

QoS em Redes IP: Arquitetura e Aplicações

Mário Meireles Teixeira
mario@deinf.ufma.br

Motivação

- ◆ Atualmente, funcionam sobre as redes IP aplicações cujos requisitos elas não foram projetadas para atender
 - Modelo de serviços de "melhor esforço"
 - Tabela de rotas determina apenas o próximo salto, sem considerar prioridades/requisitos das aplicações
- ◆ A solução não parece ser simplesmente acrescentar largura de banda às redes
 - Diferentes aplicações possuem diferentes requisitos (multimídia, tempo real, download, multicast, e-commerce...)
 - Díficil implantação
 - Tráfego em rajadas
- ◆ É preciso realizar um gerenciamento ativo da largura de banda disponível, fornecendo:
 - Serviço mais previsível e consistente
 - Diferentes níveis ou classes de serviço

Qualidade de Serviço (QoS)

- ◆ É a capacidade de fornecer a um elemento da rede algum nível de segurança de que seus requisitos de tráfego e serviço serão atendidos
- ◆ Alguns requisitos de QoS:
 - Largura de banda
 - Taxa de perda de pacotes
 - Atraso (latência ou *delay*)
 - *Jitter* (variação do atraso)
- ◆ Para atingir a QoS, faz-se necessária a cooperação de todos os elementos envolvidos (de cima a baixo, de fim a fim)
- ◆ Qualquer garantia de QoS será tão forte quanto o mais frágil elemento na cadeia entre o emissor e o receptor!
- ◆ QoS pode ser *absoluta* (métricas rígidas) ou *relativa* (diferenciação de serviços)

Caracterização do Tráfego

Tipo da Taxa	Descrição
Fluxo Contínuo	Entrega de dados a uma taxa relativamente constante (CBR — <i>Constant Bit Rate</i>)
Em Rajadas	Blocos de dados são entregues de forma imprevisível, a uma taxa de dados variável (VBR — <i>Variable Bit Rate</i>), que pode chegar a utilizar toda a largura de banda se não for empregado algum tipo de controle

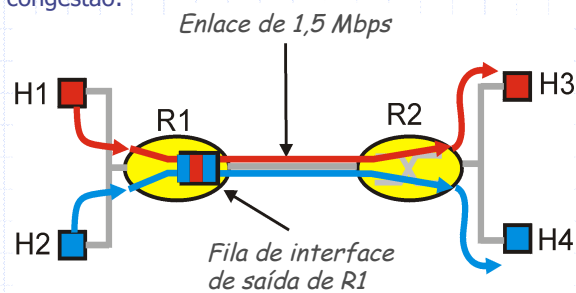
segundo a previsibilidade da taxa de dados

Tolerância a Atrasos	Tipo de Tráfego	Descrição
<i>alta</i>	Assíncrono	Tráfego não possui restrições temporais
	Síncrono	Dados são sensíveis ao tempo, mas flexíveis
	Interativo	Atrasos podem ser notados pelos usuários, mas não afetam a usabilidade e funcionalidade da aplicação
	Isócrono	Sensível ao tempo de uma forma que pode comprometer a usabilidade da aplicação
<i>baixa</i>	Missão-crítica	Atrasos podem comprometer a funcionalidade da aplicação

segundo a tolerância ao atraso e *jitter*

Provisão de QoS em Redes IP

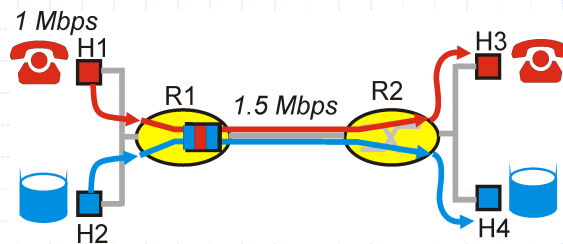
- ◆ Os grupos da IETF têm trabalhado em propostas para fornecer melhor controle de QoS em redes IP, procurando superar o serviço de melhor esforço e prover alguma garantia de QoS
- ◆ Alguns resultados recentes: Protocolo RSVP, Serviços Integrados e Serviços Diferenciados
- ◆ Cenário simples para estudos de compartilhamento e congestão:



Princípios para Garantia de QoS (1)

