obtêm maior número de recursos quando associados a um mesmo perfil de tráfego que microfluxos TCP. Um outro resultado é o de que o aumento do número de microfluxos associados a um mesmo perfil de tráfego (com granulosidade de microfluxos associados maior que 1) não possui influência no throughput médio obtido pelos fluxos: para perfis de tráfego com granulosidade 2,4,8, e 16 os resultados foram praticamente semelhantes.

Um resultado obtido não esperado foi a pequena variação do throughput médio à medida que a quantidade de microfluxos por perfil de tráfego varia entre 2 e 16. Como pode ser observado na figura 26, o throughput médio obtido pelos microfluxos para simulações com políticas de tráfegos com 2,4, 8 ou 16 microfluxos são praticamente semelhantes.

Não foram verificadas variações observáveis na latência média obtida para os pacotes marcados para a agregação com menor probabilidade de descarte (pacotes verdes ou dentro do perfil de tráfego), conforme pode ser observado na tabela 7 do Apêndice 1. Este resultado já era esperado por não ter havido durante as simulações variações nos parâmetros de reserva de recursos (r e b) dos microfluxos: o tráfego UDP em excesso foi marcado para a agregação com maior probabilidade de descarte, não influenciando a alocação de recursos dos pacotes marcados como dentro do perfil (verdes). A quantidade de pacotes descartados marcados como dentro do perfil de tráfego (verdes) foi nula ou bastante pequena em todas as simulações, o que também pode ser explicado devido ao fato dos perfis de tráfego não terem sido alterado, mantendo a quantidade de pacotes marcados como verdes sempre em torno de 60% da capacidade do enlace gargalo .

4.2 Diferenciação utilizando o Gerenciamento Ativo de Filas

O gerenciamento ativo de filas em uma classe de encaminhamento assegurado possibilita uma forma de diferenciação entre agregações utilizando apenas a probabilidade de descarte de pacotes. Uma mesma classe de encaminhamento assegurado pode possuir até 3 agregações de comportamento mapeadas para diferentes comportamentos-por-salto. O objetivo deste item é o de estudar uma implementação do suporte a microfluxos de carga controlada em uma classe de encaminhamento assegurado onde os microfluxos são mapeados para mais de uma agregação de comportamento. Este tipo de implementação é interessante por possibilitar a diferenciação do tráfego em um maior número de classes, permitindo um melhor gerenciamento.

Para a análise foi executado um grupo de 9 simulações nas quais os microfluxos foram mapeados para duas agregações de comportamento: uma com menor probabilidade de descarte para o tráfego com requisitos de qualidade (carga controlada) e outra com maior probabilidade de descarte para o tráfego sem requisitos de qualidade e o tráfego em excesso com relação ao perfil de tráfego. O tráfego sem requisitos de qualidade não possuiu perfis de tráfego associados, de modo a representar tráfego de melhor esforço (conforme [25] e [26]). Foi executado um grupo de 9 simulações nas quais a taxa de reposição de pacotes dos microfluxos com requisitos de qualidade foi variada entre 12500 bps e 112500 bps com acréscimos de 12500 bps. Cada simulação contou com 16 fluxos FTP de melhor esforço e 16 fluxos FTP de carga controlada; distribuídos em roteadores limítrofes em separado. O gerenciamento ativo de filas utilizado contou com os parâmetros do RIO 0.02/10/20 para os pacotes marcados como excedentes e os de melhor esforço e 0.1/20/40 para os pacotes marcados como dentro do perfil. Todo o tráfego foi configurado para utilizar as mesmas filas nos nós de serviços diferenciados, de modo a caracterizar uma única classe de encaminhamento assegurado. A política de tráfego utilizada foi a de Token Bucket simples, com tamanho do balde semelhante para todos os fluxos de 10 pacotes. As simulações foram executadas por um período de 100s do tempo de

simulação e os pacotes simulados foram configurados para possuir um tamanho máximo de 576 bytes.

A tabela 3 mostra o throughput médio obtido pelos microfluxos de melhor esforço e de carga controlada de acordo com a variação da taxa de reposição de pacotes do perfil de tráfego. O critério utilizado para definir o throughput considerado como justo é o mesmo já especificado na equação descrita no item 4.1.1.

Taxa de Reposição de Pacotes por Microfluxo (kbps)	Throughput médio dos Microfluxos de carga controlada (kbps)	Throughput Justo (kbps)	Throughput Médio dos Microfluxos na agregação com maior probabilidade de descarte (kbps)	Throughput Justo (kbps)
13	51 ± 13	69	37±12	56
25	89 ± 10	75	27±5	50
38	105 ± 12	81	23±10	43
50	106 ± 9	88	24±6	38
63	107 ± 7	94	22±9	31
75	110 ± 8	100	21±7	25
88	118 ± 12	106	13±7	19
100	123 ± 8	113	9±5	13
113	127 ± 6	119	3±2	6

Tabela 3: Throughput médio de microfluxos TCP simulados

Os resultados mostram que os microfluxos TCP da agregação de comportamento com menor probabilidade de descarte (representando os fluxos de carga controlada) obtêm um throughput sempre maior que o reservado pela taxa de reposição de pacotes, em prejuízo do tráfego de melhor esforço.

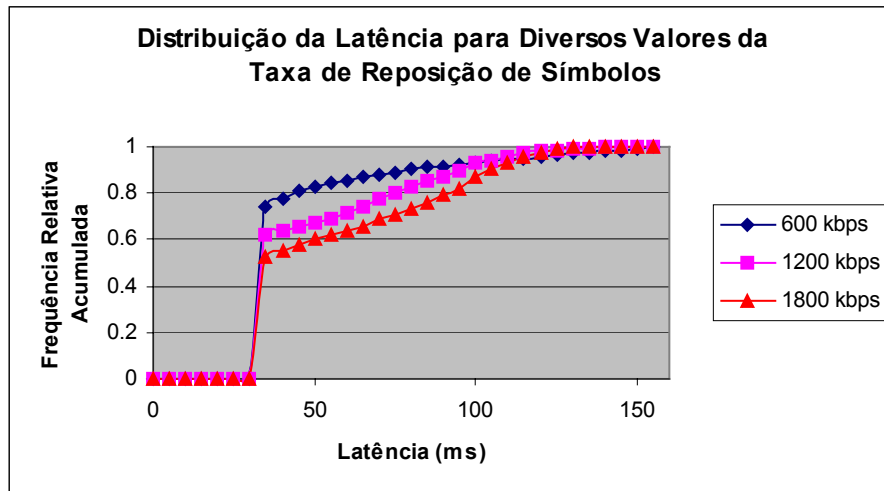


Figura 27: Frequência Relativa Acumulada da Latência para simulações com e sem tráfego de melhor esforço

As simulações também mostraram que a taxa de reposição de pacotes possui grande influência na determinação da latência média para as simulações realizadas. O gráfico 25 demonstra esta característica: à medida que a taxa de reposição de pacotes aumenta, a latência média e a dispersão também aumentam. Deve-se notar que esta característica não foi observada no item 4.1.1. Isto se deve ao aumento no tamanho médio da fila do enlace gargalo causado pelo aumento da taxa de reposição de pacotes: para baixos valores da taxa de reposição o tamanho médio da fila é menor, devido ao fato dos microfluxos TCP de melhor esforço terem grandes quantidades de pacotes descartados (agregação com maior probabilidade de descarte) e dos microfluxos TCP com requisitos de qualidade terem pequeno número de pacotes dentro do perfil. Com o aumento da taxa de reposição de pacotes passa a haver um maior número de pacotes na agregação com maior probabilidade de descarte, o que faz com que a janela TCP dos microfluxos com requisitos de qualidade aumente, aumentando assim o tamanho médio das filas associadas ao enlace gargalo.

4.3 Escalonamento entre Classes de Tráfego

Como já visto há na especificação do encaminhamento assegurado quatro

classes de tráfego, onde para cada classe devem existir recursos alocados nos roteadores do domínio de serviços diferenciados. Não há entretanto descrições sobre como deve ser implementado o relacionamento entre as classes. Isto possibilita que as implementações possam utilizar diversos tipos de disciplinas de filas, o que causa efeitos diferentes no tratamento obtido pelo tráfego em cada classe.

As simulações neste item procuraram estudar os efeitos do uso de uma disciplina de filas que provê isolamento entre as classes de tráfego (WRR) e do uso de uma disciplina de filas por prioridade (PRR) com tráfego TCP e UDP. O isolamento entre as classes possibilita que o tráfego em excesso de uma das classes não degrade o tratamento ofertado às outras classes de tráfego. Apesar do PRR não possuir esta característica, seu uso permite a obtenção de um melhor tratamento para as classes de maior prioridade, principalmente com relação à latência [29]. A análise será realizada em separado para tráfego TCP e UDP.

