

Aulas 12, 13 e 14

Redes Multimídias

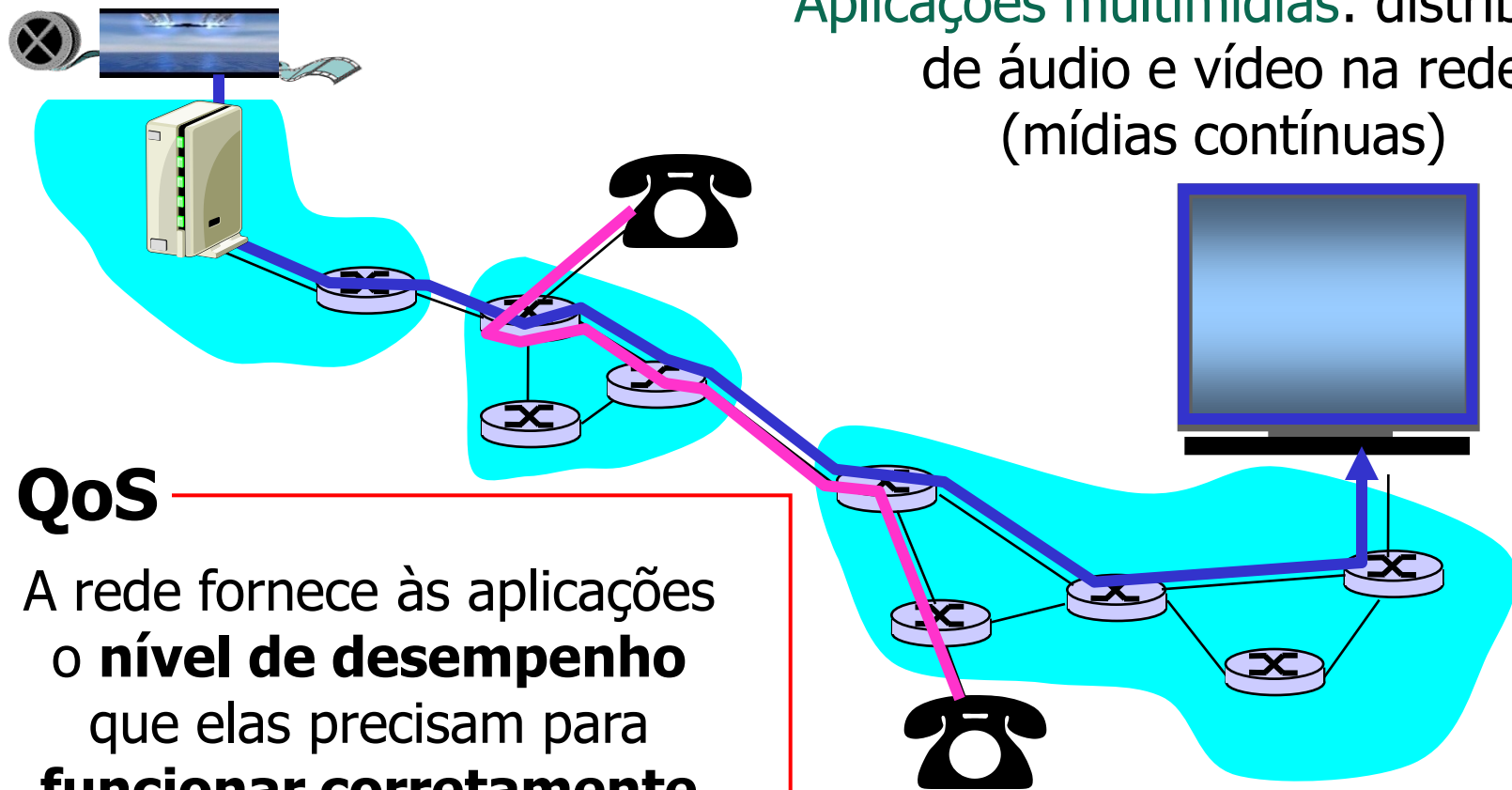
Conceitos, aplicações, requisitos e mecanismos; QoS

Igor Monteiro Moraes
Redes de Computadores II

ATENÇÃO!

- Este apresentação é contém partes baseadas nos seguintes trabalhos
 - Notas de aula do Prof. Marcelo Rubinstein, disponíveis em <http://www.lee.eng.uerj.br/~rubi>
 - Notas de aula do Prof. José Augusto Suruagy Monteiro, disponíveis em <http://www.nuperc.unifacs.br/Members/jose.suruagy/cursos>
 - Material complementar do livro Computer Networking: A Top Down Approach, 5th edition, Jim Kurose and Keith Ross, Addison-Wesley, abril de 2009
 - Computer Networks, Andrew S. Tanenbaum, 4a. Edição, Editora Prentice Hall

Multimídia? Qualidade de Serviço?



QoS

A rede fornece às aplicações o **nível de desempenho** que elas precisam para **funcionar corretamente**

- Distribuição de vídeo na Internet
 - É um tipo de aplicação multimídia
 - Grande sucesso
- Exemplos
 - YouTube
 - Compartilhamento e vídeo sob demanda
 - 20 milhões de usuários por dia
 - Tempo total de reprodução equivalente a 10 mil anos
 - CBS
 - Difusão de TV
 - 260 mil usuários simultâneos em 2006
 - Final da liga universitária de basquete

- Exemplos
 - Globo.com
 - Difusão de TV
 - 72 mil usuários simultâneos em 2006
 - Semifinal da Copa do Mundo entre Portugal e França
 - 50 mil usuários em média por jogo na Copa de 2014

Desafios da Distribuição Multimídia na Internet

1. Garantir os requisitos de QoS do conteúdo multimídia
 - Mídia contínua
 - Grande largura de banda
 - Sensível ao atraso e a perda de pacotes
2. Garantir a escalabilidade
 - Grande número de receptores
3. Atender de forma satisfatória aos receptores
 - Diferentes capacidades de processamento
 - Conectados à Internet através diferentes redes de acesso

Aplicações Multimídias

- Princípios
- Tipos de aplicações
- Serviços de rede exigidos por uma aplicação
- Mecanismos da camada de aplicação
 - Usar o melhor do serviço de melhor esforço
 - Sem modificações no núcleo da rede

Qualidade de Serviço

- Protocolos e arquiteturas
 - Protocolos específicos para o modelo de melhor esforço
 - Mecanismos para prover QoS
 - Arquiteturas para prover QoS na Internet

Aplicações Multimídias

- Na **distribuição de fluxo contínuo = *streaming***
 - Um usuário **não precisa receber e armazenar** todo o conteúdo multimídia para **reproduzi-lo**
 - O conteúdo multimídia é enviado em um **fluxo contínuo** e o seu conteúdo é **reproduzido ao ser recebido pelo usuário**
 - Exemplo
 - Distribuição de fluxo contínuo de vídeo → *video streaming*
- Deste ponto em diante
 - Distribuição
 - Expressão reduzida para distribuição de fluxo contínuo

- Características fundamentais
 - São tipicamente sensíveis ao **atraso**
 - Atraso fim-a-fim
 - Variação do atraso (*jitter*)
 - É a variabilidade dos atrasos dos pacotes de um mesmo fluxo de pacotes
 - Tolerantes a **perdas de pacotes***1
 - Se não forem frequentes, causam pequenas interrupções (*glitches*)

- Características fundamentais
 - São tipicamente sensíveis ao **atraso**
 - Atraso fim-a-fim
 - Variação do atraso (*jitter*)
 - É a variabilidade dos atrasos dos pacotes de um mesmo fluxo de pacotes
 - Tolerantes a **perdas de pacotes***1
 - Se não forem frequentes, causam pequenas interrupções (*glitches*)

Oposto da transferência de dados:
intolerante às perdas, mas tolerante ao atraso

*1 Vídeo Codificado em MPEG-4

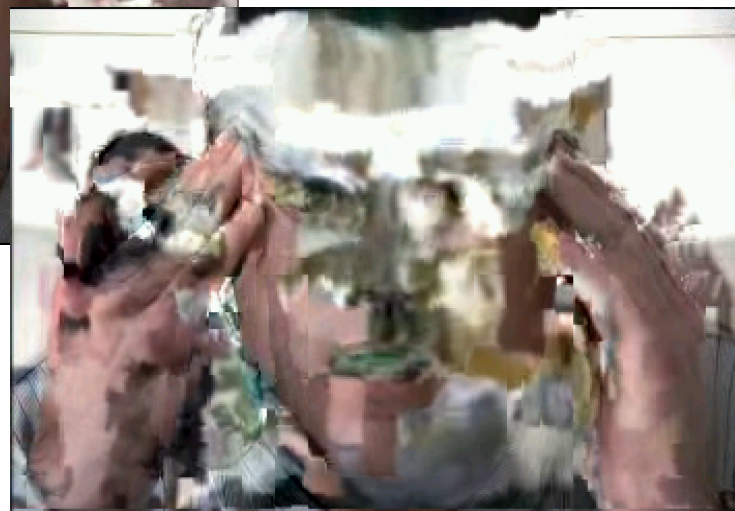
99% dos pacotes entregues



97 %

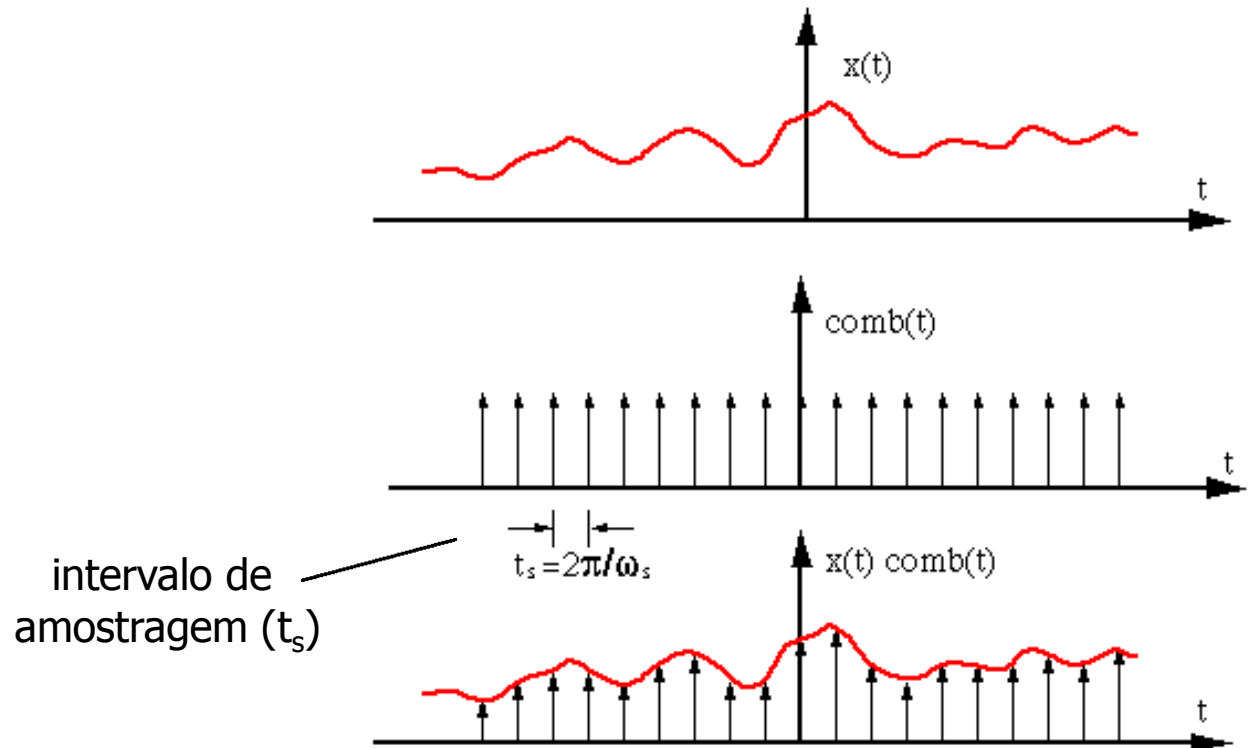


92%



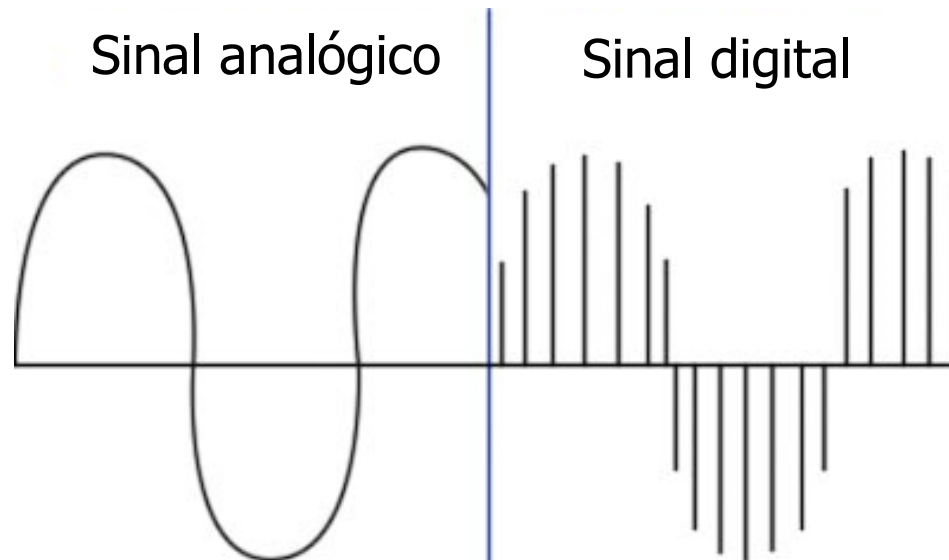
Conversão Analógico-Digital

- Amostragem
 - Sequência de amostras dos sinal analógico



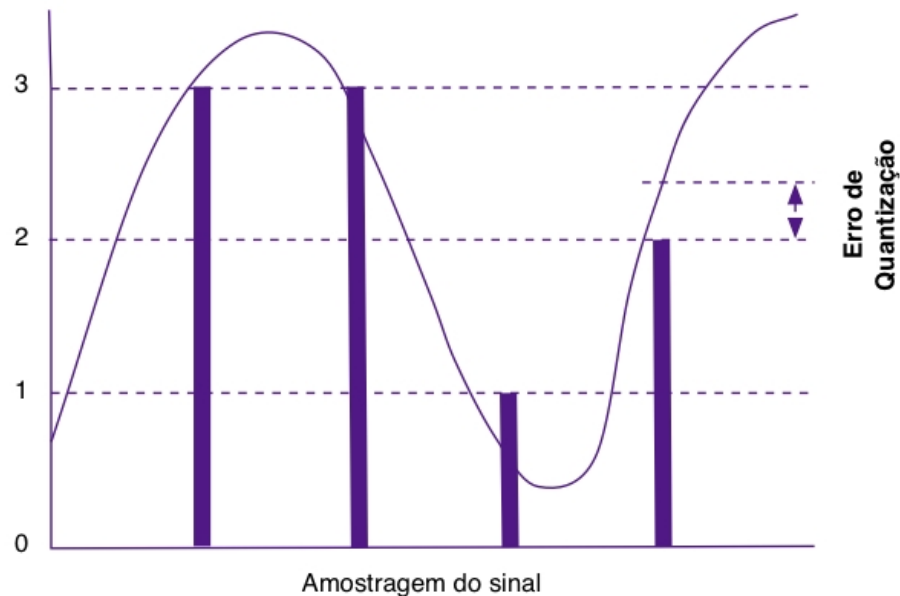
Conversão Analógico-Digital

- Amostragem
 - Sequência de amostras dos sinal analógico



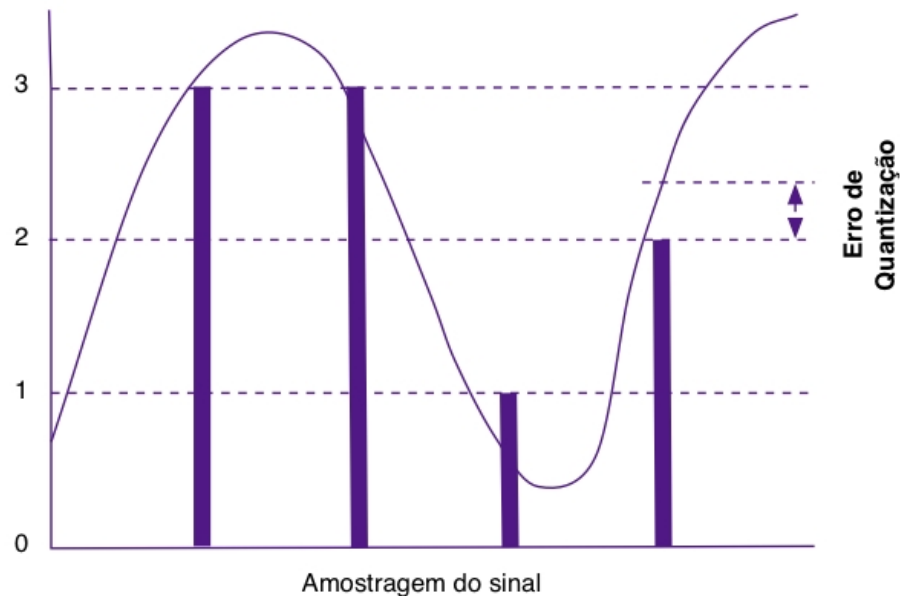
Conversão Analógico-Digital

- Quantização
 - Resolução
 - Número de bits usados para representar cada amostra
 - No exemplo, 2 bits usados para representar 4 níveis



Conversão Analógico-Digital

- Quantização
 - Erro devido ao arredondamento
 - Valor medido é aproximado para o patamar mais próximo na escala de amplitude
 - Diferença em relação ao valor do sinal original



Compressão de Áudio

- Sinal analógico amostrado a uma taxa constante
 - Telefone: 8.000 amostras/segundo
 - CD de música: 44.100 amostras/segundo
 - Cada amostra é quantizada
 - ex., $2^8=256$ possíveis valores discretos
 - Cada valor de quantização é representado por bits
 - 8 bits para 256 valores
 - Exemplo: *Pulse Code Modulation* (PCM)
 - 8.000 amostras/segundo
 - 256 valores para representar uma amostra \rightarrow 8 bits
- } **64 kb/s**

Compressão de Áudio

- Receptor converte o sinal digital novamente para analógico
 - Perda de qualidade
- Exemplo de taxas
 - CD: 1,411 Mb/s
 - MP3 (MPEG 1 *layer* 3): 96, 128, 160 kb/s
 - Telefonia IP: técnicas mais eficientes
 - GSM (13 kb/s), G.729 (8 kb/s) e G.723.3 (6,4 – 5,3 kb/s)

Compressão de Vídeo

- Vídeo
 - É uma sequência de imagens exibidas a uma taxa constante
 - Ex.: PAL-M, 30 quadros por segundo (*frame rate*)
- Imagem digital
 - É uma matriz de pixels
 - Cada pixel é representado por bits
- Compressão pode explorar dois tipos de redundância
 - Espacial
 - Dentro de uma imagem
 - Temporal
 - Entre uma imagem e as posteriores

Compressão de Vídeo

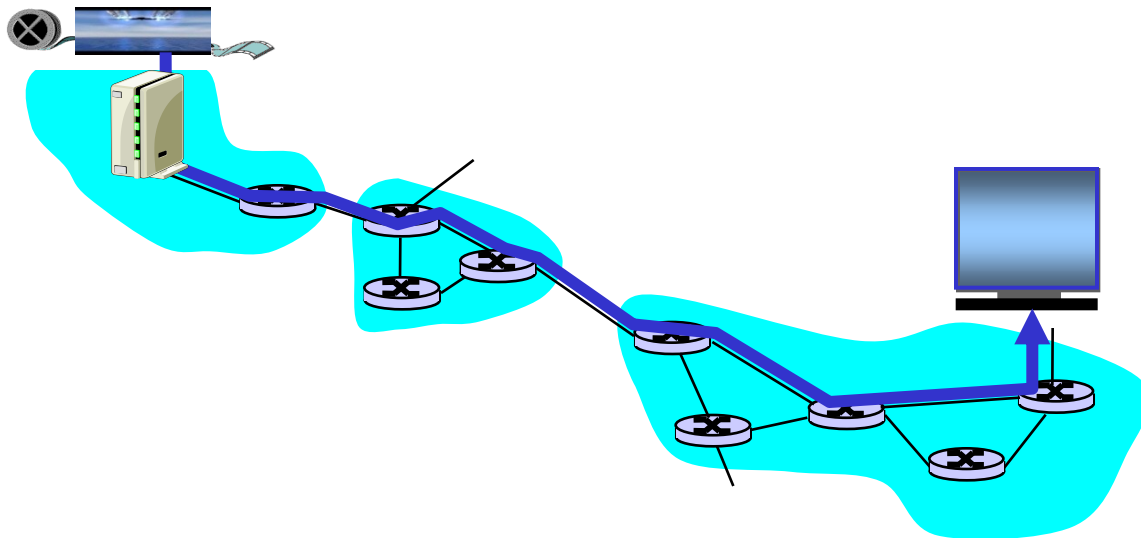
- Exemplos
 - MPEG-1 (CD-ROM) 1.5 Mb/s
 - MPEG-2 (DVD) 3-6 Mb/s
 - MPEG-4 (Internet) < 1 Mb/s
- Vídeo em camadas
 - Adaptar a qualidade do vídeo de acordo com a banda disponível
 - Receptor com mais banda disponível recebe mais camadas

Aplicações Multimídias de Rede

- Classes
 1. Distribuição de mídias armazenadas
 2. Distribuição ao vivo
 3. Distribuição interativa em tempo real

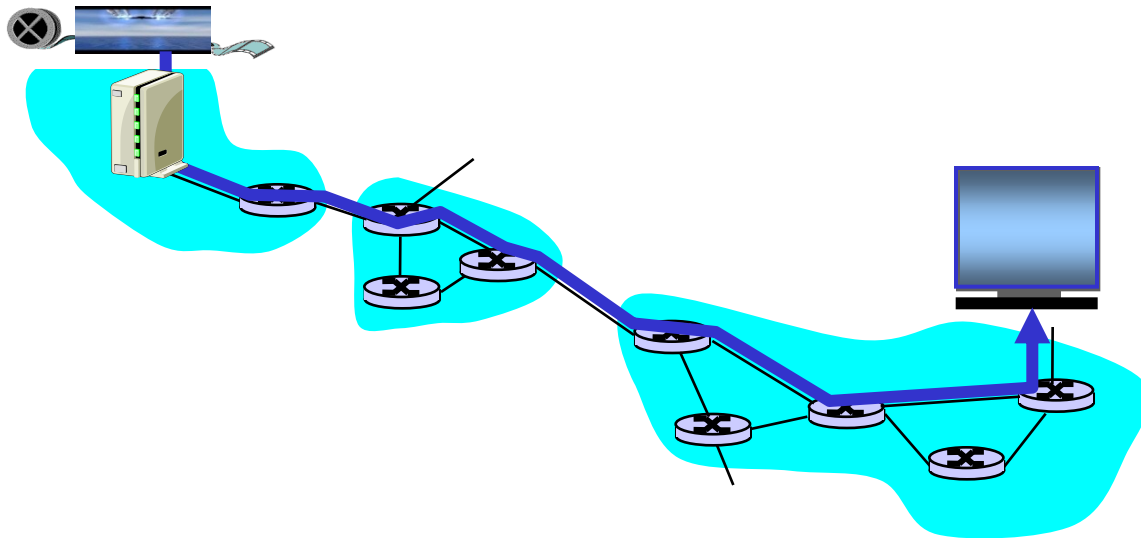
Distribuição de Mídias Armazenadas

- Mídia armazenada na fonte e transmitida para um cliente
- Cliente requisita o vídeo → **vídeo sob demanda**
- Exibição no cliente se inicia antes do recebimento da mídia por completo
 - Isso é distribuição (*streaming*)!

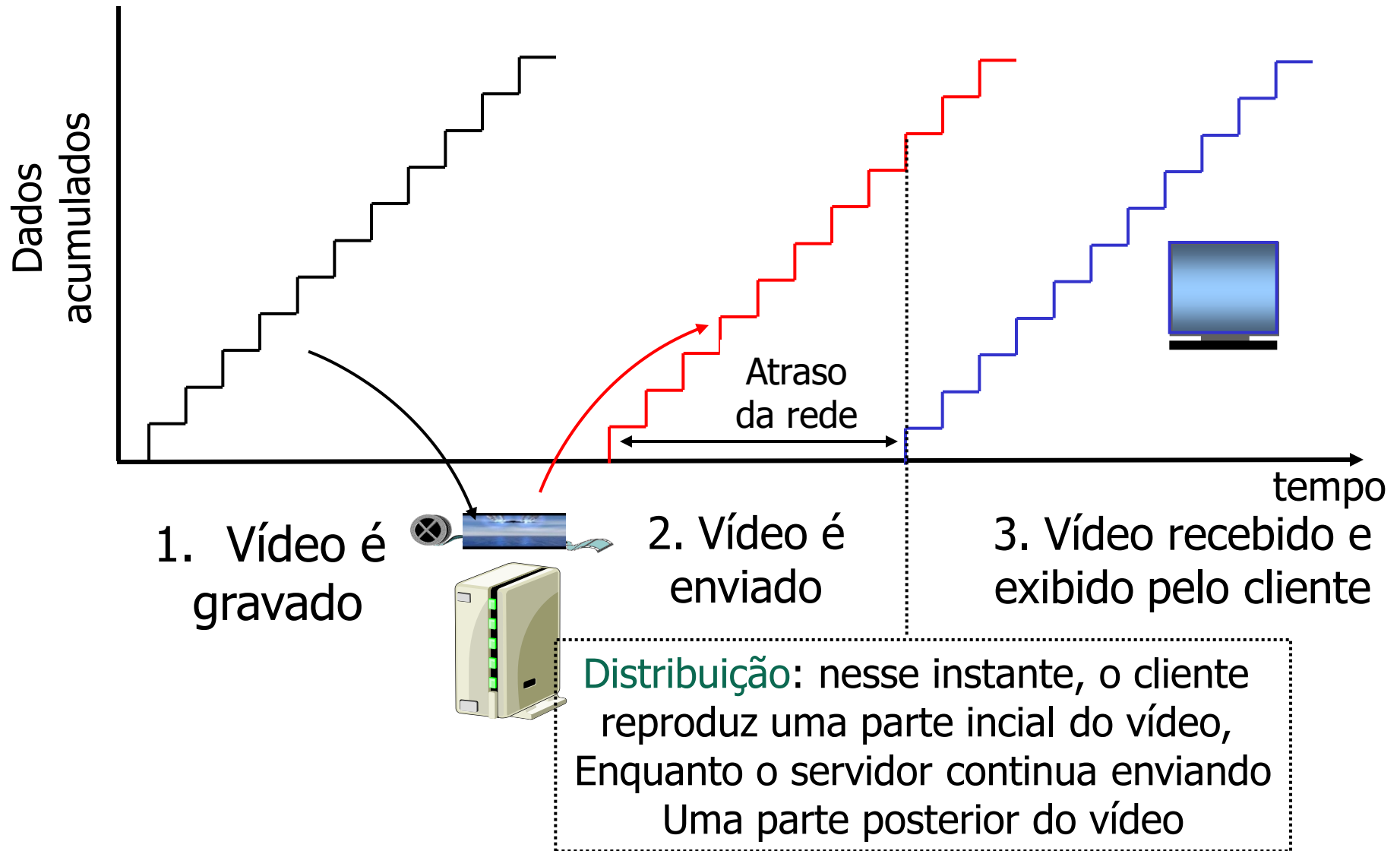


Distribuição de Mídias Armazenadas

- Requisitos de tempo
 - Dados que ainda serão transmitidos devem ser recebidos dentro do tempo útil para serem reproduzidos



Distribuição de Mídias Armazenadas



Distribuição de Mídias Armazenadas: Interatividade

- Funções de um **videocassete**
 - Clientes controlam a reprodução do vídeo
 - Podem
 - **Pausar**
 - **Retroceder**
 - **Avançar** um vídeo
 - Vídeo sob-demanda interativo

Distribuição Ao Vivo

- Semelhante à transmissão de rádio ou TV
 - Participantes recebem o mesmo conteúdo enviado para um grupo de usuários
 - Por isso é chamada de **difusão** (*broadcast*)
 - Em geral, não é possível controlar a reprodução
 - Recepção e reprodução a partir da requisição
- “Ao vivo, mas nem tanto”
 - Conteúdo pode ser previamente gravado

Distribuição Ao Vivo

- Exemplos
 - Programa de entrevistas em uma rádio da Internet
 - Eventos esportivos
- Também é uma distribuição de fluxo contínuo
 - Como ocorre para as mídias armazenadas
 - Características
 - *Buffer* de reprodução
 - Atraso de reprodução da ordem de dezenas de segundo após a transmissão
 - Também possui requisitos de tempo

Distribuição Ao Vivo

- É possível ter interatividade
 - Saltos para trás e pausa
 - Na prática, pouco usados
 - Avanço é impossível

Distribuição em Tempo-Real e Interativa

- Requisitos **estritos de tempo**
 - Atraso fim-a-fim para aplicações de áudio
 - < 150 ms é bom, < 400 ms OK
 - Inclui os atrasos das camadas de aplicação (construção dos pacotes) e rede
 - Atrasos maiores são perceptíveis e degradam a interatividade
- Exemplos
 - Telefonia IP
 - Videoconferência
 - Mundos interativos distribuídos, etc.

- Distribuição de conteúdo sob demanda
 - Participantes selecionam o conteúdo que desejam receber
 - Controlam a reprodução do conteúdo
 - Videocassete
- Distribuição de conteúdo ao vivo (difusão)
 - Participantes recebem o mesmo conteúdo enviado para um grupo de usuários
 - Em geral, não é possível controlar a reprodução
 - Transmissão de TV
- Distribuição de conteúdo em tempo real e interativa
 - Requisitos **estritos de tempo**

- Comutação de pacotes
 - Não há conexão nem reserva de recursos
 - Alocação de acordo com a disponibilidade da rede
- Encaminhamento com base no modelo de **melhor esforço** oferecido pelo IP
 - Pacotes tratados da “melhor” maneira possível
 - Sem diferenciação entres os pacotes
 - Sem garantias de atraso e entrega de pacotes
- Como garantir os requisitos de QoS das aplicações multimídias?

Multimídia na Internet Atual

- Comutação de pacotes
 - Não há conexão nem reserva de recursos
 - Alocação de acordo com a disponibilidade da rede

- Encaminhamento com base no modelo oferecido pelo IP

- Pacote

-

-

Em geral, as aplicações implementam mecanismos na camada de aplicação para reduzir os efeitos das perdas e dos atrasos de pacotes

- Atraso e entrega de pacotes

- Como garantir os requisitos de QoS das aplicações multimídias?

- Filosofia “deixa como está”
 - Sem grandes modificações nas extremidades e no núcleo da rede
 - Aumentar a capacidade dos enlaces quando necessário
 - Desenvolvimento de mecanismos na camada de aplicação
 - Redes de distribuição de conteúdo (*Content distribution Networks* – CDNs)
 - *Multicast* aplicativo
 - Sistemas par-a-par (*peer-to-peer* – P2P)

- Arquiteturas de serviços integrados
 - *Integrated Services* (IntServ)
 - Aplicações podem reservar banda fim-a-fim antes de enviarem conteúdo
 - Mudança fundamental na Internet
 - Exige modificações nos sistemas finais e nos roteadores
 - Novos e complexos programas
- Proposta em 1994
 - RFC 1633

- Arquiteturas de serviços diferenciados
 - *Differentiated Services* (DiffServ)
 - Serviços diferentes para classes de tráfego diferentes
 - Pacotes de aplicações com os mesmos requisitos são agrupados em classes
 - Escalabilidade
 - Também exigem modificações nos sistemas finais e nos roteadores
 - Menos complexas do que as exigidas pelo IntServ
- Proposta em 1998
 - RFC 2475

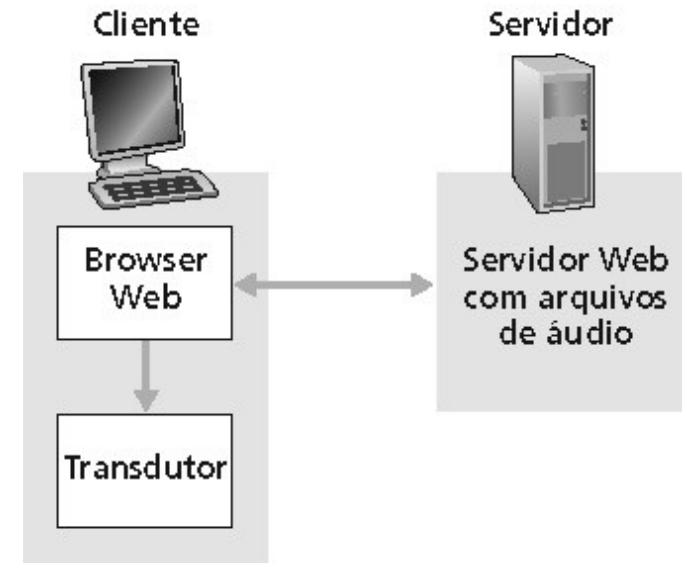
Distribuição Multimídia

Distribuição de Mídia Armazenada

- Mecanismos usados pelas aplicações para aumentar a eficiência da distribuição de conteúdo multimídia
 - *Buffers* no lado do cliente
 - UDP ao invés de TCP
 - Diferentes codificadores, etc.
- Tocador multimídia (*media player*)
 - Descompressão
 - Eliminação de variação de atraso (*jitter*)
 - Tratamento de erros
 - Interface gráfica do usuário com controles para interatividade
 - Etc.