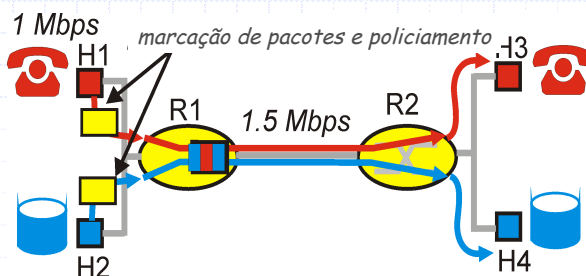
- ◆ Considere uma aplicação de áudio a 1 Mbps e uma aplicação FTP, ambas compartilhando um enlace de 1,5 Mbps:
 - rajadas de tráfego FTP podem congestionar o roteador e fazer com que pacotes de áudio sejam perdidos.
 - deseja-se dar prioridade ao áudio sobre o FTP
- ◆ **PRINCÍPIO 1: A marcação de pacotes permite que um roteador faça distinção entre pacotes de diferentes classes de tráfego – diferentes regras de roteamento**

A marcação de pacotes pressupõe uma classificação anterior



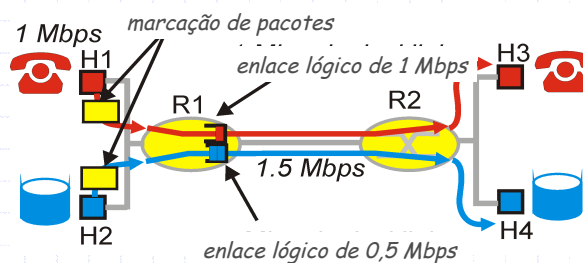
Princípios para Garantia de QoS (2)

- ◆ Deseja-se priorizar o FTP sobre o áudio
- ◆ Aplicações mal-comportadas (p.ex., áudio envia pacotes numa taxa superior a 1 Mbps)
- ◆ Exige mecanismos de **policimento** (regulação) para assegurar que as fontes obedeçam aos seus requisitos de QoS previamente acertados
- ◆ Marcação e policimento devem ser feitos nas bordas da rede
- ◆ **PRINCÍPIO 2: É desejável fornecer um certo grau de isolamento entre os diferentes fluxos de tráfego (classes)**



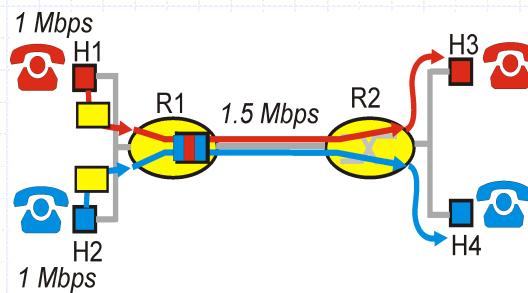
Princípios para Garantia de QoS (3)

- ◆ Alternativa à marcação e policimento: alocar uma porção fixa da taxa de transmissão a cada fluxo de aplicação; pode produzir um uso ineficiente da banda se um dos fluxos não usa toda a sua alocação
- ◆ **PRINCÍPIO 3: Embora garantindo isolamento, é necessário ter eficiência na utilização dos recursos**

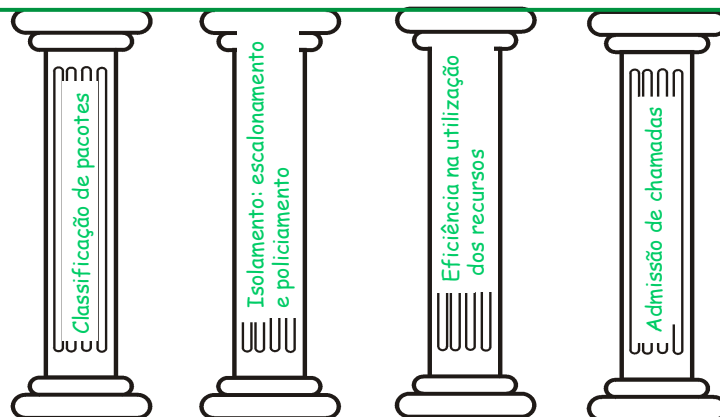


Princípios para Garantia de QoS (4)

- ◆ Não deve ser aceito tráfego além da capacidade do enlace
- ◆ PRINCÍPIO 4: **Necessita-se de um processo de admissão de chamadas**; a aplicação declara a necessidade do seu fluxo e a rede pode bloquear seu acesso caso não seja possível satisfazer os requisitos colocados