4.3.1. Tráfego TCP

Um grupo de simulações foi realizado para comparar os efeitos do uso de disciplinas de filas PRR e WRR no escalonamento entre classes de tráfego com tráfego TCP apenas. A topologia de simulação foi a mesma já descrita com a diferença de aumento da capacidade do enlace gargalo para 4 Mbps. Foram utilizados 32 microfluxos TCP com tráfego FTP, sendo 16 mapeados para uma das classes de tráfego e 16 para outra. Nas simulações utilizando WRR, a disciplina de fila foi configurada com pesos iguais para ambas as classes de tráfego, de modo que haja uma banda passante garantida de no mínimo metade do enlace gargalo para cada uma. Em uma das classes de tráfego foi variada a taxa de reposição de pacotes do perfil de tráfego dos microfluxos, de modo que a soma das taxas de reposição de todos variasse entre 200 kbps e 1800 kbps correspondendo a 10% e 45% do enlace gargalo. A outra classe de serviço teve a taxa de reposição de pacotes mantida constante em 1800 kbps. Nas simulações utilizando PRR, a classe de serviço a qual teve a banda passante reservada variada foi a de maior prioridade. Ambas as classes foram configuradas com o gerenciamento de filas RIO com os parâmetros 0.02/20/40 para o

tráfego dentro do perfil e 0.1/10/20 para o fora do perfil. O restante dos parâmetros utilizados foram mantidos semelhantes aos já descritos para o item anterior.

A tabela 4 mostra o throughput médio obtido pelas classes de tráfego nas simulações utilizando WRR e utilizando PRR. Observando os resultados, pode-se observar a diferença na qualidade obtida provocada pelo isolamento de tráfego: com o uso de disciplina de fila PRR os microfluxos TCP da classe de serviço de maior prioridade obtiveram um throughput em média cinco vezes maior do que o obtido pelos microfluxos pertencentes à classe de serviço de menor prioridade. Com o uso da disciplina de fila WRR, os throughputs obtidos por ambas as classes de tráfego foram semelhantes, com pequenas diferenças provocadas pela variação da banda passante reservada na classe 1. Uma observação interessante é a de que para as simulações utilizando PRR o throughput obtido por ambas as classes de tráfego não variou.

Banda Passante Reservada	PRR		WRR	
	Classe 1 (kbps)	Classe 2 (kbps)	Classe 1 (kbps)	Classe 2 (kbps)
10%	225 ± 43	40±15	118 ±18	148 ± 21
20%	221 ± 45	44±12	120± 20	147 ± 15
30%	225 ± 35	39±17	124±21	143 ± 20
40%	233 ± 34	33±13	128±16	138 ± 16
50%	231 ± 29	33 ±10	126±22	141 ± 15
60%	230 ± 30	36 ±15	127±17	140 ± 19
70%	239 ± 30	27 ±11	130± 18	136 ± 14
80%	244 ± 30	22 ± 9	129±14	137 ± 16
90%	234 ± 29	32 ±14	129±19	137 ± 24

Tabela 4: Throughput Médio obtido pelos Microfluxos nas simulações utilizando PRR e WRR com tráfego TCPTabela

A tabela 5 mostra os resultados obtidos para a latência. Com relação às simulações utilizando a disciplina de fila WRR, pode-se observar a latência média da classe 1 aumenta à medida que aumenta a taxa de reposição de pacotes e que a latência da classe 2 não varia, conforme o esperado pelo isolamento entre as classes de encaminhamento assegurado. O aumento da latência proporcionalmente ao aumento da banda passante reservada pode ser explicado pelo aumento do tamanho da fila dos roteadores provocado pelo maior número de pacotes dentro do perfil de tráfego existente com o aumento da banda passante. Pode ser observado que o uso de

disciplina de fila PRR permite que a classe de serviço de maior prioridade (classe 1) obtenha uma latência menor e que varia pouco (menor jitter) à medida que a banda passante reservada aumenta. Ao mesmo tempo, a classe de menor prioridade obteve uma latência muito maior do que a da classe de maior prioridade.

Banda Passante Reservada	PRR		WRR	
	Classe 1 (ms)	Classe 2 (ms)	Classe 1 (ms)	Classe 2 (ms)
10%	36±6	183±33	37±13	54±34
20%	36±6	165±30	39±15	55±35
30%	37±6	186±33	40±17	56±36
40%	37±7	216±36	42±19	57±36
50%	37±8	212±37	43±20	56±36
60%	38±8	204±38	44±22	57±36
70%	38±9	255±46	46±22	57±37
80%	39±9	295±49	47±23	58±37
90%	39±10	226±43	48±24	58±37

Tabela 5: Latência Média obtida pelos microfluxos nas simulações utilizando PRR e WRR com tráfego TCP

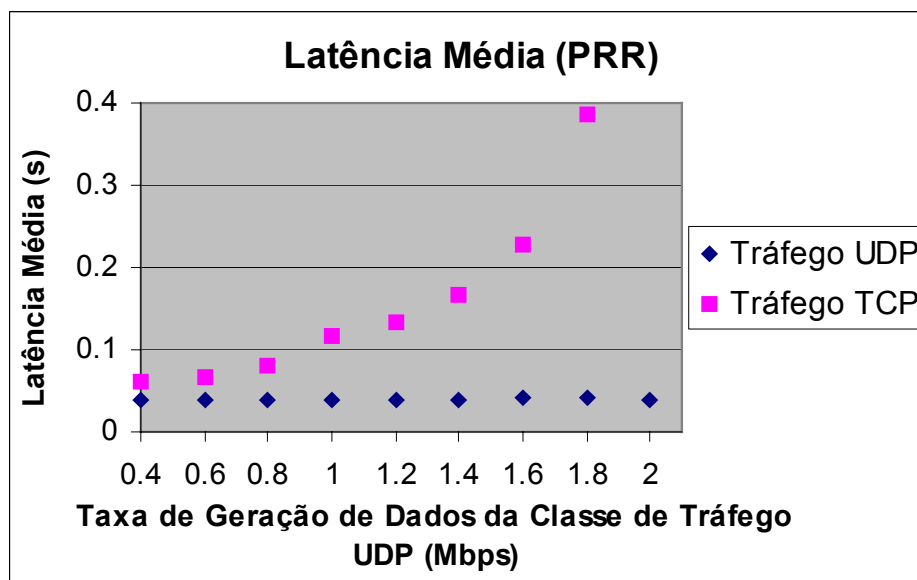


Figura 28: Distribuição Relativa da Latência para 90% de Banda Passante Reservada

4.3.2. Tráfego UDP

Um segundo grupo de simulações foi executado para estudar a influência do tráfego UDP em excesso na qualidade obtida por classes de encaminhamento assegurado escalonadas com o uso de disciplina de filas PRR, onde fluxos UDP foram mapeados para a classe de tráfego de maior prioridade e fluxos TCP para a de menor prioridade. Para efeito de comparação, foram também executadas simulações com a disciplina de filas WRR, nas quais foram configurados pesos iguais para cada classe de serviço reservando metade da banda passante para cada uma. Em cada simulação foram implementadas 16 fontes CBR e 16 fontes TCP, sendo o tráfego UDP mapeado para uma das classes de tráfego e o tráfego TCP para outra.

Foram executadas 8 simulações com disciplina de fila PRR e 8 com disciplina de fila WRR, onde a taxa de reposição de tráfego das fontes UDP foi variada entre 20% e 100% da banda passante reservada para a classe de serviço com o uso da disciplina de fila WRR no enlace gargalo, ou seja o tráfego gerado para a agregação UDP foi variado entre 400 kbps e 2Mbps. Foi utilizada uma política de tráfego por agregação, onde foram utilizadas 4 agregações simulando diferentes redes externas geradoras de tráfego, com um perfil de tráfego para cada agregação UDP com taxa de reposição de pacotes correspondendo a 400 kbps (20% da banda passante reservada para as agregações com o uso da disciplina de filas WRR). Para a classe de serviço TCP, a taxa de reposição de pacotes foi de 300 kbps para cada uma, correspondendo a um total de 1.2 Mbps ou 60% da banda passante reservada com o uso do WRR. A profundidade do balde para todos os microfluxos foi mantido em 10 pacotes ou 5760 bytes. Em cada agregação, foram simulados dois microfluxos UDP e dois TCP.

