

CENTRO PAULA SOUZA

**GOVERNO DO ESTADO DE
SÃO PAULO**

**Faculdade de Tecnologia de Americana
Curso de Análise de Sistemas e Tecnologia da Informação**

ARQUITETURAS PARA QUALIDADE DE SERVIÇO

GUSTAVO NÓBREGA CUSTÓDIO

**Americana, SP
2010**

Faculdade de Tecnologia de Americana
Curso de Análise de Sistemas e Tecnologia da Informação

ARQUITETURAS PARA QUALIDADE DE SERVIÇO

GUSTAVO NÓBREGA CUSTÓDIO

custodiognc@gmail.com

Projeto Articulador de Segurança da Informação, desenvolvido em cumprimento à exigência curricular do Curso de Análise de Sistemas e Tecnologia da Informação da Faculdade de Tecnologia de Americana, sob orientação do Prof. Benedito Aparecido Cruz

Área: Qualidade de Serviço

DEDICATÓRIA

Dedico a todos presentes nesse importante
momento na minha vida.

AGRADECIMENTOS

Agradeço a todos que me ajudaram na organização desse documento. Ao meu professor orientador pelo suporte e aos meus colegas de classe e companheiros de trabalho: Alexandre, Fred, Caio, Renato, Rafael, Diogo, Guilherme, Mariana e Robson.

CUSTÓDIO, G. N. **Arquiteturas para Qualidade de Serviço**. 2010. Americana. 38p. Monografia, Faculdade de Tecnologia de Americana

Resumo

A Internet foi concebida para transporte de tráfego não tempo real. Ela é uma rede do tipo melhor esforço, onde não há garantia de Qualidade de Serviço. A popularização da Internet resultou na demanda por novos serviços, muitos envolvendo tráfego multimídia. Esta demanda leva à necessidade de se buscar mecanismos para prover QoS em redes IP (e na Internet). Neste documento serão apresentadas as arquiteturas disponíveis para Qualidade de Serviço em Redes IP que são Serviços Integrados (*IntServ*), Serviços Diferenciados (*DiffServ*) e *Multiprotocol Label Switching (MPLS)*.

Palavras-Chave: Qualidade de Serviço, Redes IP, *MPLS*.

CUSTODIO, G. N. **Quality Service Arquitectures**. 2010. Americana. 38p.
Monograph, Technology's university of Americana

Abstract

The Internet was conceived for not real time data transport. It's a network of less effort type, where there's no guarantee of Quality of Service. The fact that the Internet became so popular came with a demand for new services, lots of them regarding multimedia traffic. This demand takes us to the need of looking for new mechanisms to provide QoS in IP networks (and Internet). This document will present the available architectures for QoS in IP networks which are Integrated Services (*IntServ*), Differentiated Services (*DiffServ*) and *Multiprotocol Label Switching (MPLS)*.

Keywords: Quality of Service, IP Networks, *MPLS*.

LISTA DE ILUSTRAÇÕES

Figura 1 - Modelo de referência para o <i>IntServ</i>	10
Figura 2 - Formato das mensagens <i>RSVP</i>	22
Figura 3 - Exemplo de estilos de reserva.	24
Figura 4 - Tipos de nós no <i>DiffServ</i>	25
Figura 5 - Arquitetura funcional da <i>DiffServ</i>	29

SUMÁRIO

1 INTRODUÇÃO.....	9
2 ARQUITETURA DE SERVIÇOS INTEGRADOS (<i>INTSERV</i>)	10
2.1 Modelo de referência	10
2.1.1 Agente de roteamento	11
2.1.2 Estabelecimento de reserva	11
2.1.3 Controle de admissão.....	12
2.1.4 Identificação do fluxo	13
2.1.5 Despacho do pacote	14
2.2 Modelos de serviços	14
2.2.1 Especificação de fluxo	15
2.2.2 Serviço garantido.....	15
2.2.2.1 Descritor de tráfego	16
2.2.2.2 Especificação do serviço	16
2.2.3 Serviço de carga controlada.....	17
2.3 <i>Resource Reservation Setup (RSVP)</i>	18
2.3.1 Características Básicas	18
2.3.2 Mensagens do <i>RSVP</i>	19
2.3.2.1 Mensagens <i>PATH</i>	20
2.3.2.2 Mensagens <i>RESV</i>	21
2.3.2.3 Formato das Mensagens	22
2.3.3 Estilos de reserva	23
2.3.3 Objeto <i>Adspec</i> e <i>OPWA</i>	24
3 ARQUITETURA DE SERVIÇOS DIFERENCIADOS (<i>DIFFSERV</i>).....	25
3.1 Tipos de Nós.....	25
3.2 Princípios Básicos	26
3.3 Tratamento de encaminhamento e Serviço	27
3.3.1 <i>Per-Hop Behaviors (PHBs)</i>	28
3.3.2 Serviços.....	28
3.3.2.1 <i>Service Level Agreement - SLA</i>	28
3.3.3 <i>Codepoint</i>	29
3.4 Arquitetura Funcional.....	30
3.4.1 Classificação.....	30
3.4.2 Condicionador.....	31
4 <i>MULTIPROTOCOL LABEL SWITCHING (MPLS)</i>	33

4.1 Princípios	33
4.2 Formas de implementação	33
4.3 Componentes	34
4.3.1 <i>Forward Equivalence Class - FEC</i>	34
4.4 Definição da rota.....	34
4.5 Estabelecimento do caminho.....	35
4.6 Alocação de <i>label</i>	35
4.7 Falha em um enlace da rede	35
5 CONSIDERAÇÕES	36
REFERÊNCIAS BIBLIOGRÁFICAS	37

1 INTRODUÇÃO

A Qualidade de Serviço (QoS) é definida como a habilidade da rede para garantir e manter certos níveis de desempenho para cada aplicação, de acordo com as necessidades específicas de cada usuário, conforme PRYCKER (1995). A QoS está relacionada com a satisfação dos usuários da rede.

Os benefícios que a QoS pode trazer dependem do tipo de serviço em questão. Por exemplo, para uma chamada telefônica podemos listar alguns parâmetros em que a QoS pode garantir ou melhorar, por consequência, melhorar a satisfação dos usuários: tempo para obter o tom de linha, tempo para completar a chamada, probabilidade de completar a chamada, dado que o usuário de destino está disponível, ausência de eco durante a comunicação, ausência de atraso significativo e ausência de ruído, interferências e distorções.

Serão apresentadas aqui as arquiteturas disponíveis para Qualidade de Serviço em Redes IP que são Serviços Integrados (*IntServ*), Serviços Diferenciados (*DiffServ*) e *Multiprotocol Label Switching (MPLS)*.

2 ARQUITETURA DE SERVIÇOS INTEGRADOS (*INTSERV*)

A Arquitetura de Serviços Integrados (*IntServ*) se baseia na reserva de recursos para cada fluxo da rede. A fonte inicia um procedimento de reserva de recurso descrevendo para a rede características de seu fluxo e os requisitos de recursos. A rede aceita o fluxo da aplicação apenas se houver recursos suficientes para atender à reserva. Uma vez feita a reserva, a fonte pode enviar seus pacotes pelo caminho reservado.