QoS para aplicações em rede

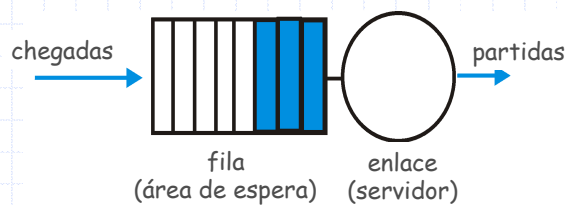


Como garantir esses quatro princípios:

classificação, isolamento, eficiência e controle de admissão

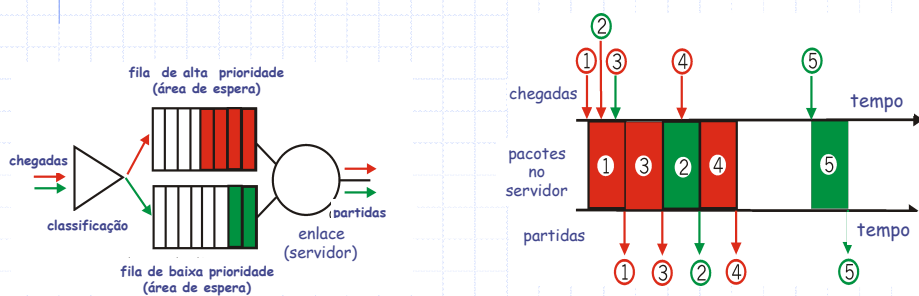
Mecanismos de Escalonamento e Policiamento

- ◆ Escalonamento: a escolha do próximo pacote para transmissão num enlace pode ser feita de acordo com várias regras
- ◆ **FIFO**: atendimento segundo a ordem de chegada na fila; se a fila estiver cheia, algum pacote será descartado



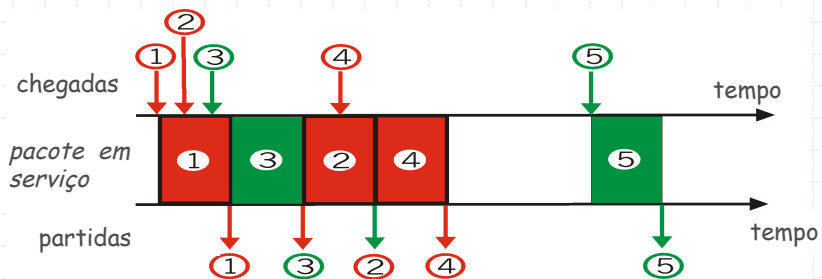
Disciplinas de Escalonamento

- ◆ **Filas com Prioridade:** classes têm diferentes prioridades; identificação da classe pode depender de marcação explícita no pacote ou de outras informações, como: endereço de origem/destino, número de porta de origem/destino, etc.
- ◆ Transmite o pacote de prioridade mais alta cuja fila não esteja vazia
- ◆ Versão **preemptiva** e **não-preemptiva**



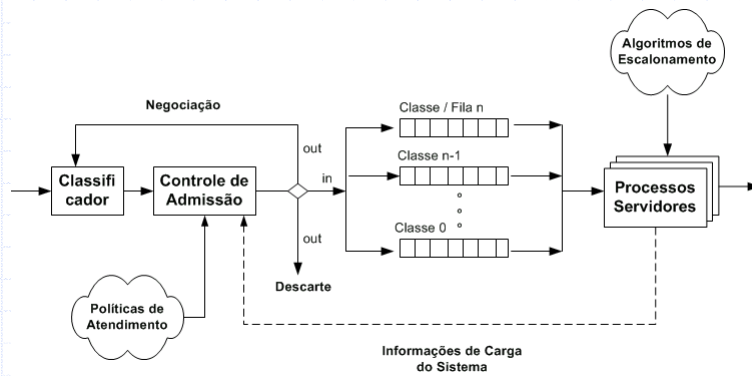
Disciplinas de Escalonamento

- ◆ **Round Robin (RR):** percorre todas as filas em seqüência, servindo um pacote de cada classe que tiver pelo menos um representante na fila