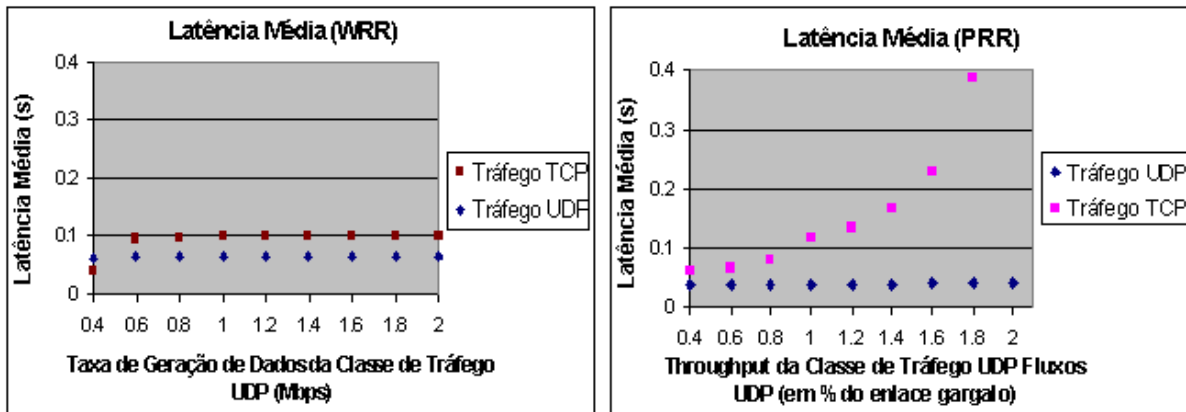


Figura 29: (a) e (b): Gráficos da Latência Média para a disciplina de filas WRR (a) e PRR (b)

A tabela 10 (Apêndice 1) mostra os resultados obtidos para a latência média, os quais são também mostrados nos gráficos 27 (a) e (b). O gráfico 27 (a) mostra os resultados obtidos para as simulações nas quais foi utilizada a disciplina de filas WRR. Como pode ser notado, não há grandes variações na latência média, tanto para os pacotes pertencentes à classe UDP quanto à classe TCP. Este resultado se deve ao isolamento entre as classes de serviço implementado pela disciplina de filas WRR, o qual não permite que a variação do tráfego gerado pela classe de serviço com fluxos UDP afete a classe com fluxos TCP. Ao mesmo tempo, os fluxos UDP durante as simulações geram tráfego sempre menor que o mínimo reservado para a classe de tráfego, o que faz com que a latência média para os pacotes UDP também não sofra grandes variações.

Os resultados referentes ao uso da disciplina de filas PRR mostram um crescimento da latência média da classe de serviço de menor prioridade (TCP) à medida que a taxa de reposição de pacotes dos microfluxos UDP é aumentada, o que já era esperado devido às características do PRR. Entretanto, há duas características nos resultados obtidos que devem ser frisadas. Uma é a de que o aumento da latência se dá de forma bem mais acentuada do que os resultados obtidos no item anterior (4.5.1) para o tráfego TCP na fila de menor prioridade. Este resultado ocorreu mesmo tendo sido utilizado um perfil de tráfego com baixa taxa de reposição de pacotes (10% do enlace gargalo) para a classe de serviço UDP de maior prioridade. Isto demonstra

que o gerenciamento ativo de filas (RIO) possui melhores resultados no isolamento entre classes de tráfego para o tráfego TCP do que para o tráfego UDP. Para baixos valores da taxa de geração de tráfego dos fluxos UDP (até 25% do enlace gargalo) a latência média obtida para a fila de baixa prioridade foi semelhante à observada com o uso da disciplina de filas WRR.

A segunda característica foi a menor dispersão obtida para a latência para a fila de maior prioridade com o uso da disciplina de filas PRR, conforme mostrado na tabela 2. A classe de encaminhamento assegurado de maior prioridade nas simulações com o uso de PRR obtém uma dispersão (indicada pelo desvio padrão) muito menor do que as classes restantes (inclusive as relacionadas com as simulações utilizando WRR); o que sugere que o PRR deve ser utilizado quando se deseja minimizar o jitter e a latência.

Com relação ao throughput, os resultados sumarizados na tabela 11 (apêndice) demonstram novamente o isolamento entre as classes de tráfego existente no WRR e inexistente no PRR: para as classes de encaminhamento assegurado implementadas utilizando-se WRR a variação da taxa de geração de pacotes UDP não causa grandes variações no throughput médio dos microfluxos, ao passo que para as classes implementadas utilizando-se PRR esta variação é bastante acentuada com a classe de serviço de maior prioridade (UDP) obtendo um maior número de recursos em detrimento da classe de menor prioridade (TCP). O fato de que nas simulações utilizando-se WRR o throughput médio obtido pelos microfluxos UDP não obteve grandes variações mesmo com o aumento da taxa de geração de dados pode ser explicado pelo gerenciamento ativo de filas (RIO) o qual descartou com maior probabilidade os pacotes UDP em excesso com relação ao perfil de tráfego.

Capítulo 5 - Trabalhos Futuros e Conclusões

5.1. Trabalhos Futuros

A interoperação entre as arquiteturas de serviços diferenciados e de serviços integrados é uma área de pesquisa relativamente nova, apresentando diversos tópicos que podem ser objeto de estudos futuros. Os tópicos podem tanto se referir à expansão dos resultados apresentados nesta dissertação quanto ao estudo de áreas relacionadas que, por questão de tempo ou de escopo, não puderam ser analisadas.

Com relação à expansão dos resultados apresentados, algumas possibilidades de trabalhos futuros seriam:

- simulações com maior número de microfluxos: a interoperação entre os serviços integrados e os serviços diferenciados deverá ser de grande aplicação nas regiões de maior tráfego em redes, onde a escalabilidade é um fator limitante. Simulações com um maior número de microfluxos permitiriam aproximar o modelo simulado do que será observado nas implementações da arquitetura híbrida, onde as agregações de comportamento deverão contar com grande número de microfluxos. Permitiria também a obtenção de resultados complementares aos realizados, principalmente os relacionados com as conseqüências da agregação de fluxos de serviços integrados;
- estudo do impacto do uso de diferentes implementações do controle de congestionamento do TCP: como já citado, as simulações realizadas neste trabalho foram baseadas apenas na implementação do TCP do tipo Reno. Entretanto, a quantidade de implementações de TCP do tipo Sack e New Reno em uso na Internet está em constante expansão, sendo interessante analisar o impacto do uso destas na arquitetura híbrida;
- estudo da influência do uso de políticas de tráfego por agregação de microfluxos na alocação de recursos entre fluxos TCP com diferentes taxas de

reposição de pacotes: como citado no item 4.1.2, fluxos TCP em agregações de encaminhamento assegurado com políticas de tráfego Token Bucket com menor taxa de reposição de pacotes (r) têm facilidade em obter um throughput maior que a taxa de reposição. Simulações podem ser utilizadas para estudar como o uso de políticas de tráfego por agregação afetam a alocação de recursos entre microfluxos TCP onde o parâmetro r é variado;

- validação experimental dos resultados das simulações: apesar do modelo utilizado nas simulações ser bastante próximo de um sistema real, há parâmetros e características que só podem ser observados através de experimentos. Bons exemplos são as influências do uso de recursos nos roteadores (processador e memória) no tratamento ofertado ao tráfego. A implementação do modelo de simulação em um sistema real poderia não só corroborar os resultados obtidos como também indicar a influência de fatores não analisados durante as simulações.

Com relação a áreas não abordadas diretamente por esta dissertação, algumas sugestões são:

- suporte ao serviço MPLS e suas características;
- simulação de um ambiente ATM utilizando MPLS.

5.2. Conclusões

Nesta dissertação foi realizada uma apresentação dos principais conceitos relacionados com qualidade de serviço; incluindo-se métricas, políticas de tráfego e disciplinas de fila. A seguir foram descritas as principais características das arquiteturas de serviços diferenciados e de serviços integrados, bem como da proposta de interoperação entre ambas.

Na parte experimental foram realizadas simulações com vistas ao estudo do suporte à arquitetura de serviços integrados em domínios de serviços diferenciados. As simulações focaram-se no serviço de carga controlada da arquitetura de serviços

integrados e no serviço de encaminhamento assegurado da arquitetura de serviços diferenciados.