2.1 Modelo de referência

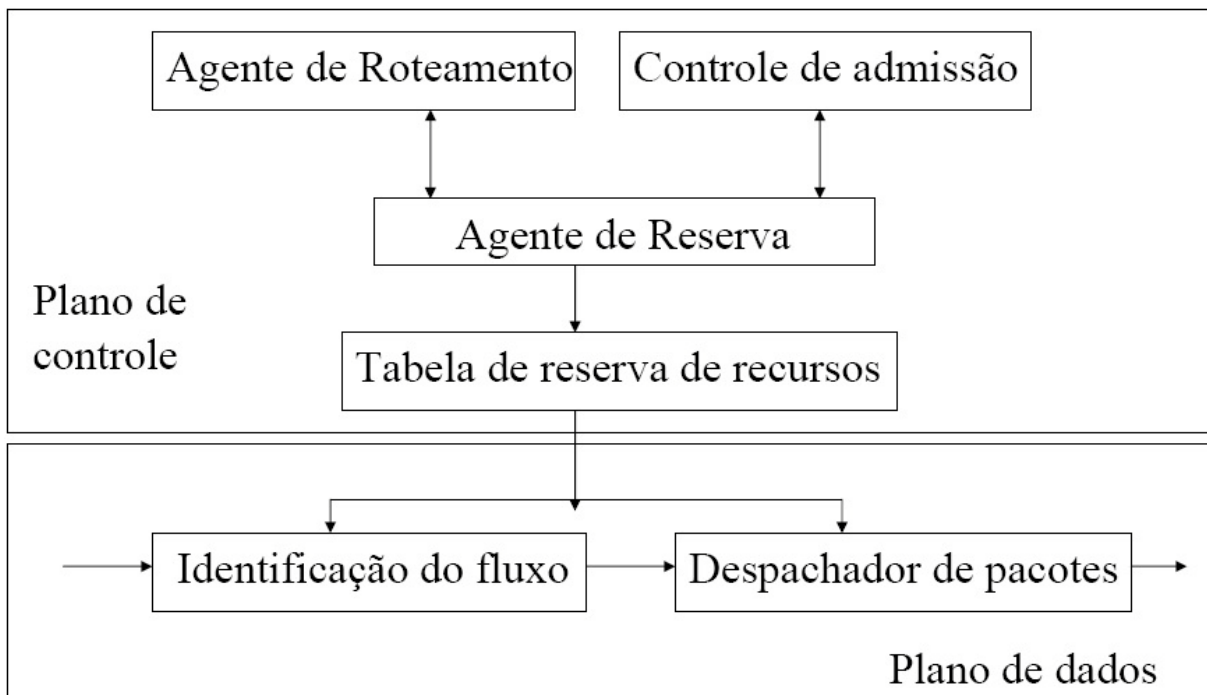


Figura 1 – Modelo de referência para o *IntServ*.

(BRITO, 1996)

A Figura mostra os componentes do modelo de referência para a *IntServ*. O modelo pode ser dividido logicamente em duas partes: o plano de controle e o plano de dados. O plano de controle estabelece a reserva dos recursos; e o plano de dados encaminha os pacotes com base no estado da reserva.

Para fazer uma reserva, a aplicação primeiro caracteriza o seu fluxo e especifica os requisitos de QoS, um processo denominado Especificação de Fluxo. A requisição de reserva é então enviada à rede. Quando o roteador recebe a requisição, ele executa duas

funções: primeiro, ele interage com o módulo de roteamento para determinar o próximo roteador (*next hop*) para o qual a requisição deve ser encaminhada; segundo, ele utiliza o controle de admissão para decidir se já recursos suficientes para atender os requisitos. Se a reserva pode ser estabelecida com sucesso, a informação sobre o fluxo reservado é armazenada na tabela de reservas. Esta informação é utilizada para configurar o módulo de identificação de fluxo e o módulo de despacho de pacote no plano de dados. Quando um pacote chega, o módulo de identificação do fluxo define a que reserva o pacote pertence e o encaminha para a fila apropriada; o despachador de pacotes aloca os recursos para o fluxo com base nas informações de reserva.

2.1.1 Agente de roteamento

A arquitetura *IntServ* desacopla a função de roteamento do processo de reserva. A arquitetura assume que o módulo de roteamento no roteador irá informar qual é o próximo roteador (*next hop*). Isto pode ser feito utilizando-se o roteamento IP tradicional.

Idealmente, a rota deveria ser escolhida levando-se em conta os recursos solicitados. O roteamento IP normal não leva em conta a quantidade de recurso disponível em cada enlace ao longo do caminho, mas sim outros parâmetros como número de saltos ou atraso estimado.

2.1.2 Estabelecimento de reserva

A reserva é feita através de um protocolo de estabelecimento de reserva, que percorre o caminho entre origem e destino (*hop by hop*) para instalar os estados de reserva nos roteadores. O protocolo transporta as características do tráfego e requisitos de recursos, de modo que cada nó ao longo do caminho possa avaliar se a requisição de reserva pode ser atendida ou não. O protocolo de estabelecimento de reserva deve acompanhar as mudanças topológicas na rede. Por exemplo, se um enlace cai, o protocolo de reserva deve cancelar a reserva anterior e estabelecer uma nova reserva.

O uso de reserva de recursos pode envolver questões financeiras. Por exemplo, um usuário que contrata um serviço com reserva de recurso é tarifado de forma distinta de um usuário que contrata um serviço do tipo melhor esforço. Assim, associado à reserva de recursos existe um conjunto de questões relacionadas a autorização, autenticação e cobrança.

O protocolo de reserva de recursos para a Internet, utilizado na *IntServ*, é o *RSVP*. Este protocolo é baseado na abordagem de inicialização pelo receptor e permite operação com *multicast*. Ele permite diferentes estilos de reserva e utiliza o conceito de *soft state* para lidar com mudanças nas rotas. O protocolo *RSVP* será tratado com mais detalhe à frente.

2.1.3 Controle de admissão

Para garantir a reserva de recursos na rede, um mecanismo de controle de admissão é necessário. Este mecanismo monitora e mede os recursos disponíveis na rede e decide se uma reserva solicitada pode ser aceita ou não, com base nas políticas de controle de admissão definidas.

Há duas abordagens básicas para o controle de admissão: baseada em parâmetros e baseada em medida.

Na abordagem baseada em parâmetros um conjunto de parâmetros é utilizado para caracterizar os fluxos; o agente de controle de admissão calcula os recursos necessários com base nestes parâmetros e verifica se há recursos disponíveis ou não. O problema com esta abordagem é que nem sempre o modelo reflete precisamente o comportamento da fonte, e situações onde a quantidade de recursos reservados é maior do que a efetivamente utilizada pode ocorrer.

Na abordagem baseada em medida a rede mede a carga de tráfego real e utiliza esta informação para o controle de admissão. Como as fontes de tráfego não são estáticas, o esquema baseado em medida é naturalmente probabilístico e não pode ser usado para dar garantias estritas de recursos. No entanto, para fontes em aplicações que toleram

pequenas flutuações no atraso, esta abordagem oferece uma boa relação entre o nível de recurso reservado e o utilizado. Quando existe um grande número de fluxos na rede, a multiplexagem estatística melhora significativamente o grau de utilização dos recursos.