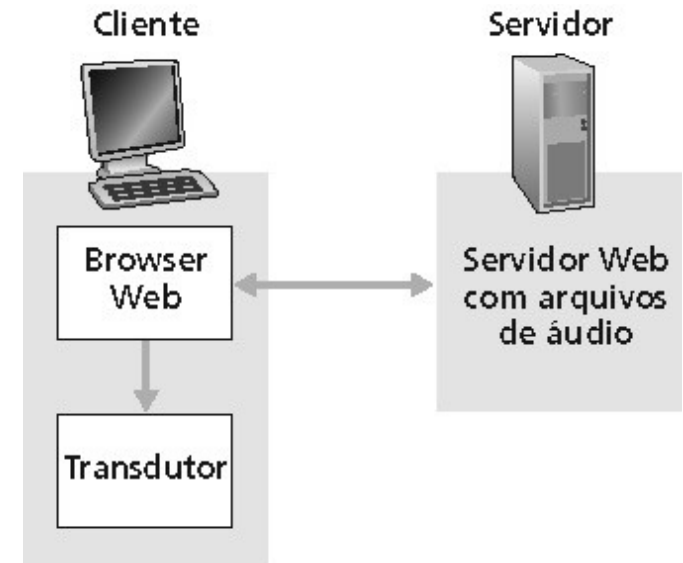
Abordagem Mais Simples

- Receber áudio e vídeo
 - Áudio ou vídeo armazenados em um arquivo
 - Arquivos transferidos como objetos HTTP
 - Recebidos por **inteiro** no cliente
 - Só depois são passados para o tocador (transdutor)



Abordagem Mais Simples

- Receber áudio e vídeo
 - Áudio ou vídeo armazenados em um arquivo
 - Arquivos transferidos como objetos HTTP
 - Recebidos por **inteiro** no cliente
 - Só depois são passados para o tocador (transdutor)

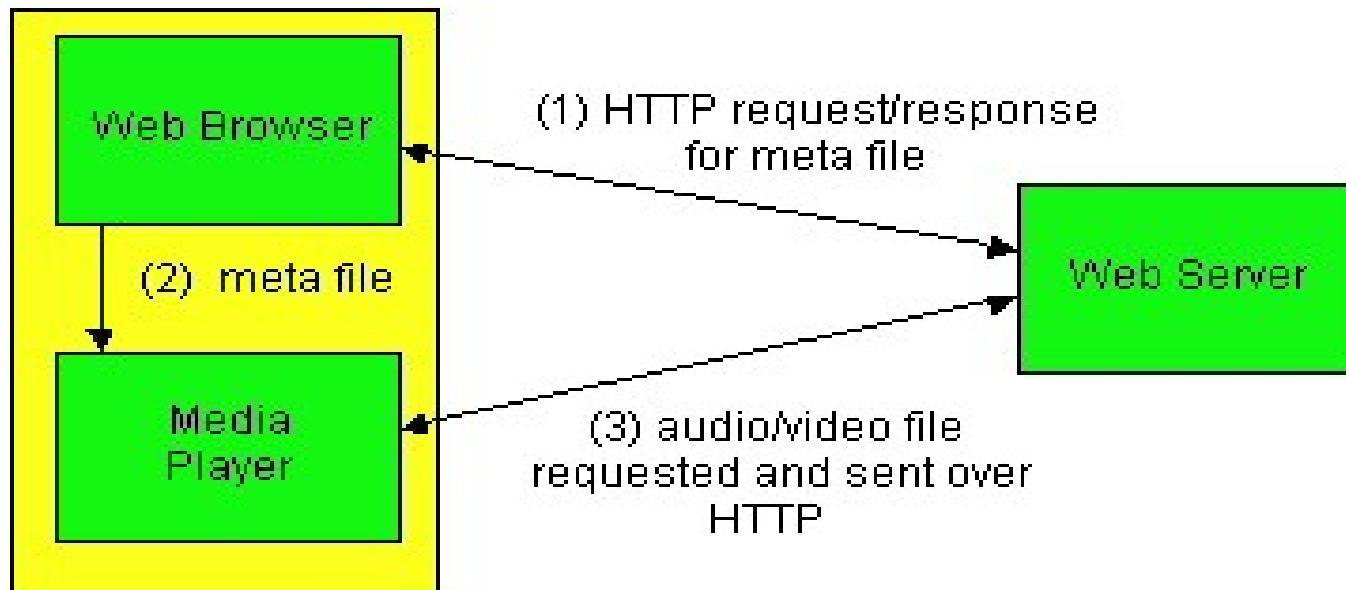


Não é uma distribuição de fluxo contínuo!

Não há cadeia de montagem (*pipelining*) → longos atrasos até a reprodução

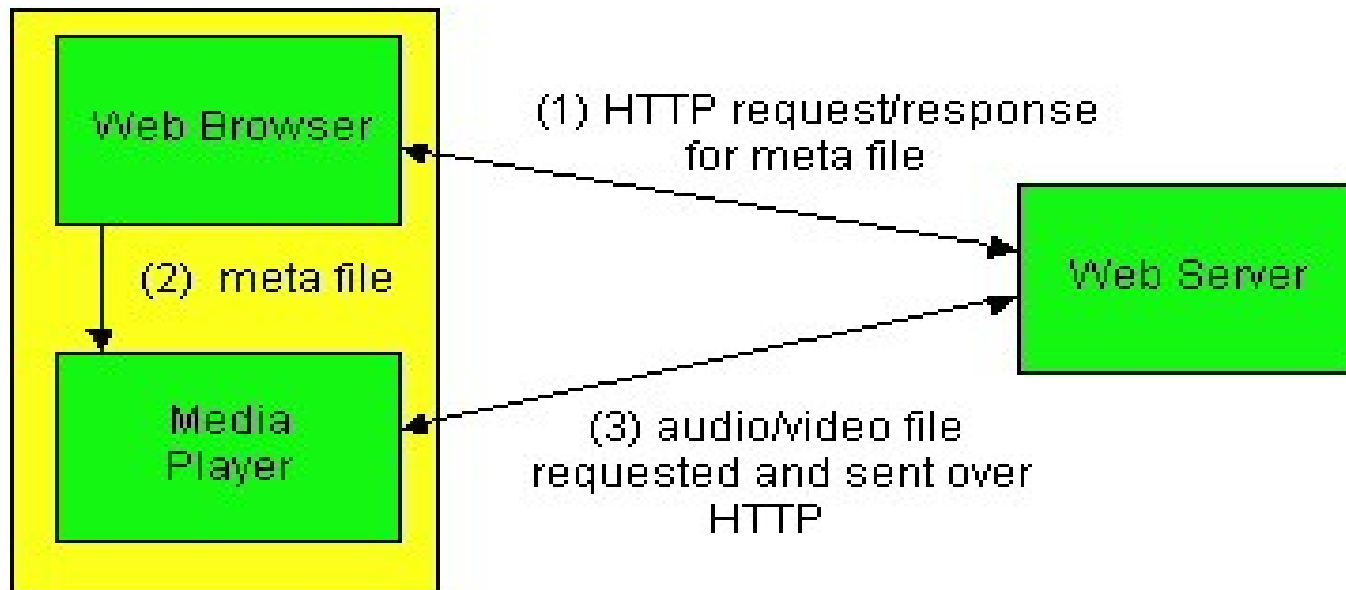
Distribuição de Mídia Armazenada

- Abordagem usando fluxos contínuos (*streams*)
 1. Navegador solicita metarquivo
 2. Navegador inicia o tocador encaminhando o metarquivo



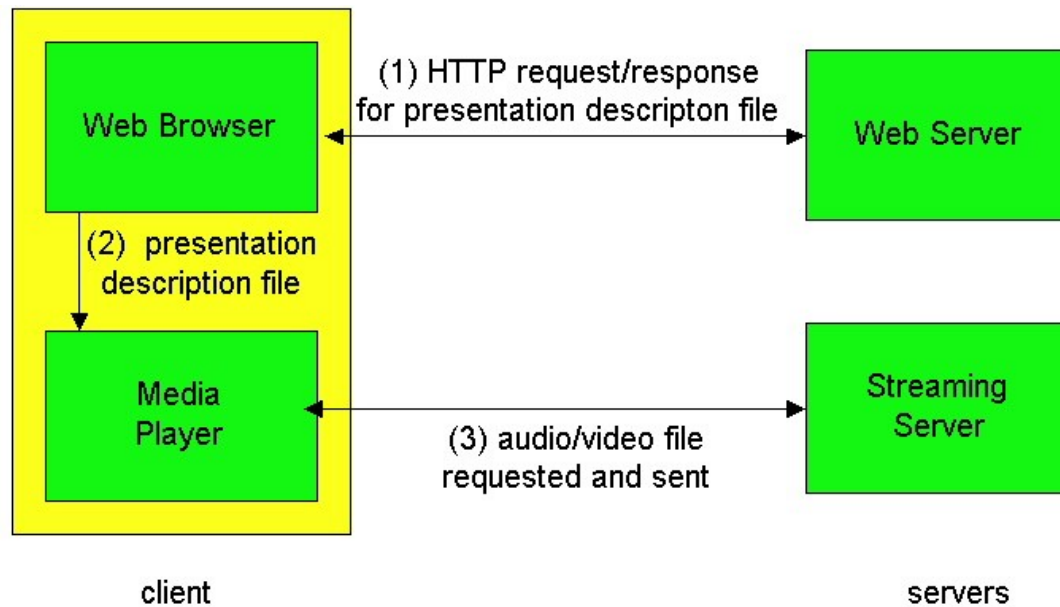
Distribuição de Mídia Armazenada

- Abordagem usando fluxos contínuos (*streams*)
 3. Tocador contata o servidor
 4. Servidor cria o fluxo de áudio/vídeo e o distribui para o tocador

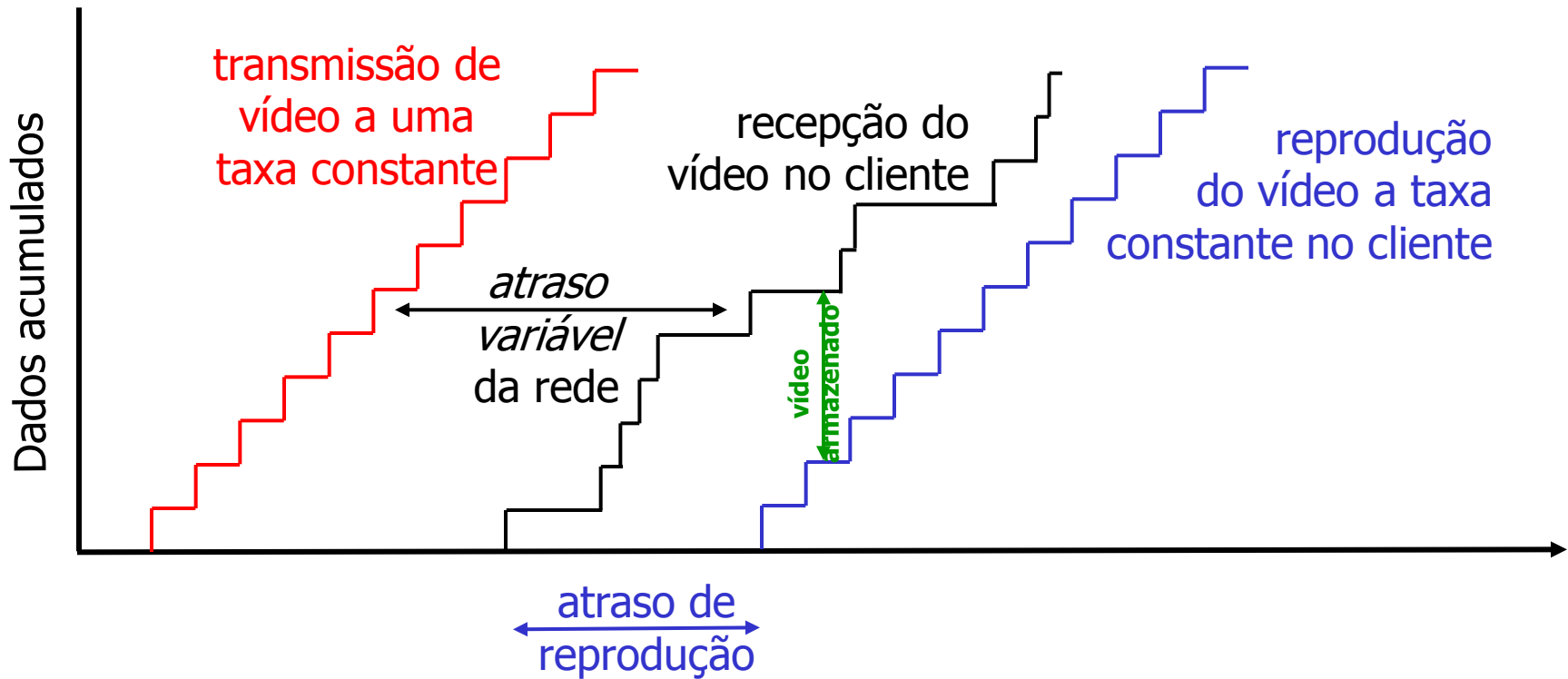


Distribuição de Mídia Armazenada

- Uso de um servidor de fluxos contínuos (*streaming server*)
 - Dedicado
 - Permite o uso de protocolos não-HTTP entre o servidor e o tocador
 - Também pode usar UDP ao invés do TCP

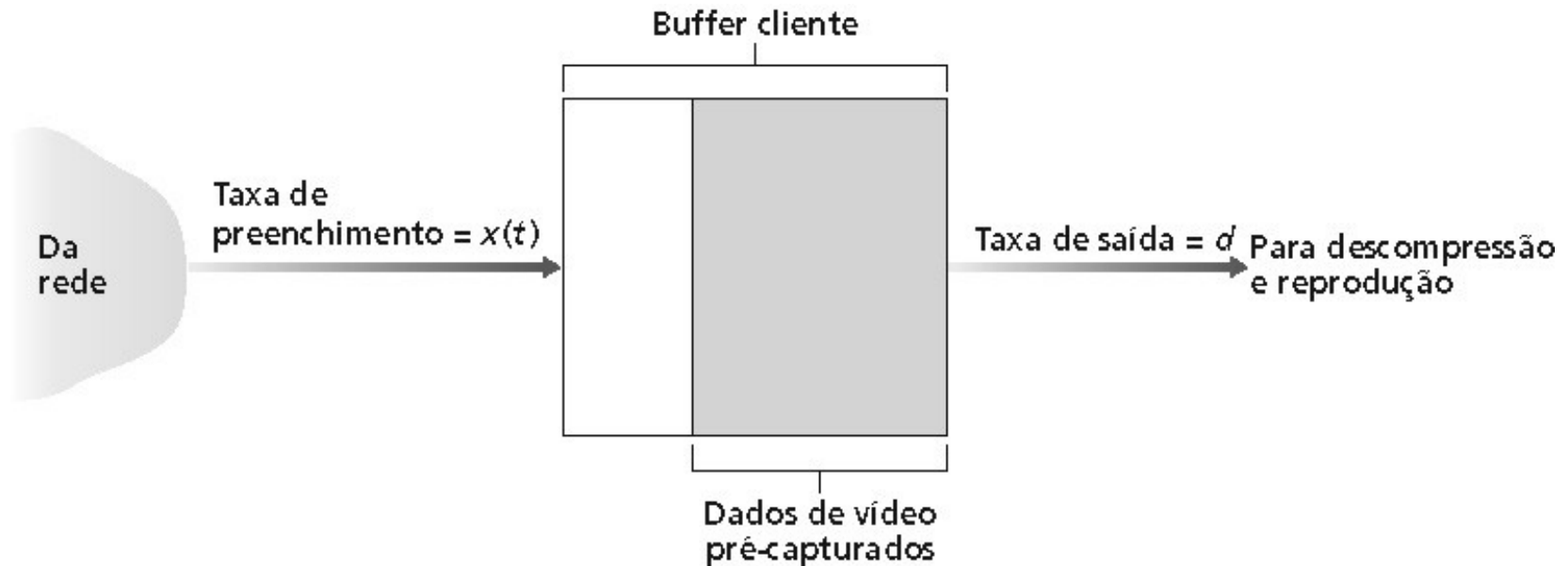


Armazenamento no Cliente



- Armazenamento no lado do cliente, o atraso de reprodução compensa a variação do atraso (*jitter*) provocados pela rede

Armazenamento no Cliente



- Armazenamento no lado do cliente, o atraso de reprodução compensa a variação do atraso (*jitter*) provocados pela rede → **amortecimento**

UDP ou TCP?

- UDP

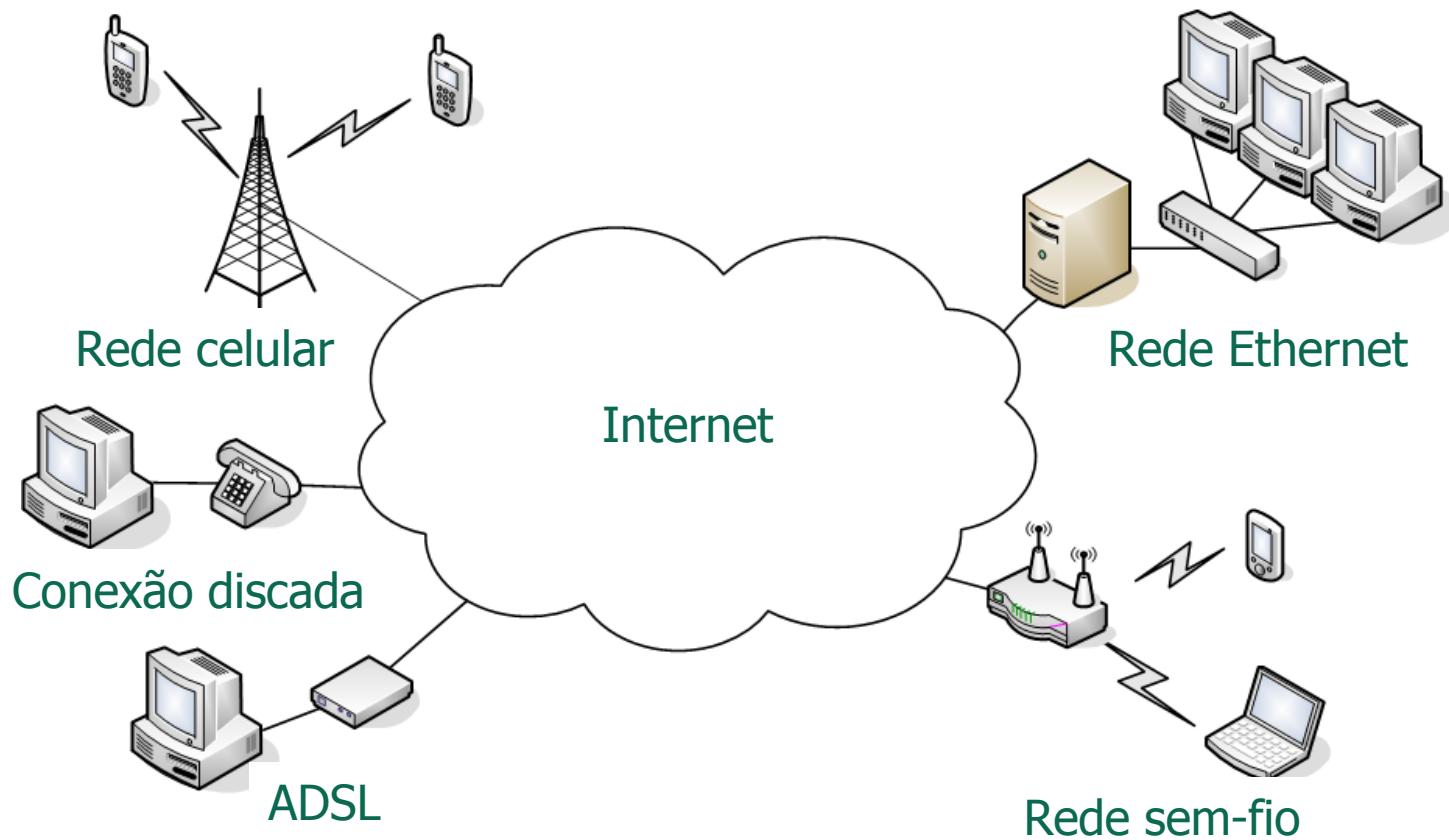
- Servidor envia a uma taxa apropriada para o cliente
 - Sem se importar com o congestionamento da rede
 - Em geral
 - taxa de transmissão = taxa de codificação = constante
 - taxa de preenchimento = taxa constante – perda de pacotes
- Pequeno atraso de reprodução para compensar a variação do atraso da rede
 - 2-5 segundos
- Recuperação de erros
 - Somente quando há tempo disponível

UDP ou TCP?

- TCP
 - Fonte envia na taxa máxima permitida pelo TCP
 - Taxa de preenchimento do *buffer* varia em virtude do controle de congestionamento do TCP
 - Maior atraso de reprodução
 - Taxa de envio do TCP cresce lentamente
 - Comprimento do cabeçalho maior
 - Mais tempo para criar um pacote
 - HTTP/TCP passam mais facilmente por *firewalls*

Heterogeneidade dos Receptores

- Receptores com diferentes capacidades e redes de acesso



Heterogeneidade dos Receptores

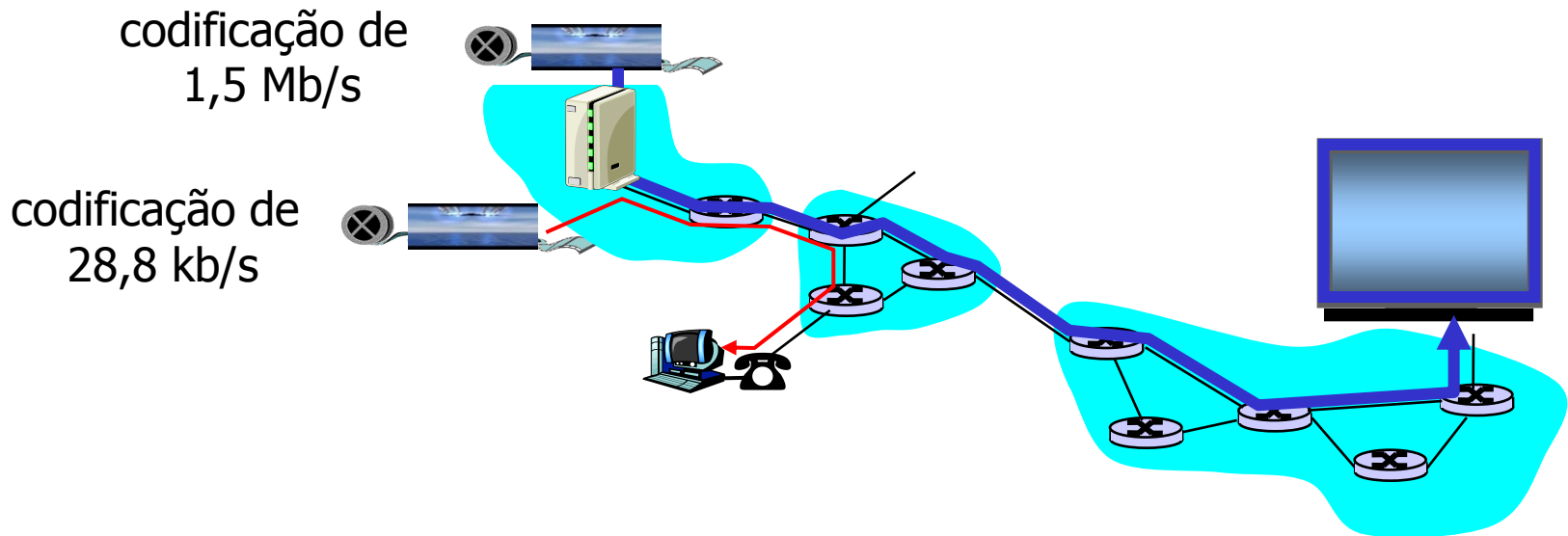
- Diversidade na capacidade dos receptores
 - Banda passante
 - Processamento
- Soluções inadequadas
 - Enviar na taxa ao receptor de maior capacidade
 - Congestionamento
 - Enviar na taxa ao receptor de menor capacidade
 - Insatisfação com a qualidade do vídeo recebido



Técnicas adaptativas para atender os receptores

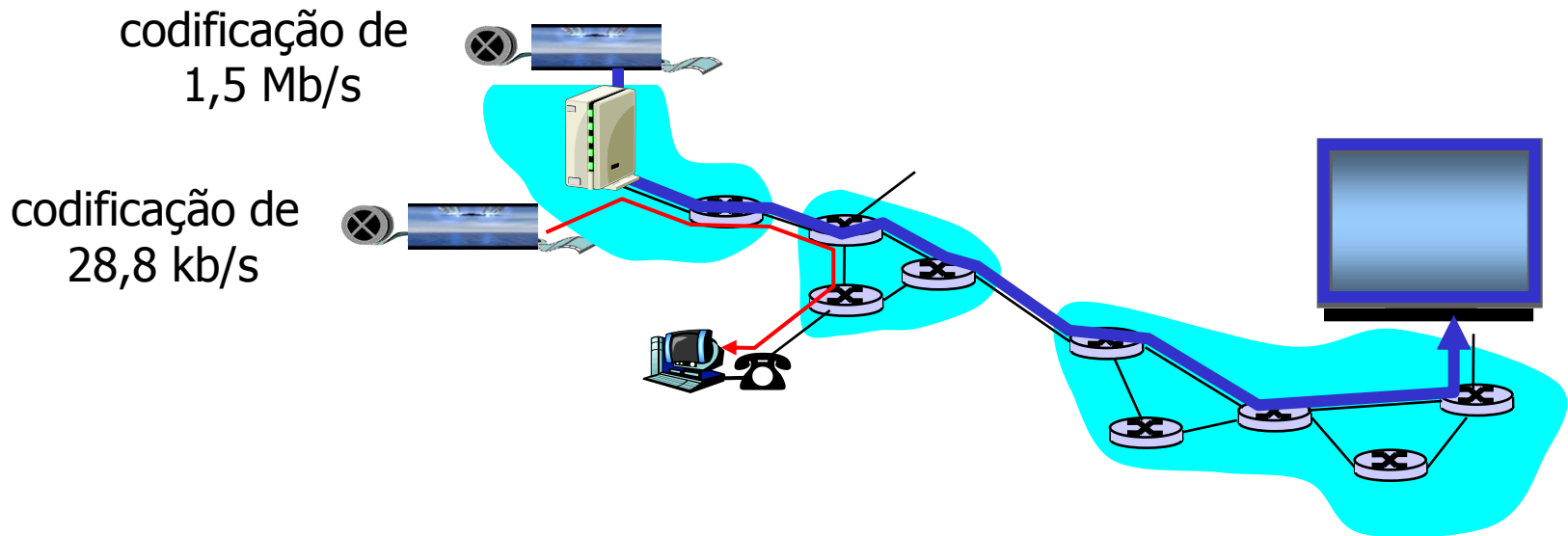
Heterogeneidade dos Receptores

- Como lidar com diferentes capacitações de taxa de recepção do cliente?
 - Exemplo para dois clientes
 - Acesso discado de 28,8 kb/s e Ethernet de 100 Mb/s



Heterogeneidade dos Receptores

- Como lidar com diferentes capacitações de taxa de recepção do cliente?
 - Exemplo para dois clientes
 - Acesso discado de 28,8 kb/s e Ethernet de 100 Mb/s



Servidor armazena e transmite **múltiplas cópias** do vídeo, codificadas em **taxas diferentes**

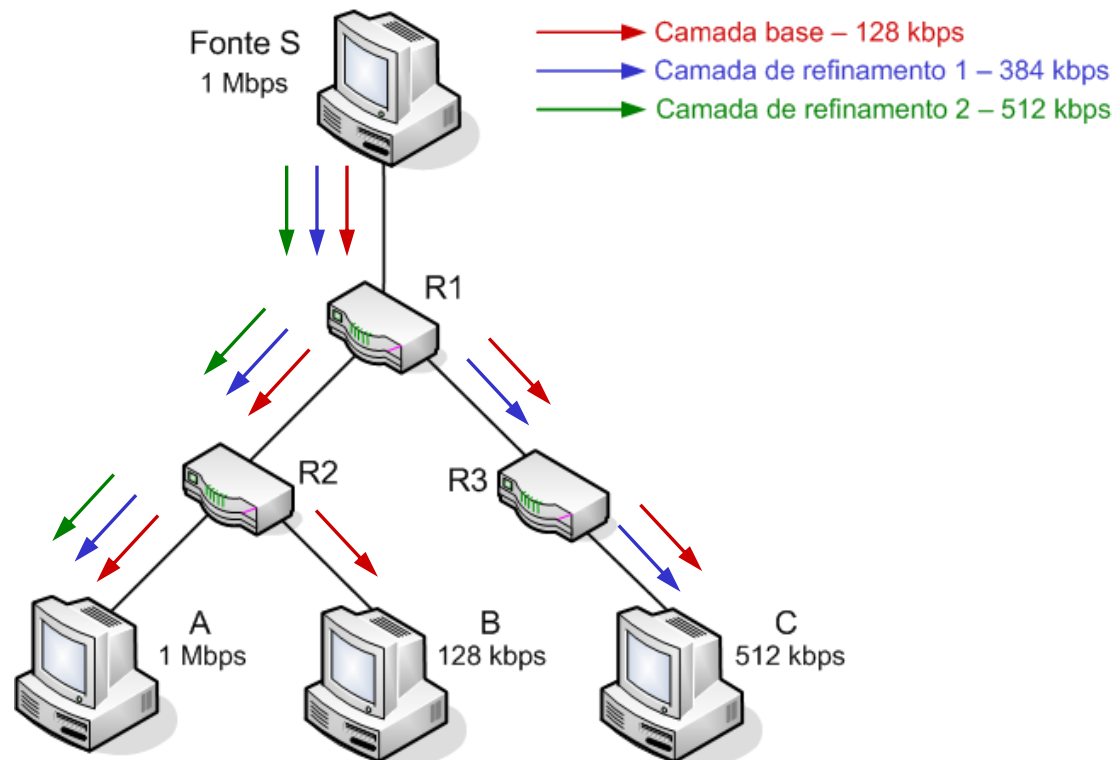
- Lidar com a heterogeneidade dos receptores
- Adaptar a taxa de transmissão do vídeo às condições dos receptores
 - Congestionamentos
 - Receptores com capacidades diferentes
 - Banda passante e processamento
- Classificação
 - Taxa de transmissão
 - Taxa única ou multitaxa
 - Local da adaptação
 - Nos receptores ou no núcleo da rede

Codificação em Camadas

- Vídeo codificado em camadas
 - Hierárquicas e com diferentes prioridades
- Camada base
 - Informações essenciais
 - Sem elas não é possível reconstruir o vídeo
 - Mais prioritária
 - Todo receptor deve ser capaz de recebê-la
- Camadas de refinamento
 - Somadas à camada base
 - Prioridade decrescente
 - Aumentam progressivamente a qualidade do vídeo recebido