As simulações foram divididas em três grupos. No primeiro foram realizados estudos relativos ao suporte a fluxos de carga controlada quando estes representam o único tráfego em uma classe de encaminhamento assegurado. Foram realizados experimentos neste grupo para verificar a influência da quantidade de microfluxos TCP e da taxa de reposição de símbolos na qualidade de serviço obtida pelos microfluxos. Os resultados mostraram que a quantidade de microfluxos influencia diretamente na latência dos pacotes marcados como dentro do perfil de tráfego: quanto maior é o número de microfluxos TCP maior é a latência média e sua dispersão. Não foram observados durante as simulações influência da taxa de reposição de símbolos na qualidade de serviço obtida, bem como dos parâmetros estudados na quantidade média de pacotes descartados e no throughput médio obtido pelos microfluxos.

Ainda no primeiro grupo de simulações foi estudada a influência da agregação de perfis de tráfego de microfluxos TCP na justiça na alocação de recursos entre os mesmos. A alocação de recursos foi observada utilizando-se como parâmetro o throughput médio obtido pelos microfluxos durante as simulações. Os resultados mostraram que microfluxos TCP com maior latência física média são prejudicados na disputa por recursos com a agregação de perfis de tráfego. Finalmente, foi estudada a influência da agregação de perfis de tráfego na alocação de recursos entre microfluxos TCP e UDP; com a geração de tráfego UDP em excesso em relação à taxa de reposição de símbolos de cada microfluxo. As simulações mostraram que a agregação de perfis de tráfego permite que o tráfego UDP em excesso obtenha maiores vantagens em relação aos microfluxos TCP; levando vantagens no throughput médio observado.

No segundo grupo foram colocados em uma mesma classe de encaminhamento assegurado dois diferentes tipos de tráfego com requisitos de qualidade de serviço diferentes: um de maior prioridade representando tráfego de microfluxos de carga controlada e outro de menor prioridade representando tráfego de microfluxos sem requisitos de qualidade (“melhor esforço”). As simulações foram realizadas apenas com

tráfego TCP, tendo sido mostrado que a taxa de reposição de símbolos possui influência na latência média e em sua dispersão, em contraste com o observado nas simulações do grupo 1. Com relação ao throughput observaram-se resultados que aparentemente contrastaram com os obtidos em outros trabalhos, mas que puderam ser explicados pelas características das simulações realizadas.

No terceiro grupo foram estudadas agregações de tráfego em diferentes classes de encaminhamento assegurado com o uso de disciplinas de fila PRR e WRR para implementar a diferenciação entre as classes, sendo utilizadas nas simulações duas. Inicialmente foram realizadas simulações apenas com microfluxos TCP, tendo sido variada a taxa de reposição de símbolos de uma das classes de tráfego, a qual para as simulações utilizando PRR correspondeu à de maior prioridade. Os resultados mostraram que para as simulações utilizando-se WRR; há um isolamento entre as classes de tráfego que possibilita que a variação da taxa de reposição de símbolos dos microfluxos de uma das classes de tráfego não afete a outra. Com o uso do PRR foi mostrado que o throughput de uma das classes de tráfego é bem menor que a da outra, não tendo sido observada grandes variações desta métrica devido à variação da taxa de reposição de símbolos. Para a latência essa influência foi observada: o aumento da taxa de reposição de símbolos gerou um aumento da latência média da classe de menor prioridade. Observou-se ainda que com o uso do PRR a latência média da fila de maior prioridade apresentou-se bem menor que a observada para as classes de tráfego nas simulações utilizando WRR.

Finalmente foram realizadas simulações que estudaram tráfego UDP e TCP em duas classes de tráfego em separado utilizando disciplinas de fila PRR e WRR. As simulações foram realizadas com a variação da taxa de geração de pacotes das fontes de tráfego UDP. Observou-se que, para as simulações utilizando o WRR, as variações na taxa de geração de pacotes UDP não causaram modificações na latência média e no throughput da classe de tráfego com tráfego TCP. Já para as simulações disciplinas de filas PRR, foi observada um aumento da latência média dos pacotes pertencente à classe de tráfego TCP, o qual revelou-se mais acentuado que o observado para as simulações utilizando-se apenas tráfego TCP e disciplina PRR. Observou-se ainda que

a latência média dos pacotes UDP sofreu pouca variação, mantendo-se com valores baixos. Com relação ao throughput, os resultados corroboraram que, com o uso da disciplina PRR o tráfego da classe de serviço de menor prioridade (TCP) é prejudicado em favor da classe de maior prioridade.

Apêndice 1: Resultados das Simulações

Tabela A.1: Variação do Número de Fluxos TCP em um mesmo comportamento-por-nó de Serviço Assegurado com Política de Tráfego Token Bucket Simples (referente ao Item 4.3.1)

Taxa de Reposição de Símbolos por Microfluxo (bps)	Throughput Médio por fluxo para 8 Microfluxos (kbps)	Throughput Médio por fluxo para 16 Microfluxos (kbps)	Throughput Médio por fluxo para 24 Microfluxos (kbps)	Throughput Médio por fluxo para 32 Microfluxos (kbps)
15625	263 ± 20	133±13	89±8	67±10
31250	260 ± 18	133±14	89±10	67±11
46875	261 ± 14	130±11	88±10	66±10
62500	258 ± 18	133±10	88±10	66±10

Tabela A.2: Variação do Número de Fluxos TCP em um mesmo comportamento-por-nó de Serviço Assegurado com Política de Tráfego por Agregação (referente ao item 4.3.1)

Taxa de Reposição de Símbolos por Microfluxo (bps)	Throughput Médio para 8 Microfluxos (kbps)	Throughput Médio para 16 Microfluxos (kbps)	Throughput Médio para 24 Microfluxos (kbps)	Throughput Médio para 32 Microfluxos (kbps)
15625	260±37	133±18	89±13	67±12
31250	262±41	133±14	89±14	67±12
46875	256±27	132±31	89±21	67±11
62500	261±21	133±24	89±24	67±13

Tabela A.3: Variação da Latência de acordo com o número de fluxos TCP em um mesmo comportamento-por-nó de Serviço Assegurado com Política de Tráfego Token Bucket Simples (referente ao Item 4.3.1)

Taxa de Reposição de Símbolos por Microfluxo (bps)	Latência Média por microfluxo para 8 Microfluxos (ms)	Latência Média por microfluxo para 16 Microfluxos (ms)	Latência Média por microfluxo para 24 Microfluxos (kbps)	Latência Média por microfluxo para 32 Microfluxos (kbps)
15625	49±2	70±3	76±5	79±5
31250	49±2	69±3	77±5	79±5
46875	49±2	70±4	78±5	79±5
62500	49±2	69±4	77±5	78±5

Tabela A.4: Variação da Latência de acordo com o número de fluxos TCP em um mesmo comportamento-por-nó de Serviço Assegurado com Política de Tráfego Token Bucket por agregação (referente ao Item 4.3.1)

Taxa de Reposição de Símbolos por Microfluxo (bps)	Latência Média por microfluxo para 8 Microfluxos (ms)	Latência Média por microfluxo para 16 Microfluxos (ms)	Latência Média por microfluxo para 24 Microfluxos (kbps)	Latência Média por microfluxo para 32 Microfluxos (kbps)
15625	49 ± 2	75±4	78±5	79±5
31250	50±2	76±5	78±5	79±5
46875	50 ±2	75±5	79±5	80±5
62500	50±2	75±4	79±5	80±5

Tabela A.5: Throughput médio obtido por microfluxos TCP com Latências física diferentes e políticas de tráfego Token Bucket (referente ao Item 4.3.2)

Latência Física do Microfluxo (ms)	Throughput Médio para taxa de reposição de símbolos de 18.75 kbps (kbps)	Throughput Médio para taxa de reposição de símbolos de 37.5 kbps (kbps)	Throughput Médio para taxa de reposição de símbolos de 56.25 kbps (kbps)
20	41±9	51±5	55±5
60	44±8	51±5	55±4
100	42±8	50±6	56±4
140	45±7	51±5	56±4
180	43±7	51±5	54±5
220	41±7	54±5	56±5
260	44±7	52±6	56±5
300	42±6	54±5	56±5
340	42±8	48±5	56±5
380	44±9	52±5	55±4
420	41±5	49±5	55±4
460	42±8	50±4	55±4
500	40±7	50±5	54±5
540	40±7	51±5	56±5
580	44±7	50±4	55±4
620	45±8	52±4	55±3

Tabela A.6: Throughput médio obtido por microfluxos TCP com Latências física diferentes e políticas de tráfego Token Bucket por agregação de 32 microfluxos (referente ao Item 4.3.2)