Vários métodos têm sido propostos para decidir se um novo fluxo pode ser admitido pela rede. Os algoritmos mais comuns são:

- Soma Simples: a abordagem mais simples é garantir que a soma da banda requisitada por todos os fluxos correntes e pelo novo fluxo não excede a capacidade do enlace. Esta é a abordagem mais conservadora.
- Soma Medida: esta abordagem utiliza uma medida da carga associada aos fluxos existentes, ao invés de se basear na banda requisitada por eles. Esta abordagem leva em conta o fato de que a carga real é usualmente menor do que a soma das bandas requisitadas.
- Região de Aceitação: nesta técnica busca-se o aumento na utilização dos recursos da rede, admitindo-se a perda de pacotes. Dado um modelo estatístico das fontes de tráfego, uma região de aceitação para um tipo de tráfego pode ser definida.
- Banda Equivalente: esta é uma abordagem baseada em modelagem estatística, onde a cada fluxo associa-se, com base nas características do mesmo, uma banda equivalente. O somatório das bandas equivalentes não deve exceder a capacidade do enlace.

2.1.4 Identificação do fluxo

Os roteadores da rede devem examinar os pacotes entrantes e decidir se o pacote pertence a algum fluxo com reserva de recurso (fluxo *RSVP*). Um fluxo IP é identificado pelos cinco campos do cabeçalho IP: endereço de fonte, endereço de destino, protocolo ID, porta de fonte e porta de destino. Para determinar se um pacote pertence a um fluxo *RSVP*,

o mecanismo de identificação de fluxo deve comparar este conjunto de parâmetros do pacote entrante com os conjuntos de parâmetros de todos os fluxos na tabela de reservas.

Uma vez identificado o fluxo *RSVP* ao qual o pacote pertence, o pacote é encaminhado para o despachador de pacotes com o estado da reserva associado ao fluxo.

A identificação do fluxo deve ser feita a cada pacote. Se o número de fluxos é muito grande o processo de identificação de fluxo pode tornar-se lento. Esta característica limita o uso desta abordagem em backbones de alta velocidade, onde o número de fluxos é elevado e o tempo disponível para processar um pacote é pequeno.

2.1.5 Despacho do pacote

O despacho do pacote é um dos passos mais importantes no processo de reserva de recurso. O algoritmo de despacho seleciona o pacote a ser transmitido a cada ciclo de transmissão. A forma de operação do algoritmo influencia diretamente no atraso do pacote e, conseqüentemente, no nível de QoS oferecido à aplicação.

O algoritmo *WFQ* (*Weighted Fair Queuing*) é o algoritmo base para a arquitetura *IntServ*. De fato, o *WFQ* representa uma classe de algoritmos de despacho que compartilham uma abordagem comum mas diferem nos detalhes da implementação.

2.2 Modelos de serviços

Os modelos de serviço descrevem a interface entre a rede e seus usuários na arquitetura de alocação de recursos; isto é, que tipo de serviço o usuário pode solicitar da rede e que tipo de garantia a rede pode oferecer ao usuário. A arquitetura *IntServ* padronizou dois tipos de serviço: Serviço Garantido e Serviço de Carga Controlada.

2.2.1 Especificação de fluxo

Para fazer uma reserva, a aplicação deve caracterizar o tráfego que ela irá injetar na rede e especificar os requisitos do serviço para aquele fluxo. Estas ações são denominadas de especificação do fluxo. A especificação do fluxo, em essência, é um contrato de serviço que define o tráfego que a fonte irá enviar e os serviços que a rede promete oferecer.

Os parâmetros mais comuns para caracterização do tráfego são:

- Taxa de pico: mais alta taxa de tráfego que a fonte pode gerar.
- Taxa média: taxa média de transmissão dentro de um intervalo de tempo.
- Tamanho de *burst*: máxima quantidade de dados que pode ser injetada na rede à taxa de pico.

A *IntServ* utiliza o algoritmo *Token Bucket* para descrever o tráfego, que é uma definição formal de uma taxa de transferência.

Os requisitos de QoS são específicos para cada aplicação. Os seguintes parâmetros são comumente utilizados para caracterizar a QoS em redes de pacotes

- Banda mínima: mínima quantidade de banda requerida por um fluxo.
- Atraso: pode ser especificado em termos de atraso médio ou atraso máximo.
- Jitter de atraso: variação do atraso de pacote; diferença entre o máximo e o mínimo atraso de pacote.
- Taxa de perda de pacotes: razão entre o número de pacotes perdidos e o número de pacotes transmitidos.

2.2.2 Serviço garantido

O Serviço Garantido provê, para o tráfego conforme, banda garantida e limites estritos para o atraso de enfileiramento fim-a-fim. O atraso controlado é o atraso de enfileiramento máximo (não há controle quanto aos atrasos mínimo e médio). Este tipo de serviço é adequado para aplicações em tempo real sensíveis a atraso.

Uma aplicação solicita o Serviço Garantido especificando um descritor de tráfego (*TSpec*) e uma especificação de serviço (*RSpec*).

2.2.2.1 Descritor de tráfego

O descritor de tráfego utiliza os seguintes parâmetros para especificar o tráfego:

- Taxa de fichas (r) (*bytes/segundo*): taxa com a qual as fichas chegam no *bucket*.
- Taxa de pico (p) (*bytes/segundo*): taxa máxima em que os pacotes são transmitidos.
- Profundidade do *bucket* (b) (*bytes*): capacidade do *bucket*.
- Unidade mínima policiada (m) (*bytes*): qualquer pacote com tamanho menor do que m será contado como m *bytes*.
- Tamanho máximo de pacote (M) (*bytes*): tamanho máximo de pacote que será aceito.

2.2.2.2 Especificação do serviço

A especificação do serviço é feita com base nos seguintes parâmetros:

- Taxa de serviço (R) (*bytes/segundo*): requisito de banda para o serviço.
- *SLAck* Term (S) (*microsegundos*): atraso extra que um nó pode adicionar sem violar os requisitos de atraso fim-a-fim (veja protocolo *RSVP* na seção 2.3).

O termo de relaxamento é a diferença entre o atraso desejado e o atraso real (obtido com a reserva de banda corrente). Em outras palavras, o termo de relaxamento é o crédito em atraso (a quantidade de atraso que pode ser incrementada sem violar o atraso fim-a-fim) nos *hops* anteriores que pode ser utilizado pelos outros *hops*.

Para entender a idéia do termo de relaxamento seja o seguinte exemplo: um receptor deseja reservar 2.5 Mbps em cada enlace do caminho, que é composto de três enlaces. O primeiro enlace tem banda disponível de 10 Mbps, o segundo enlace tem banda disponível

de 2 Mbps e o terceiro enlace tem banda disponível de 6 Mbps. Como o segundo enlace não tem 2.5 Mbps de banda disponível o pedido de reserva iria falhar. No entanto, como há sobra de recursos nos outros dois enlaces, a rede poderia reservar mais banda do que o especificado nos enlaces 1 e 3 e reservar menos banda do que o especificado no enlace 2, mantendo-se o atraso fim-a-fim compatível com a reserva de 2.5 Mbps em todos os enlaces. O termo de relaxamento é utilizado para transportar a folga de atraso, decorrente da possível alocação de banda além do especificado, nos *hops* anteriores.

2.2.3 Serviço de carga controlada

O serviço de carga controlada não oferece garantias quantitativas de banda ou atraso. Ao invés disto, o serviço tenta emular uma rede levemente carregada para as aplicações que requisitaram o serviço. Este tipo de serviço permite multiplexagem estatística e pode ser implementado de uma forma mais eficiente que o serviço garantido. Suas características são adequadas a aplicações adaptativas que requeiram algum grau de garantia de desempenho, mas que não exijam limites absolutos.