Codificação em Camadas

- Cada camada é associada a um endereço de grupo
 - Receptores se associam os grupos
 - Mais grupos → maior qualidade



Codificação em Camadas

- Hierarquia → dificulta a adoção
 - Camada base deve ser entregue sem erros
 - Pacotes tratados diferenciadamente
 - Retransmissão de pacotes perdidos
 - Nem sempre é possível
 - Configuração da rede
 - Atraso tolerado pela aplicação

Múltiplos Descritores (MDC)

- Simplificação da codificação em camadas
 - Não há hierarquia
- Descritores
 - Subfluxos independentes do vídeo
 - Mesma taxa e mesma prioridade
 - Pacotes não exigem diferenciação e nem são retransmitidos
- Cada descritor pode ser decodificado independentemente
 - Um único descritor é suficiente para a reconstrução
 - Facilita a manutenção da continuidade da recepção
 - Mais descritores → maior qualidade

Aplicações de Tempo-Real e Interativas

Aplicações de Tempo-Real e Interativas

- Telefonia PC-para-PC
 - Skype
- Telefonia PC-para-telefone
 - Dialpad
 - Net2phone
 - Skype
- Videoconferência com *webcams*
 - Skype
 - Polycom

Telefonia na Internet

- Para um cenário de telefonia PC-para-PC
 - Áudio do emissor da voz
 - Alterna rajadas (surtos) de voz com períodos de silêncio
 - 64 kb/s durante surto de voz
 - Pacotes gerados apenas durante os surtos de voz
 - Pedacos de 20 ms codificados a 64 kb/s → 60 bytes por pedaço
 - Cabeçalho da camada de aplicação é adicionado a cada pedaço
 - Pedaço+cabeçalho empacotado num segmento UDP
 - Por exemplo, a aplicação envia segmentos UDP a cada 20 ms durante um surto de voz
 - Cadência

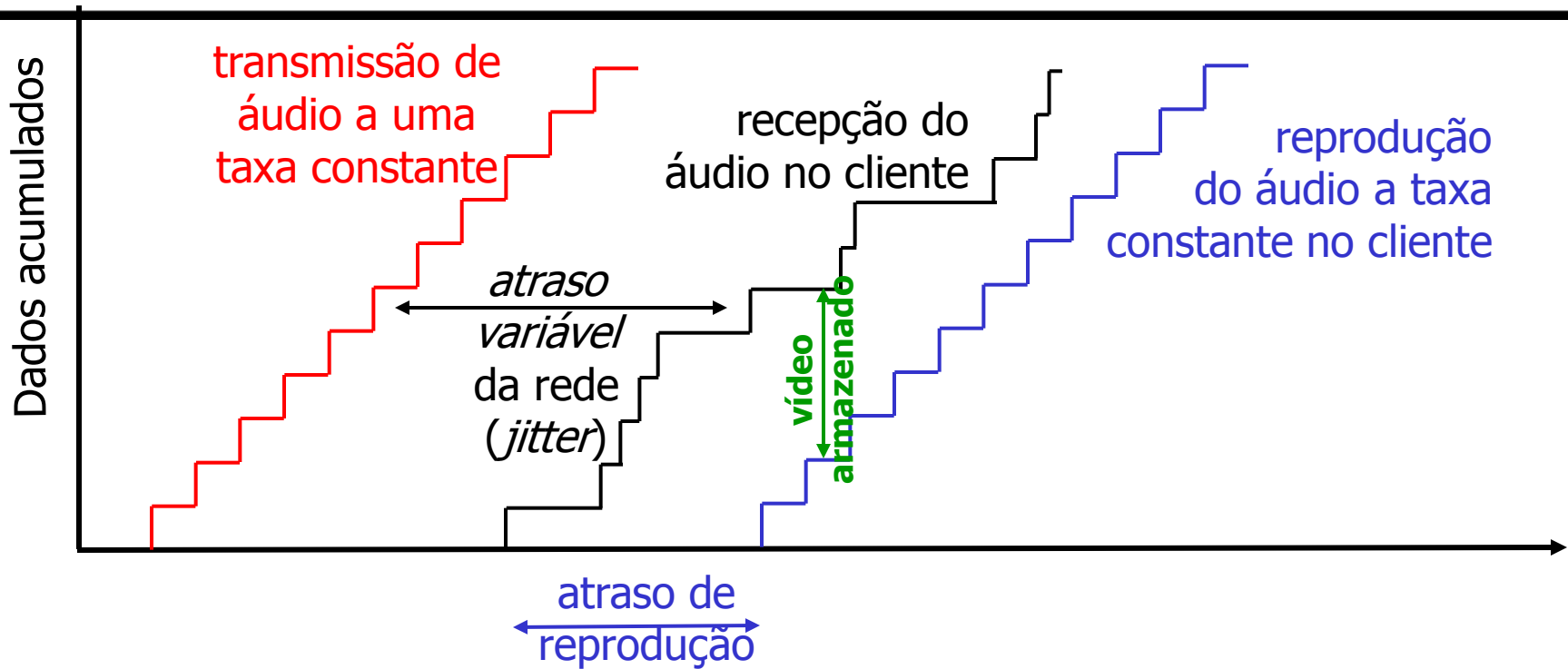
- Perdas e atrasos de pacotes
 - Perda pela rede
 - Datagrama IP perdido devido a congestionamento da rede (estouro do *buffer* do roteador)
 - Perda por atraso
 - O datagrama IP chega muito tarde para ser tocado no receptor
 - Após o tempo de reprodução, no pior caso
 - Atrasos: processamento, enfileiramento na rede; atrasos do sistema terminal (transmissor, receptor)
 - Atraso máximo tolerável típico: 400 ms

Telefonia na Internet

- Perdas e atrasos de pacotes
 - Tolerância a perdas
 - Dependendo da codificação da voz, as perdas podem ser imperceptíveis
 - Taxas de perdas de pacotes entre 1% e 10%

Mecanismos para Compensar Atrasos e Perdas

Variação do Atraso (*jitter*)



- Considere o atraso fim a fim de dois pacotes consecutivos: a diferença pode ser maior ou menor do que 20 ms

Variação do Atraso (*jitter*)

- Como eliminar a variação do atraso no receptor para áudio?
 - Preceder cada pedaço com um **número de seqüência**
 - Incrementado em um para cada pacote gerado
 - Preceder cada pedaço com uma **marca de tempo**
 - Cada pedaço contém a informação do instante em que foi gerado
 - **Atrasar a reprodução** dos pedaços no receptor
 - Suficientemente grande para receber os pacotes antes do tempo de reprodução



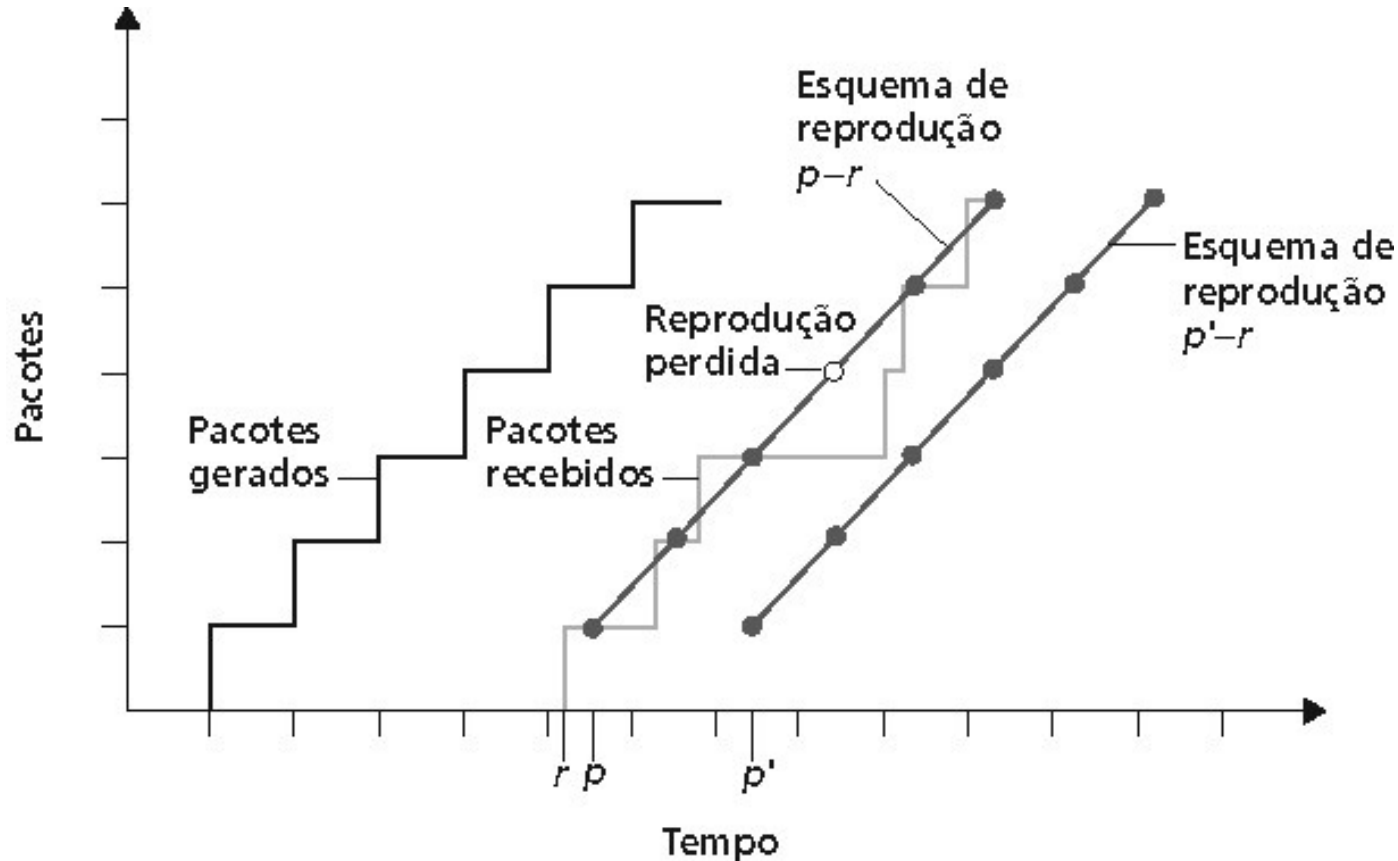
Combinação das três técnicas!

Atraso de Reprodução Fixo

- O receptor tenta reproduzir cada pedaço exatamente q milissegundos após o pedaço ter sido gerado
- Se o pedaço contiver uma marca de tempo t , o receptor reproduzirá o pedaço no instante $t + q$
- Se o pedaço chegar após o instante $t + q$, o receptor o descartará
- Compromissos para q
 - q grande \rightarrow menos perda de pacotes
 - q pequeno \rightarrow menor atraso \rightarrow melhor experiência interativa

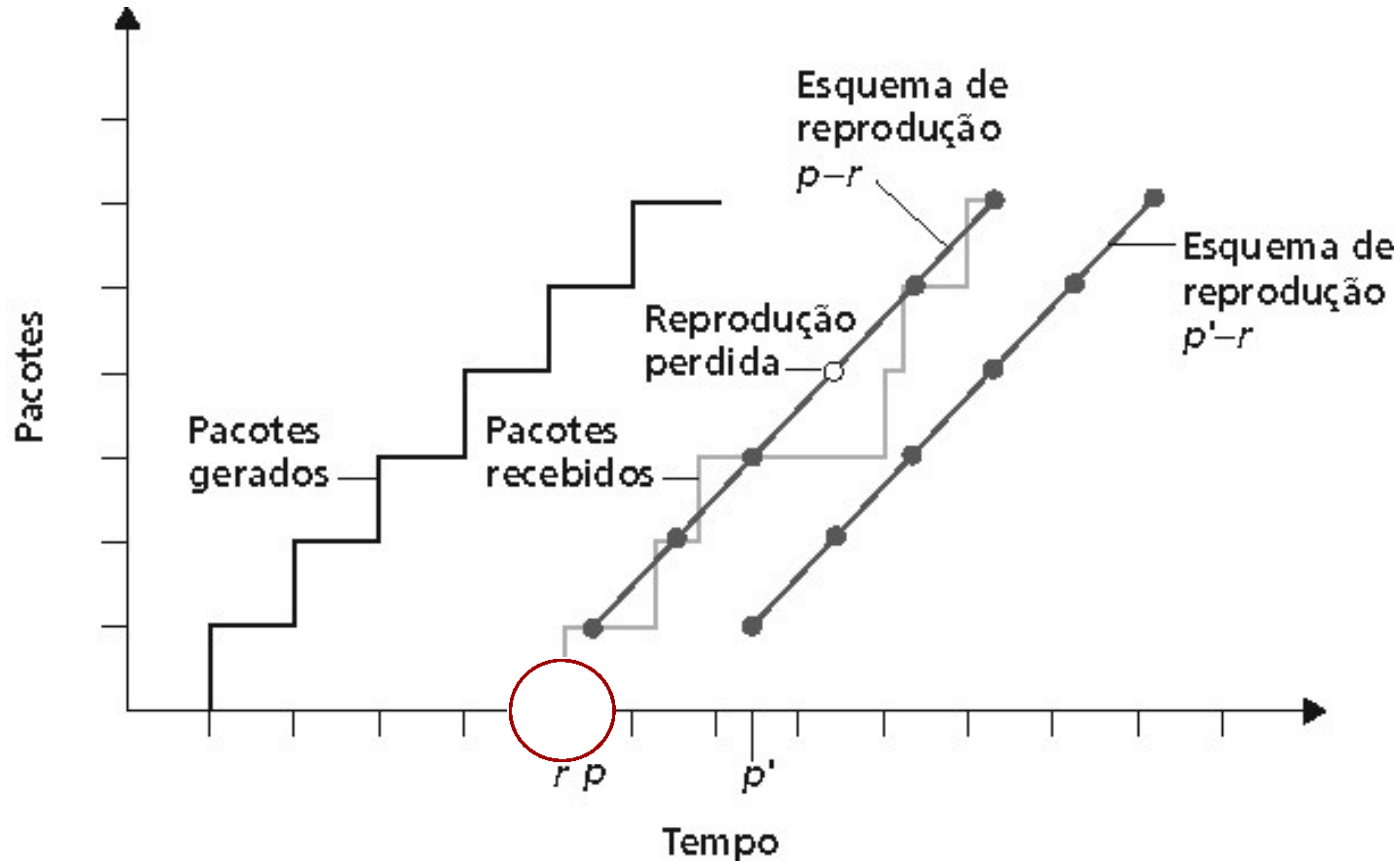
Atraso de Reprodução Fixo

- Transmissor gera pacotes a cada 20 ms durante uma rajada de voz



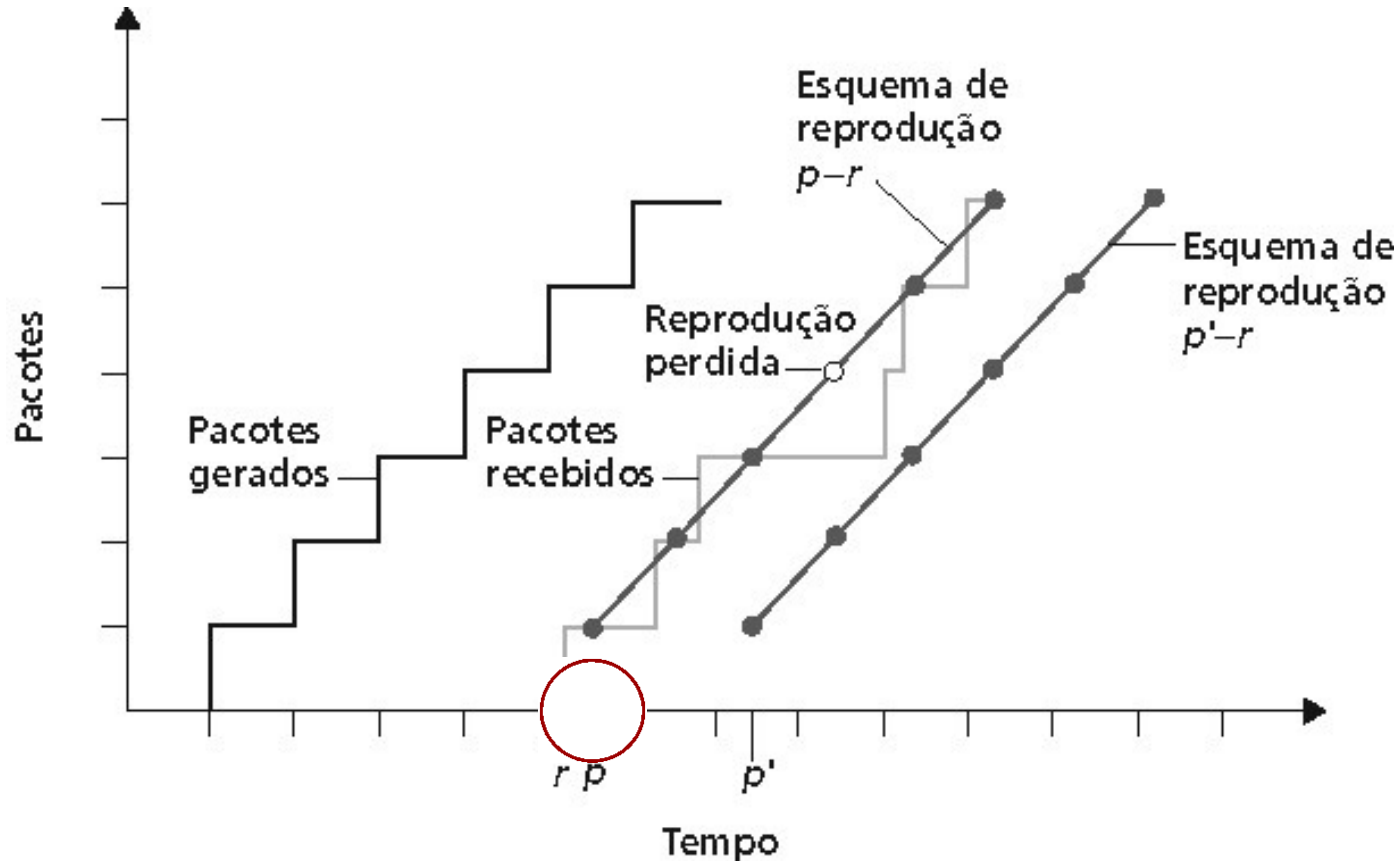
Atraso de Reprodução Fixo

- O primeiro pacote é recebido no instante r



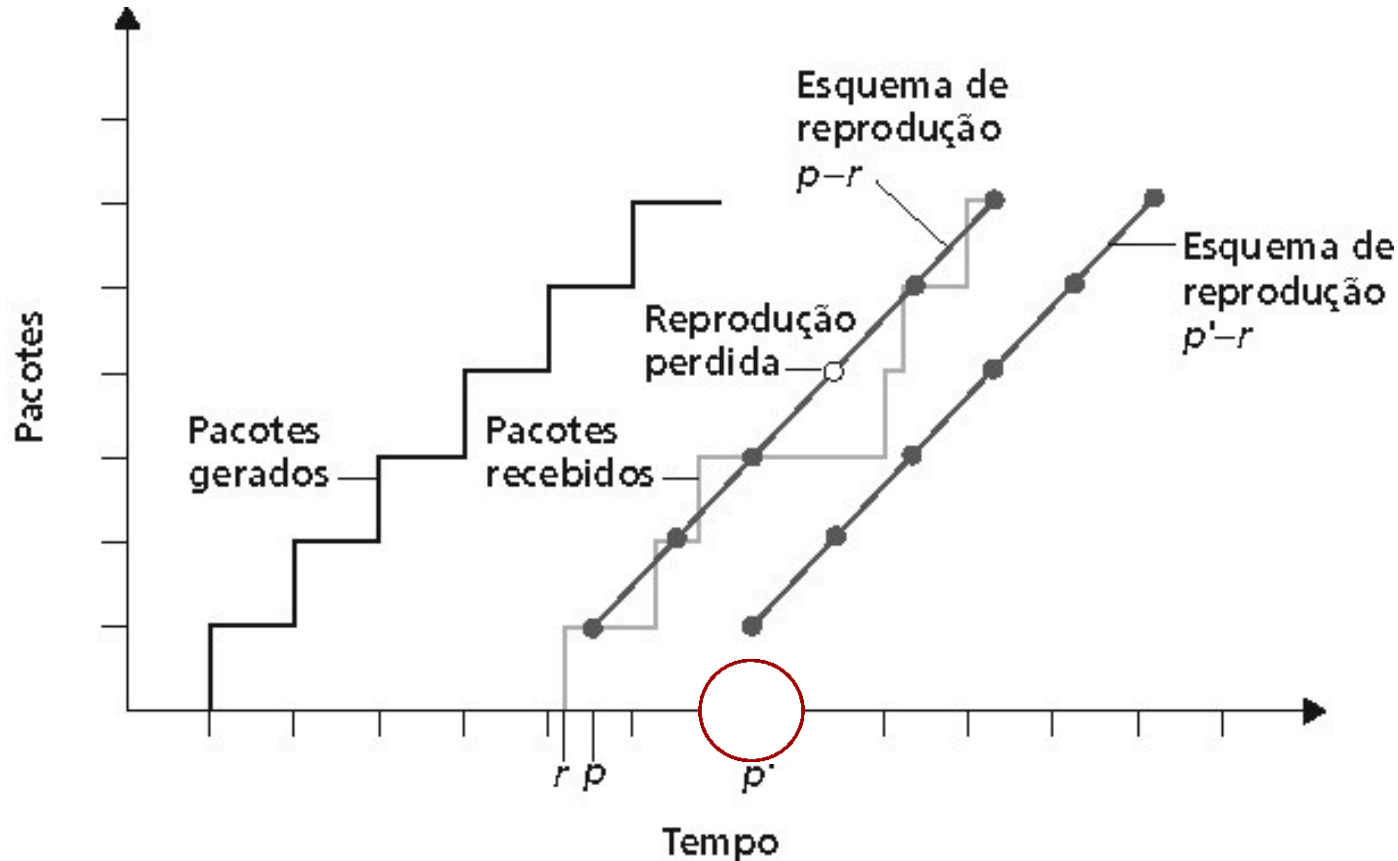
Atraso de Reprodução Fixo

- A primeira reprodução é programada para iniciar no instante p



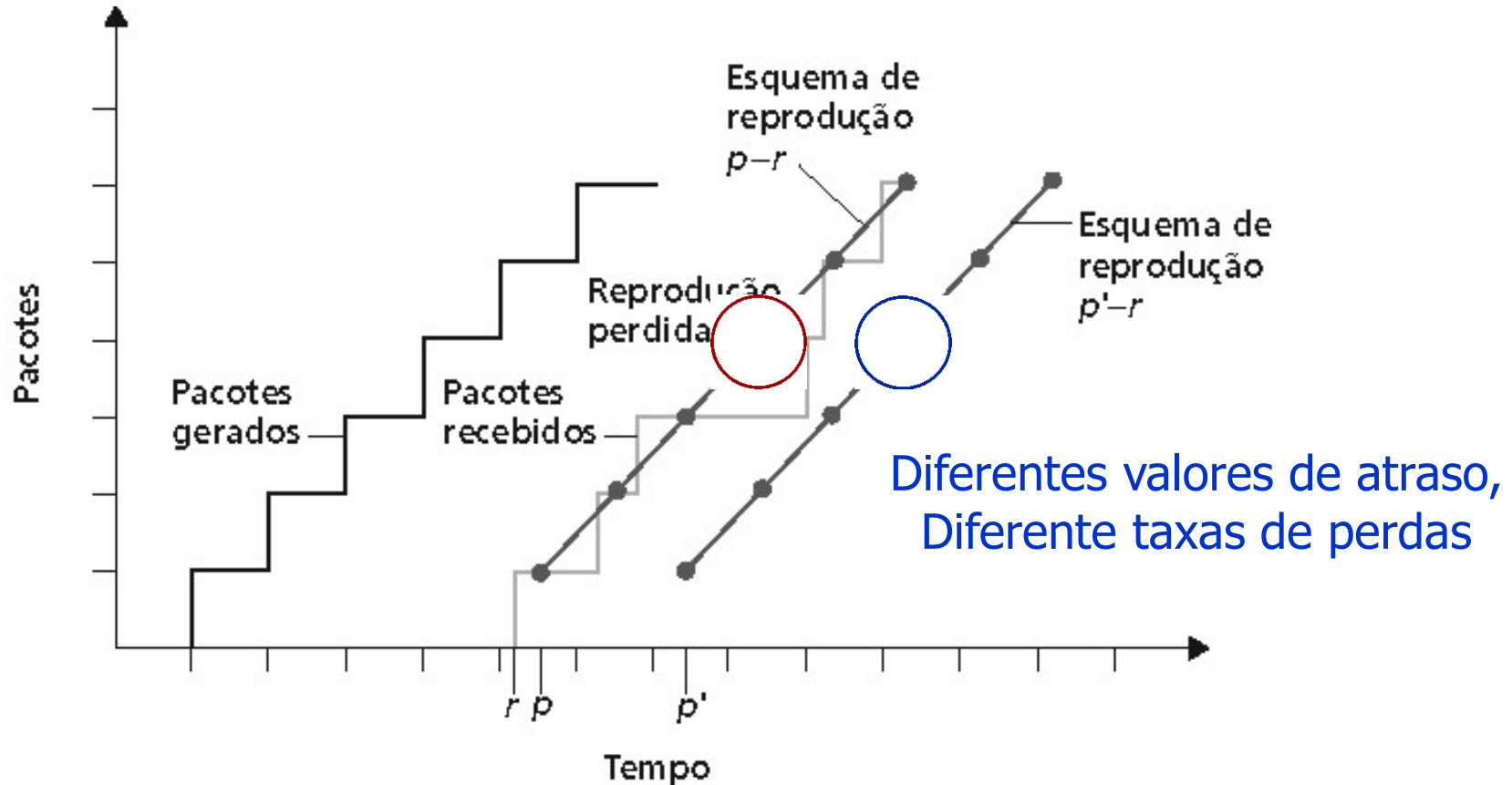
Atraso de Reprodução Fixo

- A segunda reprodução é programada para iniciar no instante p'



Atraso de Reprodução Fixo

- A segunda reprodução é programada para iniciar no instante p'



Atraso de Reprodução Adaptativo

- Objetivos
 - Minimizar o atraso de reprodução
 - Baixa a taxa de perdas

Atraso de Reprodução Adaptativo

- Funcionamento
 - Estima o atraso da rede e ajusta o atraso de reprodução no início de cada rajada de voz
 - Períodos de silêncio são comprimidos e alongados
 - Os pedaços ainda são reproduzidos a cada 20 ms durante uma rajada de voz

t_i = marca de tempo do i - ésimo pacote

r_i = instante em que o pacote i é recebido pelo receptor

p_i = instante em que o pacote i é reproduzido no receptor

$r_i - t_i$ = atraso da rede para o i - ésimo pacote

d_i = estimativa do atraso médio da rede após receber o i - ésimo pacote



u : constante fixa

Atraso de Reprodução Adaptativo

- Também é útil estimar o desvio médio do atraso v_i

$$v_i = (1 - u)v_{i-1} + u |r_i - t_i - d_i|$$

- As estimativas d_i e v_i são calculadas para cada pacote recebido
 - Apesar de serem usadas apenas para determinar o ponto de reprodução para o primeiro pacote de uma rajada de voz

Atraso de Reprodução Adaptativo

- Para o primeiro pacote de uma rajada de voz, o tempo de reprodução é

$$p_i = t_i + d_i + Kv_i$$

- K é um constante positiva
 - Por exemplo, $K = 4$
- Os pacotes restantes em uma rajada de voz são reproduzidos periodicamente

Atraso de Reprodução Adaptativo

- Como o receptor determina se um pacote é o primeiro de uma rajada de voz?
 - Se **não há perdas**, o receptor pode verificar as marcas de tempo sucessivas
 - Diferença entre marcas sucessivas > 20 ms \rightarrow início da rajada
 - Se **há perdas**, o receptor deve verificar as marcas de tempo e os números de seqüência
 - Diferença entre marcas sucessivas > 20 ms e números de seqüência sem “buracos” \rightarrow início da rajada de voz

Recuperação de Perdas

- Em geral, para a distribuição de **conteúdo armazenado**
 - Requisitos de tempo menos estritos do que tempo-real
- Idéia: usar mecanismos de *Forward Error Correction* (FEC)

- Mecanismo OU-EXCLUSIVO simples (*Simple XOR*)
 - Criar um pedaço redundante para cada grupo de n pedaços
 - OU-exclusivo dos n pedaços originais
 - Transmitir $n+1$ pedaços
 - Aumenta-se a largura de banda por um fator de $1/n$
 - É possível reconstruir os n pedaços originais se houver no máximo um pedaço perdido dentre os $n + 1$ pedaços
 - Atraso de reprodução deve ser suficientemente grande para receber $n + 1$ pedaços

Recuperação de Perdas

- Exemplo: Grupo de 3 pedaços (01, 10, 11)
 - $XOR(01,10,11) = 01 XOR 10 XOR 11 = 00$
 - Transmitir todos os pedaços e o resultado do OU-EXCLUSIVO
 - 01, 10, 11 e 00
 - Suponha que o pedaço 10 foi perdido
 - Para recuperá-lo o receptor faz o OU-EXCLUSIVO do que recebeu
 - $XOR(01,00,11) = 01 XOR 00 XOR 11 = 10$
 - Preço pago é aumento taxa de transmissão
 - 6 bits por unidade de tempo \rightarrow 8 bits por unidade de tempo
 - 33% \rightarrow $1/n$, $n=3$ pedaços

Recuperação de Perdas

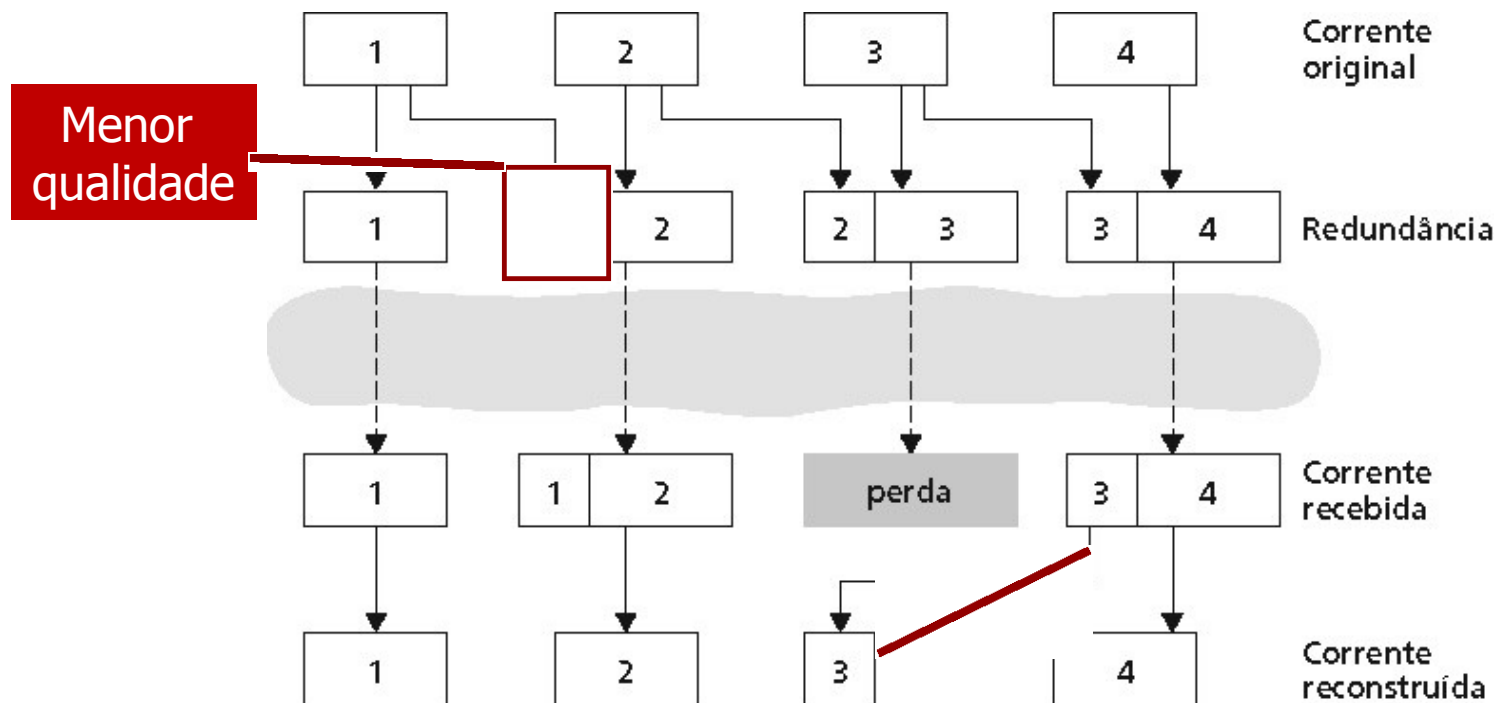
- Compromisso
 - Maior n , menos desperdício de banda
 - Maior n , atraso de reprodução mais longo
 - É preciso esperar todos os pedaços de um grupo para calcular o XOR
 - Maior n , maior probabilidade de que 2 ou mais pedaços sejam perdidos