Latência Física do Microfluxo (ms)	Throughput Médio para taxa de reposição de símbolos de 18.75 kbps (kbps)	Throughput Médio para taxa de reposição de símbolos de 37.5 kbps (kbps)	Throughput Médio para taxa de reposição de símbolos de 56.25 kbps (kbps)
20	51±17	64±18	69±12
60	50±16	58±13	69±13
100	56±22	65±16	66±12
140	54±17	65±19	64±14
180	56±22	67±19	67±11
220	54±18	60±16	66±14
260	49±16	64±16	68±12
300	55±17	62±19	66±14
340	56±23	68±19	66±13
380	52±20	64±16	66±13
420	47±14	65±18	66±13
460	53±18	62±14	64±16
500	50±17	60±17	66±14
540	51±20	63±16	61±15
580	50±20	63±18	61±15
620	55±19	61±18	63±14

Tabela A.7: Throughput médio obtido por tráfego UDP e TCP em uma mesma classe de serviço com diferenciação por gerenciamento ativo de filas (referente ao Item 4.3.3)

Número de Microfluxos política	Tráfego UDP em Excesso por Microfluxo (bps)	Throughput médio dos Microfluxos TCP (kbps)	Throughput Médio dos Microfluxos UDP (kbps)
1	0	71 ±11	35± 9
	12500	73 ± 7	44±2
	25000	70 ± 7	53±2
	37500	65 ± 8	62±2
	50000	59 ±6	68 ± 3
	62500	56 ±8	70±3
2	0	69±8	35±1
	12500	45 ± 12	46 ±2
	25000	34±7	58±3
	37500	28±6	70±2
	50000	24±7	79±4
	62500	25±10	91±3
4	0	78±11	35 ±1
	12500	51 ±10	46±2
	25000	35±8	58 ±3
	37500	30±9	69±3
	50000	26±8	78±5
	62500	24±7	88±7
8	0	82±17	35 ±1
	12500	59 ±16	46±2
	25000	41±10	56 ±4
	37500	32±10	69±3
	50000	27±10	78±7
	62500	27±10	89±10
16	0	85±16	34±2
	12500	61±18	46±2
	25000	43±15	57 ±4
	37500	32±10	69±3
	50000	27±10	78±7
	62500	27±12	89±10
32	0	98±14	32 ±2
	12500	87±18	43±3
	25000	77±21	53 ±4
	37500	65±9	64±2
	50000	54±13	74±4
	62500	47±16	81±4

Tabela A.8: Latência média obtida por tráfego UDP e TCP em uma mesma classe de serviço com diferenciação por gerenciamento ativo de filas (referente ao Item 4.3.3)

Número de Microfluxos por política	Tráfego UDP em Excesso por Microfluxo (bps)	Latência Média dos Pacotes dentro do Perfil de Tráfego (ms)
1	0	54 ±27
	12500	53 ±23
	25000	50 ±20
	37500	53±17
	50000	53 ±14
	62500	53 ±14
2	0	55±28
	12500	54 ±26
	25000	51±24
	37500	50±22
	50000	49±21
	62500	50±17
4	0	54±26
	12500	53 ±26
	25000	50±24
	37500	50±22
	50000	49±21
	62500	50±19
8	0	53±24
	12500	52 ±25
	25000	51±23
	37500	49±22
	50000	49±21
	62500	50±18
16	0	52±23
	12500	52±24
	25000	50±23
	37500	49±21
	50000	49±21
	62500	50±18
32	0	77 ±32
	12500	76 ± 3
	25000	76 ± 3
	37500	74 ±22
	50000	74 ±19
	62500	73 ±18

Tabela A.9: Latência Média obtida pelos Pacotes dentro do Perfil de Tráfego em simulações com agregação de tráfego de melhor esforço (referente ao Item 4.4)

Taxa de Reposição de Símbolos por Microfluxo (kbps)	Latência média dos pacotes dos Microfluxos de Carga Controlada dentro do perfil (kbps)
13	47 ± 30
25	46 ± 28
38	47 ± 26
50	49 ± 26
63	50 ± 27
75	53 ± 27
88	54 ± 27
100	57 ± 29
113	59 ± 30

Tabela A.10: Comparação do uso de Disciplinas de Filas CBQ e PRR com tráfego UDP com relação à Latência (referente ao Item 4.5.2)

Tráfego UDP (em % do enlace gargalo)	PRI		WRR	
	Latência Média da Fila UDP (ms)	Latência Média da Fila TCP (ms)	Latência Média da Fila UDP	Latência Média da Fila TCP
20	38±1	60±30	60±30	40±3
30	38±1	60±40	60±40	100±20
40	38±1	80±50	60±40	100±20
50	39±1	100±90	60±40	100±20
60	39±1	100±100	60±40	100±20
70	39±1	200±100	70±40	100±20
80	40±2	200±200	60±40	100±20
90	42±3	400±300	60±40	100±20
100	40±1	1.000±1000	60±40	100±20

Tabela A.11: Comparação do uso de Disciplinas de Filas CBQ e PRR com tráfego UDP com relação ao throughput (referente ao Item 4.5.2)

Tráfego UDP (em % do enlace gargalo)	PRR		WRR	
	Throughput Médio dos Microfluxos TCP (kbps)	Throughput Médio dos Microfluxos UDP (kbps)	Throughput Médio dos Microfluxos TCP (kbps)	Throughput Médio dos Microfluxos UDP (kbps)
20	110±13	24±1	100±19	24±1
30	94±16	36±1	99±17	33±3
40	83±12	48±1	100±15	33±16
50	70±13	60±2	96±13	33±9
60	58±10	70±2	99±12	33±12
70	45±12	83±2	98±17	33±10
80	32±9	95±3	98±11	32±21
90	19±10	110±3	98±16	35±17
100	6±8	120±4	98±9	33±17

Bibliografia

- [1] Melo, E., *Qualidade de Serviço em Redes IP com DiffServ: Avaliação através de Medições*, Dissertação de mestrado UFSC, maio 2001.
- [2] Coultart, J. P., *Qualidade de Serviço no mundo IP*, Minicurso SBRC 2001, maio 2001.
- [3] Braden, S., Clark, D. e Shenker, S., *Integrated Services in the Internet Architecture: an Overview*, IETF RFC 1633, Junho 1994.
- [4] Blake, S. et al, *An Architecture for Differentiated Services*, IETF RFC 2475, Dezembro 1998.
- [5] Santiago, M. F., *Um Ambiente de Simulação de Serviços Integrados e Diferenciados*, dissertação de mestrado UnB, setembro 2001.
- [6] Wroclawski, J., *The use of RSVP with IETF Integrated Services*, IETF RFC 2210, setembro 1997.
- [7] Silva, M. P. , *Estabelecimento de um SLA através de métricas de Performance para Redes sem Fio*, draft, dezembro 2002
- [7] Bernet, Y. et al, *A Framework for Integrated Services Operation over DiffServ Networks*, IETF RFC 2598, Maio 2000.
- [8] Charny, A. e Wroclawski, J., *Integrated Service Mappings for Differentiated Service Networks*, IETF Draft, Março 2000
- [9] Paxson, V. et al, *A Framework for IP Performance Metrics*, IETF RFC 2330, Maio 1999.

- [10] Westerinen, A. et al, *Terminology-Policy Framework Working Group*, IETF Draft, Abril 2001.
- [11] Shenker, S. e Wroclawski, J., *General Characterization Parameters for Integrated Services Network Elements*, IETF RFC 2215, Setembro 1997.
- [12] Prior, L. M, *Qualidade de Serviço em redes de comutação de pacotes*, Dissertação de mestrado, Universidade do Porto, Portugal, março 2001.
- [13] Fergunson, P. e Huston, G., *Quality of Service: Delivering QoS in the Internet and in Corporate Networks*, John Wiley & Sons, 1998.
- [14] Almes, G., Kalidindi, S. e Zekauskas, M., *A One-way Delay Metric for IPPM*, IETF RFC 2679, Setembro 1999.
- [15] Clark, D. et al, *Recommendations on Queue Management and Congestion Avoidance in the Internet*, IETF RFC2309, Abril 1998.
- [16] Ramakrishnan, K., Chiu, D. e Chain, R., *Congestion Avoidance in Computer Networks with a Connectionless Network Layer*, Digital Equipment Corporation, Technical Report TR-510, Novembro 1987.
- [17] Comer, D. E., *Internetworking with TCP/IP – 3rd Edition*, Prentice Hall, 1995.
- [18] Postel, J., *Transmission Control Protocol: DARPA Internet Protocol Specification*, IETF RFC 791, Setembro 1981.
- [19] Padhye, J. e Floyd, S., *On Inferring TCP Behavior*, Proceedings of SIGCOMM, Agosto 2001.
- [20] Fall, K. e Floyd, S., *Simulation-based Comparisons of Tahoe, Reno and SACK TCP*, Computer Communication Review, Vol. 26, no. 3, pp. 5-21, Julho 1996.
- [21] Floyd, S. e Henderson, T., *The NewReno Modification to TCP's Fast Recovery Algorithm*, IETF RFC 2589, Abril 1999.