O Serviço de Carga Controlada tenta criar um modelo de serviço entre o melhor esforço e o serviço garantido. A rede oferece à aplicação um comportamento similar ao de uma rede melhor esforço levemente carregada, ou seja:

- A grande maioria dos pacotes transmitidos serão entregues pela rede com sucesso para os receptores.
- A maior parte dos pacotes sofre atraso com valor próximo ao mínimo possível.

O serviço de carga controlada não aceita o estabelecimento de valores alvos para os parâmetros de atraso e perda. Uma requisição de serviço controlado deve ser aceita apenas se a rede possui recursos suficientes para acomodar o tráfego sem causar congestionamento.

Na solicitação do serviço de carga controlada a fonte deve especificar o comportamento do tráfego, através de parâmetros do *Token Bucket*. A rede deve aceitar a requisição de serviço apenas se a quantidade de recursos disponíveis for suficiente para atender ao tráfego especificado, sem resultar em congestionamento.

O algoritmo de controle de admissão para decidir se o fluxo pode ser aceito não é especificado, ficando a definição a cargo de cada implementador.

2.3 Resource Reservation Setup (RSVP)

Em uma rede do tipo melhor esforço uma aplicação pode enviar pacotes sempre que desejar. Na arquitetura *IntServ* a aplicação deve reservar recursos da rede antes de iniciar a transmissão de seus pacotes. Para tal, é necessária a utilização de um protocolo de reserva de recursos. O *RSVP (Resource Reservation Setup)* é um protocolo desenvolvido pelo IETF para estabelecer reservas de recursos em redes IP. O *RSVP* é utilizado pelos hosts para comunicar à rede os requisitos de serviço, e é utilizado pelos roteadores da rede para estabelecer os estados de reserva ao longo do caminho.

2.3.1 Características Básicas

O *RSVP* possui as seguintes características básicas:

- A reserva é unidirecional (simplex): em uma comunicação bidirecional, as duas extremidades devem estabelecer reservas, cada uma em uma direção. As reservas de cada direção são independentes.
- Definição da reserva é feita pelo receptor: o *RSVP* suporta comunicação *multicast*. Para oferecer mais flexibilidade e eficiência no suporte a grandes grupos *multicast* o *RSVP* é orientado ao receptor; ou seja, os receptores são responsáveis por decidir quais os recursos serão solicitados e iniciam o processo de reserva.

- Independente do roteamento: o *RSVP* foi definido para operar em conjunto com o roteamento unicast e *multicast* das redes IP. A decisão da rota a ser seguida pelas mensagens *RSVP* é tomada pelo roteamento IP. Extensões do *RSVP* utilizadas para engenharia de tráfego permitem a utilização de roteamento explícito, onde o transmissor define o caminho das mensagens *RSVP*. De qualquer forma, o *RSVP* não inclui funções de roteamento.
- *Soft state*: A cada estado de reserva está associado um timer; se o timer expira o estado é apagado. Assim, para manter uma reserva, o *RSVP* deve revalidá-la periodicamente. Esta abordagem permite ao *RSVP* se adaptar a mudanças na topologia da rede e a mudanças nos membros de um grupo *multicast*.
- Estilos de reserva: o *RSVP* possui vários estilos de reserva, que caracterizam como a reserva deve ser tratada.

2.3.2 Mensagens do *RSVP*

O *RSVP* possui sete mensagens:

- Mensagem *PATH*: detalhada a seguir
- Mensagem *RESV*: detalhada a seguir
- *PATHErr*: indicação de erro em resposta a uma mensagem *PATH*
- *RESVErr*: indicação de erro em resposta a uma mensagem *RESV*
- *PATHTear*: remoção de caminho (remove todos os estados associados à mensagem *PATH* correspondente)
- *RESVTear*: remoção de reserva (remove todos os estados associados à reserva correspondente)
- *RESVConf*: confirmação de reserva.

O *RSVP* possui dois tipos de mensagens: mensagens *PATH* e mensagens *RESV*. As mensagens *PATH* são enviadas do transmissor para o receptor e estabelecem o caminho com reserva de recurso. As mensagens *RESV* são enviadas do receptor para o transmissor. Estas mensagens definem os recursos requisitados e efetivam, se possível, o processo de reserva.

2.3.2.1 Mensagens *PATH*

As mensagens *PATH* são enviadas da fonte para o destino, seguindo o caminho definido pelo roteamento IP, e são utilizadas para:

- Informar os receptores sobre a fonte de tráfego.
- Passam adiante as características do caminho.
- Instalar os estados (*PATH state*) necessários para que as mensagens *RESV* possam ir do receptor ao transmissor efetuando as reservas. Estes estados contém o endereço IP do roteador prévio (*upstream*).

Após receber a mensagem *PATH* o receptor envia de volta, pelo mesmo caminho, uma mensagem *RESV*. As mensagens *RESV* especificam os recursos solicitados e efetivam, se possível, as reservas nos roteadores ao longo do caminho. Após receber as mensagens *RESV*, o transmissor pode iniciar o envio dos pacotes pelo caminho reservado.

A mensagem *PATH* tem o seguinte conteúdo:

- *Previous Hop (PHOP)*: contém o endereço IP do roteador que enviou a mensagem *PATH*.
- *Sender Template*: contém informações que identificam inequivocamente o fluxo associado à reserva que está sendo iniciada. Especifica o endereço IP do transmissor e, opcionalmente, a porta UDP/TCP transmissora, e assume o protocolo ID especificado para a sessão.

- *Sender TSpec*: caracteriza o tráfego que o transmissor irá gerar. Esta informação é utilizada para decidir a quantidade de recursos que será reservada.
- *Adspec*: elemento opcional utilizado para transportar um *OPWA* (One Pass with Advertising). O conteúdo do *Adspec* é passado para o controle de tráfego local em cada nó, que o atualiza. Portanto, o conteúdo do *Adspec* pode mudar a cada nó. O *Adspec* é utilizado para transportar informações sobre o nível de recursos disponíveis ao longo do caminho.

2.3.2.2 Mensagens *RESV*

As mensagens *RESV* são requisições de reserva enviadas pelo receptor para o transmissor, pelo mesmo caminho trilhado pelas mensagens *PATH*. Uma mensagem *RESV* contém informação sobre o estilo da reserva, um objeto *Flow Spec* (especificação do fluxo), e um objeto *Filter Spec* (especificação do filtro); este par de objetos (*Flow* e *Filter Spec*) é denominado de descritor do fluxo.

O *Flow Spec* especifica a *QoS* desejada. O *Filter Spec*, junto com a especificação da sessão, define o fluxo que irá receber a *QoS* definida pelo *Flow Spec*.

O *Flow Spec* pode incluir a classe de serviço e dois conjuntos de parâmetros: a especificação da reserva (*RSpec*), que define a *QoS* desejada e a especificação de tráfego (*TSpec*), que define o comportamento do fluxo. O parâmetro *RSpec* não está presente se o serviço requisitado é do tipo carga controlada.

2.3.2.3 Formato das Mensagens

Versão	Flags	Tipo de mensagem	RSVP checksum
Send TTL		Reservado	Comprimento RSVP

Cabeçalho comum

Comprimento	Número da classe	C type
Conteúdo do objeto (comprimento variável)		

Cabeçalho do objeto

Figura 2 – Formato das mensagens *RSVP*.

(BRITO, 1996)

Toda mensagem *RSVP* inicia com um cabeçalho comum seguido por uma série de objetos *RSVP* de comprimento variável.