Recuperação de Perdas

- Esquema de FEC mais avançado
 - Transmissão de “carona” de um fluxo de menor qualidade
 - Enviar um fluxo de áudio de baixa resolução como informação redundante
 - Por exemplo, fluxo nominal PCM a 64 kb/s e fluxo redundante GSM a 13 kb/s

Recuperação de Perdas

- Esquema de FEC mais avançado
 - Transmissão de “carona” de um fluxo de menor qualidade



Recuperação de Perdas

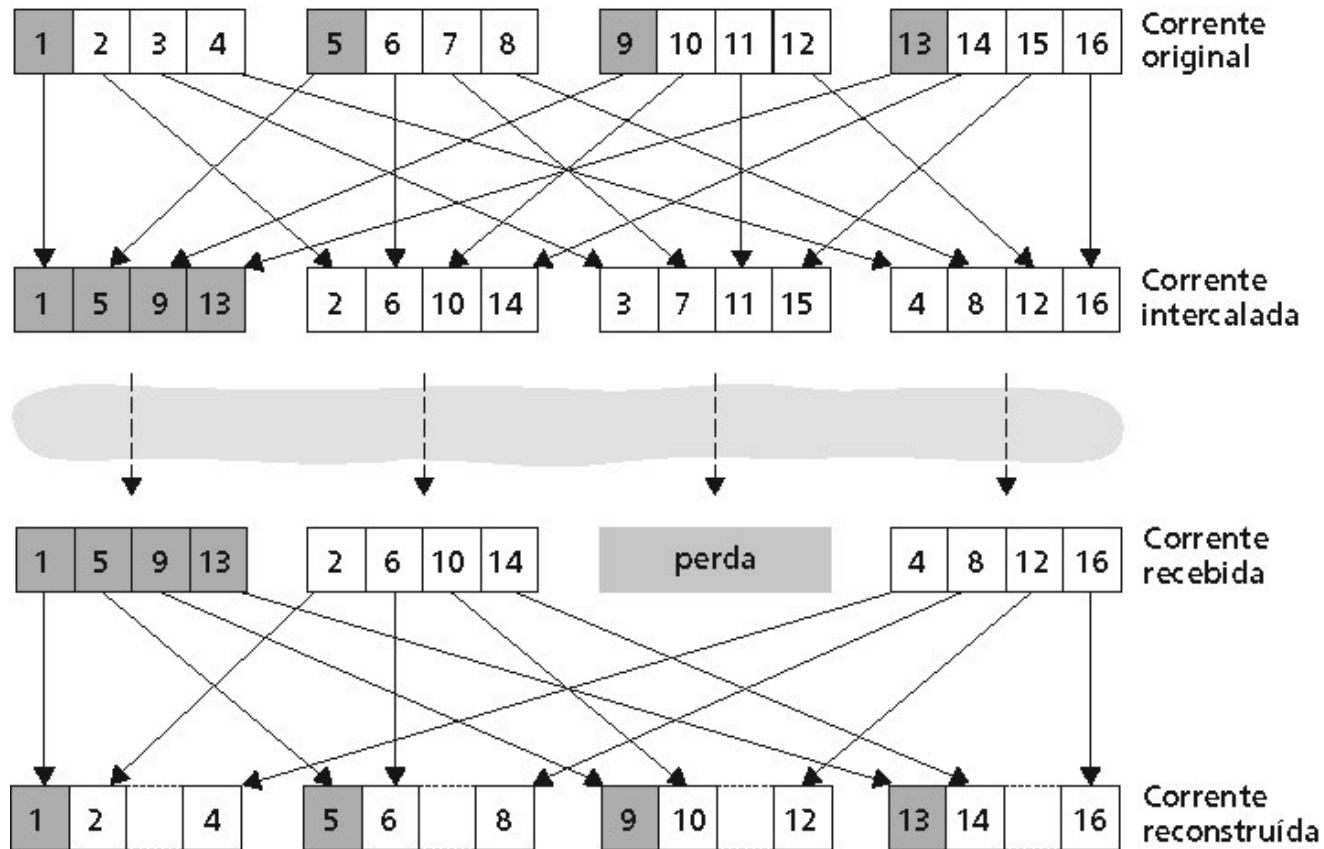
- Esquema de FEC mais avançado
 - Sempre que houver uma perda **não consecutiva**, o receptor poderá recuperar a perda
 - Pode também adicionar o (n-1)-ésimo e a (n-2)-ésimo pedaço de baixa taxa de transmissão

- Esquema de Intercalação (*interleaving*)
 - Pedacos são quebrados em unidades menores
 - Por exemplo, quatro unidades de 5 ms por pedaço
 - Pacote agora contém pequenas unidades de pedacos diferentes
 - Se o pacote se perder, ainda temos muito de cada pedaço
 - Não tem sobrecarga de redundância
 - Mas aumenta o atraso de reprodução

Evitar a perda em rajada!

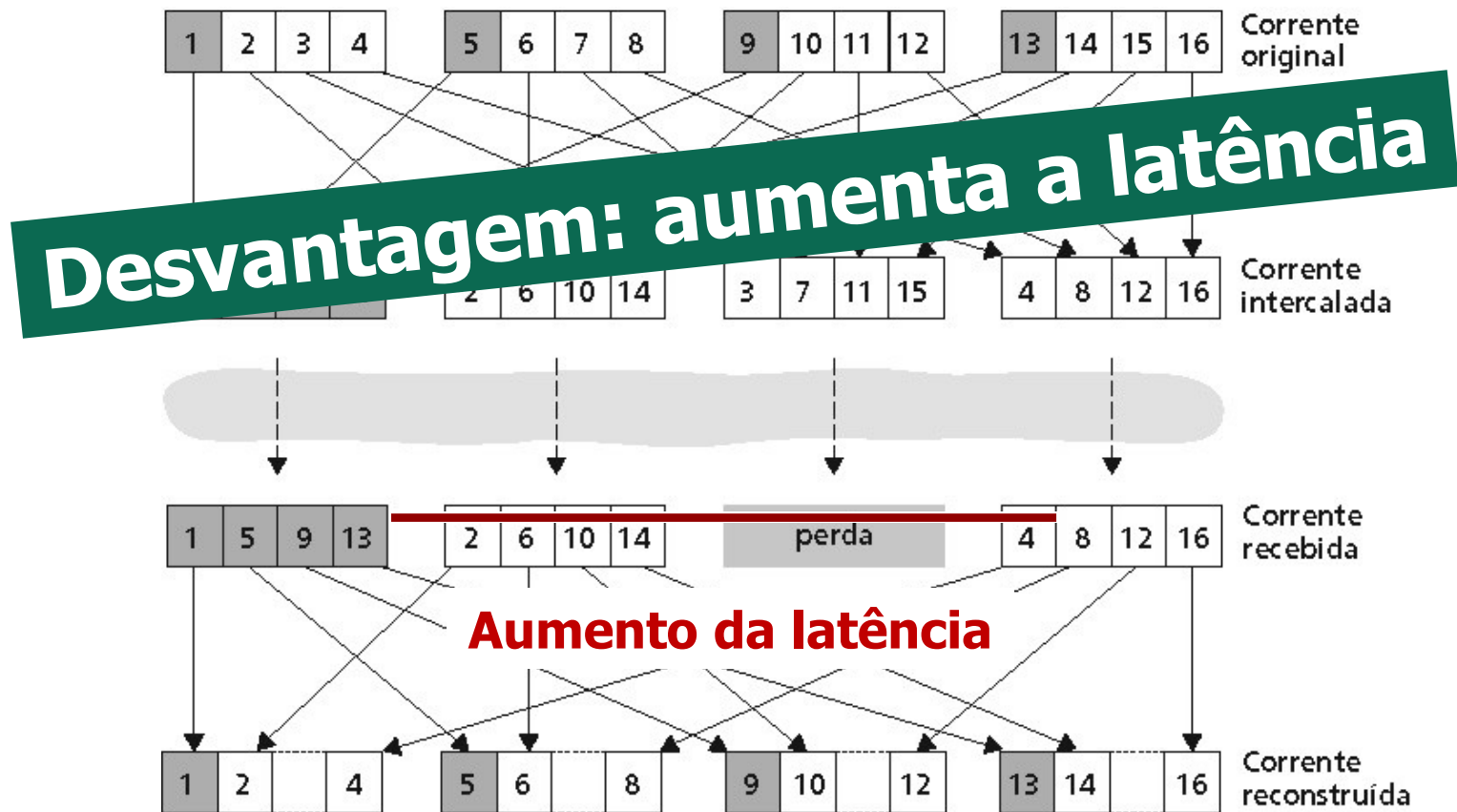
Recuperação de Perdas

- Esquema de Intercalação (*interleaving*)



Recuperação de Perdas

- Esquema de Intercalação (*interleaving*)



Escalabilidade

Comunicação Multidestinatória

- Garantir a escalabilidade
 - Envio de uma mesma informação para um grupo de usuários
 - Difusão de TV
- A fonte envia uma única cópia da informação
 - A rede deve replicar essa informação quando necessário
 - Evitar o consumo desnecessário de banda passante

- Proposto em 1989
- Comunicação multidestinatória na **camada de rede**
- Enviar um datagrama para um endereço IP de grupo
 - Todos os membros do grupo recebem esse datagrama
- Lenta implantação
 - Complexidade de gerenciamento e configuração
 - O grupo é conversação aberta muitos-para-muitos
 - Não há controle de acesso nem um sistema global de alocação de endereços
 - Interferência e colisão

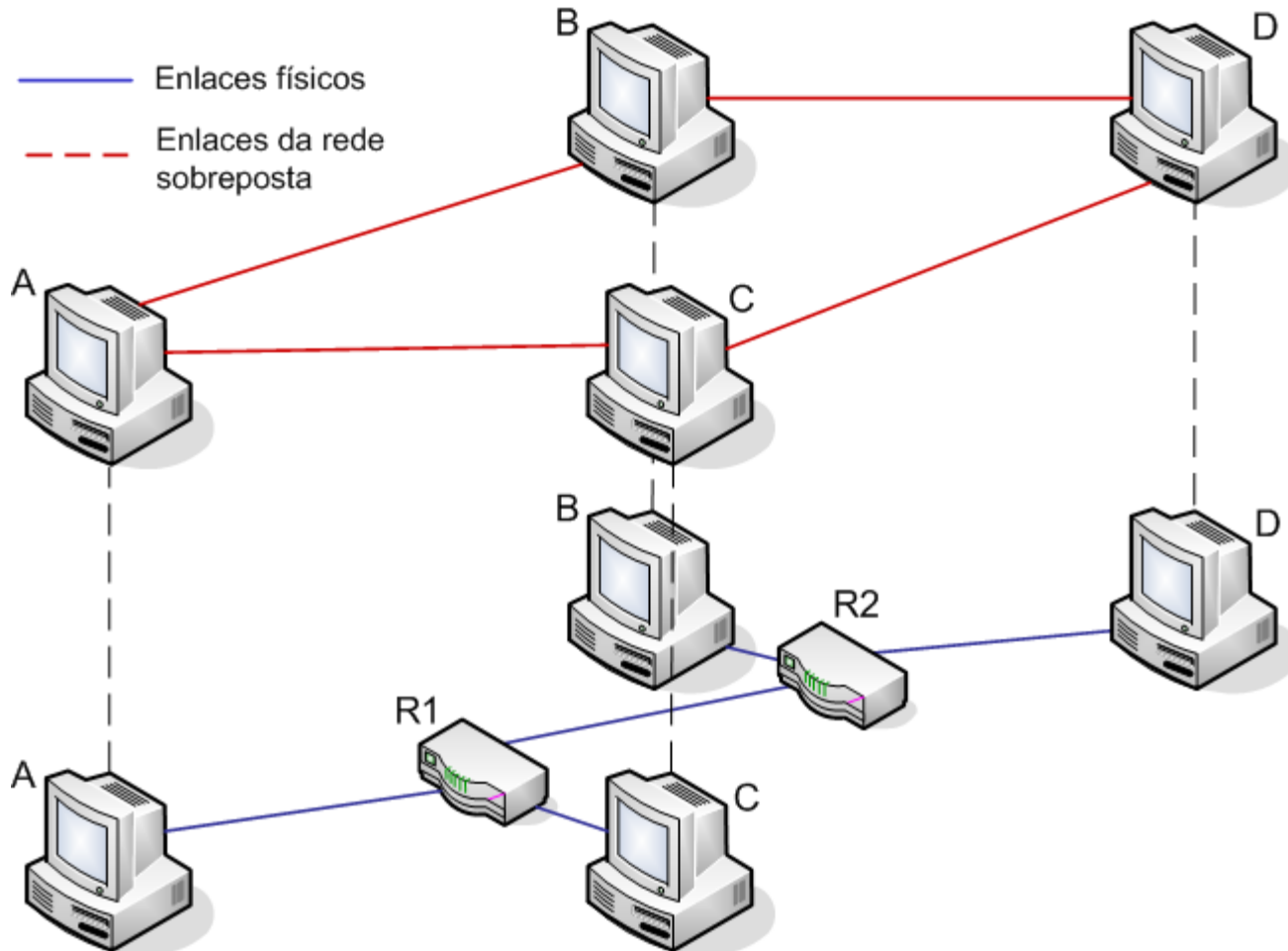
- Proposto em 1989
- Comunicação multidestinatória na **camada de rede**
- Enviar um datagrama para um endereço IP de grupo
 - Todos os membros do grupo recebem esse datagrama
- Lenta implantação
 - Complexidade de configuração

Exige modificações no núcleo da rede!

- Conversação aberta muitos-para-muitos
 - Não há controle de acesso nem um sistema global de alocação de endereços
 - Interferência e colisão

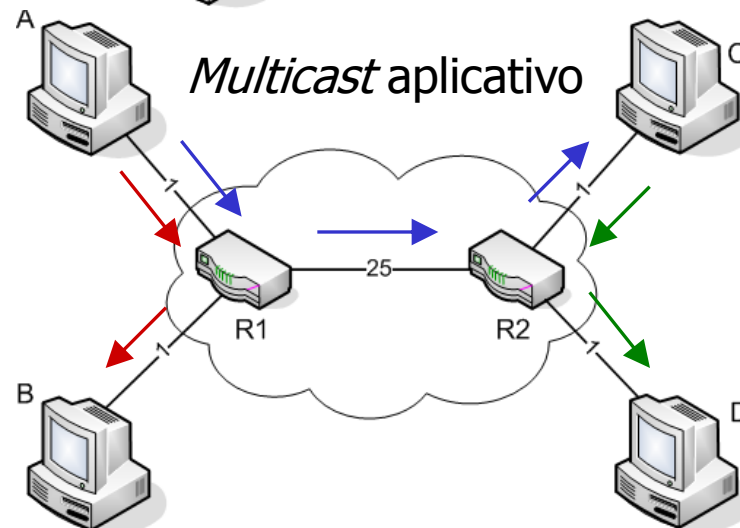
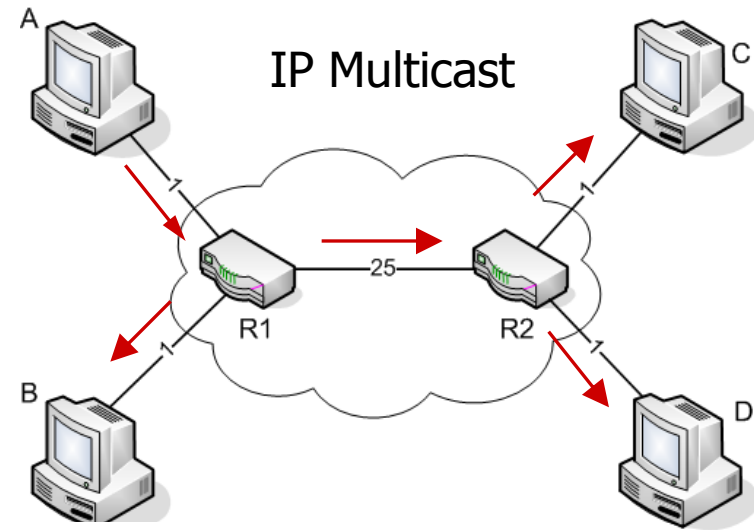
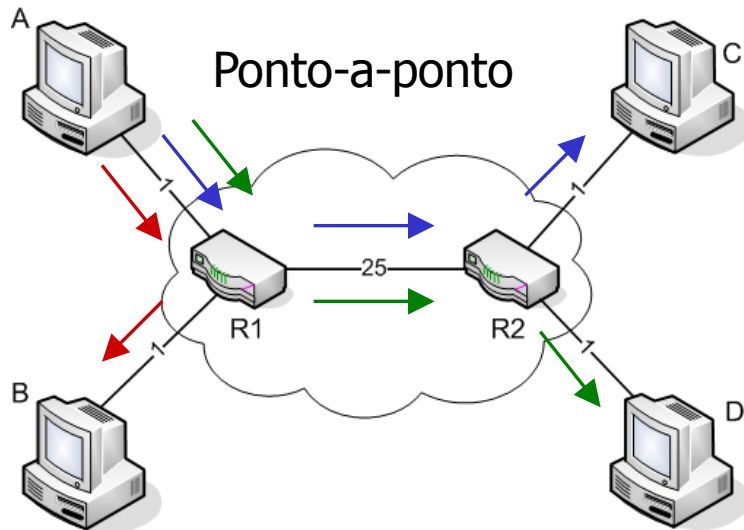
- Comunicação multidestinatória na camada de aplicação
- Os participantes são responsáveis pelo
 - Roteamento
 - Gerenciamento de participantes
 - Replicação da informação
- Existência apenas da comunicação ponto-a-ponto
- Construção de uma rede sobreposta (*overlay*)
 - Enlaces são túneis ponto-a-ponto
 - Não é necessário modificar a infra-estrutura de rede

Multicast Aplicativo



- Duas formas para construir a rede sobreposta
 - Conjunto de servidores e *proxies*
 - Nós âncoras
 - Sistemas par-a-par
 - Não há nós dedicados
 - Compartilhamento de recursos x utilização do serviço
- Desvantagem
 - Desempenho em relação a comunicação multidestinatória na camada de rede
 - Redundância nos enlaces físicos
 - Aumento da latência

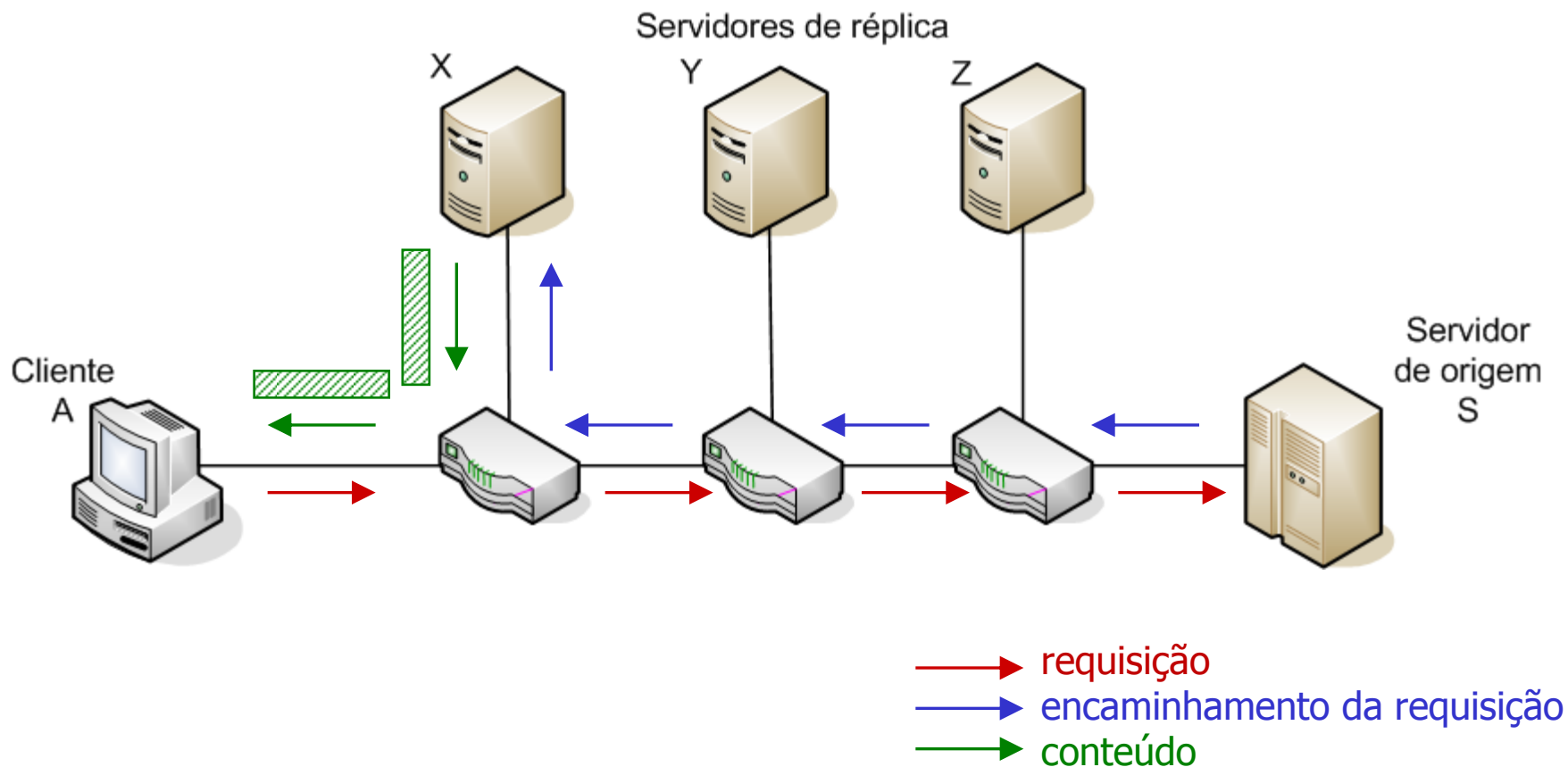
Desempenho



Redes de Distribuição de Conteúdo (*Content Distribution Networks* - CDNs)

- Tornar o modelo cliente-servidor mais eficiente e escalável
 - Distribuição de vídeo
- Conjunto de servidores auxiliares
 - Espalhados geograficamente
 - Pertencem a diferentes *backbones*
- Replicar o conteúdo do servidor de origem
 - Reencaminhar uma requisição para servidores auxiliares mais próximos do cliente
 - Maior taxa de transferência
 - Menor latência
 - Transparente para o cliente

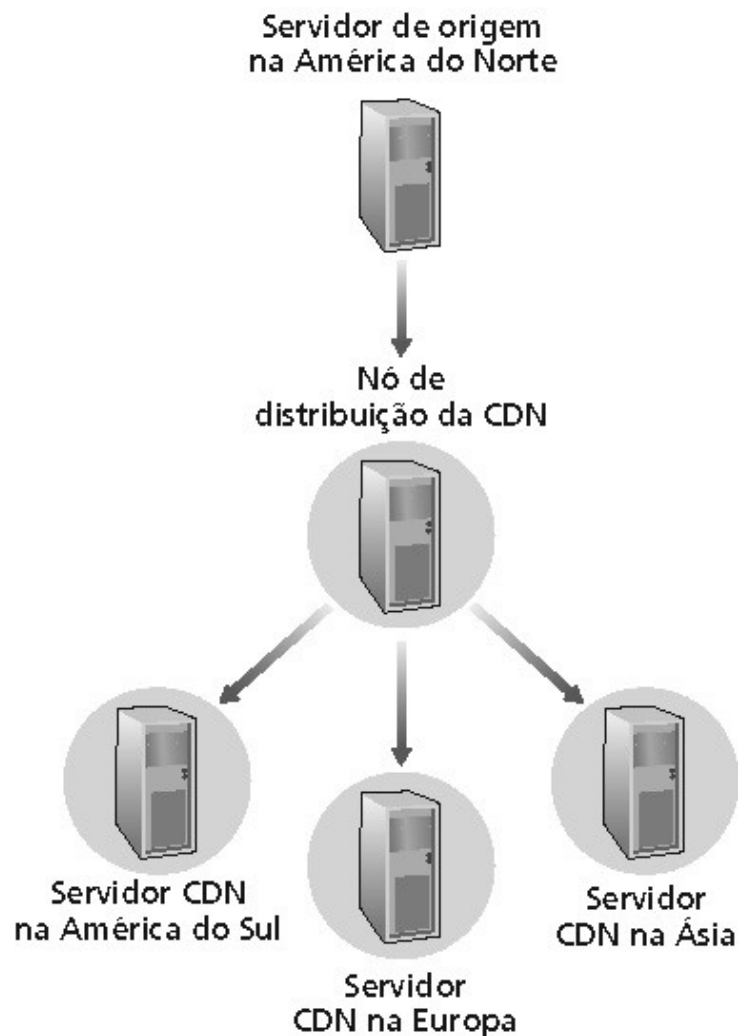
Redes de Distribuição de Conteúdo (*Content Distribution Networks - CDNs*)



Redes de Distribuição de Conteúdo (*Content Distribution Networks* - CDNs)

- Replicação de conteúdo
 - Desafio transmitir fluxo de grandes arquivos (ex, vídeo) de um único servidor origem em tempo real
- Solução
 - Replicar o conteúdo em centenas de servidores através da Internet
 - Conteúdo carregado antecipadamente nos servidores CDN
 - Colocando o conteúdo “perto” do usuário evita impedimentos (perda, atraso) com o envio do conteúdo sobre caminhos longos.
 - Servidor CDN tipicamente posicionado na borda da rede

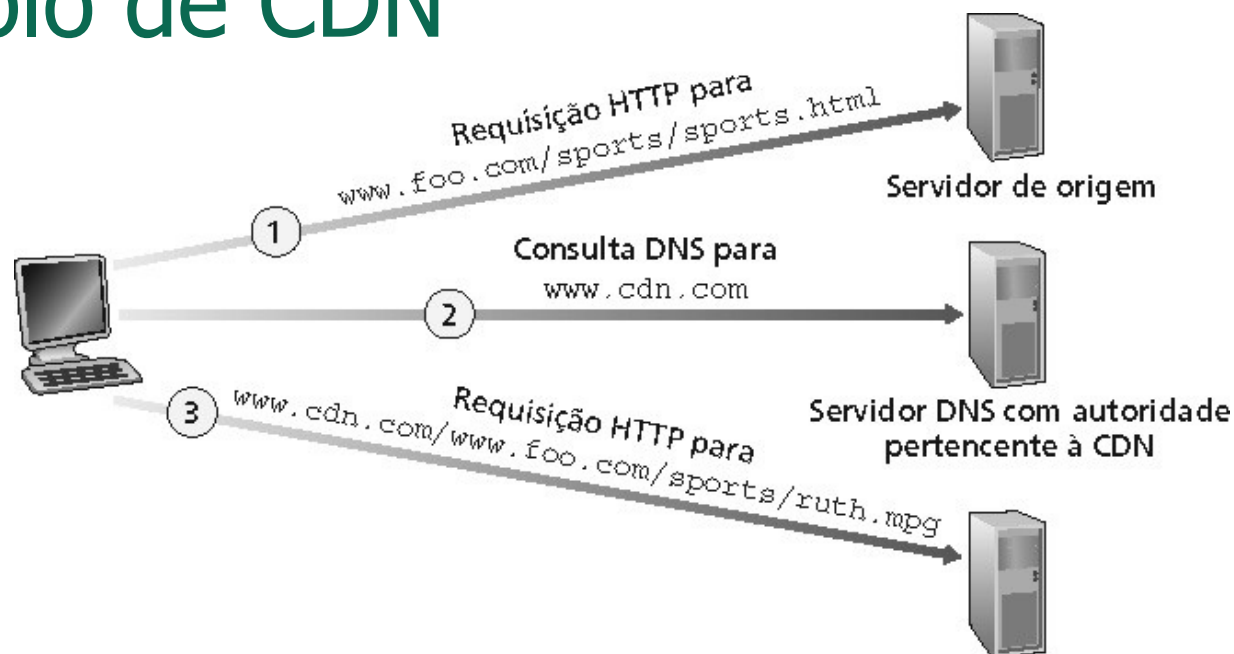
Redes de Distribuição de Conteúdo (*Content Distribution Networks - CDNs*)



Redes de Distribuição de Conteúdo (*Content Distribution Networks* - CDNs)

- Replicação de conteúdo
 - Usuário de uma CDN (ex, Akamai) é o provedor de conteúdo (ex. CNN)
 - A CDN replica o conteúdo do usuário em servidores CDN
 - Quando o provedor atualiza o conteúdo, a CDN atualiza os servidores

Exemplo de CDN



Servidor origem: `www.foo.com`

- Distribui HTML
- Substitui:
`http://www.foo.com/sports.ruth.gif`
por
`http://www.cdn.com/www.foo.com/sports/ruth.gif`

Empresa CDN: `cdn.com`

- Distribui arquivos `.gif`
- usa o seu servidor DNS oficial para redirecionar os pedidos

Redes de Distribuição de Conteúdo (*Content Distribution Networks* - CDNs)

- Desafios
 - Encaminhamento da requisição
 - Escolha do servidor de réplica
 - Replicação do conteúdo
- Desvantagem
 - A eficiência depende do número de servidores auxiliares
 - Alto custo
- Exemplo: Akamai
 - 19 mil servidores na Internet
 - Transmissão do concerto Live Earth
 - 237 mil usuários simultâneos e 15 milhões de fluxos no total

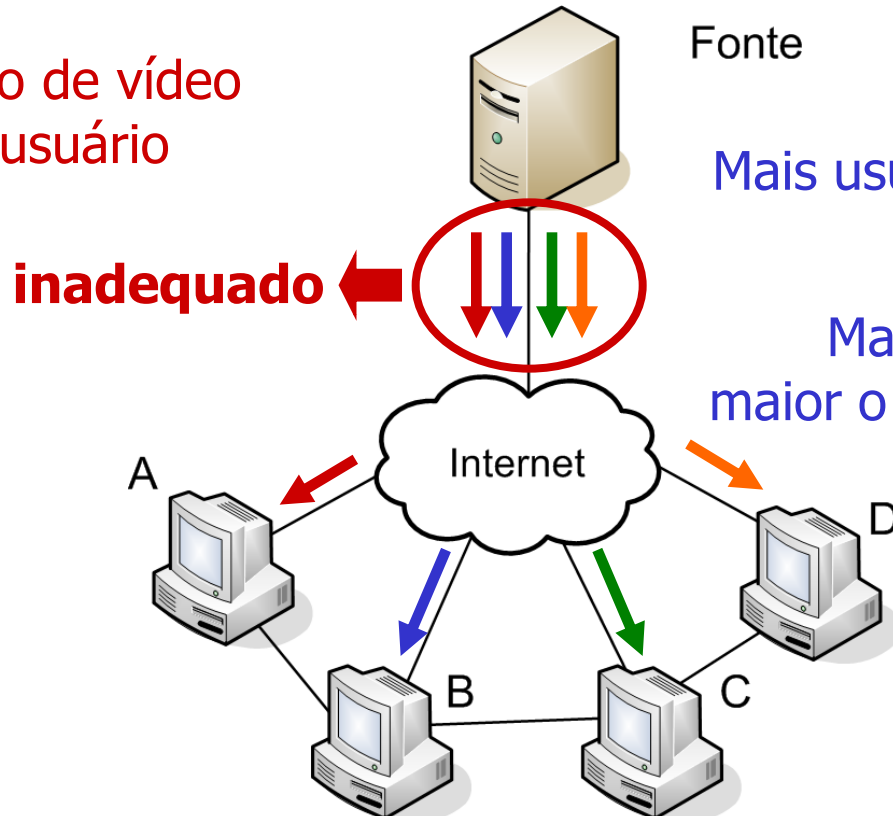
Roteamento de Pedidos em CDNs

- A CDN cria um “mapa”
 - Indica as distâncias entre os ISPs folhas e os nós CDN
- Quando a solicitação chega em um servidor DNS oficial
 1. O servidor determina qual é o ISP de onde provém o pedido
 2. Usa o “mapa” para determinar qual o melhor servidor CDN
- Nós da CDN criam uma rede sobreposta na camada de aplicação