- [22] Stevens, W., *TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms*, IETF RFC2001, Setembro 1999.
- [23] Zhang, L. e Clark, D., *Oscillating Behavior of Network Traffic: a case study simulation*, Internetworking: Research and Experience, Vol.1 , pp. 101-112, 1990.
- [24] Jacobson, V. e Floyd, S., *Random Early Detection Gateways for Congestion Avoidance*, IEEE/ACM Transactions in Networking, vol.1, no.4, pp. 397-413, Agosto 1993.
- [25] Ibanez, J. e Nichols, K., *Preliminary Simulation Evaluation of an Assured Service*, IETF Draft, Agosto 1998.
- [26] Rezende, J., *Avaliação do Serviço Assegurado para a Diferenciação de Serviços na Internet*, Anais do XVII Simpósio Brasileiro de Redes de Computadores, SBRC'99 pp. 339-353. Maio 1999.
- [27] Cisco Corporation, *Distributed WRED Technical report*, <http://www.cisco.com/univercd/cc/td/doc/product/software/ios111/cc111/wred.htm> , Outubro 1999.
- [28] Shenker, S. e Wroclawski, J., *Network Element Service Specification Template*, IETF RFC 2216, setembro 1997.
- [29] F. Baker et al, *Aggregation of RSVP for IPv4 and IPv6 Reservations*, IETF RFC 3175, Setembro 2001.
- [30] White, P., *RSVP and Integrated Services in the Internet: a Tutorial*, IEEE Communications Magazine, pp. 100-106, Maio 1997.
- [31] Shenker, S., Partridge, C. e Guerin, R., *Specification of the Guaranteed Quality of Service*, IETF RFC 2212. Setembro 1997.

- [32] Wroclawski, J., *Specification of the Controlled-Load Network Element Service*, IETF RFC 2211, Setembro 1997.
- [33] Almquist, P., *Type of Service in the Internet Protocol Suite*, IETF RFC 1349, Julho 1992.
- [34] Carpenter, L. et al, *Per Hop Behavior Identification Codes*, IETF RFC 2836, Maio 2000.
- [35] Fang, W., Seddigh, N. e Nandy, B., *A Time Sliding Window Three Color Marker (TSWTCM)*, IETF RFC 2859, Junho 2000.
- [36] Ghanvani, A. et al, *A framework for Integrated Services over Shared and Switched IEEE 802 LAN Technologies*, IETF RFC 2816, Maio 2000
- [37] Shalwani, F. et al, *A Network Simulator Differentiated Services Implementation*, Documentação Técnica, <http://www7.nortel.com:8080/CTL> , Julho 2000.

Uma proposta para prover QoS em redes IP

Angela Maria Erdtmann

Walter Felix Cardoso Neto

Introdução

Nesta fase de projeto ainda não há nada implementado, pois só foram realizados estudos sobre o assunto a ser desenvolvido.

Este trabalho inicial tem como função explicitar o conceito de Qualidade de Serviço sobre as aplicações em redes, abordando, separadamente, os modelos IntServ, DiffServ e uma arquitetura híbrida, juntamente com os protocolos e componentes de cada um destes modelos. Serão vistas também, as vantagens e desvantagens.

1 - Modelos de Qualidade de Serviço

Devido ao crescimento da utilização de aplicações multimídia na

internet, certos recursos foram introduzidos tornando mais viável e segura a transmissão dos dados. Um desses recursos é *Quality of Service* - QoS, que se torna imprescindível quando se trata de uma aplicação de missão crítica, como por exemplo, a telemedicina. Atualmente, na transmissão dos dados na Internet, é utilizada a filosofia "*Best Effort*", que não provê nenhum tipo de garantia de que os pacotes enviados na rede chegarão ao seu destino. Como muitos hosts estão conectados na rede ao mesmo tempo, os limites da banda de transmissão são excedidos. Isto é, os usuários compartilham a largura de banda com os fluxos de dados de outros usuários e de acordo com a quantidade de banda disponível e definição das rotas, os dados chegam ao seu destino. Entretanto, quando há

congestionamento, pacotes são descartados aleatoriamente não garantindo que a aplicação seja executada com eficiência. Por outro lado, com a introdução do QoS podemos reservar banda para tipos diferentes de fluxo de dados, onde os pacotes não são descartados e a banda não excede valores pré-definidos. Por exemplo, se quisermos transmitir voz em tempo real, uma quantidade X de Mb/s será reservada para tal aplicação. Dentro do QoS na Internet temos diferentes tipos de modelos propostos pelo *Internet Engineering Task Force - IETF*, que se adequam de acordo com o tipo de aplicação e arquitetura da rede.

Neste ponto são apresentadas algumas tentativas de fornecer às redes IP algumas funcionalidades de qualidade de serviço.

CAR (*Committed Access Rate*): esta técnica implementada por *roteador* de determinados fabricantes, limita a largura de banda consumida numa ligação por determinadas aplicações. Por exemplo, pode ser definido que o tráfego HTTP, SMTP ou telnet, ocupe apenas 50% da largura de banda de uma determinada ligação. O restante é para aplicações de VoIP;

CBQ (*Class-Based Queuing*): CBQ classifica como vária o tráfego de uma rede em categorias e atribui-lhes uma determinada percentagem de largura de banda disponível. As classes podem ser fluxos individuais de pacotes ou representar uma categoria inteira de aplicações. Podem ser definidos com base em grupos de endereços IP, protocolos, portas TCP ou UDP que representam as aplicações;

CoS (*Class of Service*): CoS está definido na especificação IEEE 802.1p. Usa 3 bits da trama *Ethernet* para atribuir sete níveis de prioridade a essas tramas. Estes níveis podem ser mapeados em níveis do campo ToS (*type of service*) do pacote IP;

DiffServ (*Differentiated Services*): redefine 6 dos 8 bits do campo ToS do pacote IP para permitir que este campo seja utilizado para a diferenciação de serviços. Estes 6 bits podem ser combinados de forma a constituírem 64 classes de serviço, que representam várias categorias de aplicações. Esta funcionalidade é interessante, mas necessita que todos os *roteador* entendam as categorias do DiffServ. O DiffServ não apresenta garantias absolutas de QoS, como por exemplo, no caso da VoIP, o melhor que

o DiffServ pode fazer é garantir que os pacotes são colocados primeiro nas filas de espera;

IP Precedence: este método compete com o anterior uma vez que também recorre à alteração do campo ToS do pacote IP. Desta vez, este campo é alterado com valores de 0 a 7, sendo o 7 o mais prioritário;

MPLS (Multiprotocol Label Switching): este método é um *standard* do IETF. O DiffServ fornece um mecanismo de identificar classes de serviço mas deixa a implementação dessas classes a cargo das aplicações. MPLS fornece um possível mecanismo exigindo que os roteador passem a ser comutadores (*switches*) de *Layer 3*. Uma das formas de fazer isso é de juntar um *router* e um *switch* ATM. O MPLS requer que exista uma infraestrutura que processe “etiquetas” usadas no protocolo;

Filas QoS: ou também filas de classes de serviço. Neste método, os roteador ou *switches* de uma rede têm um número de filas para cada porta de saída de tráfego. Os pacotes são identificados com as prioridades dos campos ToS e colocados nas filas conforme a sua prioridade. As filas com

maior prioridade são as que são mais rapidamente são atendidas;

RED (Random Early Discard): este método baseia-se em regras definidas para que o *router* possa descartar pacotes de uma fila a partir de um determinado nível de ocupação das filas. Por exemplo, um *router* pode começar a descartar pacotes de uma fila a partir de um valor de 80% de ocupação da mesma. O objetivo é evitar que a fila fique cheia e comece a dispensar pacotes de maior prioridade como os de VoIP. Desta forma, é preferível perder pacotes com prioridade inferior. Este método pode ser combinado com outras técnicas de QoS e não precisa de ser implementado em todos os roteador para ser eficiente;