O cabeçalho comum possui os seguintes campos:

- Versão do protocolo
- *Flags*: não definido
- Tipo de mensagem: define a mensagem *RSVP*
- Checksum
- Send TTL: contém o valor do TTL utilizado pelo transmissor no cabeçalho IP
- Comprimento *RSVP*: comprimento total da mensagem, em *bytes*, incluindo-se os objetos *RSVP* que seguem o cabeçalho comum.

O objeto é composto por um campo de conteúdo, com comprimento variável, e um cabeçalho, que contém os seguintes campos:

- Comprimento: comprimento total do objeto, em *bytes*.

- Número da Classe: identifica a classe do objeto. Algumas classes de objetos, já vistas, são: *RSVP_HOP* (*PHOP* ou *NHOP*), *STYLE*, *FLOWSPEC*, *FILTER_SPEC*, *SENDER_TEMPLATE*, *SENDER_TSPEC*, *ADSPEC*
- C type: distingue subtipos com o mesmo Class-Num e é único dentro de uma classe.

2.3.3 Estilos de reserva

O *RSVP* possui três estilos de reserva:

- Wild-card Filter (WF): neste estilo a reserva é compartilhada e não há seleção explícita do transmissor. Todos os receptores compartilham uma única reserva, cujo tamanho é a maior das requisições de recursos feitas pelos receptores. Todos os transmissores a *upstream* podem utilizar a reserva.
- Fixed Filter (FF): este estilo é o oposto do estilo WF; ele implica em reservas distintas e seleção explícita do transmissor.
- Shared Explicit (SE): o estilo SE implica em reserva compartilhada com seleção explícita do transmissor.

Os estilos compartilhados (WF e SE) são adequados para uma aplicação *multicast* onde seja improvável que todas as fontes transmitam simultaneamente. Por exemplo, em uma áudio-conferência, tipicamente uma ou duas pessoas, no máximo, falam ao mesmo tempo. Assim, reservar recursos suficientes para a transmissão do sinal de duas fontes pode ser suficiente.

Na simbologia associada a cada reserva S_1, S_2, \dots, S_n representa as fontes especificadas, Q_1, Q_2, \dots, Q_n representa o recurso reservado para cada fonte; Q representa recurso reservado compartilhado por diversas fontes; e "*" representa qualquer fonte.

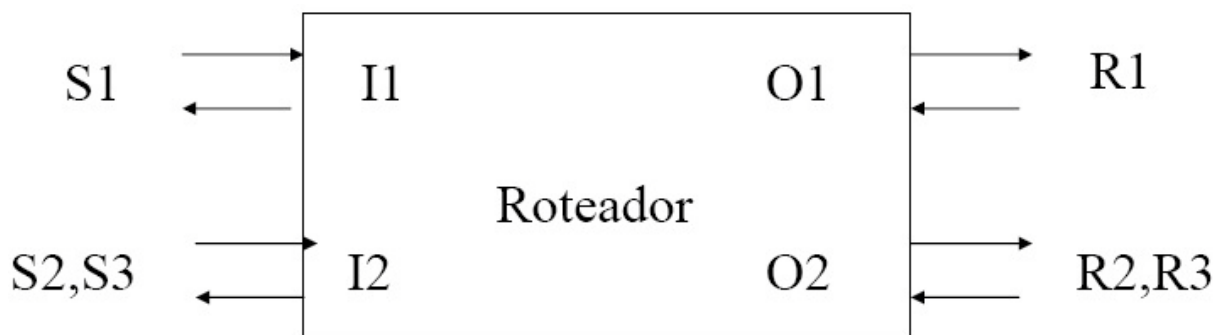


Figura 3 – Exemplo de estilos de reserva.

(BRITO, 1996)

A figura representa um roteador com duas interfaces de entrada e duas interfaces de saída. Associado à interface I1 temos o transmissor S1; associado à interface I2 temos os transmissores S2 e S3; o receptor R1 está associado à interface O1; e os receptores R2 e R3 estão associados à interface O2.

As requisições de reserva vêm dos receptores para os transmissores. A unidade básica de reserva de recurso é B.

2.3.3 Objeto *Adspec* e *OPWA*

O *OPWA* (One Pass with Advertising) é um modelo de reserva onde o transmissor inclui um objeto *Adspec* na mensagem *PATH*. Este objeto contém alguns parâmetros relacionados aos recursos disponíveis ao longo do caminho, tais como: estimativa de latência e atraso, banda disponível, número de *hops* que são capazes de suportar a *IntServ*, presença de nó que não é capaz de suportar a *IntServ*, unidade máxima de transmissão (MTU) do caminho, etc.

Cada nó ao longo do caminho modifica o *Adspec*, atualizando as informações. O receptor utiliza o conteúdo dos diversos parâmetros para gerar o pedido de reserva a ser encaminhado pela rede.

3 ARQUITETURA DE SERVIÇOS DIFERENCIADOS (*DIFFSERV*)

A arquitetura de Serviços Diferenciados (*DiffServ*) é mais simples que a arquitetura *IntServ*.

Na *DiffServ* o tráfego é dividido em um pequeno número de classes de serviço ou classes de encaminhamento. Os recursos são alocados para cada classe, e não mais para cada fluxo, como é na *IntServ*. A classe de encaminhamento à qual o pacote pertence está definida no cabeçalho do pacote. O tratamento dispensado pela rede para cada classe de encaminhamento é pré-definido. Não há necessidade de se prover estabelecimento de reserva de recursos a cada fluxo.

3.1 Tipos de Nós

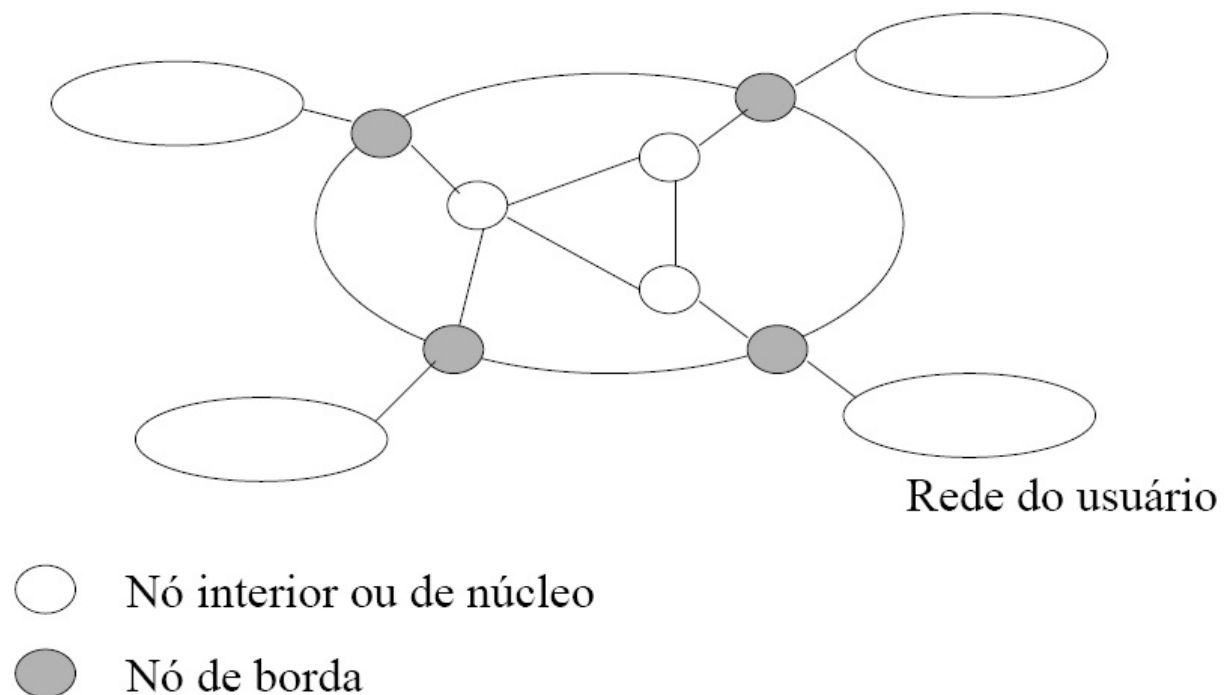


Figura 4 – Tipos de nós no *DiffServ*.