Sistemas Par-a-Par

- Cenário atual
 - Modelo cliente-servidor e comunicação ponto-a-ponto

Um fluxo de vídeo
por usuário



Fonte

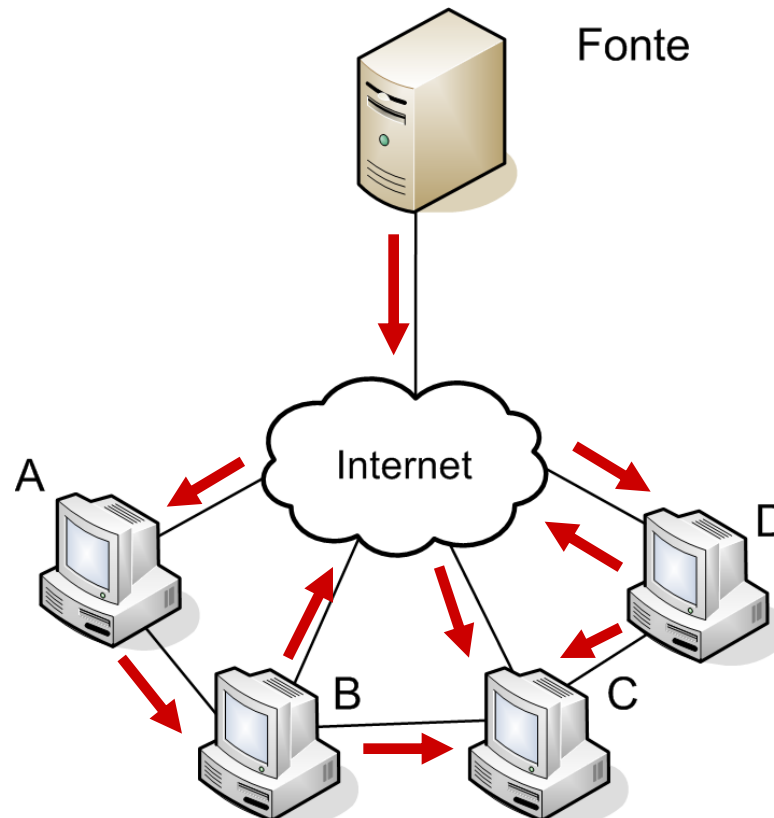
Mais usuários e maior qualidade



Mais banda passante e
maior o custo para os provedores

Sistemas Par-a-Par

- Participantes colaboram para o funcionamento e manutenção do sistema



- Participantes colaboram para o funcionamento e manutenção do sistema
 - Compartilhamento de recursos
 - Banda passante, processamento e armazenamento
 - Mais participantes → maior a capacidade
- Não exigem modificações no núcleo da rede
 - Participantes constroem a estrutura de comunicação
- Exemplos
 - Áudio: Skype
 - Vídeo: PPLive, SopCast,

- <http://www.sopcast.com>



- <http://www.synacast.com/en/>



Protocolos para Aplicações Multimídias

Real-Time Protocol (RTP)

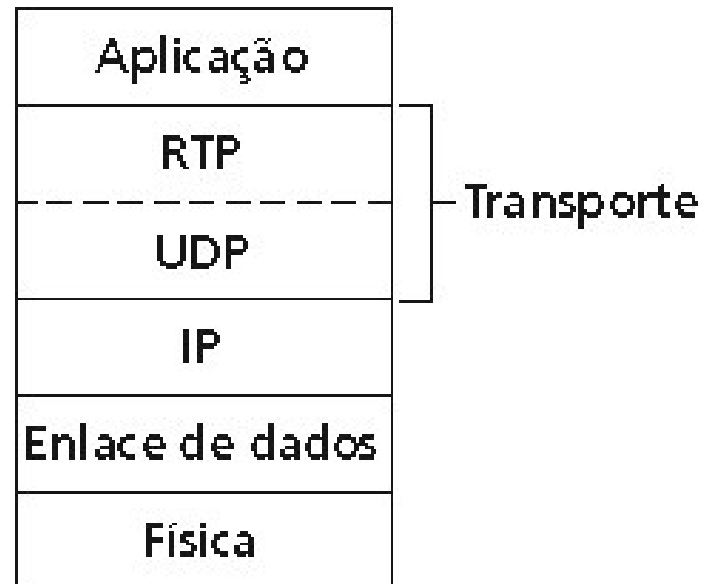
- Definido pela RFC 3550
- Especifica uma estrutura de pacotes
 - Para pacotes que contém dados de áudio e vídeo
- Provê
 - Identificação do tipo de dados contido no pacote
 - Números de sequência
 - Estampilha de tempo

Real-Time Protocol (RTP)

- Executado pelos sistemas finais
- Pacotes RTP são encapsulados em segmentos UDP
- Interoperabilidade
 - Se duas aplicações de telefonia IP rodam RTP, elas podem se comunicar

Real-Time Protocol (RTP)

- Bibliotecas **RTP** provêm uma interface da camada de transporte que **estende o UDP**
 - Números de portas, endereços IP
 - Verificação de erro através de segmentos
 - Identificação do tipo da carga
 - Numeração da seqüência de pacotes
 - Estampilha de tempo



Exemplo com o RTP

- Aplicação
 - Envia voz codificada em PCM a 64 kb/s sobre RTP
 - Coleta dados codificados em pedaços (*chunks*)
 - A cada 20 ms = pedaços de 160 bytes
- Pedaço de áudio + cabeçalho RTP → pacote RTP
 - Pacote RTP é encapsulado em um segmento UDP
- O cabeçalho RTP
 - Indica o tipo da codificação de áudio em cada pacote:
 - Os transmissores podem mudar a codificação durante a conferência
 - Contém um número de seqüência e uma estampilha de tempo

- O encapsulamento RTP é visto apenas nos sistemas finais
 - Não é visto por roteadores intermediários
 - Roteadores provêm o serviço de melhor esforço
 - Não fazem nenhum esforço adicional para garantir que os pacotes RTP cheguem ao destino em tempo

Cabeçalho RTP

- Tipo da carga útil (7 bits)
 - Usado para indicar o tipo de codificação que está sendo usada
 - Se o transmissor modificar a codificação no meio de uma conferência, o transmissor informará o receptor através do campo do tipo de carga útil

Tipo de
carga útil

Número de
seqüência

Marca de
tempo

Identificador de
sincronização da fonte

Campos
variados

Cabeçalho RTP

- Tipo da carga útil (7 bits)
 - Exemplos
 - Tipo de carga 0: PCM, 64 kbps
 - Tipo de carga 3: GSM, 13 kbps
 - Tipo de carga 7: LPC, 2,4 kbps
 - Tipo de carga 26: Motion JPEG
 - Tipo de carga 31: H.261
 - Tipo de carga 33: vídeo MPEG2

Tipo de
carga útil

Número de
seqüência

Marca de
tempo

Identificador de
sincronização da fonte

Campos
variados

Cabeçalho RTP

- Número de sequência (16 bits)
 - É incrementado em uma unidade a cada pacote RTP enviado
 - Pode ser usado para detectar perdas de pacotes e recuperar uma sequência de pacotes

Tipo de carga útil	Número de sequência	Marca de tempo	Identificador de sincronização da fonte	Campos variados
--------------------	---------------------	----------------	---	-----------------

- Estampilha ou marca de tempo (32 bits)
 - Reflete o instante de amostragem do primeiro byte no pacote de dados RTP
 - Para áudio, o relógio de estampilha de tempo é incrementado em uma unidade a cada período de amostragem
 - Ex.: a cada 125 ms para um relógio de amostragem de 8 kHz
 - Se uma aplicação de áudio gera pedaços de 160 amostras codificadas
 - Quando a fonte esta ativa, a estampilha aumenta de 160 para cada pacote RTP enviado
 - Quando a fonte está inativa, o relógio da estampilha de tempo continua a aumentar a uma taxa constante

Tipo de carga útil	Número de seqüência	Marca de tempo	Identificador de sincronização da fonte	Campos variados
--------------------	---------------------	----------------	---	-----------------

Cabeçalho RTP

- Identificador de sincronização da fonte - SSRC (32 bits)
 - Identifica a fonte de um fluxo RTP
 - Cada fluxo em uma sessão RTP deve possuir um SSRC distinto

Tipo de carga útil	Número de seqüência	Marca de tempo	Identificador de sincronização da fonte	Campos variados
--------------------	---------------------	----------------	---	-----------------

Real-Time Control Protocol (RTCP)

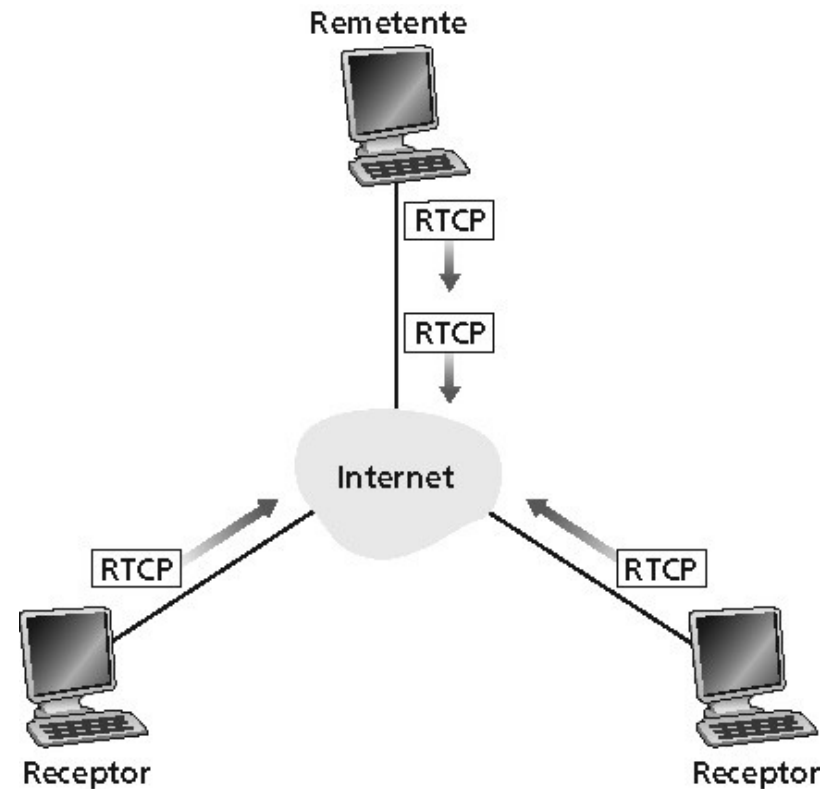
- Trabalha em **conjunto** com o RTP
- Cada participante em uma sessão RTP transmite periodicamente pacotes de controle RTCP para todos os outros participantes
- Cada pacote RTCP contém **relatórios** do emissor e/ou receptores
 - Estatísticas úteis para aplicação
 - Número de pacotes enviados
 - Número de pacotes perdidos
 - Variação do atraso entre os pacotes recebidos
 - Etc.

Real-Time Control Protocol (RTCP)

- Realimentação pode ser usada para controlar o desempenho das aplicações
 - Emissor pode modificar sua taxa de transmissão/codificação baseado nas informações dos relatórios
 - Mecanismos não definidos pelo RTCP
 - Responsabilidade da aplicação

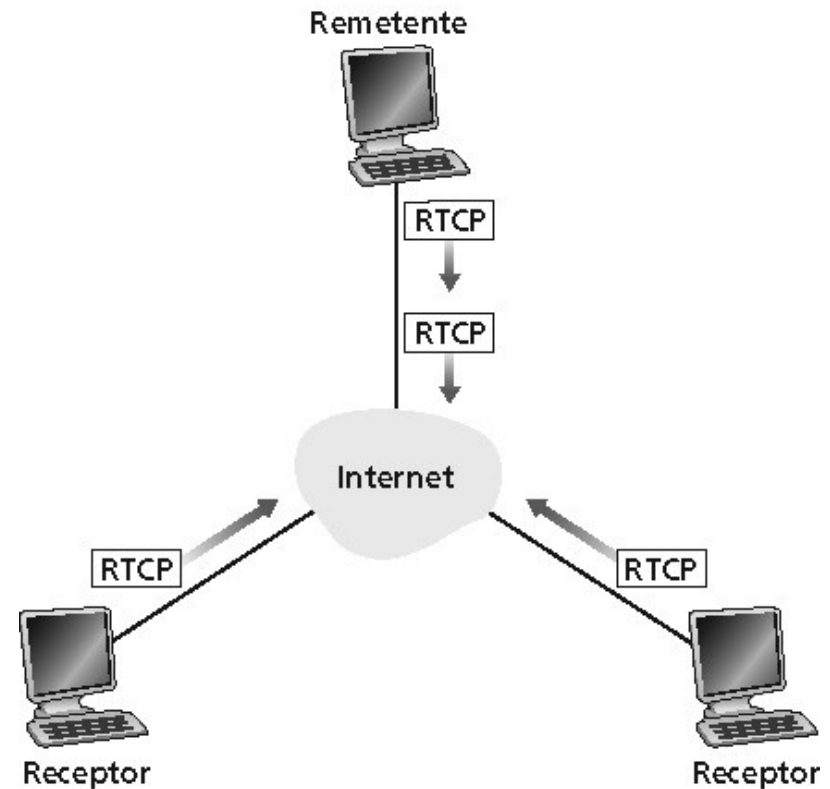
Real-Time Control Protocol (RTCP)

- Para uma sessão RTP com muitos usuários há tipicamente um único endereço *multicast*
 - Todos os pacotes RTP e RTCP pertencentes à sessão usam o mesmo endereço *multicast*
- Pacotes RTP e RTCP são diferenciados uns dos outros através do uso de números de portas distintos



Real-Time Control Protocol (RTCP)

- Para limitar o tráfego, cada participante reduz o seu tráfego RTCP à medida que cresce o número de participantes da sessão



- Relatório do receptor
 - Fração dos pacotes perdidos, último número de seqüência, *jitter* entre chegadas
- Relatório do transmissor
 - SSRC do fluxo RTP, estampilha de tempo, número de pacotes e número de bytes enviados
- Descrição da fonte
 - Endereço de e-mail do remetente, nome do remetente, o SSRC do fluxo RTP associado
 - Os pacotes provêm um mapeamento entre o SSRC e o nome do usuário/estação

RTCP: Sincronização dos Fluxos

- O RTCP pode ser usado para sincronizar diferentes fluxos multimídias dentro de uma sessão RTP
 - Exemplo: uma aplicação de videoconferência

RTCP: Sincronização dos Fluxos

- Exemplo: uma aplicação de videoconferência
 - Cada transmissor gera um fluxo RTP para vídeo e outro para áudio
 - As estampilhas de tempo nos pacotes RTP estão vinculadas aos relógios de amostragem de vídeo e de áudio
 - Não estão vinculadas ao “relógio real”
 - Cada relatório RTCP do transmissor contém (para o pacote gerado mais recentemente em um fluxo RTP):
 - Estampilha de tempo do pacote RTP
 - Instante do relógio real no qual o pacote RTP foi criado
 - Receptores associam essas duas informações para sincronizar a reprodução dos fluxos de áudio e vídeo

RTCP: Ajuste de Banda

- O RTCP tenta limitar o seu tráfego a 5% da largura de banda da sessão
- Exemplo
 - Transmissor envia um vídeo a 2 Mb/s
 - RTCP tenta limitar o seu tráfego a 100 kb/s
 - 75% desta taxa para os receptores → 75 kb/s
 - 25% da taxa para o transmissor → 25 kb/s

RTCP: Ajuste de Banda

- Ainda para o exemplo
 - 75 kb/s é igualmente compartilhado pelos receptores
 - Com R receptores, cada receptor pode transmitir pacotes RTCP a uma taxa igual a $75/R$ kb/s
 - O transmissor envia pacotes RTCP a uma taxa de 25 kb/s
 - Um receptor/transmissor determina o período de transmissão dos pacotes RTCP
 - Calcula dinamicamente do tamanho médio de um pacote RTCP, ao longo de toda a sessão
 - Divide o tamanho médio do pacote RTCP pela sua taxa alocada

- *Real-Time Streaming Protocol*
 - Definido pela RFC 2326
- É um protocolo de **controle das ações de reprodução**
 - É um “controle remoto”
 - Possui comandos para retroceder, avançar, pausar a reprodução, entre outros
 - Baseado na arquitetura cliente-servidor
 - Implementado na camada de aplicação
- Usado em substituição ao HTTP na distribuição multimídia
 - HTTP não é específico para essa finalidade e não possui comandos para avanço, pausa, etc.

- **NÃO** é responsável por
 - Definir como os fluxos de áudio e vídeo são encapsulados para a distribuição
 - Definir o protocolo de transporte usado na distribuição
 - Especificar como o tocador armazena os fluxos de áudio e vídeo

- Usa um canal de controle **fora da banda**
 - Similar ao FTP
 - Mensagens de controle do RTSP usam números de porta diferentes dos usados pelos fluxos multimídias
 - Por isso, “fora da banda”
 - Porta 554
 - O fluxo multimídia está “dentro” da banda, nesse caso

Exemplo de Uso do RTSP

- Cenário
 - Metarquivo enviado para o navegador
 - Navegador inicia o tocador
 - Tocador estabelece uma conexão de controle RTSP e uma conexão de dados para o servidor de fluxo contínuo

Exemplo de Metarquivo

```
<title>Twister</title>
```

```
<session>
```

```
  <group language=en lipsync>
```

```
    <switch>
```

```
      <track type=audio
```

```
        e="PCMU/8000/1"
```

```
        src = "rtsp://audio.example.com/twister/audio.en/lofi">
```

```
      <track type=audio
```

```
        e="DVI4/16000/2" pt="90 DVI4/8000/1"
```

```
        src="rtsp://audio.example.com/twister/audio.en/hifi">
```

```
    </switch>
```

```
  <track type="video/jpeg"
```

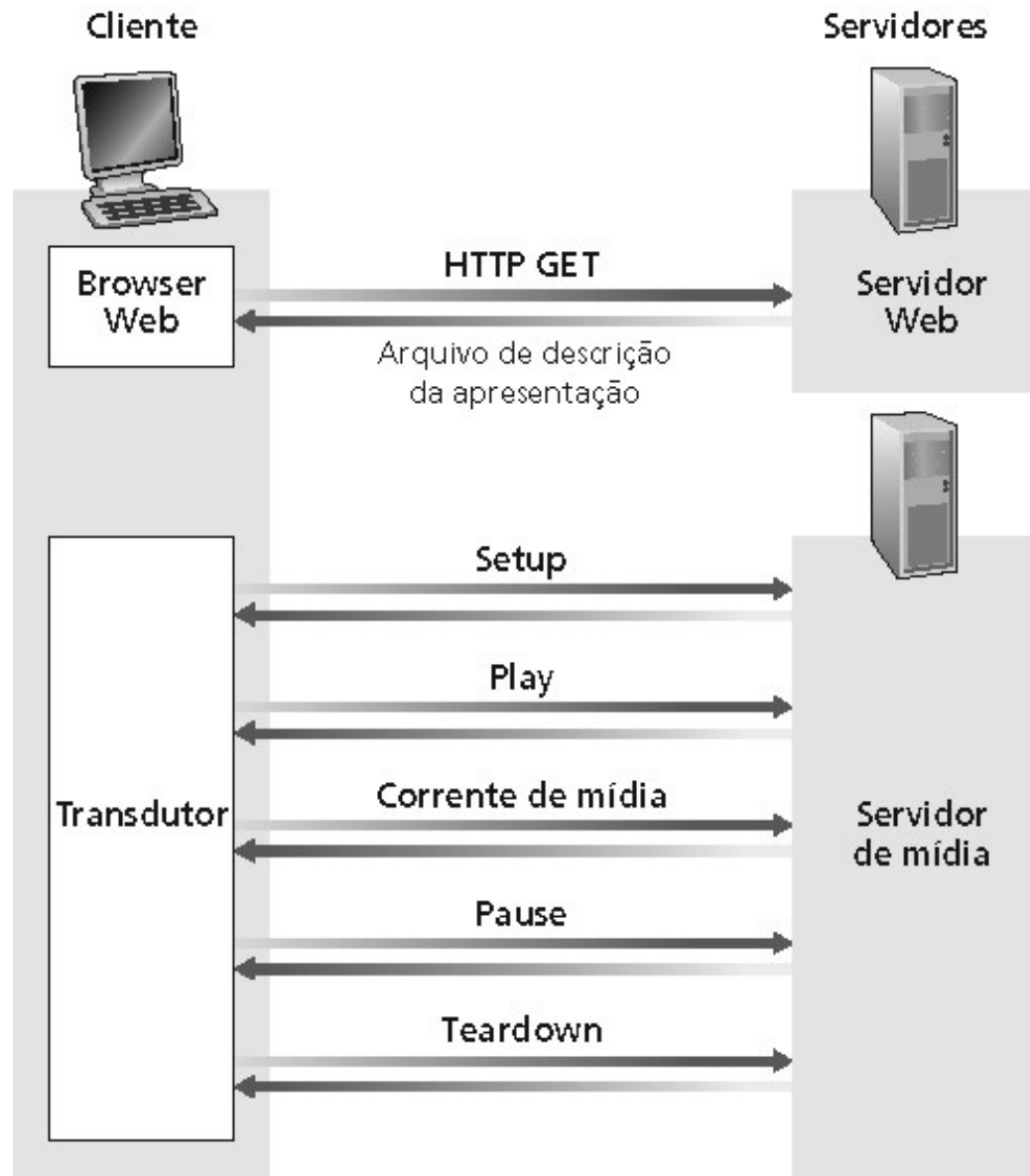
```
    src="rtsp://video.example.com/twister/video">
```

```
</group>
```

```
</session>
```

**Um fluxo de vídeo e
dois fluxos de áudio
(diferentes qualidades)**

Operação do RTSP



Exemplo de Troca de Mensagens do RTSP

```
C: SETUP rtsp://audio.example.com/twister/audio RTSP/1.0
  Transport: rtp/udp; compression; port=3056; mode=PLAY

S: RTSP/1.0 200 1 OK
  Session 4231

C: PLAY rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
  Session: 4231
  Range: npt=0-

C: PAUSE rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
  Session: 4231
  Range: npt=37

C: TEARDOWN rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
  Session: 4231

S: 200 3 OK
```

Session Initiation Protocol (SIP)

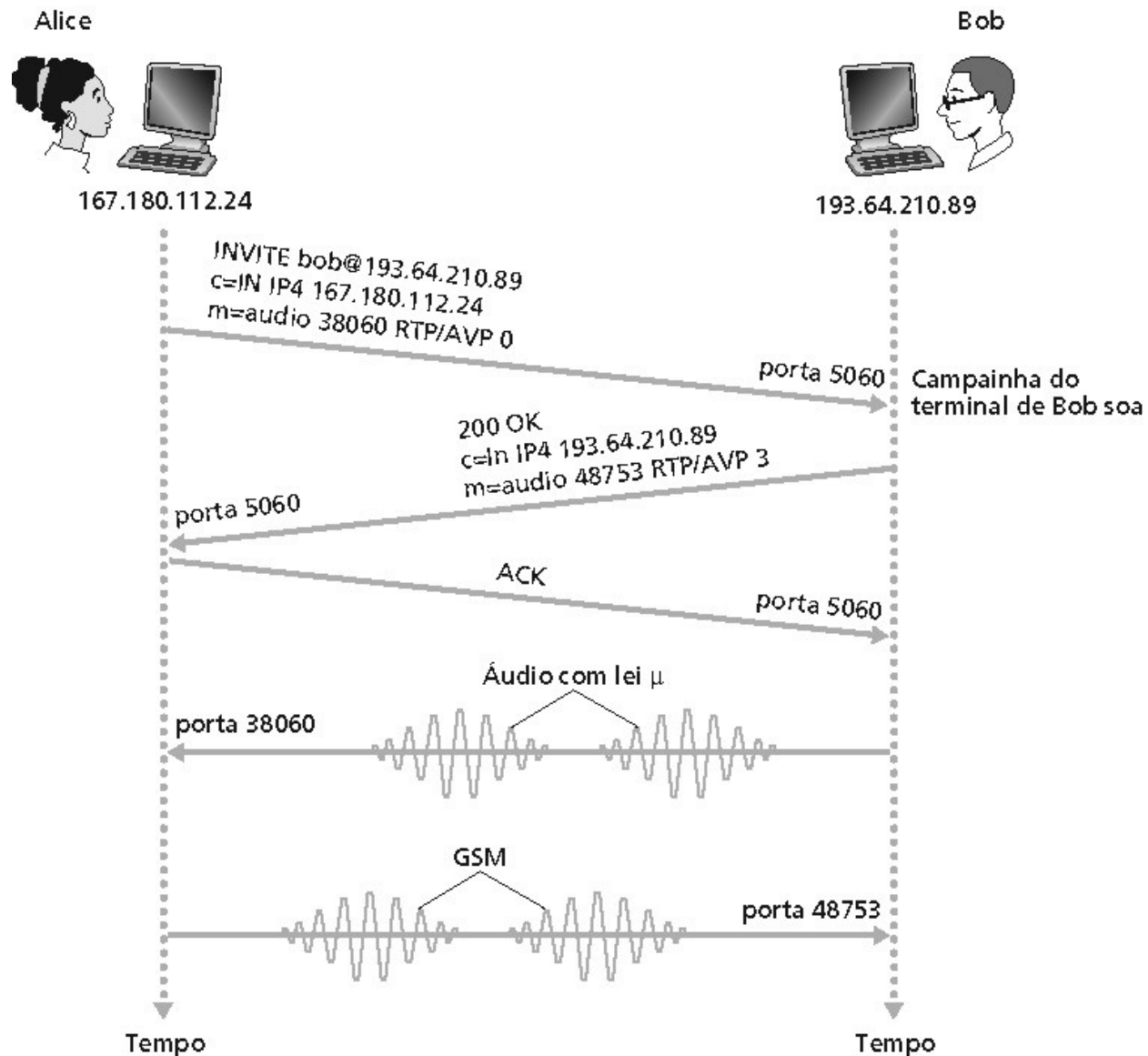
- Definido pela RFC 3261
 - É um protocolo do IETF → “Internet”
- Visão de longo prazo do SIP
 - Todas as chamadas telefônicas e de videoconferência se realizam sobre a Internet
 - Pessoas são identificadas por nomes ou endereços de e-mail, ao invés de números de telefone
 - Você pode alcançar o destinatário de uma chamada, não importa onde ele esteja, em qualquer dispositivo IP que o destinatário esteja usando no momento

- Ao estabelecer a conexão, provê mecanismos
 - Para o “chamador” sinalizar ao “chamado” que ele deseja estabelecer uma conexão
 - Para que chamador e chamado acordem sobre o tipo de mídia e codificação a serem usados em uma chamada
 - Para encerrar uma chamada
- Determinar o endereço IP atual do chamado (destinatário)
 - Mapeamento: identificador mnemônico para o endereço IP atual

- Gerenciamento de chamada
 - Adicionar novos fluxos multimídias durante uma chamada
 - Modificar a codificação durante uma chamada
 - Convidar outros participantes
 - Transferir e “pausar” chamadas

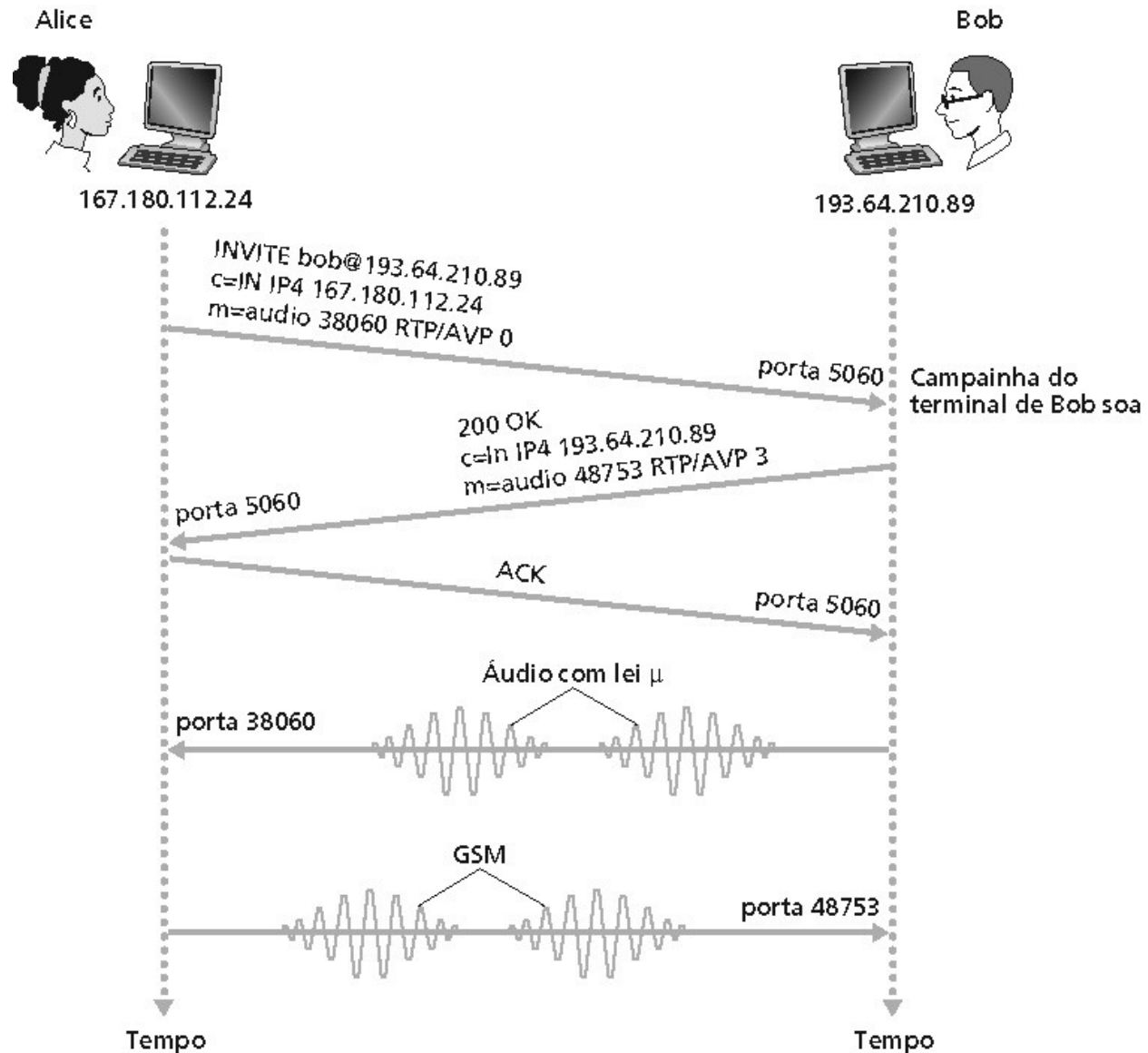
SIP: Conexão Para um Endereço IP Conhecido

- Mensagem SIP de convite de Alice indica o seu número de porta e endereço IP
 - Indica a codificação em que Alice prefere receber (lei m do PCM)