RSVP (Resource Reservation Protocol): há uns tempos atrás, este protocolo liderava as hipóteses de se tornar uma norma para acrescentar funcionalidades de QoS às redes IP. Um equipamento terminal suportando RSVP poderia fazer pedidos muito específicos de QoS à rede e os roteador com RSVP poderiam garantir esses pedidos. Desta forma, o RSVP precisa que para além dos *routers*, também os terminais implementem o protocolo. Atualmente, muitas das potencialidades esperadas

pelo RSVP passaram para implementações de DiffServ;

ToS (*Type of Service*): o cabeçalho IP contém um campo de 8 bits designado por *type of service* que era supostamente para ser usado para indicar a prioridade de pacotes. A maioria dos fabricantes de *roteador* ignora este campo porque a maioria das aplicações não o usa. Este campo é reutilizado no DiffServ;

Traffic Shaping: utilizando este método, o tráfego é deformado, isto é, ao tráfego que entra numa rede é retirada a componente de rajada que este possa ter. Este processo é feito recorrendo a *buffers*. Desta forma a componente de rajada do tráfego é dispersa ao longo do tempo, garantindo que não existem picos de tráfego em determinadas alturas;

Weighted Fair Queuing (WFQ): este método aplica-se à largura de banda que uma aplicação recebe nas filas de saída. A cada fluxo de pacotes a que o WFQ é aplicado é colocado em filas separadas e recebe largura de banda de uma forma pesada e variável;

WRED (*Weighted Random Early Discard*): é uma variante pesada do RED. Num *router* RED, os pacotes que são descartados são escolhidos aleatoriamente. Neste caso, essa escolha

não é arbitrária, tentando-se escolher os pacotes com mais baixa prioridade.

Podemos também, ter qualidade de serviço direccionada para o ATM, que é uma tecnologia que tem como aplicação nativa o QoS. Porém, devido ao fato de ser uma tecnologia altamente complexa, pode mostrar dificuldades quanto à implementação deste recurso. No entanto, os testes com QoS sobre ATM ainda se encontram em fase experimental, gerando grandes expectativas em torno dos resultados.

2- Integrated Services - IntServ

Este modelo de qualidade de serviço é caracterizado essencialmente pela reserva de recursos (largura de banda, atraso e jitter), antes do estabelecimento da comunicação. Este serviço utiliza o protocolo de sinalização RSVP, que será abordado com mais detalhes no próximo sub-item. Na sinalização RSVP existe troca de mensagens de controle entre emissor e receptor de forma que num determinado período de tempo possamos alocar uma faixa da largura de banda para a transmissão dos dados. Neste modelo

temos alocação para dois tipos de serviços, além do “Best Effort”:

- Serviços Garantidos - aplicações que necessitam de um atraso constante.
- Serviços de Carga Controlada – aplicações que necessitam de segurança e um limite de variação de atraso (jitter), eliminando a idéia de “best effort”.

Aplicações que exigem esses tipos de serviço devem configurar caminhos e reservar recursos antes da transmissão dos dados. A implementação do IntServ é feita por quatro componentes:

1. protocolo de sinalização(RSVP)
2. rotina de controle de admissão,
3. classificador,
4. escalonador de pacotes.

Esses componentes têm por função organizar os pacotes de forma que a Qualidade de Serviço seja aplicada.

2.1 – Resource Reservation Protocol - RSVP

O RSVP é usado para gerenciar recursos ao longo do caminho no qual deseja-se utilizar aplicações que necessitem de QoS. Ele não realiza

transporte de dados, é apenas um protocolo de sinalização que atua juntamente com o **ICMP** (*Internet Control Management Protocol*) e **IGMP**(*Internet Group Management Protocol*). O processo de sinalização se dá antes da transmissão dos dados e é renovado sempre que necessário. Para haver a requisição dos recursos, existem mensagens que são trocadas entre o receptor e o transmissor, são elas: *PATH* e *RESV*.

2.2-Rotina de Controle de Admissão

O controle de admissão tem somente a função de determinar se um fluxo de dados poderá ser aceito ou não, de acordo com a banda disponível. Este componente é requisitado de forma que sua decisão não interfira nos fluxos previamente aceitos pelo roteador.

2.3- Classificador

Com a introdução dos parâmetros de QoS, foi necessária uma forma de classificação mais específica dos pacotes. Além de analisarmos o endereço do destino, levamos também em consideração a porta e número de protocolo.

Poderemos tomar como exemplo uma seqüência de música que seria reconhecida por uma porta particular. Os pacotes são marcados de modo que possamos reservar banda para determinado fluxo. E este vai ser atendido de acordo com sua prioridade de fila dentro do roteador. Quem cuida das prioridades da fila é o escalonador, que implementa algoritmos que selecionam os pacotes que serão atendidos. Isto é claro, ocorre de acordo com a complexidade do algoritmo e marcação dos pacotes. Existindo dois fluxos com a mesma classificação, se o estilo de reserva permitir, eles se unirão. Caso contrário um dos fluxos será tratado de forma específica.

2.4-Escalonador

Como foi citado acima, o papel do escalonador é estabelecer políticas de enfileiramento e prevenção de congestionamento nas interfaces dos roteadores e switches de nível 3, aqueles que também *roteiam*, para atender as prioridades do fluxo. O escalonador trabalha com algoritmos que fazem tais implementações de acordo com a necessidade de QoS para determinados serviços. O mecanismo mais conhecido é o *FIFO – First In First Out*, onde os pacotes são tratados de acordo com a

ordem de chegada. Uma fila FIFO é um mecanismo de repasse, não implementando nenhum tipo de classificação. É fácil percebermos, então, que quando estamos utilizando qualidade de serviço, este tipo de mecanismo não é adequado. Outro algoritmo que também merece ser citado é o *WFQ – Weighted Fair Queueing*, onde é possível ponderar os tipos de fluxo. Isto é, são associados pesos para determinados tipos de fluxo, também de acordo com as prioridades de cada um. Ele trabalha da seguinte forma:

- O WFQ coloca para o início da fila o tráfego que tem maior prioridade, reduzindo o tempo de resposta desse fluxo. Ao mesmo tempo, o WFQ compartilha banda com outros fluxos de menores prioridades, porém alocando uma largura de banda menor, já que os de menor prioridade têm também menor peso junto ao WFQ.

Este algoritmo se adapta automaticamente às mudanças das condições de tráfego. Podemos também citar o *PQ – Priority Queueing*, onde o tráfego de entrada é classificado em quatro níveis de prioridade: alta, média, baixa e normal. Os pacotes que não são marcados levam configuração padrão, isto é, são tratados de acordo com a

prioridade normal. Neste mecanismo o tráfego classificado e marcado como prioritário tem preferência absoluta em relação aos outros fluxos. Esta é uma das desvantagens do PQ, pois isso pode causar um aumento de jitter e atrasos consideráveis em aplicações de menor prioridade.

Numa situação extrema pode acontecer até de um fluxo com menor prioridade nunca chegar a ser enviado se o fluxo de maior prioridade ocupar toda largura de banda. Isso ocorre em conexões de baixa velocidade. Outra desvantagem do PQ é que se um fluxo não receber classificação ele pode também não ser enviado. Por isso a necessidade da habilitação de uma fila padrão, isto é, com prioridade normal.

As classificações de um fila PQ pode ser por protocolo (IP, IPX, DecNet, SNA, etc), por interface de entrada ou por *access list*.

2.2- Desvantagens do IntServ

1. A quantidade de informação de estado aumenta com o número de fluxos exigindo enorme espaço de armazenamento e gerando sobrecarga de processamento nos roteadores.

2. Todos os roteadores devem implementar RSVP, controle de admissão, classificação e escalonamento

de pacotes em todos os elementos da rede: transmissores, receptores e roteadores.

3- Differentiated Services - DiffServ

De acordo com os problemas encontrados com a implantação do IntServ o *IETF-Internet Engineering Task Force* introduziu o DiffServ. Um modelo onde os pacotes são previamente marcados de acordo com os tipos de serviços desejados.

3.1- Differentiated Service Field – DS Field

No cabeçalho do pacote IP existe um campo de oito bits, anteriormente chamado de ToS, e que mudou recentemente para *DS Field*, em virtude da ampliação dos serviços e o tratamento que pode ser dado a ele. É no *DS Field* que são codificadas as classes para diferenciação de serviços. Na verdade, o ToS, que foi inicialmente definido e reservado para indicar tipos de serviços nunca foi utilizado de fato para nenhuma implementação. Com a introdução da Qualidade de Serviço foi necessário um tratamento mais específico, fazendo assim com que os pacotes sejam

classificados de forma que possamos, então, obter funcionalidades para o pacote IP. Podemos então dividir o byte do ToS do pacote IP em: *IP Precedence*, *ToS field* e *MBZ(must be zero)*.