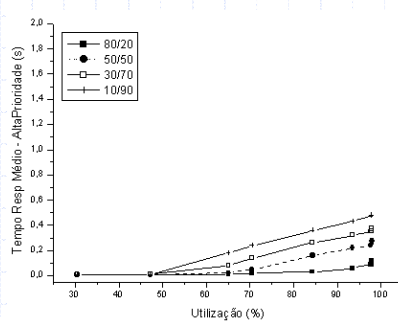
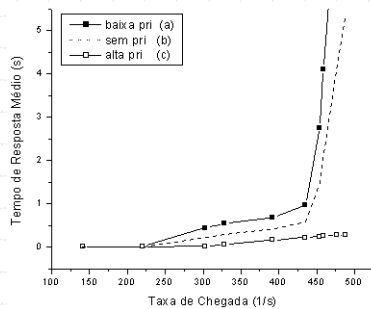


SWDS

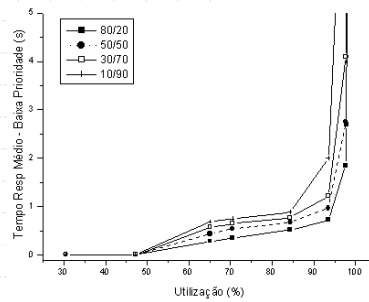
Servidor Web com Diferenciação de Serviços



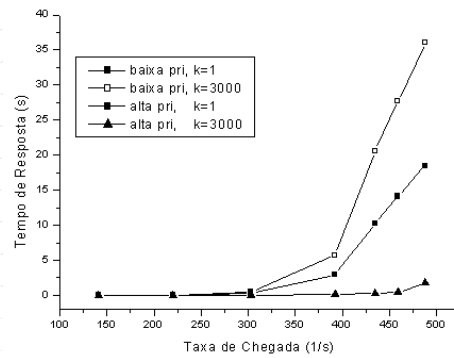
Filas de Prioridades em Servidores Web



80/20

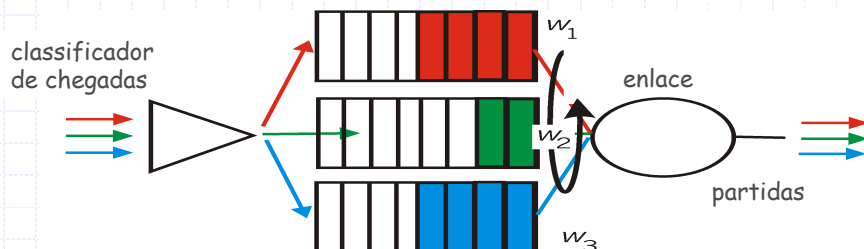


Fila de Prioridades Adaptativa



Disciplinas de Escalonamento

- ◆ **Weighted Fair Queuing (WFQ):** procura fornecer uma quantidade diferenciada de serviço a cada classe, num dado período de tempo, atribuindo-lhes pesos diferentes
- ◆ Assim, uma classe i , de peso w_i , receberá uma fração de serviço $\frac{w_i}{\sum w_j}$ e cada fluxo conseguirá uma vazão de, no mínimo, $R \cdot \frac{w_i}{\sum w_j}$
- ◆ RR é um caso específico do WFQ, com todos os pesos iguais a 1

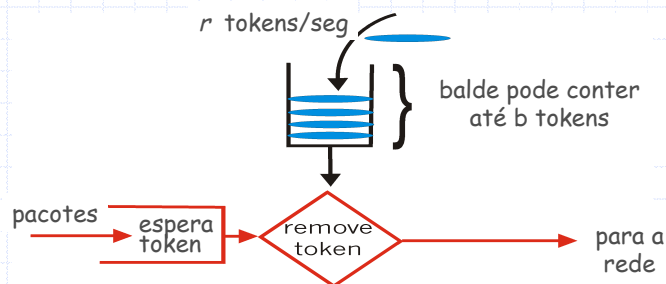


Mecanismos de Policiamento

- ◆ Policiamento está relacionado à taxa com a qual se permite que um fluxo (aplicação) injete pacotes na rede
- ◆ Três critérios:
 - **Taxa Média:** (100 pacotes por segundo ou 6000 pacotes por minuto?); o aspecto crucial é o tamanho do intervalo
 - **Taxa de Pico:** 6000 pacotes por minuto, na média; 1500 pacotes por segundo, no pico
 - **Tamanho da Rajada** (*burst size*): número máximo de pacotes enviados consecutivamente, num período de tempo extremamente curto