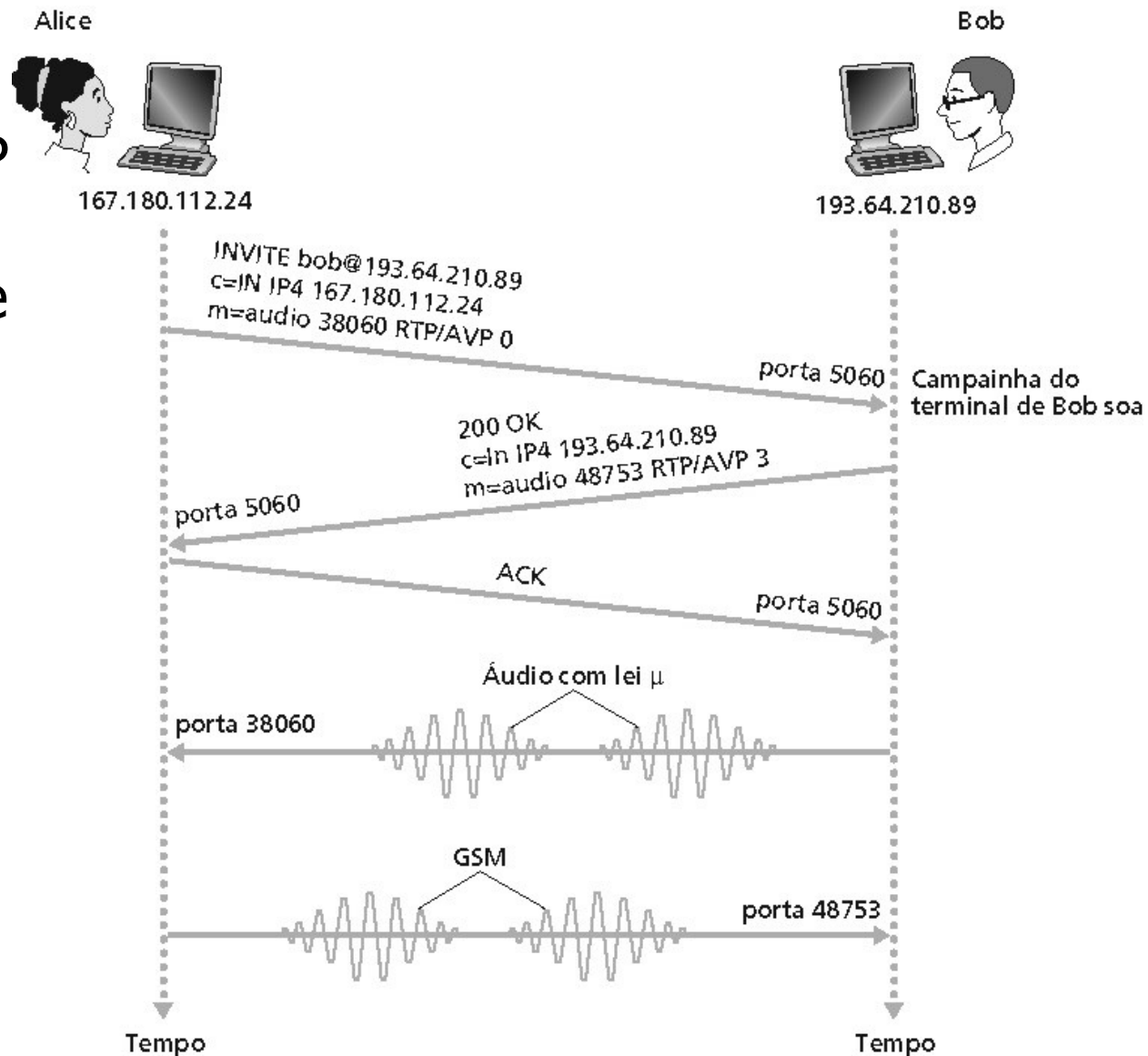
SIP: Conexão Para um Endereço IP Conhecido

- A mensagem 200 OK de Bob indica o seu número de porta, endereço IP e codificação preferida (GSM)



SIP: Conexão Para um Endereço IP Conhecido

- As mensagens SIP podem ser transmitidas sobre TCP ou UDP
 - No exemplo, está sendo enviada sobre RTP/UDP
- O número de porta padrão do SIP é a 5060



SIP: Conexão Para um Endereço IP Conhecido

- Negociação do codificador
 - Caso Bob não possua um codificador lei μ do PCM
 - Bob responderá com um código "606 Not Acceptable Reply" e lista os codificadores que ele pode usar
 - Alice pode então enviar uma nova mensagem INVITE, anunciando um codificador apropriado
- Rejeição de uma chamada
 - Bob pode rejeitar com uma resposta do tipo "busy" (ocupado), "gone" (encerrado), "payment required" (é necessário pagamento), "forbidden" (proibido)
- Fluxos multimídias podem ser enviados sobre o RTP ou outro protocolo

Exemplo de uma Mensagem SIP

- Nesse caso, Alice não conhece o endereço IP de Bob
 - Servidores SIP intermediários são necessários
- Alice envia e recebe mensagens SIP usando a porta padrão 5060
- Alice especifica em "Via:" o cabeçalho que o cliente SIP envia, recebe mensagens SIP sobre UDP

```
INVITE sip:bob@domain.com SIP/2.0
Via: SIP/2.0/UDP 167.180.112.24
From: sip:alice@hereway.com
To: sip:bob@domain.com
Call-ID: a2e3a@pigeon.hereway.com
Content-Type: application/sdp
Content-Length: 885

c=IN IP4 167.180.112.24
m=audio 38060 RTP/AVP 0
```

Notas

- Sintaxe de uma mensagem HTTP
- sdp → *Session Description Protocol*
- Call-ID é único para cada chamada

Tradução de Nomes e Localização de Usuários

- Chamador deseja chamar o chamado, mas possui apenas o nome ou o endereço de e-mail do chamado
- É necessário obter o endereço IP da estação atual do chamado
 - Usuário pode
 - Se movimentar
 - Obter um endereço IP através de DHCP
 - Possuir diferentes dispositivos IP:
 - PC, PDA, dispositivo no veículo, etc.

Tradução de Nomes e Localização de Usuários

- Resultado pode depender
 - Da hora do dia
 - Trabalho, casa, etc.
 - Do chamador
 - Ex.: Não deseja que o chefe o chame em casa
 - Da situação atual do chamado
 - Ex.: chamadas enviadas para correio de voz quando o chamado já estiver falando com alguém
- Serviços providos por servidores SIP
 - Entidade registradora SIP
 - Servidor *proxy* do SIP

Entidade Registradora SIP

- Quando Bob inicia o cliente SIP
 - Cliente envia uma mensagem SIP REGISTER para a entidade registradora de Bob
 - Função similar à necessário em serviços de envio de mensagens instantâneas

Mensagem de registro

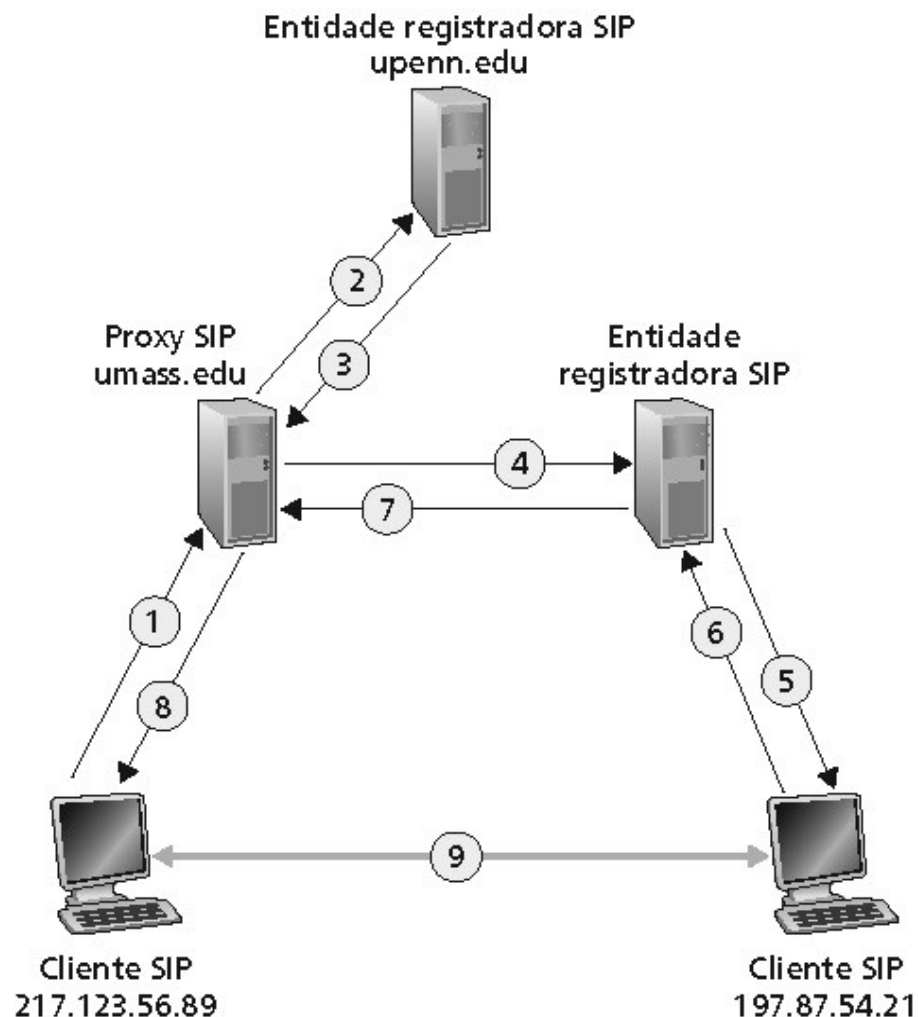
```
REGISTER sip:domain.com SIP/2.0  
Via: SIP/2.0/UDP 193.64.210.89  
From: sip:bob@domain.com  
To: sip:bob@domain.com  
Expires: 3600
```

- Alice envia mensagens de convite para o seu servidor *proxy*
 - Contém o endereço `sip:bob@domain.com`
- O *proxy* é responsável por rotear as mensagens SIP para o chamado (destinatário)
 - Possivelmente através de múltiplos *proxies*
- O chamado envia uma resposta de volta através do mesmo conjunto de *proxies*
- O proxy retorna uma mensagem de resposta SIP para Alice
 - Contém o endereço IP de Bob
- O proxy é análogo ao servidor DNS local

Exemplo

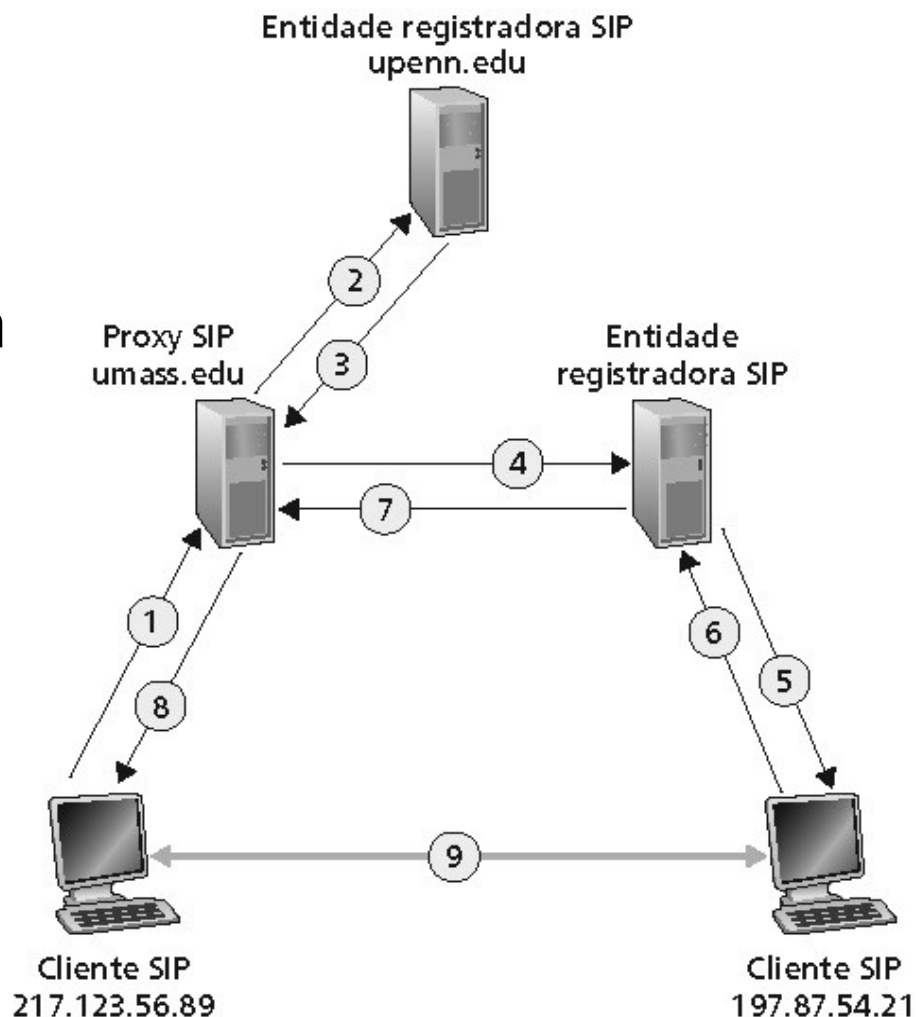
- Chamador `jim@umass.edu` inicia uma chamada para `keith@upenn.edu`

1. Jim envia mensagem INVITE para o *proxy* SIP da UMass
2. *Proxy* encaminha o pedido para o servidor de registro da Upenn



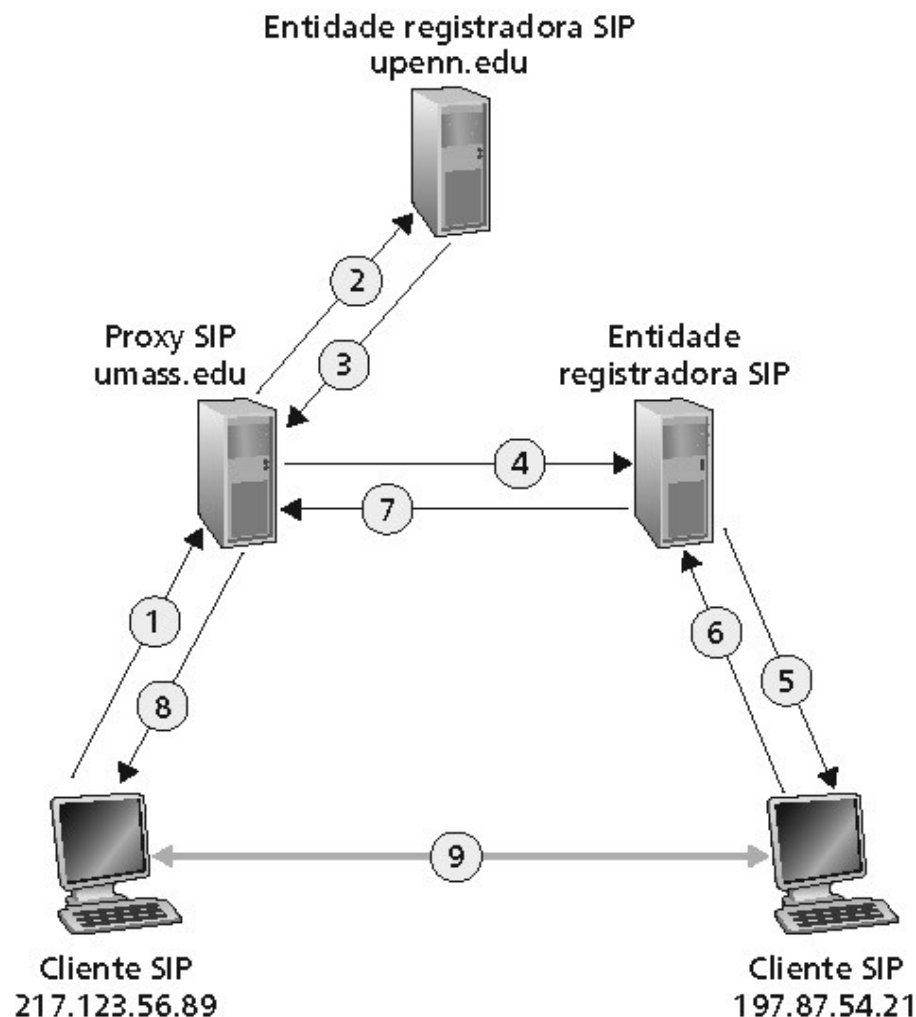
Exemplo

- Chamador `jim@umass.edu` inicia uma chamada para `keith@upenn.edu`
- 3. O servidor da UPenn retorna resposta de redirecionamento, indicando que deve tentar `keith@eurecom.fr`
- 4. O proxy da UMass envia INVITE para o registro da eurecom.



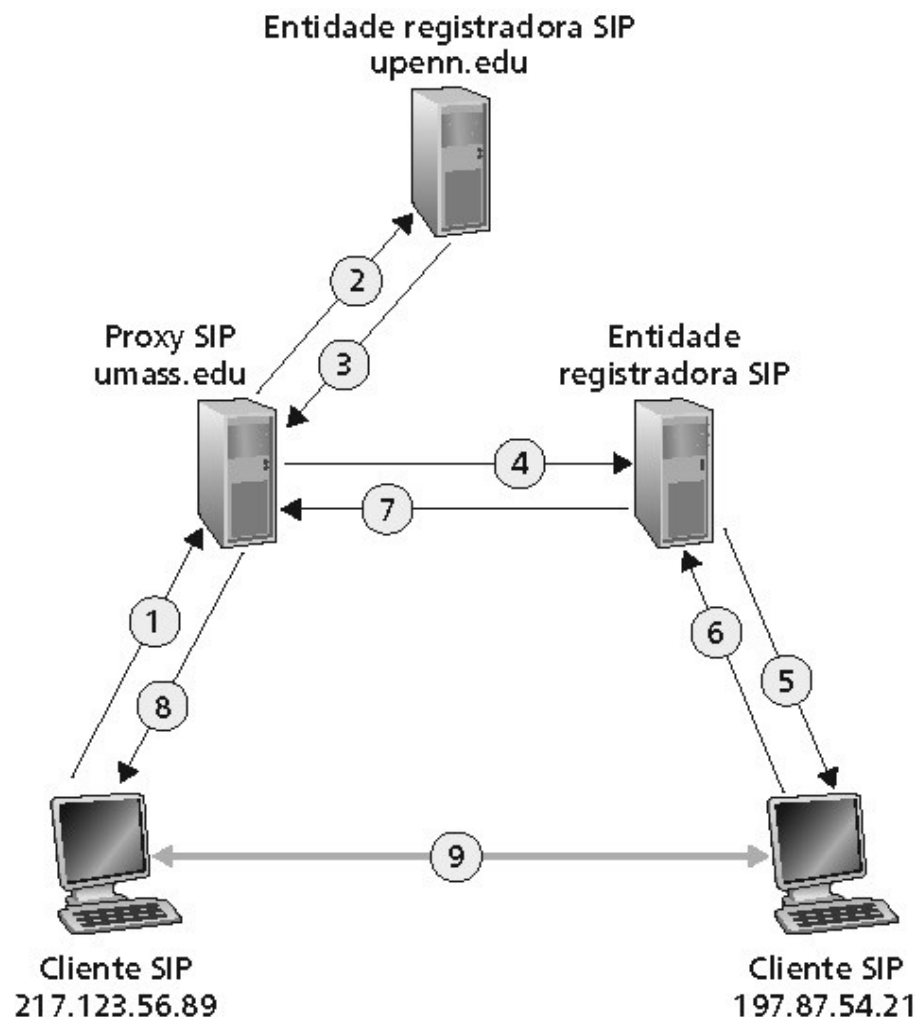
Exemplo

- Chamador `jim@umass.edu` inicia uma chamada para `keith@upenn.edu`
- 5. Registro da eurecom encaminha o INVITE para 197.87.54.21, que está rodando o cliente SIP de Keith
- 6. a 8. Retorno da resposta SIP



Exemplo

- Chamador `jim@umass.edu` inicia uma chamada para `keith@upenn.edu`
- 9. Mídia enviada diretamente entre clientes
- Nota
 - Mensagens ACK do SIP não apresentadas



- H.323 também é um protocolo de sinalização para aplicações multimídias de tempo-real e interativas
- H.323 é um **conjunto integrado de protocolos** para conferência multimídia
 - É um **guarda-chuva**
 - Sinalização, registro, controle de admissão, transporte e codificadores
- SIP é um **único componente**
 - Trabalha com RTP, mas não é obrigatório
 - Pode ser combinado com outros protocolos e serviços

SIP vs. H.323

- H.323 é um padrão da ITU
 - Telefonia
- SIP é um padrão IETF
 - Internet
 - Pega emprestado muitos dos seus conceitos do HTTP
- SIP tem o “sabor” Web e o H.323 tem o “sabor” da telefonia
- SIP usa o princípio KISS: “*Keep it simple stupid*”

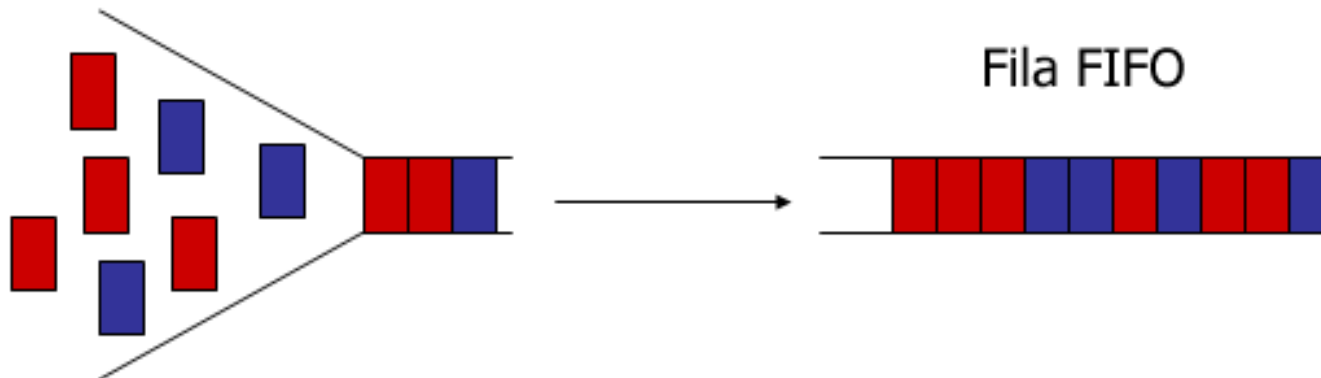
Qualidade de Serviço

Qualidade de Serviço

- Definição
 - Nível de desempenho que as aplicações precisam para funcionar corretamente
- Na Internet, é difícil **garantir** desempenho
 - Não há reserva de recursos
 - O meio é compartilhado por diferentes aplicações
- Mas, **diferenciar** o desempenho é mais fácil
 - Classificar e marcar os pacotes
 - Tratar diferenciadamente a partir da marcação

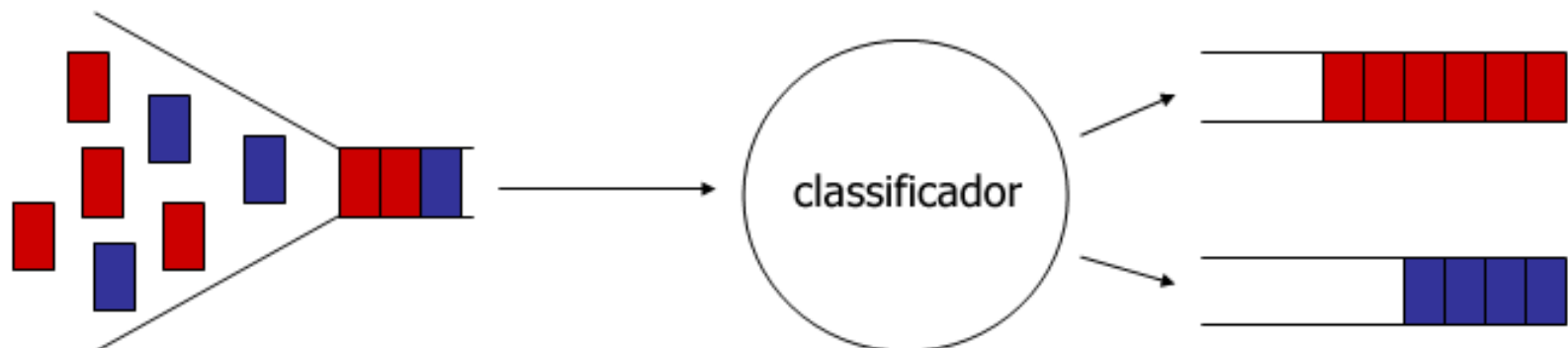
Classes de Serviço

- Modelo de serviço de melhor esforço
 - Sem diferenciação
 - “Tamanho único” → “*one size fits all*”



Classes de Serviço

- Uma alternativa é definir múltiplas **classes de serviço**
 - Tráfego com requisitos semelhantes são agrupados em classes
 - Rede trata de forma diferenciada as diferentes classes de tráfego
 - Analogia com as classes em um avião
 - Primeira, executiva, econômica

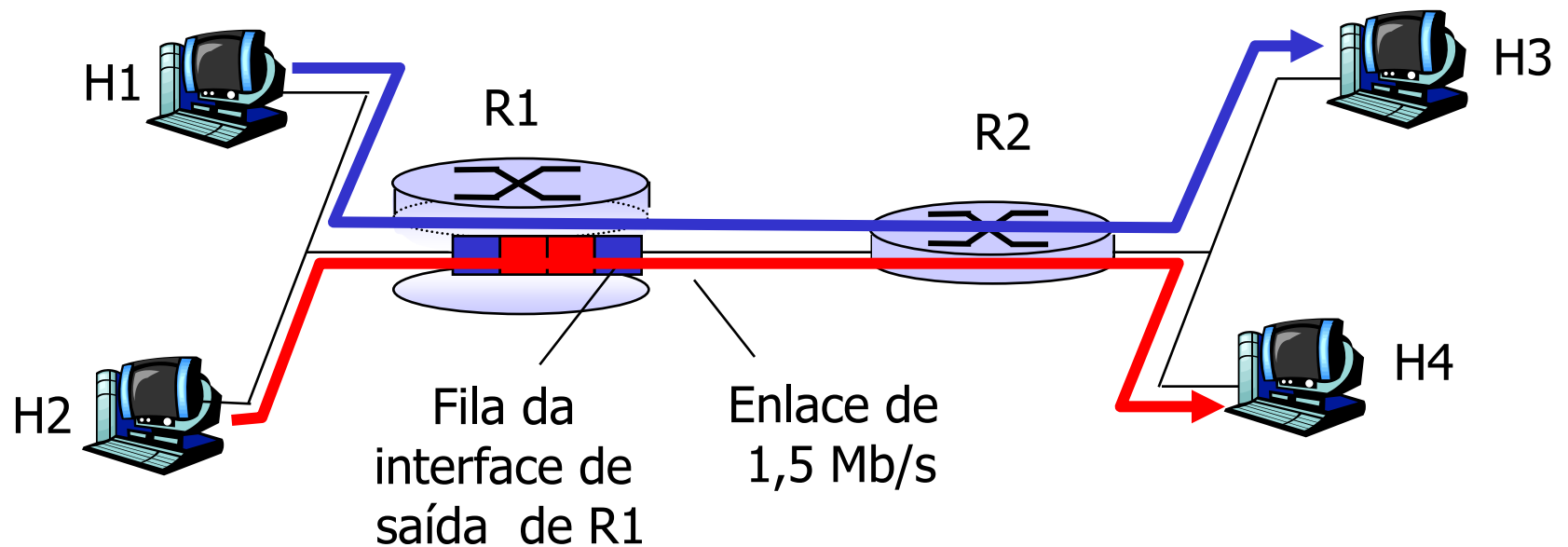


Classes de Serviço

- Granularidade → Escalabilidade
 - Diferenciação entre as classes e não para conexões individuais
- Bits de tipo de serviço (ToS) do cabeçalho IP são um indicativo

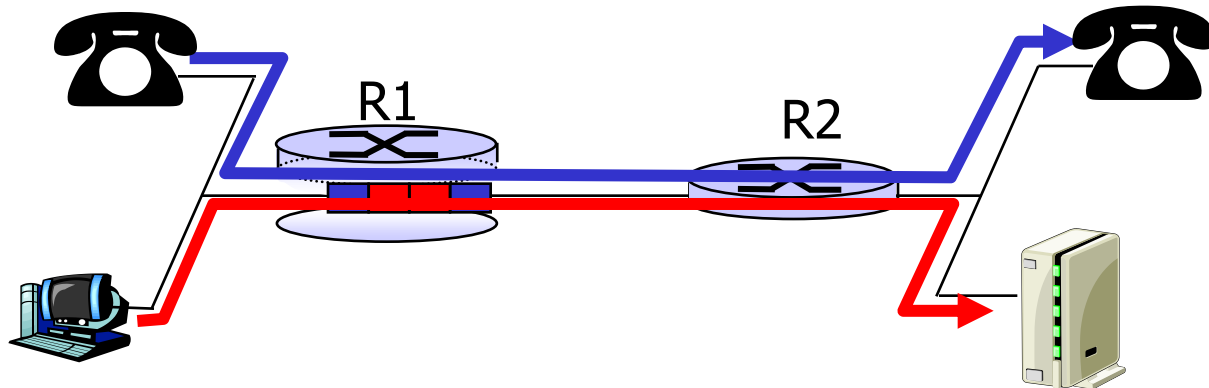
Classes de Serviço

- Cenário com duas classes



Princípios para Provisão de QoS

- Exemplo: Fluxo de telefonia IP de 1 Mb/s compartilha em enlace de 1,5 Mb/s com um fluxo FTP
 - Rajadas FTP podem congestionar a fila do roteador e causar perdas de pacotes de áudio
 - Deseja-se priorizar o fluxo de áudio em relação ao FTP



Princípios para Provisão de QoS

- Exemplo: Fluxo de telefonia IP de 1 Mb/s compartilha em enlace de 1,5 Mb/s com um fluxo FTP
 - Rajadas FTP podem congestionar a fila do roteador e causar

Princípio 1: É necessário

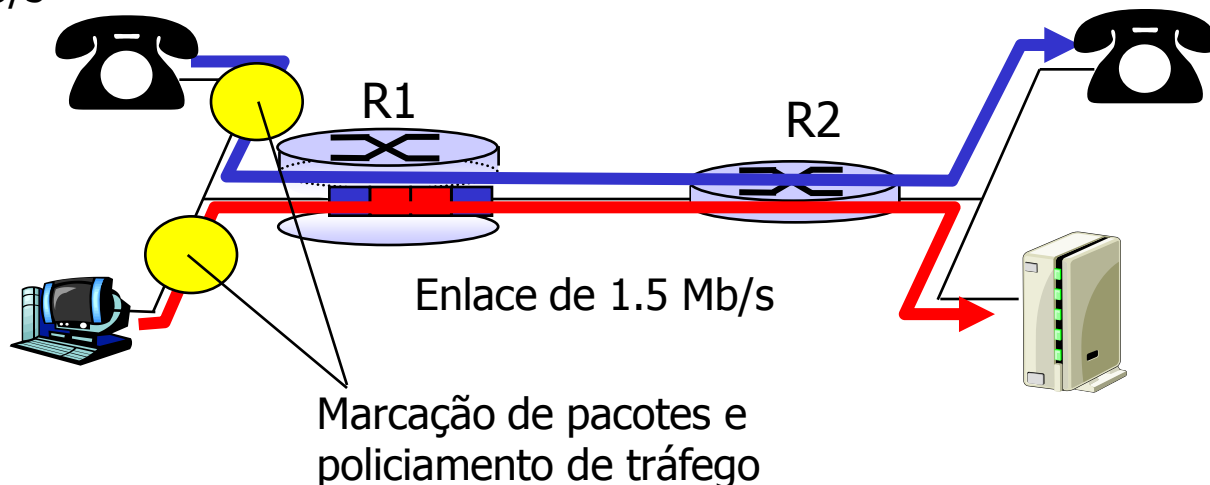
- **Marcar os pacotes** para que um roteador diferencie as classes de serviço
- **Definir uma nova política** no roteador para tratar os pacotes de forma diferenciada



Princípios para Provisão de QoS

- Uma aplicação pode ser mal comportada
 - Ex.: áudio envia pacotes a uma taxa superior da declarada
 - **Policimento**
 - Força que as fontes respeitem as alocações de banda
 - Implementado na borda da rede junto com a marcação
 - Similar à UNI (*User Network Interface*) do ATM