O IP Precedence são três bits que podem ser classificados de 0 a 7 de acordo com a prioridade do fluxo de pacotes. Isto é: se um pacote tem prioridade 7 (serviço de missão crítica), e outro com prioridade 5, com certeza o de maior prioridade será atendido. Outro caso é quando temos um pacote marcado com valor zero: este receberá prioridade mais baixa, podendo nunca ser atendido.

Temos também, mais quatro bits reservados para o ToS field que vão tratar exatamente de: *delay*, *throughput* e *reability*. Sendo que os bits 6 e 7 quase não são usados, servindo também para aplicações de controle e gerência da rede.

Como foi citado acima o bit 7 não é usado, por isso o nome de *MBZ(must be zero)*. Cada campo DS corresponde a um tratamento diferente de encaminhamento (*PHB - Per Hop Behavior*), que será tratado posteriormente. Este caminho é marcado através do *DSCP - Differentiated Selector Codepoints*. Os roteadores

ordenam os pacotes de entrada em diferentes classes de encaminhamento, de acordo com os valores correspondentes de DS Field. O DSCP preserva o IP Precedence e os PHBs, porém não valor do ToS. Existem dois tipos de classificação dentro do DiffServ:

- *BA(behavior aggregate)* que é baseado somente no DS code-point.
- *MF(Mult-field)*, baseado em vários parâmetros do pacote: endereço do destino, endereço da fonte, número da porta e a própria classificação do DS field.

O DiffServ parte do princípio de que domínios adjacentes tenham acordo sobre os serviços disponibilizados. Os clientes podem marcar o campo DS de pacotes individuais para indicar o serviço desejado, ou podem ser marcadas pelo roteador folha (ou de borda); e assim mandado para o receptor.

No entanto dessa forma não sabemos quanto de banda disponível nós temos para utilizar, já que não foi feita nenhuma alocação. Podendo até um pacote com DS Field marcado chegando a um roteador que não provê qualidade de serviço ser remarcado, de forma que passe a ser um pacote de “best effort” podendo ser descartado. Para isso foi

inserido um componente para gerenciar os recursos do domínio, que tem como função básica controlar a largura de banda, as políticas e as prioridades dentro e entre as organizações. Este componente é o Controlador de banda (*BB - Bandwidth Broker*).

3.2- Service Level Agreement – SLA

Antes de falarmos sobre *Bandwidth Broker* é importante ressaltarmos **SLA**, que o gerencia. Um SLA determina que classes de serviços são suportadas e a quantidade de tráfego na banda entre os domínios. É um acordo feito entre o transmissor e o receptor determinando os limites dos parâmetros utilizados na aplicação.

O SLA pode ser estático ou dinâmico. É chamado de estático quando negociado de forma regular, por um determinado tempo. É chamado de dinâmico quando é necessário o uso de um protocolo de sinalização e controle para o gerenciamento da banda, por exemplo, RSVP.

As regras de classificação, policiamento, condicionamento e escalonamento usadas nos roteadores são determinadas também pela SLA, tanto quanto o espaço nos *buffers* de cada roteador.

3.3 - Bandwidth Broker – BB

Quando há solicitação de um fluxo, o BB é um componente que verifica a disponibilidade de recursos e a autorização do cliente para a conexão dentro do domínio QoS. Se encarrega também de fazer as alocações necessárias para a comunicação dentro do seu domínio e solicita ao BB adjacente, caso o pedido seja para fora do domínio. Esse processo de solicitação de alocação de recursos é contínuo entre os BBs adjacentes até que se chegue ao domínio do receptor.

Pode-se usar o RSVP para alocação de recursos entre BBs. Cada controlador de banda possui uma tabela de políticas estabelecidas através da SLA (*Service Level Agreement*) que é consultada a cada solicitação de QoS para o BB, por parte dos BBs vizinhos ou de outros domínios.

O BB deve operar em intra-domínio (mesmo ambiente em que é oferecida a qualidade de serviço) e inter-domínio (ambientes diferentes). Sendo limitado pelas políticas, que dizem quais usuários podem usar e quanto dos recursos do seu domínio.

3.4 – Desvantagens do DiffServ

1. Os serviços diferenciados dão segurança ao desempenho das aplicações somente em termos relativos, isto é, é mais seguro aquele que foi determinado que tem maior prioridade.
2. Esquemas de prioridade relativa garantem que uma aplicação gerando tráfego de determinada prioridade terá melhor desempenho que outra gerando tráfego de menor prioridade. Entretanto, dependendo da carga da rede, ambas as aplicações podem ter um desempenho muito aquém do que suas reais necessidades.

4 - Arquitetura Híbrida

As diferentes características dos serviços integrados e dos serviços diferenciados fazem com que nenhuma destas arquiteturas possa ser vista como uma solução completa para a implementação de qualidade de serviço na Internet.

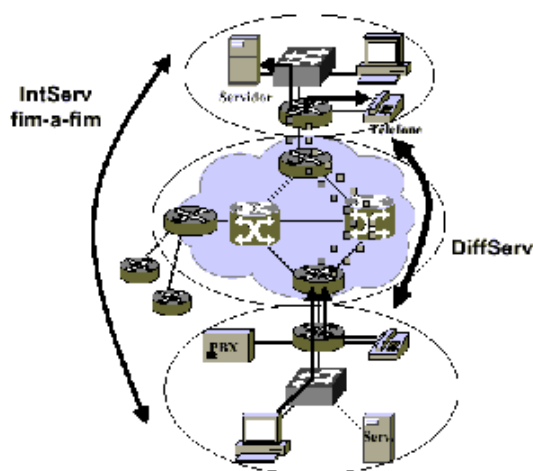
Enquanto os serviços integrados apresentam problemas de escalabilidade, dificultando sua implementação em redes de grande porte, os serviços

diferenciados apresentam mecanismos simples e voltados para a agregação de microfluxos, o que dificulta o gerenciamento e a alocação de recursos. Uma arquitetura que permita a interoperação entre os serviços integrados e os serviços diferenciados possibilitaria a adoção das arquiteturas nos locais onde seu uso é mais vantajoso e a obtenção das melhores características de cada uma para a implementação de QoS.

4.1 - Descrição da Arquitetura Híbrida

A arquitetura para a interoperação entre os serviços diferenciados e os serviços integrados especificada pelo IETF envolve a adoção do modelo de reserva de recursos com RSVP utilizado pelos serviços integrados através de uma rede contendo um ou mais domínios de serviços diferenciados. Os domínios DiffServ podem mas não são obrigados a participar da sinalização e da reserva de recursos RSVP: as regiões com serviços diferenciados podem tanto suportar o RSVP e serem capazes de efetuar sinalização e controle de admissão por fluxos, quanto serem capazes apenas de encaminhar mensagens RSVP de forma transparente.

Nesta arquitetura, os domínios DiffServ atuam como componentes de uma rede fim-a-fim implementando QoS utilizando os padrões de sinalização, reserva de recursos e de classes de serviço do modelo de serviços integrados. Em outras palavras, para as aplicações que requisitam qualidade de serviço uma rede utilizando a arquitetura híbrida deve ser semelhante a redes implementando apenas os serviços integrados, mesmo que haja uma ou mais regiões que não suportem seus mecanismos. Com este fim, os diferentes componentes existentes em nós QoS capazes devem ser mapeados para mecanismos implementáveis em regiões com serviços diferenciados. A figura abaixo mostra de forma simplificada a utilização de um domínio de serviços diferenciados como um elemento de rede de serviços integrados.



Descrição da arquitetura híbrida

4.2 – Desvantagens

As principais dificuldades relacionadas com a implementação da arquitetura híbrida estão diretamente relacionadas com a relativa juventude das arquiteturas de qualidade de serviço em IP, e principalmente dos serviços diferenciados. A falta de mecanismos de obtenção dinâmica de informações sobre o uso dos recursos no interior de domínios de diferenciação de serviços e de reserva de recursos tornam difícil o suporte aos requisitos quantitativos dos serviços de controle QoS.

Um segundo problema envolve a interoperação entre redes com diferentes administrações como as que constituem a Internet: para a garantia de suporte da reserva de recursos é necessário que a reserva seja adequadamente provisionada em todo caminho entre cada transmissor e o receptor.