Mecanismos de Policiamento

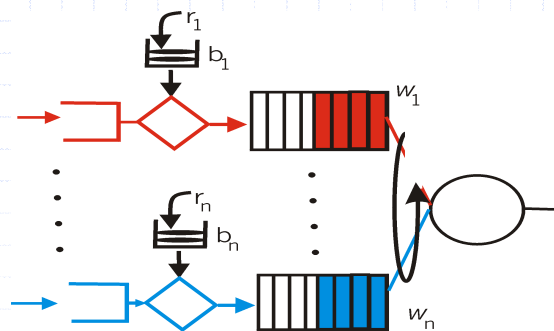
- ◆ **Token Bucket** (balde de permissões)
 - Oferece um meio de limitar a entrada de pacotes na rede, que ficam sujeitos a um tamanho de rajada e a uma taxa média pré-determinados



Mecanismos de Policiamento

- ◆ Balde pode armazenar até b tokens; os tokens são gerados a uma taxa de r tokens/seg, exceto se o balde estiver cheio
- ◆ Em qualquer intervalo de tempo, o número de pacotes admitidos na rede será sempre menor ou igual a $(r t + b)$
- ◆ Token bucket e WFQ podem ser combinados para garantir um limite superior ao atraso:

$$d_{\max} = \frac{b_i}{R \sum w_j}$$



Arquiteturas para QoS em Redes IP

- ◆ A introdução de QoS nas redes IP atuais visa fornecer algum nível de previsibilidade e controle além do serviço de melhor esforço
- ◆ Duas estratégias para provisão de QoS na Internet:
 - Reserva de Recursos
 - ◆ Os recursos são atribuídos às aplicações segundo suas demandas de QoS, antes da transmissão dos dados
 - ◆ Serviços Integrados – *IntServ*
 - Priorização
 - ◆ O tráfego na rede é classificado segundo suas características de demanda, nos pontos de ingresso na rede e a QoS é garantida a cada *hop*
 - ◆ Serviços Diferenciados – *DiffServ*