Telefonia IP
Fluxo de 1 Mb/s

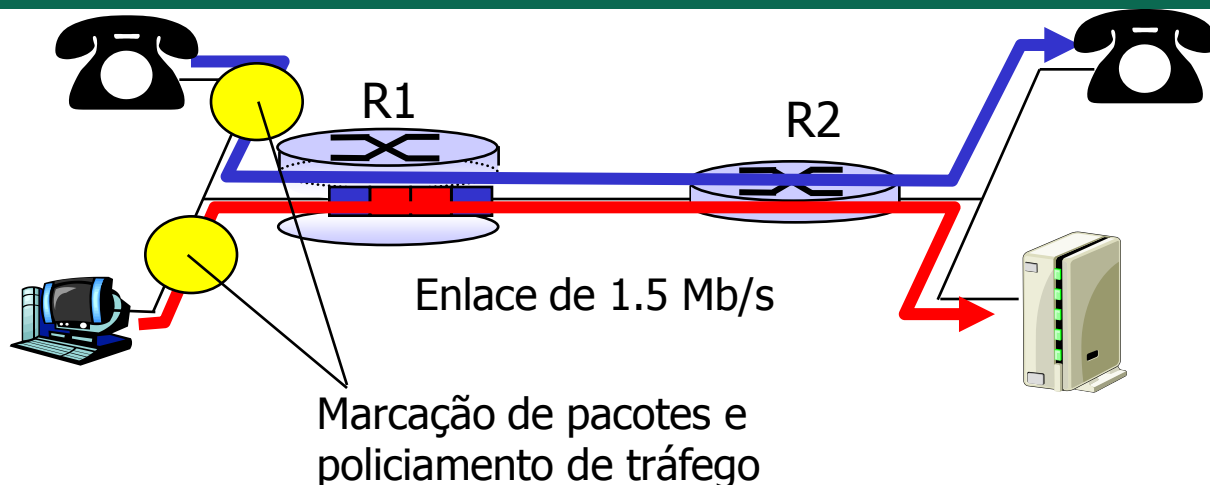


Princípios para Provisão de QoS

- Uma aplicação pode ser mal comportada
 - Ex.: áudio envia pacotes a uma taxa superior da declarada
 - **Policimento**

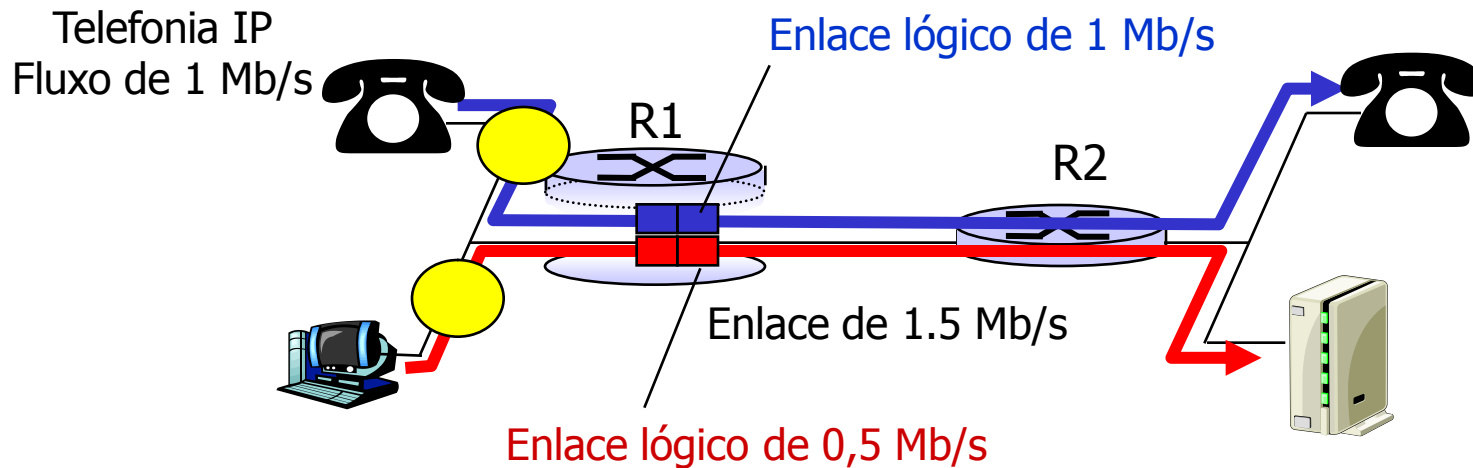
Princípio 2:

- É necessário **prover isolamento** entre as classes de serviço
 - O tráfego de uma classe não deve interferir no serviço acordado para outro classe



Princípios para Provisão de QoS

- Alternativa: alocar uma parte fixa e não-compartilhada da banda para um fluxo
 - É ineficiente se os fluxos não usam toda sua parte durante todo o tempo

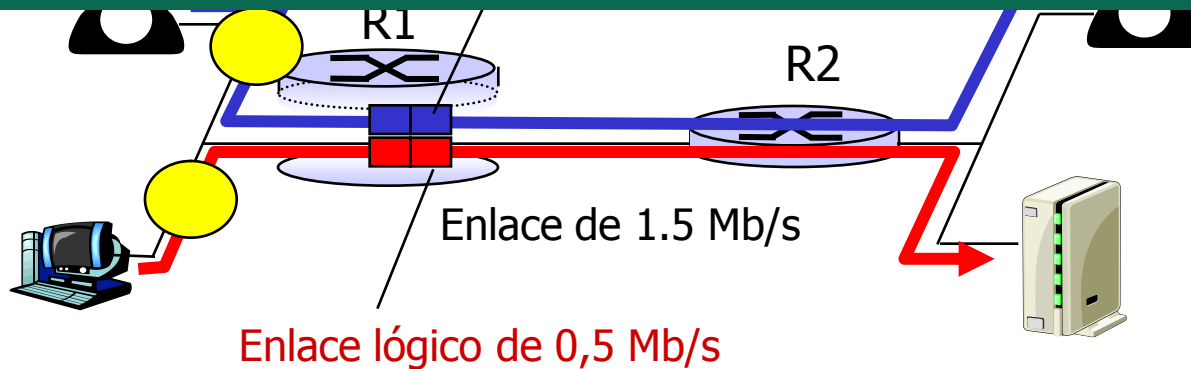


Princípios para Provisão de QoS

- Alternativa: alocar uma parte fixa e não-compartilhada da banda para um fluxo
 - É ineficiente se os fluxos não usam toda sua parte durante

Princípio 3:

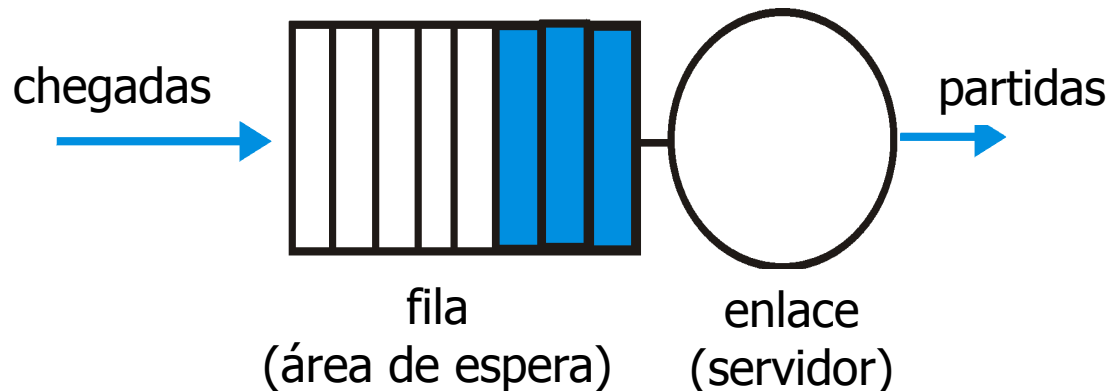
- O uso dos recursos deve ser o mais eficiente possível
- Ao garantir isolamento entre as classes



- Mecanismos de escalonamento
 - Determinam o **próximo pacote da fila** que será **transmitido** em um enlace
 - Também chamados de disciplinas de fila
- Exemplos
 - FIFO
 - Prioridade simples
 - *Round-Robin* (RR)
 - *Weighted Fair Queuing* (WFQ)
 - Etc.

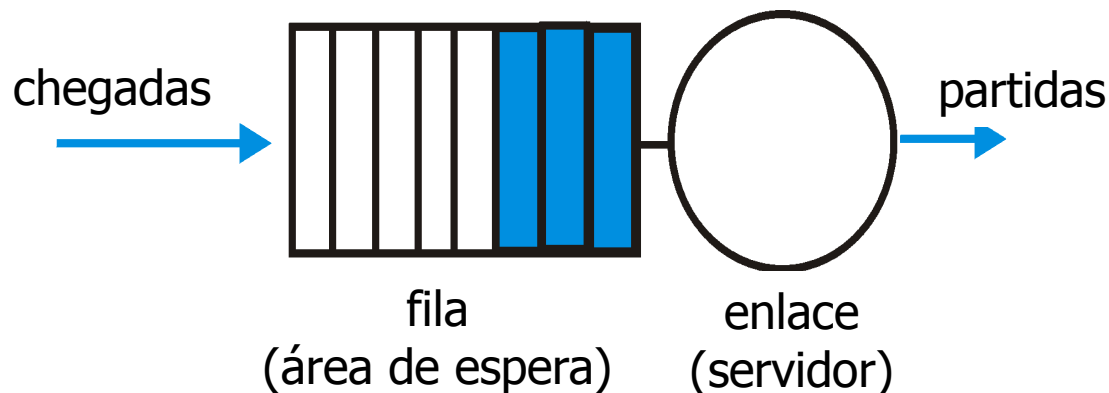
FIFO (*First In First Out*)

- Exemplo mais comum
- Funcionamento
 - O primeiro pacote a entrar na fila é o primeiro a ser tratado
 - Pacotes são transmitidos na **ordem de chegada** à fila



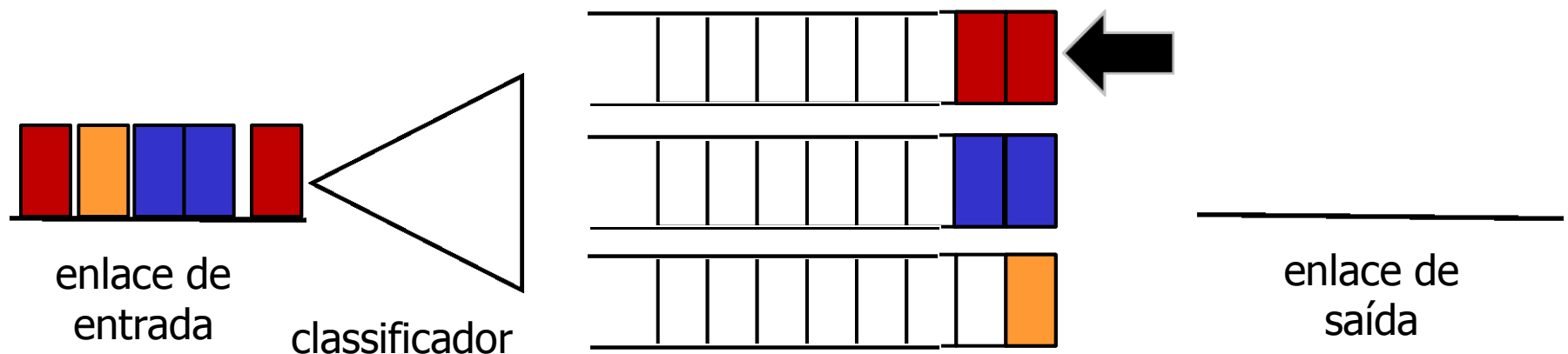
FIFO (*First In First Out*)

- Políticas de descarte
 - Se os pacotes ao chegarem encontrarem a fila cheia: quem deve ser descartado?
 - Descarta o último (*tail drop*) : descarta o pacote que acabou de chegar
 - Prioridade: descarta/remove baseado na prioridade
 - Aleatório: descarta/remove aleatoriamente



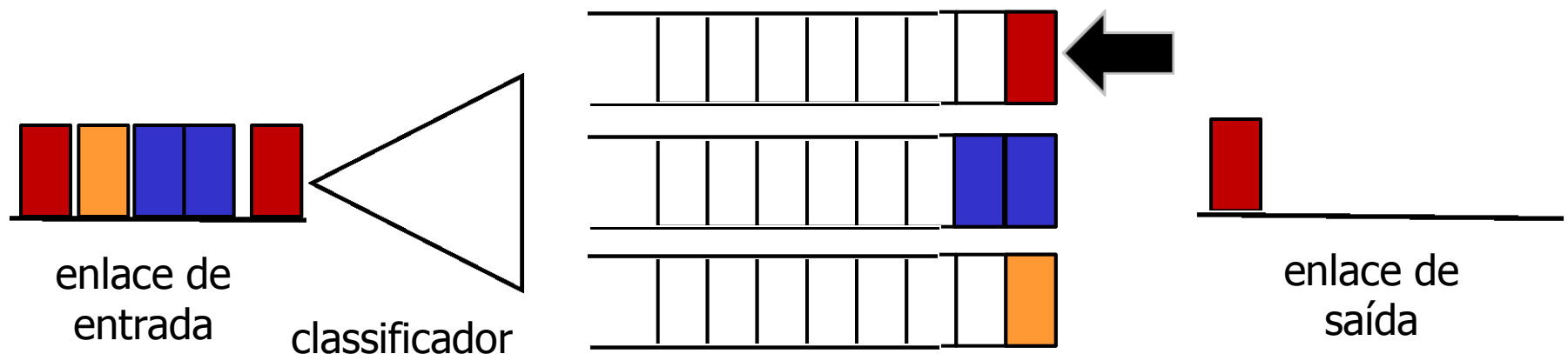
Escalonamento com Prioridades

- Pacotes enfileirados com prioridade mais alta são transmitidos primeiro
- Múltiplas classes com **diferentes prioridades**
 - Classes identificadas pela marcação ou por outra informação do cabeçalho
 - Ex.: endereços IP de origem e destino, números de porta de origem e destino, etc.



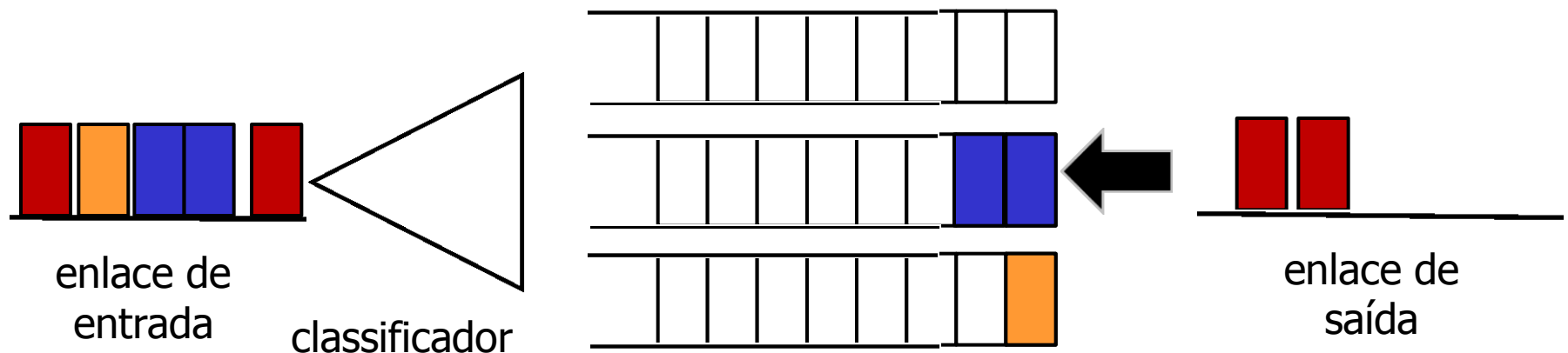
Escalonamento com Prioridades

- Pacotes enfileirados com prioridade mais alta são transmitidos primeiro
- Múltiplas classes com **diferentes prioridades**
 - Classes identificadas pela marcação ou por outra informação do cabeçalho
 - Ex.: endereços IP de origem e destino, números de porta de origem e destino, etc.



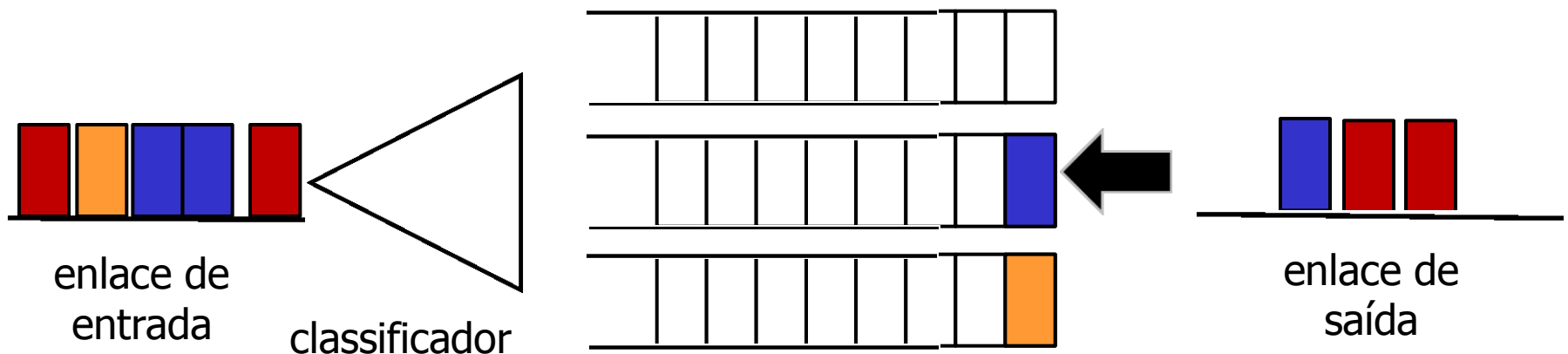
Escalonamento com Prioridades

- Pacotes enfileirados com prioridade mais alta são transmitidos primeiro
- Múltiplas classes com **diferentes prioridades**
 - Classes identificadas pela marcação ou por outra informação do cabeçalho
 - Ex.: endereços IP de origem e destino, números de porta de origem e destino, etc.



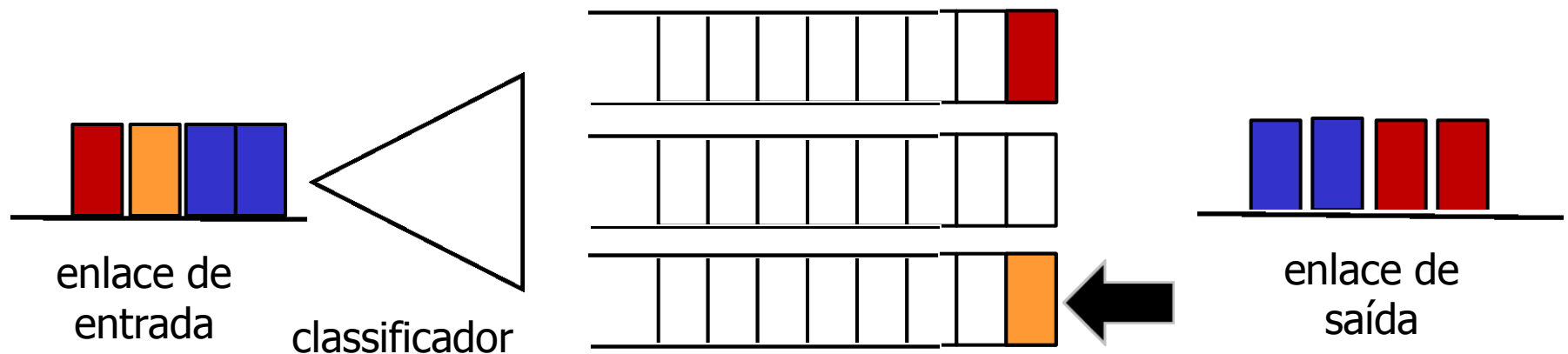
Escalonamento com Prioridades

- Pacotes enfileirados com prioridade mais alta são transmitidos primeiro
- Múltiplas classes com **diferentes prioridades**
 - Classes identificadas pela marcação ou por outra informação do cabeçalho
 - Ex.: endereços IP de origem e destino, números de porta de origem e destino, etc.



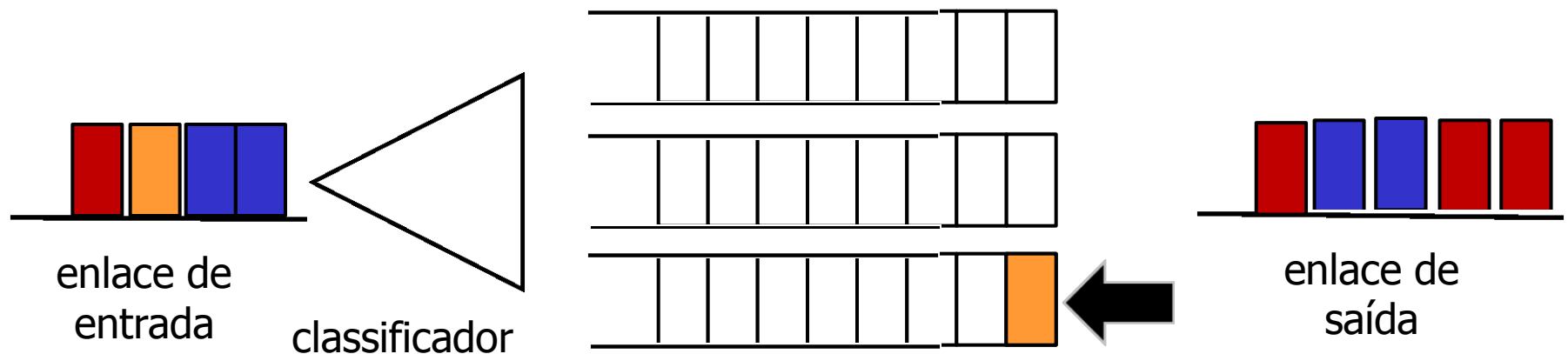
Escalonamento com Prioridades

- Pacotes enfileirados com prioridade mais alta são transmitidos primeiro
- Múltiplas classes com **diferentes prioridades**
 - Classes identificadas pela marcação ou por outra informação do cabeçalho
 - Ex.: endereços IP de origem e destino, números de porta de origem e destino, etc.



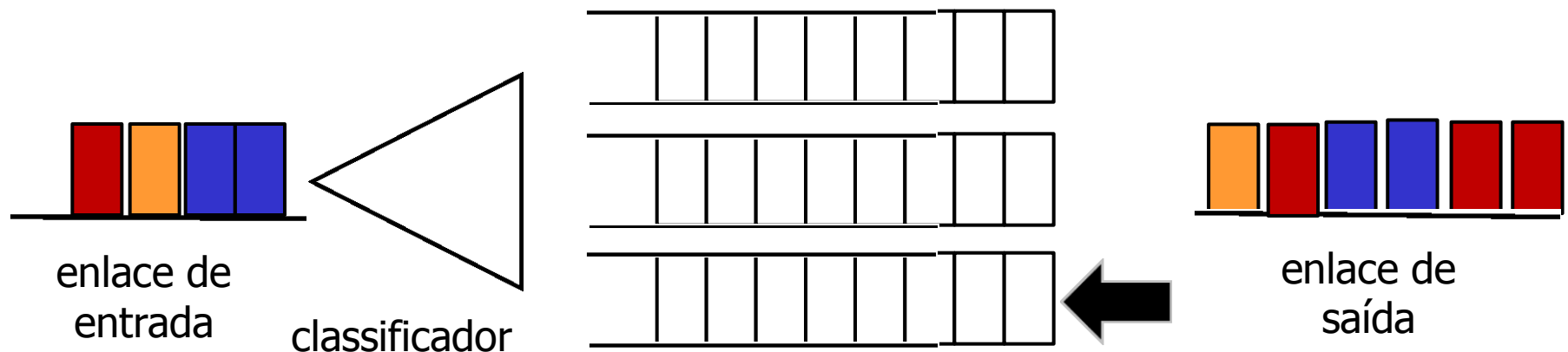
Escalonamento com Prioridades

- Pacotes enfileirados com prioridade mais alta são transmitidos primeiro
- Múltiplas classes com **diferentes prioridades**
 - Classes identificadas pela marcação ou por outra informação do cabeçalho
 - Ex.: endereços IP de origem e destino, números de porta de origem e destino, etc.



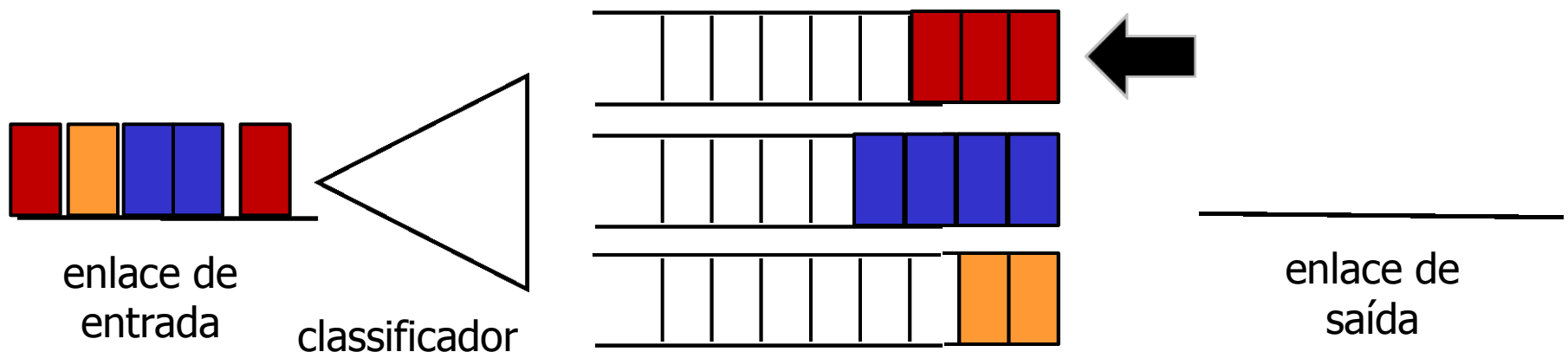
Escalonamento com Prioridades

- Pacotes enfileirados com prioridade mais alta são transmitidos primeiro
- Múltiplas classes com **diferentes prioridades**
 - Classes identificadas pela marcação ou por outra informação do cabeçalho
 - Ex.: endereços IP de origem e destino, números de porta de origem e destino, etc.



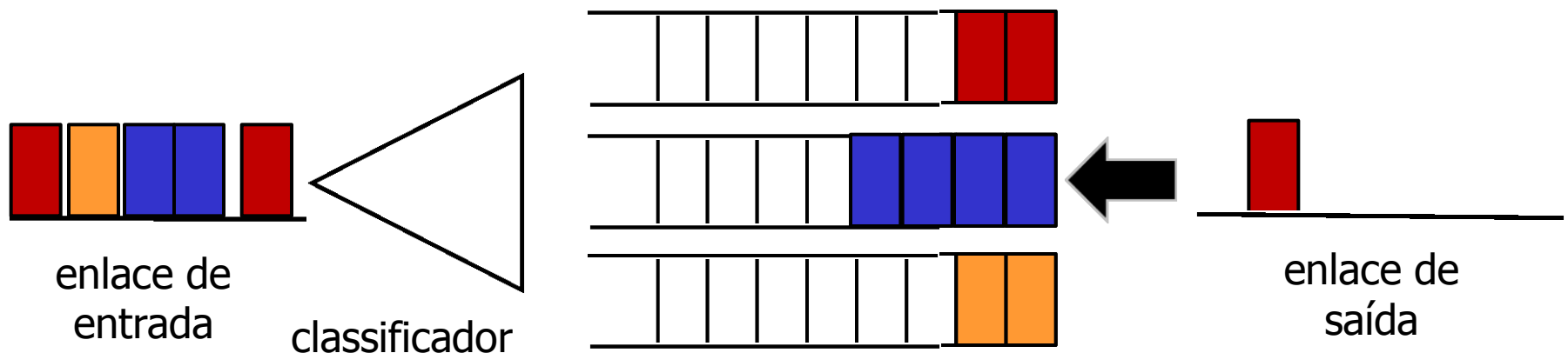
Varredura Cíclica (*Round Robin*)

- Usado quando existem múltiplas classes
- Varre as filas de cada classe
- Transmite um pacote por vez de cada classe cuja fila não estiver vazia



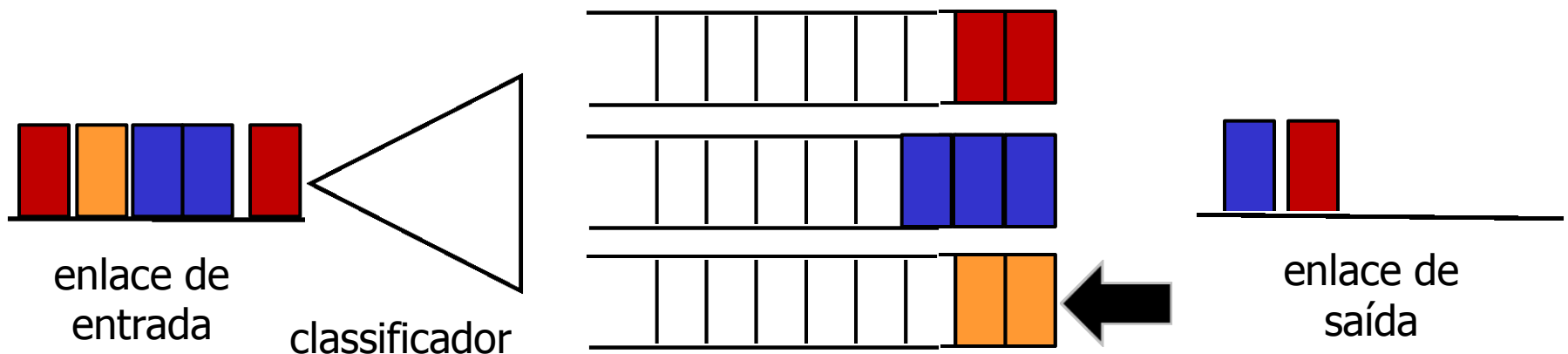
Varredura Cíclica (*Round Robin*)

- Usado quando existem múltiplas classes
- Varre as filas de cada classe
- Transmite um pacote por vez de cada classe cuja fila não estiver vazia



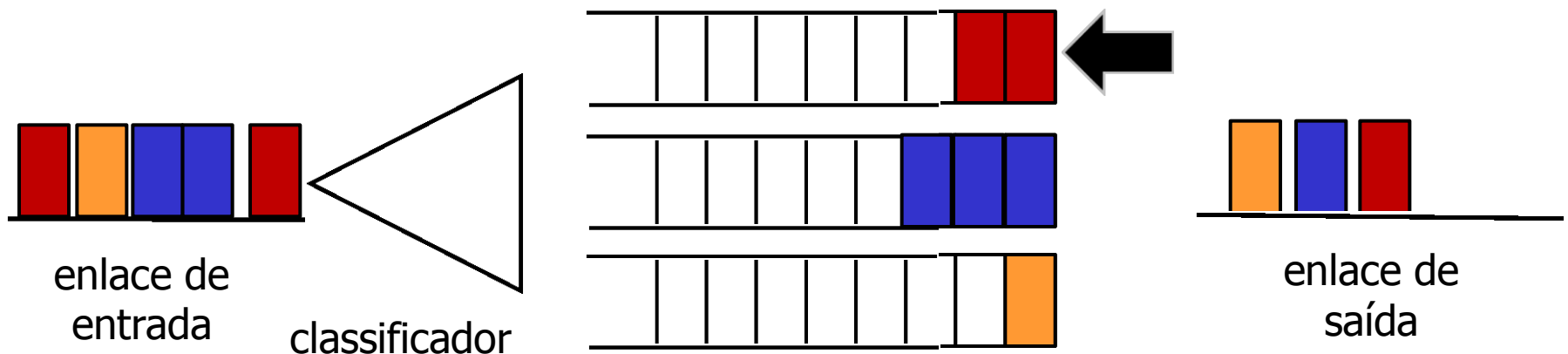
Varredura Cíclica (*Round Robin*)

- Usado quando existem múltiplas classes
- Varre as filas de cada classe
- Transmite um pacote por vez de cada classe cuja fila não estiver vazia