(BRITO, 1996)

A figura ilustra os tipos de nó na arquitetura *DiffServ*:

- Nós de borda: são responsáveis por classificar os pacotes, associando-os a uma das classes de encaminhamento definidas e de policiar o tráfego, para verificar a conformidade dos fluxos com os contratos de serviço.
- Nós de núcleo: os nós no interior da rede encaminham os pacotes baseado na classe de encaminhamento contida no cabeçalho do pacote.

3.2 Princípios Básicos

A arquitetura *DiffServ* tem os seguintes princípios básicos:

- Alocação de recurso para tráfego agregado: os recursos são alocados para classes individuais que representam tráfego agregado. A garantia de desempenho para fluxos individuais em uma classe de encaminhamento é provida por priorização e provisionamento, e não por reserva para cada fluxo.
- Policiamento de tráfego na borda e encaminhamento baseado na classe no núcleo: somente os nós na borda da rede classificam o tráfego e marcam os pacotes. No interior da rede, os nós utilizam a classe de encaminhamento codificada no cabeçalho do pacote para determinar o tratamento que o mesmo vai receber. Na arquitetura *IntServ* todos os nós fazem classificação de pacote, para identificar a que fluxo o pacote pertence, qual a reserva associada e como o pacote deve ser tratado.
- Define comportamento de encaminhamento e não serviço: a *DiffServ* define tratamento de encaminhamento, e não serviço fim-a-fim. Cada classe de encaminhamento representa um tratamento e não um serviço. Os serviços, contudo, podem ser estabelecidos combinando-se classes de encaminhamento e controle de admissão.
- Garantia por provisionamento e não por reserva: através da alocação de recursos para as classes de encaminhamento e do controle da quantidade de

tráfego para estas classes, a *DiffServ* cria diferentes níveis de serviço e garantia de recursos, mas não oferece garantia absoluta de banda e atraso para fluxos individuais.

- Ênfase no *SLA* (Service Level Agreement): o objetivo da *DiffServ* é garantir que os *SLAs* entre clientes e provedores de serviço sejam cumpridos. Na *IntServ* as aplicações solicitam reservas de recurso (utilizando o *RSVP*) sob demanda.
- Foco em um único domínio: a utilização da *DiffServ* na Internet pode ser incremental. As classes de encaminhamento podem ser definidas para um único domínio, e entre domínios diferentes os provedores de serviço podem estender ou mapear suas definições através de acordos bilaterais. O modelo da *IntServ* é inerentemente fim-a-fim. A reserva de recurso para um fluxo particular irá falhar se alguma parte do caminho não suportar a *IntServ*.

3.3 Tratamento de encaminhamento e Serviço

No contexto da *DiffServ*, o tratamento de encaminhamento se refere a um comportamento, observável externamente, de um algoritmo específico ou mecanismo que é implementado em um nó. Serviço é definido pelo desempenho global que o tráfego do cliente recebe.

Serviços e tratamento de encaminhamento estão diretamente relacionados. Por exemplo, um algoritmo de priorização de pacotes implementado nos nós da rede, aqui denominado de despacho expresso, é um tratamento de encaminhamento e não um serviço. Este tratamento de encaminhamento, contudo, pode ser utilizado para implementar determinados serviços. Por exemplo, um serviço denominado "sem perda", que garante que não haverá perda de pacotes dos usuários, pode ser implementado através do uso do tratamento de encaminhamento despacho expresso e de algum mecanismo de controle de admissão que limite a quantidade de tráfego no serviço "sem perda".

3.3.1 *Per-Hop Behaviors (PHBs)*

Na *DiffServ*, os tratamentos de encaminhamento em um nó são descritos em termos de *per-hop behavior (PHB)*. Cada *PHB* é representado por um campo de 6 bits no cabeçalho IP denominado *Differentiated Services Codepoint (DSCP)*. Todos os pacotes com o mesmo *Codepoint* recebem o mesmo tratamento de encaminhamento. Serviços fim-a-fim podem ser construídos combinando-se *PHBs* com condicionamento de tráfego e provisionamento de recursos.

Os *PHBs* podem descrever os tratamentos de encaminhamento em termos absolutos ou relativos. Por exemplo, os *PHBs* podem ser especificados em termos da banda mínima garantida para um comportamento agregado (*behavior aggregate*). Um exemplo de *PHB* com tratamento de encaminhamento relativo é um *PHB* que aloca a banda do enlace proporcionalmente. Um conjunto de *PHBs* podem formar um grupo de *PHB*.

3.3.2 Serviços

Um serviço descreve o tratamento global que o tráfego de um cliente irá receber dentro de um domínio DS ou fim-a-fim. Um serviço é aquilo que é visível para o usuário, enquanto os *PHBs* estão escondidos dentro dos elementos da rede. Criar um serviço requer a combinação de várias ações: mapear o tráfego para o *PHB* adequado e condicionar o tráfego na borda da rede, provisionar recursos na rede, encaminhar os pacotes com base no *PHB* no interior da rede.

Na arquitetura *DiffServ* os serviços são definidos na forma de um *Service Level Agreement (SLA)* entre o cliente e seu provedor de serviço.

3.3.2.1 *Service Level Agreement - SLA*

Um *SLA* especifica os detalhes do serviço que será oferecido ao cliente. Um elemento importante do *SLA* é o *TCA (Traffic Conditioning Agreement)*; o *TCA* detalha os

parâmetros do serviço, o perfil do tráfego e as ações de policiamento. O conteúdo básico do *TCA* inclui:

- Perfil do tráfego: parâmetros do *Token bucket* para cada classe de tráfego.
- Métricas de desempenho: vazão, atraso e prioridade de descarte.
- Policiamento: ações a serem tomadas para o tráfego não-conforme.
- Serviços adicionais: serviços de marcação e conformação providos pelo provedor de serviço.

Além do *TCA*, um *SLA* pode conter outras características do serviço e parâmetros negociais relacionados a : disponibilidade, segurança, auditoria, contabilidade, tarifação, cobrança, etc.

Os *SLAs* podem ser estáticos ou dinâmicos. A maioria dos *SLAs* são negociados entre o cliente e o provedor antes que o serviço seja iniciado, sofrendo mudanças apenas ocasionalmente. Os *SLAs* podem também ser dinamicamente ajustados ou negociados. Por exemplo, um cliente pode desejar banda adicional para um evento especial. *SLAs* dinâmicos requerem sistemas de gerenciamento que permitam automatizar o processo de negociação e mudança do conteúdo do *SLA*.

3.3.3 Codepoint

O *Codepoint* para definição do *PHB* é codificado no campo ToS do cabeçalho IP. Seis bits deste campo, DSCP, são redefinidos e indiciam o *PHB*; os outros dois bits, CU, não são utilizados.