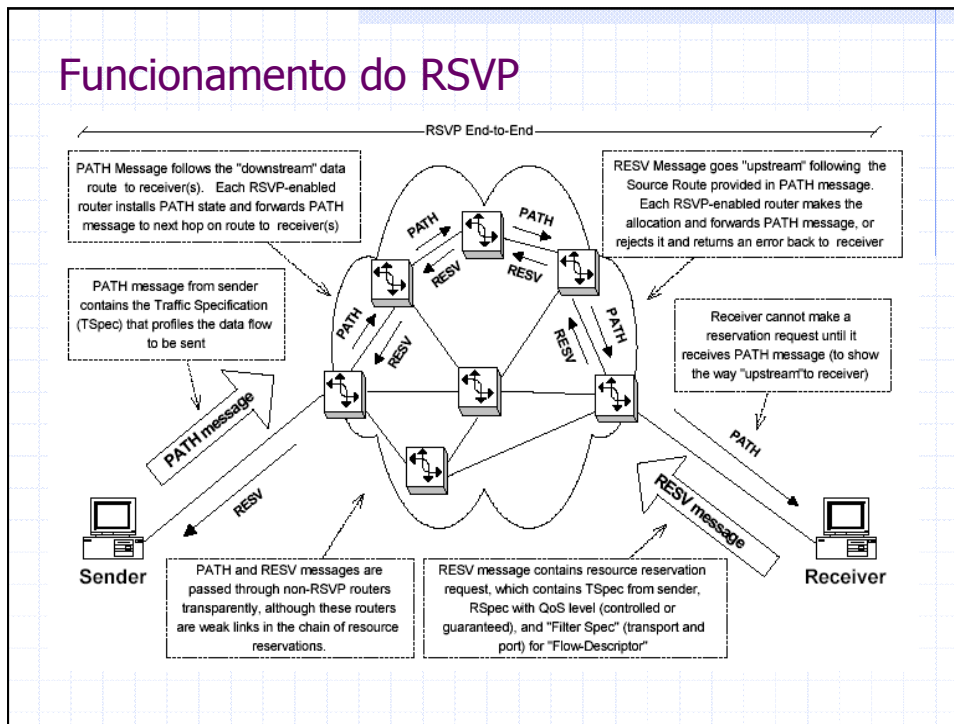
Serviços Integrados

- ◆ Primeira das arquiteturas propostas pela IETF para dar suporte a QoS na Internet (RFC 1633)
- ◆ Visa fornecer, em uma rede de pacotes, o serviço mais próximo possível da abstração de circuitos virtuais
- ◆ Baseia-se na idéia de reserva de recursos, fim-a-fim, antes do início da transmissão dos dados
- ◆ **Protocolo RSVP** – protocolo de controle e sinalização responsável pela reserva de caminhos e recursos na sub-rede de comunicação (RFC 2205)
 - Trabalha com fluxos individuais → sobrecarga
 - Roteadores precisam manter informações de estado relacionadas aos fluxos
 - Problemas de escalabilidade e implantação na Internet

Protocolo RSVP

- ◆ Características
 - Protocolo de controle e sinalização que atua na camada de rede, transportando informações relativas à reserva de recursos ao longo de um *path* ou *spanning tree*
 - Transmissão *unicast* e *multicast*
 - Transmissão unidirecional
 - Reserva iniciada pelo receptor – especifica os requisitos de largura de banda, atraso e *jitter*
 - *Soft state* – é preciso renovar a informação sobre a reserva de recursos periodicamente nos roteadores, responsabilidade dos *end hosts*
 - Operação transparente através de roteadores não-RSVP
 - Suporte para IPv4 (*Type-of-Service*) e IPv6 (*Flow Label*)

Funcionamento do RSVP



Classes de Serviço – IntServ

◆ Serviço Assegurado (RFC 2212)

- Fornece um limite superior rígido para o atraso fim-a-fim, além de garantir a disponibilidade de largura de banda
- Alto nível de QoS na Internet
- Voltado a aplicações com requisitos rígidos de tempo real (vídeo-conferência, voz sobre IP)

◆ Serviço de Carga Controlada (RFC 2211)

- Serviço equivalente ao modelo de melhor esforço em uma rede pouco carregada, com quase nenhuma perda ou atraso
- Em situações de sobrecarga, a largura de banda será compartilhada entre múltiplos fluxos de maneira controlada → serviço melhor que o de melhor esforço
- Não oferece garantia de atraso máximo, apenas um limiar probabilístico. Não assegura que pacotes não serão perdidos

Problemas com Intserv

◆ Escalabilidade

- Intserv trabalha sobre **fluxos individuais**
- Exige a manutenção das reservas para cada fluxo que passa por um roteador – **sobrecarga** numa rede do tamanho da Internet!

◆ Modelos de serviço não flexíveis

- Intserv fornece apenas um pequeno número de classes de serviço pré-definidas
- Não comporta diferenciação de serviços relativa: classes Ouro, Prata e Bronze, por exemplo

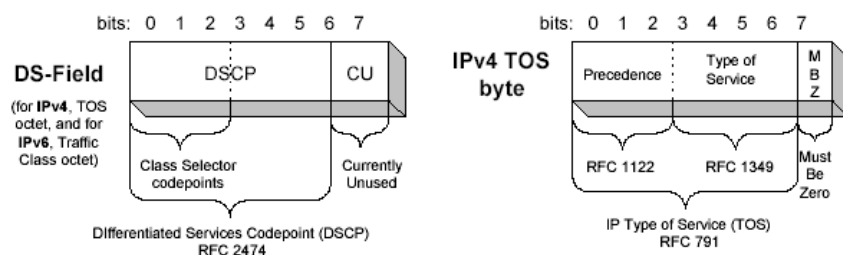
27

Serviços Diferenciados - Diffserv

- ◆ Baseia-se na idéia de **agregação de fluxos** em classes de serviço (RFC 2475)
- ◆ Abordagem: **classificação** e **marcação** de pacotes
- ◆ Os pacotes são marcados nos pontos de ingresso na rede – complexidade na borda da rede
- ◆ Redefine-se o layout de um campo do datagrama IP – especifica como é feito o encaminhamento de pacotes nos roteadores
- ◆ Diffserv fornece diferenciação de serviços local para grandes agregados de tráfego; IntServ trabalha com fluxos individuais fim-a-fim

28

Differentiated Services Code Point (DSCP)