3.4 Arquitetura funcional

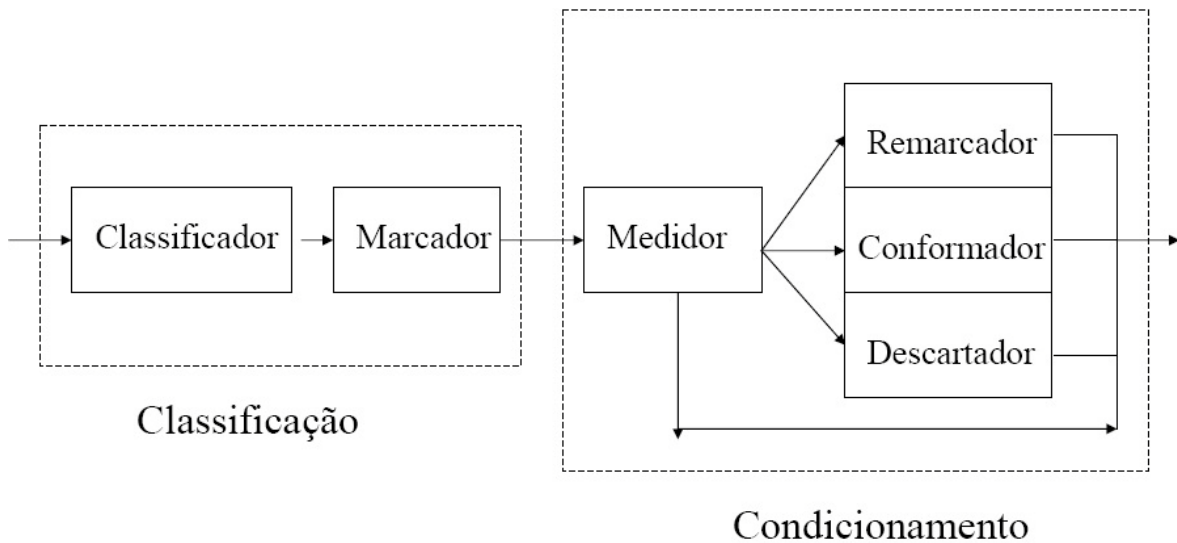


Figura 5 – Arquitetura funcional da *DiffServ*.

(BRITO, 1996)

Na *DiffServ*, os nós de borda são responsáveis por mapear pacotes para uma classe de encaminhamento e garantir que o tráfego esteja conforme o *SLA* associado. Dentro da rede os recursos são alocados e os pacotes são encaminhados com base na classe de encaminhamento.

As funções executadas pelos nós de borda são denominadas de classificação e condicionamento de tráfego. A função de classificação compreende o classificador e o marcador da figura acima, enquanto a função de condicionamento compreende o medidor e o condicionador, que executa a função de remarcação, conformação ou descarte dos pacotes.

3.4.1 Classificação

O classificador divide o fluxo de pacotes em múltiplos grupos, baseado em regras pré-definidas. Existem dois tipos de classificadores: *Behavior Aggregate (BA)* e *Multifield (MF)*. O classificador tipo BA é o mais simples, ele seleciona os pacotes baseado apenas no valor do DSCP contido no cabeçalho. Este tipo de classificador normalmente é utilizado

quando o DSCP associado ao pacote já foi configurado antes que o pacote chegue ao classificador.

O valor do DSCP pode ser configurado de várias formas. Se o cliente suporta a *DiffServ*, é desejável que os pacotes sejam marcados pelo host ou pelo roteador da rede local. Clientes que não suportam *DiffServ* devem solicitar ao provedor de serviço que faça a marcação dos pacotes.

O classificador MF utiliza um ou mais campos do cabeçalho do pacote (endereços de fonte e destino, portas de fonte e destino, protocol ID) para a classificação do pacote. A classificação pode levar em conta o tipo de aplicação, a rede ou host que está gerando o pacote, a rede ou host de destino do pacote, etc. O classificador MF define o valor do DSCP associado ao pacote e o configura no cabeçalho do pacote utilizando o marcador.

3.4.2 Condicionador

O condicionador de tráfego executa funções de policiamento de tráfego para garantir que o *TCA* entre o cliente e o provedor de serviço da rede seja cumprido. O condicionador consiste de quatro elementos básicos: medidor, remarcador, conformador e descartador.

- Medidor: monitora o tráfego de um cliente e o compara com o perfil de tráfego definido no *TCA*. Os pacotes considerados conformes podem entrar na rede, enquanto os pacotes não-conformes sofrem a ação de um dos três elementos seguintes: remarcador, conformador e descartador. O perfil do tráfego é tipicamente descrito através de parâmetros do *Token bucket*.
- Remarcador: os pacotes não-conformes podem ter o valor do seu DSCP alterado para indicar sua não-conformidade. Estes pacotes serão os primeiros a serem descartados caso a rede entre em situação de congestionamento. A remarcação pode também ser necessária na fronteira entre dois domínios administrativos que utilizam diferentes DSCPs.

- Conformador: atrasa os pacotes não-conformes, de modo que os mesmos passem a ter o comportamento definido no perfil de tráfego associado. A função de conformação pode também ser necessária na fronteira entre dois domínios administrativos: o perfil de tráfego tende a mudar quando os pacotes atravessam a rede, e o nó de egresso do domínio a *upstream* pode ter de fazer a conformação dos pacotes.
- Descartador: descarta os pacotes considerados não-conformes,

4 MULTIPROTOCOL LABEL SWITCHING (MPLS)

O *MPLS* (*Multiprotocol Label Switching*) é um padrão da *IETF* (*Internet Engineering Task Force*) onde os pacotes são encaminhados pela rede com base em um *label*, e não com base no endereço IP.

4.1 Princípios

O *MPLS* separa os planos de roteamento e encaminhamento. No plano de roteamento os pacotes são enviados pela rede com base no endereço IP de destino, utilizando-se o roteamento IP normal, enquanto no plano de encaminhamento os pacotes são enviados com base em um *label* acrescentado ao pacote.

O envio dos pacotes pelo plano de encaminhamento requer que um caminho seja estabelecido entre origem e destino (*LSP - Label Switched PATH*) e que *labels* sejam atribuídos a cada enlace ao longo deste caminho. O *LSP* está associado a uma *FEC* (*Forward Equivalence Class*), que foi associada aos dados gramas entre as estações pelo roteador de borda.

No *MPLS* os pacotes que chegam a um roteador de borda são analisados e associados a uma *FEC* (*Forward Equivalence Class*). A cada *FEC* está associado um *LSP* (*Label Switched PATH*), e a cada *LSP* está associado um conjunto de *labels*. A comutação dos pacotes dentro da rede é feita com base nos *labels*, e não no endereço IP.

4.2 Formas de implementação

O *MPLS* pode ser implementado como uma solução de nível 3, onde o *label* é adicionado ao datagrama IP; como uma solução de IP sobre Frame Relay, onde o *label* é o DLCI do cabeçalho Frame Relay; ou como uma solução de IP sobre ATM, onde o datagrama é transportado em células ATM e o *label* é composto pelos campos VPI/VCI do cabeçalho da célula ATM.

4.3 Componentes

Os LERs estão na interface entre redes não-MPLS e a rede MPLS. Eles são responsáveis por iniciar o processo de estabelecimento de um caminho (*LSP*) na rede. O LER associa os datagramas às *FECs* estabelecidas e acrescenta o *label* inicial, encaminhando o datagrama para o próximo nó no caminho.

Os LSRs comutam os pacotes ao longo do *LSP* tomando como base o *label*. A comutação baseada em *label* permite a implementação de LSRs de alto desempenho.

4.3.1 *Forward Equivalence Class - FEC*

Pacotes associados a uma mesma *FEC* são encaminhados pela rede da mesma maneira, através do mesmo *LSP*.

A estratégia para se definir uma *FEC* pode ser simples, levando em conta apenas o endereço do destino, por exemplo, ou complexa, onde diversos outros fatores podem ser considerados.

Por exemplo, um provedor pode definir classes de serviço, tais como serviço premium e serviço olímpico (ouro, prata e bronze), com tarifas diferenciadas. Neste caso, a *FEC* pode levar em conta o endereço do destino e o tipo de serviço que está sendo solicitado pelo cliente, ou seja, datagramas entrando pelo mesmo LER de ingresso e destinados ao mesmo LER de egresso podem estar associados a *FECs* distintas e serem encaminhados por *LSPs* distintos (no mesmo caminho físico ou não).

4.4 Definição da rota

O roteamento explícito torna-se viável com o MPLS pois, uma vez criado o *LSP*, os datagramas não mais precisam transportar os endereços dos roteadores ao longo do caminho, bastando associar ao *LSP* a característica de operação baseada em roteamento explícito. Nesta solução, mudanças no roteamento IP não afetarão o *LSP*, e o mesmo se aproxima em termos de comportamento de um circuito virtual.

4.5 Estabelecimento do caminho

No método de alocação de *label* disparado por topologia, um *label* é associado em resposta ao processamento normal do protocolo de roteamento. Se uma rota existe, um *label* é associado a ela e distribuído, sendo a quantidade de tráfego de controle proporcional ao número de rotas. Neste tipo de solução o tráfego chegando por uma dada rota pode ser associado imediatamente a um *label*, sem latência.

No método disparado por fluxo a detecção de um fluxo dispara o processo de estabelecimento de um *LSP* e distribuição de *labels*. A quantidade de tráfego de controle gerado é proporcional ao número de fluxos. Esta característica faz com que redes com fluxos recorrentes e com vida curta possam ter um excesso de tráfego de controle para o estabelecimento dos caminhos. Neste método existe uma latência entre o aparecimento de um fluxo e a alocação de um *label*.

4.6 Alocação de *label*

No método de alocação do *label* a *downstream* um LSR faz a distribuição de *labels* para LSRs que estejam a *upstream*, mesmo que os nós a *upstream* não tenham solicitado explicitamente.

No método *downstream* sob demanda, um LSR solicita explicitamente a seu nó subsequente (*downstream*) que ele faça a associação de *label* para uma determinada *FEC*.

4.7 Falha em um enlace da rede

A falha de um enlace em uma rede *MPLS* pode ser detectada através da operação normal do roteamento, ou através das mensagens keep-alive do LDP. Todos os *LSPs* que estiverem utilizando o enlace em falha serão desfeitos. Outros *LSPs* serão criados através de rotas alternativas definidas pelo roteamento.

5 Considerações

Com esse documento é possível obter conhecimento sobre as características das arquiteturas disponíveis para Qualidade de Serviço. Pode-se concluir que no ambiente corporativo o uso da arquitetura *IntServ* pode ser uma boa alternativa. No backbone, devido ao grande número de fluxos, a arquitetura *DiffServ*, possivelmente associada ao uso de *MPLS* com roteamento explícito, pode ser a melhor opção.

REFERÊNCIAS BIBLIOGRÁFICAS

PRYCKER, Martin De. Asynchronous Transfer Mode, Solution for Broadband ISDN. Third Edition. Prentice Hall, 1995.

RFC 1889 - RTP: A Transport Protocol for Real-Time Applications – January, 1996.

Planning for Quality of Service. Documento elaborado pela Cisco, disponível em www.cisco.com. Acesso em 02/08/2010

CROLL, Alistair e Packman, Eric. Managing Bandwidth, Deploying QoS in Enterprise Networks. Prentice Hall, 2000.

Quality of Service Solutions Configuration Guide, Classification Overview. Documento elaborado pela Cisco, disponível em www.cisco.com. Acesso em 03/08/2010

Quality of Service Solutions Configuration Guide, Congestion Management Overview. Documento elaborado pela Cisco, disponível em www.cisco.com. Acesso em 03/08/2010

MAGALHÃES, Maurício F. e Cardoso, Eleri. Qualidade de Serviço na Internet – Versão Draft. Unicamp, 1999.

Quality of Service (QoS) Networking. Documento elaborado pela Cisco, disponível em www.cisco.com. Acesso em 27/08/2010

Quality of Service Solutions Configuration Guide, Congestion Avoidance Overview. Documento elaborado pela Cisco, disponível em www.cisco.com. Acesso em 27/08/2010

Quality of Service (QoS) An Overview Advanced Services. Documento elaborado pela Cisco, disponível em www.cisco.com. Acesso em 27/08/2010

Quality of Service in Cisco Networks. Documento elaborado pela Cisco, disponível em www.cisco.com. Acesso em 27/08/2010

Quality of Service Solutions Configuration Guide, Policing and Shaping Overview. Documento elaborado pela Cisco e disponível em www.cisco.com. Acesso em 02/09/2010

Quality of Service Solutions Configuration Guide, Link Efficiency Mechanisms Overview. Documento elaborado pela Cisco e disponível em www.cisco.com. Acesso em 02/09/2010

BRITO, José Marcos Câmara. Qualidade de Serviço em Redes de Pacotes. Apostila de curso. Inatel, 1996.

RFC 1633. Integrated Services in the Internet Architecture: an Overview. July 1994. Documento disponível em www.ietf.org

FERGUSON, Paul e Huston, Geoff. Quality of Service, Delivering QoS on the Internet and in Corporate Networks. John Wiley & Sons, 1998.

RFC 2212. Specification of Guaranteed Quality of Service. September 1997. Documento disponível em www.ietf.org/rfc/rfc2212.txt Acesso em 17/09/2010

A Framework for *Multiprotocol Label Switching*. Internet Draft, July 1999. Documento disponível em www.ietf.org/internet-drafts/draft-ietf-MPLS-framework-04.txt Acesso em 17/09/2010

BAKER, F., Aggregation of *RSVP* for IPv4 and IPv6 Reservations, RFC3175, IETF, Setembro, 2001.

BLACK, D., An Architecture for Differentiated Services, RFC 2475, IETF, Dezembro, 1998.

IntServ workgroup charters, <http://www.ietf.org/html.charters/IntServ-charter.html>. Acesso em 30/09/2010

DiffServ workgroup charters, <http://www.ietf.org/html.charters/DiffServ-charter.html>. Acesso em 30/09/2010

GRANADO FILHO, Arlindo Garcia. *MPLS, Multiprotocol Label Switching*. Trabalho da disciplina IA 368, dezembro de 1998. Unicamp, SP.

RFC 2205 – Resource ReSerVation Protocol (*RSVP*), Functional Specification. Documento disponível em www.ietf.org/rfc/rfc2205.txt Acesso em 12/10/2010

Bernet, Y., Format of the *RSVP* DCLASS Object, RFC2996, IETF, Novembro, 2000.

QoS Protocols & Architectures. White Paper produzido por Stardust Inc, disponível em www.QoSforum.com. Acesso em 21/10/2010

XIAO, Xipeng and Ni, Lionel M. – Internet QoS: A Big Picture. Department of Computer Science, Michigan State University.