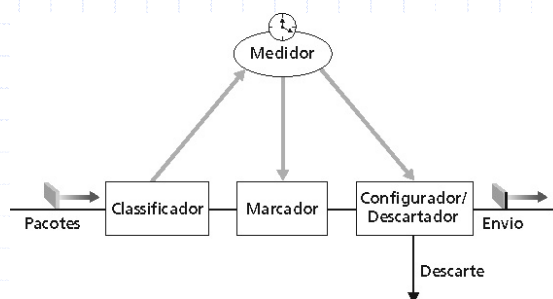


PHB: encaminhamento de pacotes

- ◆ Campo DSCP define como será feito o encaminhamento de pacotes em cada roteador
 - **PHB** (*Per-Hop Behavior*)
- ◆ A marcação dos pacotes permite à rede tratá-los de forma diferenciada
- ◆ Abordagem do DiffServ: **prioridades relativas**
- ◆ Diffserv não necessita de protocolo próprio (facilidade de implantação)
- ◆ Roteadores não precisam guardar informações sobre cada fluxo (escalabilidade)

Diffserv: classificação e condicionamento de tráfego

- ◆ Pode-se desejar limitar a entrada de tráfego em alguma classe de serviço
- ◆ Usuário deve declarar seu perfil de tráfego (p.ex., taxa de transmissão e tamanho das rajadas)
- ◆ Tráfego fora do perfil contratado pode ser descartado



31

Classes de Serviço – DiffServ

- ◆ **Envio Expresso** (RFC 2598)
 - Também denominado de *Premium Service* ou *EF PHB*
 - Fornece o equivalente a uma linha privada virtual, com largura de banda fixa, entre dois hosts. Indicado para telefonia sobre IP, videoconferência e criação de VPNs
 - Oferece garantias absolutas de perda, atraso e *jitter*
 - Requer alguma forma de isolamento de tráfego entre as classes. Tráfego não conforme com o perfil contratado é descartado
 - Tem implementação mais simples que o *Serviço Garantido* do IntServ
 - Os pacotes desta classe são os primeiros a serem encaminhados, em qualquer situação. Baixa probabilidade de descarte para o tráfego em conformidade com o perfil contratado

Classes de Serviço – DiffServ

◆ Envio Assegurado (RFC 2597)

- Também denominado de *Assured Service* ou *AF PHB*
- Fornece um serviço melhor que o de melhor esforço, mas sem garantias rígidas de QoS (ideal para aplicações C/S)
- Não oferece limites superiores para o atraso e *jitter*
- Tráfego é dividido em N classes, cada uma com um mínimo garantido de largura de banda e M níveis de precedência de descarte (atualmente, N=4, M=3)
- Serviço fornecido por uma classe independe do serviço das demais, sendo função dos recursos alocados àquela classe
- Pacotes do serviço AF são os últimos a ser descartados em situações de congestionamento

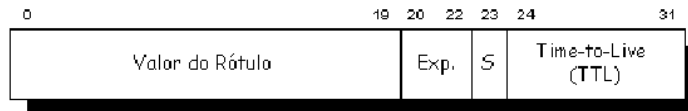
IntServ vs. Diffserv

IntServ	DiffServ
Reserva de recursos	Classificação de pacotes
Fluxos individuais	Agregação de fluxos
Gerenciamento mais complexo	Gerenciamento mais simples
QoS intra-domínio	QoS em redes maiores
Complexidade no núcleo da rede	Complexidade nos pontos externos da rede
Oferece garantias absolutas de QoS	Diferenciação de serviços relativa entre as classes
Coordenação de QoS fim-a-fim	Decisões tomadas localmente
Necessita de um protocolo próprio	Usa um campo do protocolo IP
Preocupa-se com as demandas de serviço de cada aplicação	Leva em consideração as propriedades do tráfego gerado

Protocolo MPLS

Multi-Protocol Label Switching

- ◆ Especificado na RFC 3031
- ◆ Assim como o DiffServ, marca os pacotes nos pontos de ingresso na rede e os desmarca na saída
- ◆ Marcação de pacotes
 - DiffServ – atribuição de prioridades
 - MPLS – informação utilizada para roteamento
- ◆ O rótulo MPLS determina completamente o caminho a ser seguido pelo pacote, criando canais com largura de banda fixa (como circuitos virtuais em redes ATM ou *Frame Relay*)
- ◆ MPLS opera entre as camadas de rede e enlace, portanto pode funcionar sobre outros protocolos além do IP



Cooperação entre as Arquiteturas