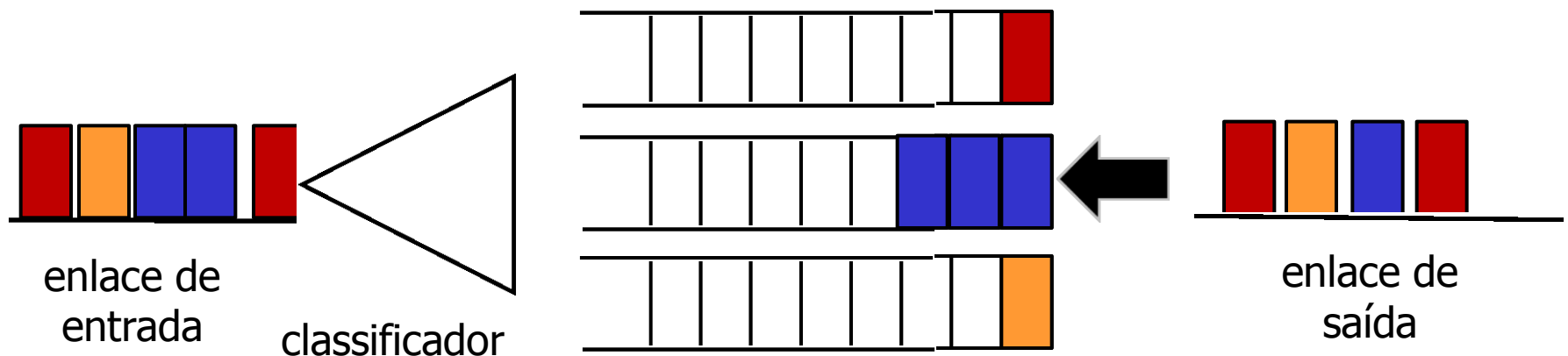
Varredura Cíclica (*Round Robin*)

- Usado quando existem múltiplas classes
- Varre as filas de cada classe
- Transmite um pacote por vez de cada classe cuja fila não estiver vazia



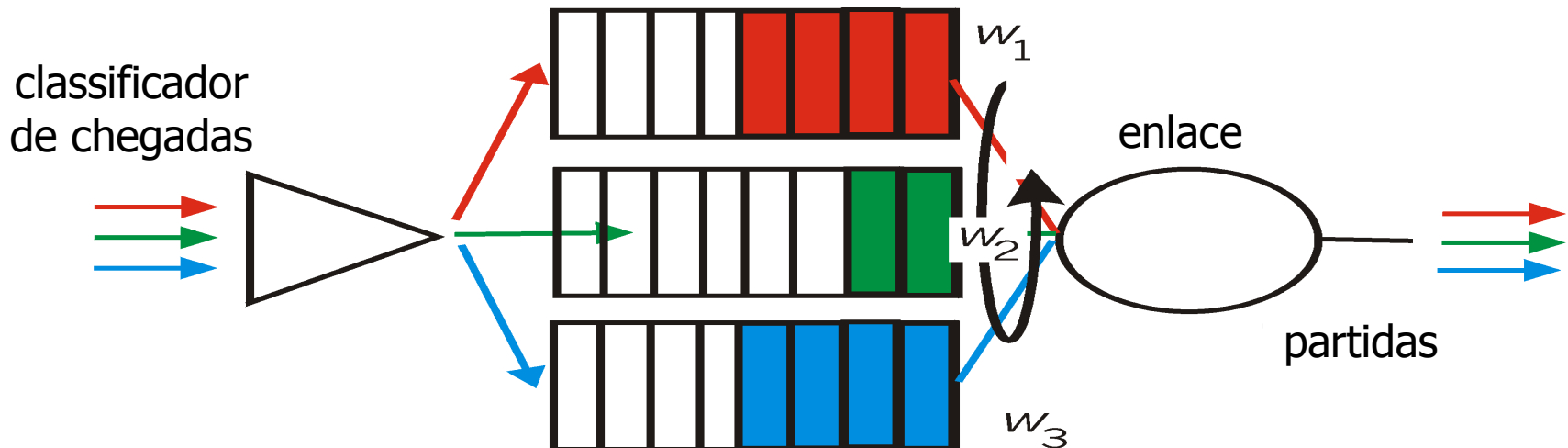
Varredura Cíclica (*Round Robin*)

- Usado quando existem múltiplas classes
- Varre as filas de cada classe
- Transmite um pacote por vez de cada classe cuja fila não estiver vazia



Enfileiramento Justo Ponderado

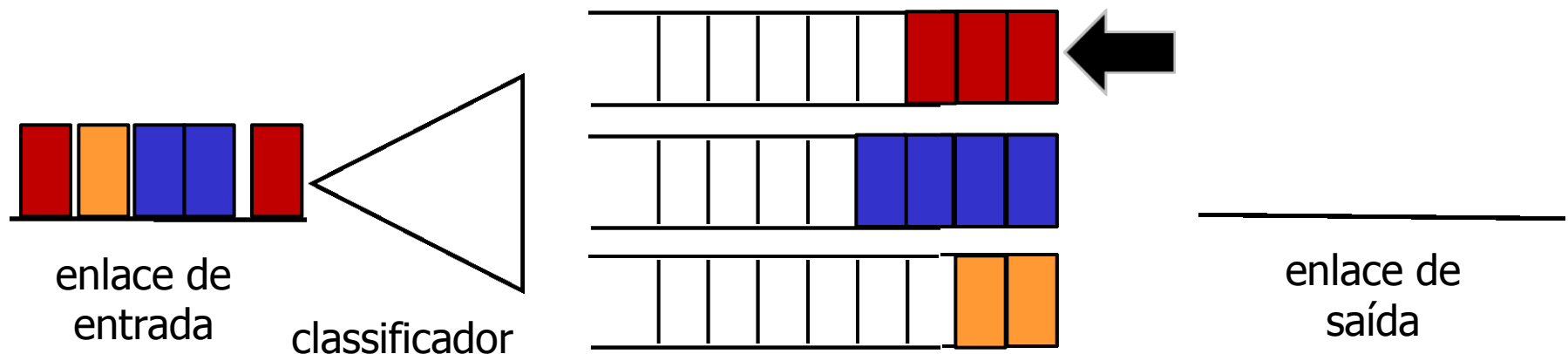
- *Weighted Fair Queuing* (WFQ)
- É uma generalização da varredura cíclica
 - Cada classe recebe um tempo de serviço diferenciado em cada ciclo



Enfileiramento Justo Ponderado

- *Weighted Fair Queuing* (WFQ)
- É uma generalização da varredura cíclica
 - Cada classe recebe um tempo de serviço diferenciado em cada ciclo

$$R = 10 \text{ pacotes/s} \quad w_0 = 6 \quad w_1 = 3 \quad w_2 = 1$$

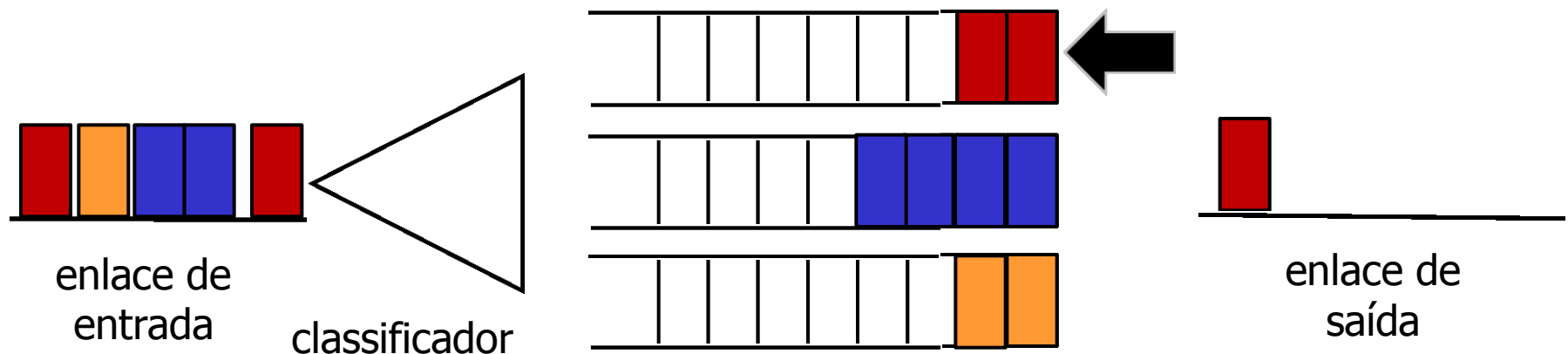


$$R * (w_0 / (w_0 + w_1 + w_2)) = R * (6 / (6 + 3 + 1)) = 0,6 * R = 6 \text{ pacotes/s}$$

Enfileiramento Justo Ponderado

- *Weighted Fair Queuing* (WFQ)
- É uma generalização da varredura cíclica
 - Cada classe recebe um tempo de serviço diferenciado em cada ciclo

$$R = 10 \text{ pacotes/s} \quad w_0 = 6 \quad w_1 = 3 \quad w_2 = 1$$

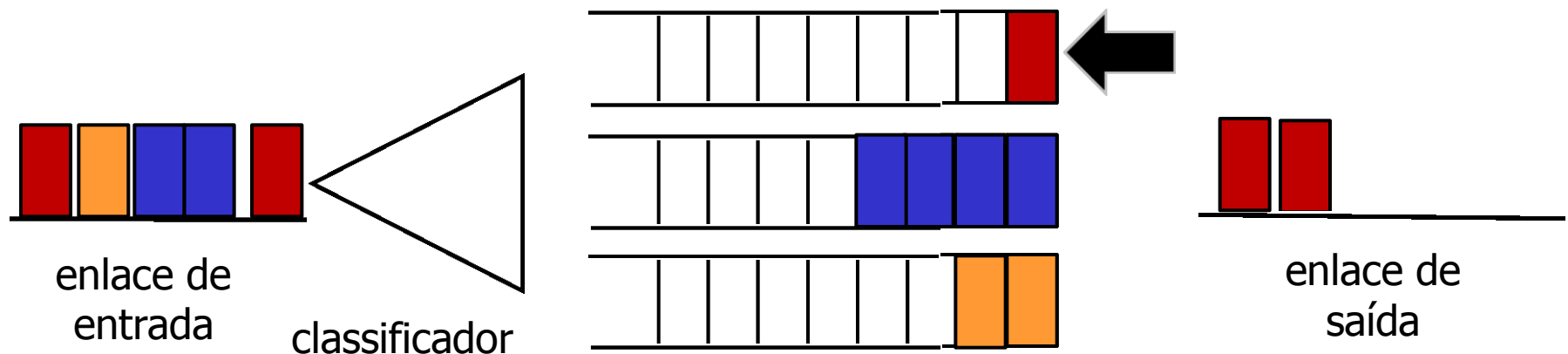


$$R * (w_0 / (w_0 + w_1 + w_2)) = R * (6 / (6 + 3 + 1)) = 0,6 * R = 6 \text{ pacotes/s}$$

Enfileiramento Justo Ponderado

- *Weighted Fair Queuing* (WFQ)
- É uma generalização da varredura cíclica
 - Cada classe recebe um tempo de serviço diferenciado em cada ciclo

$$R = 10 \text{ pacotes/s} \quad w_0 = 6 \quad w_1 = 3 \quad w_2 = 1$$

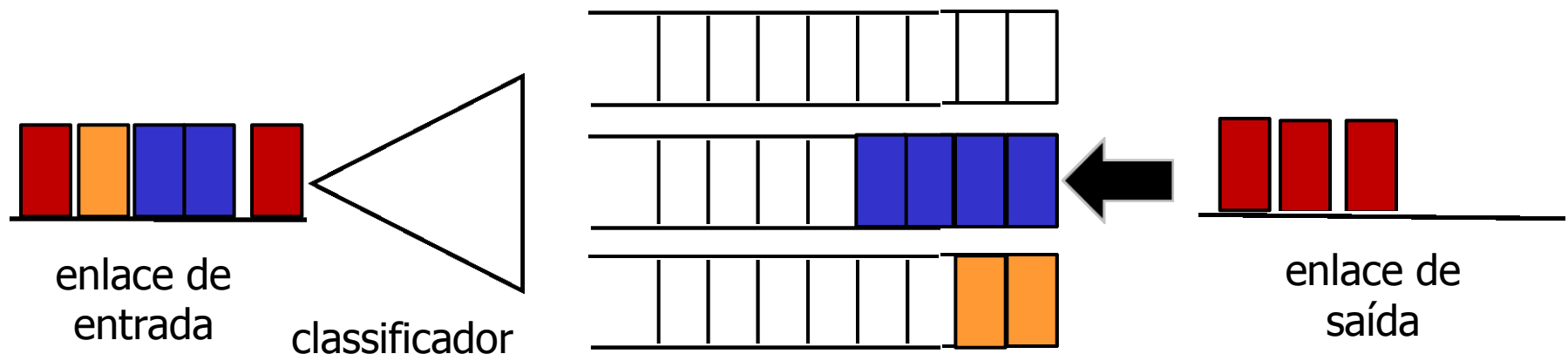


$$R * (w_0 / (w_0 + w_1 + w_2)) = R * (6 / (6 + 3 + 1)) = 0,6 * R = 6 \text{ pacotes/s}$$

Enfileiramento Justo Ponderado

- *Weighted Fair Queuing* (WFQ)
- É uma generalização da varredura cíclica
 - Cada classe recebe um tempo de serviço diferenciado em cada ciclo

$$R = 10 \text{ pacotes/s} \quad w_0 = 6 \quad w_1 = 3 \quad w_2 = 1$$

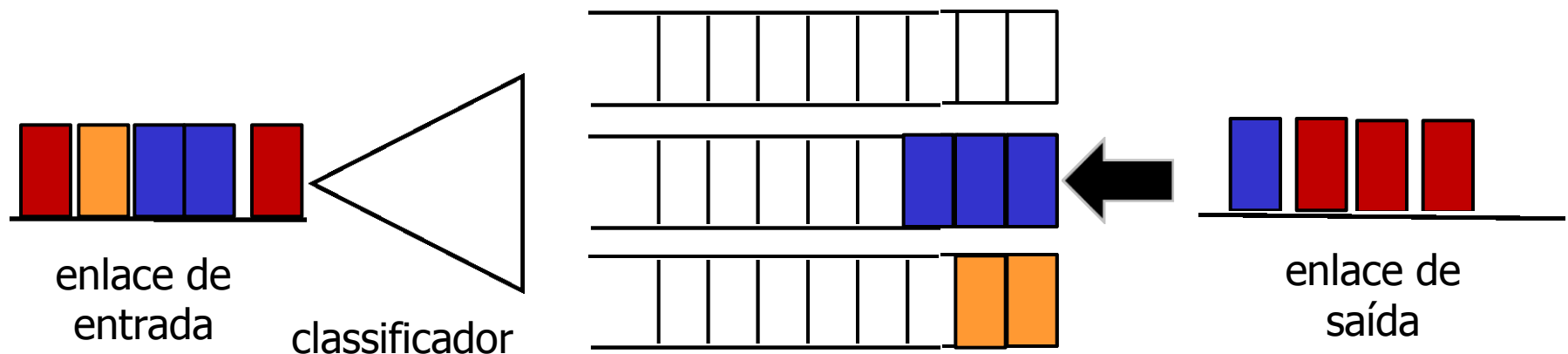


$$R * (w_0 / (w_0 + w_1 + w_2)) = R * (6 / (6 + 3 + 1)) = 0,6 * R = 6 \text{ pacotes/s}$$

Enfileiramento Justo Ponderado

- *Weighted Fair Queuing* (WFQ)
- É uma generalização da varredura cíclica
 - Cada classe recebe um tempo de serviço diferenciado em cada ciclo

$$R = 10 \text{ pacotes/s} \quad w_0 = 6 \quad w_1 = 3 \quad w_2 = 1$$

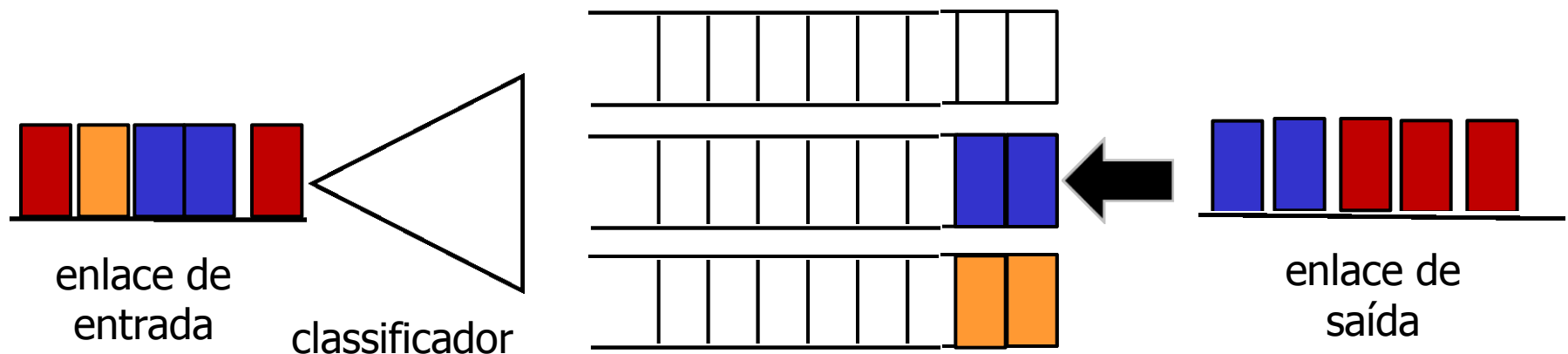


$$R * (w_1 / (w_0 + w_1 + w_2)) = R * (3 / (6 + 3 + 1)) = 0,3 * R = 3 \text{ pacotes/s}$$

Enfileiramento Justo Ponderado

- *Weighted Fair Queuing* (WFQ)
- É uma generalização da varredura cíclica
 - Cada classe recebe um tempo de serviço diferenciado em cada ciclo

$$R = 10 \text{ pacotes/s} \quad w_0 = 6 \quad w_1 = 3 \quad w_2 = 1$$

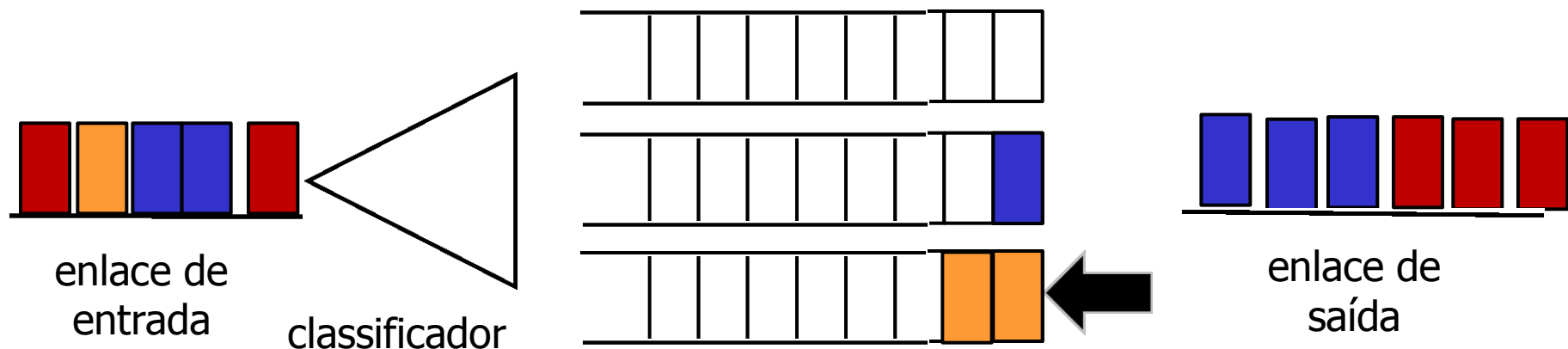


$$R * (w_1 / (w_0 + w_1 + w_2)) = R * (3 / (6 + 3 + 1)) = 0,3 * R = 3 \text{ pacotes/s}$$

Enfileiramento Justo Ponderado

- *Weighted Fair Queuing* (WFQ)
- É uma generalização da varredura cíclica
 - Cada classe recebe um tempo de serviço diferenciado em cada ciclo

$$R = 10 \text{ pacotes/s} \quad w_0 = 6 \quad w_1 = 3 \quad w_2 = 1$$

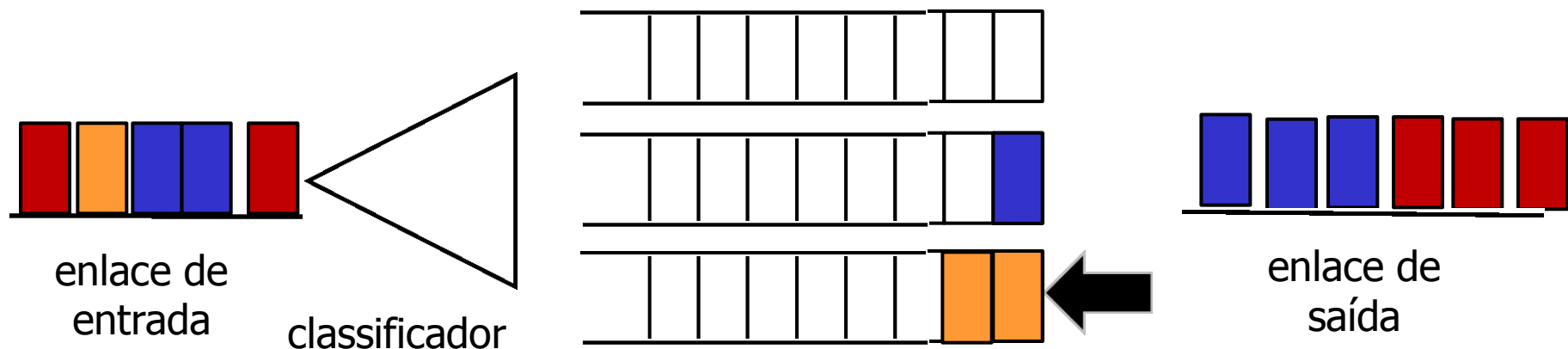


$$R * (w_1 / (w_0 + w_1 + w_2)) = R * (3 / (6 + 3 + 1)) = 0,3 * R = 3 \text{ pacotes/s}$$

Enfileiramento Justo Ponderado

- *Weighted Fair Queuing* (WFQ)
- É uma generalização da varredura cíclica
 - Cada classe recebe um tempo de serviço diferenciado em cada ciclo

$$R = 10 \text{ pacotes/s} \quad w_0 = 6 \quad w_1 = 3 \quad w_2 = 1$$

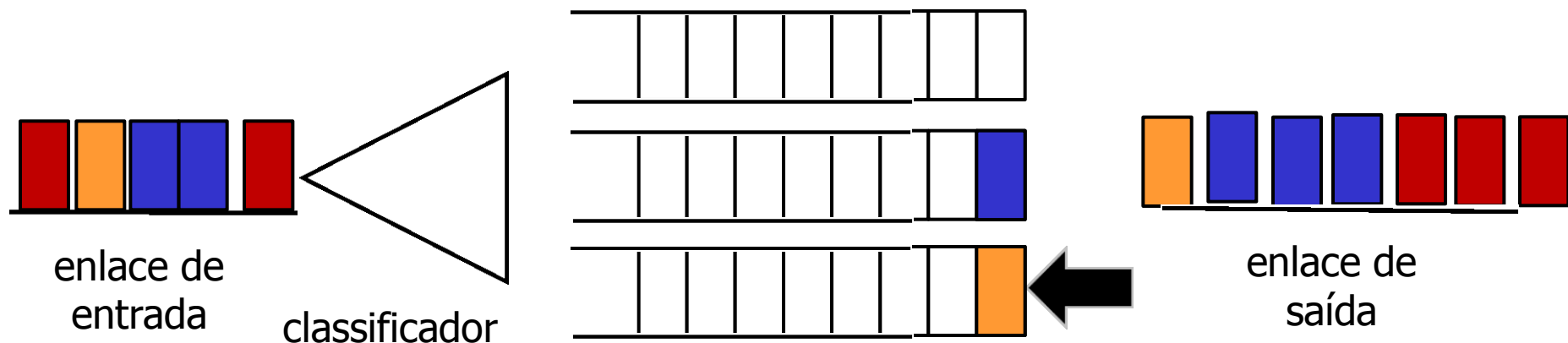


$$R * (w_2 / (w_0 + w_1 + w_2)) = R * (1 / (6 + 3 + 1)) = 0,1 * R = 1 \text{ pacote/s}$$

Enfileiramento Justo Ponderado

- *Weighted Fair Queuing* (WFQ)
- É uma generalização da varredura cíclica
 - Cada classe recebe um tempo de serviço diferenciado em cada ciclo

$$R = 10 \text{ pacotes/s} \quad w_0 = 6 \quad w_1 = 3 \quad w_2 = 1$$

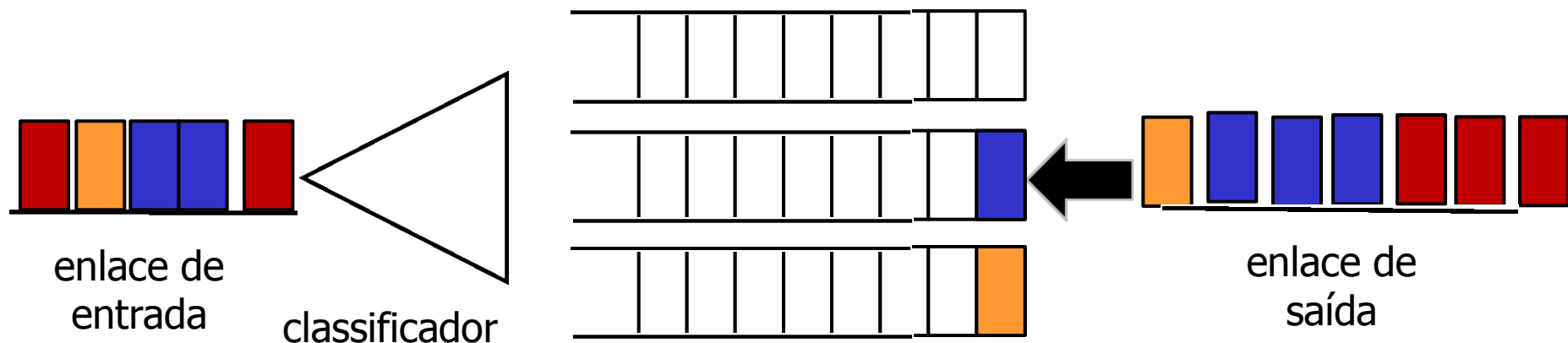


$$R * (w_2 / (w_0 + w_1 + w_2)) = R * (1 / (6 + 3 + 1)) = 0,1 * R = 1 \text{ pacote/s}$$

Enfileiramento Justo Ponderado

- *Weighted Fair Queuing* (WFQ)
- É uma generalização da varredura cíclica
 - Cada classe recebe um tempo de serviço diferenciado em cada ciclo

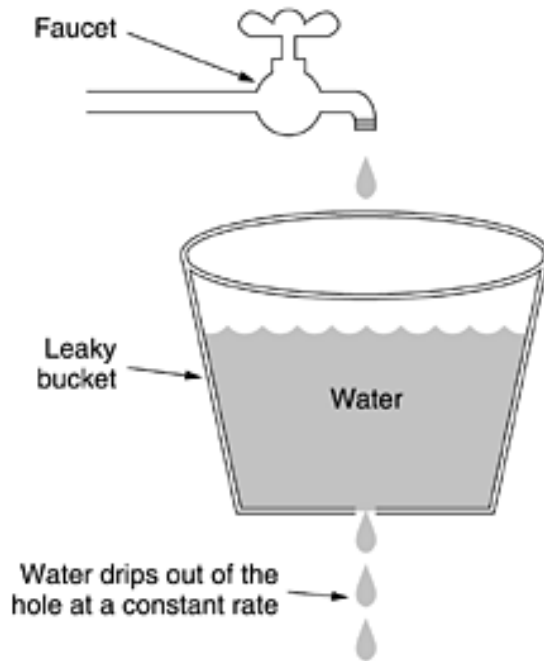
$$R = 10 \text{ pacotes/s} \quad w_0 = 6 \quad w_1 = 3 \quad w_2 = 1$$



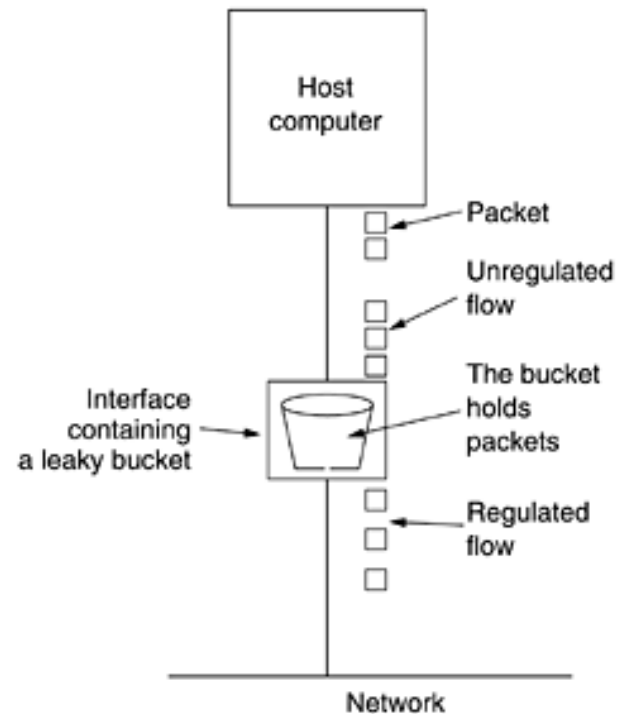
- Mecanismos de policiamento
 - Limitam o tráfego para que ele não exceda os parâmetros acordados
 - Também chamados de modeladores de tráfego (*shapers*)
- Exemplos
 - Balde furado (*Leaky bucket*)
 - Pacotes enviados a taxa constante: regula a taxa média
 - Quando o balde está cheio, pacotes podem ser descartados
 - Balde de permissões (*Token bucket*)
 - Um pacote é transmitido quando recebe uma permissão
 - Quando o balde está cheio, permissões são descartadas
 - Etc

Leaky Bucket

- Balde furado
 - Regula a taxa de envio dos pacotes (“tamanho do furo”)



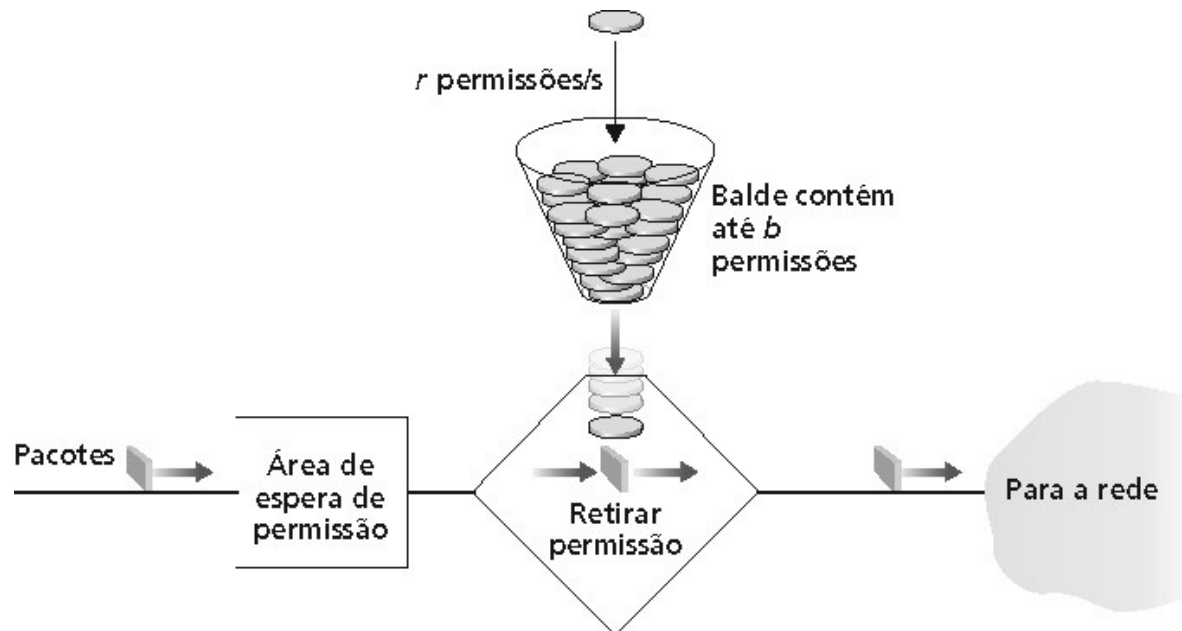
(a)



(b)

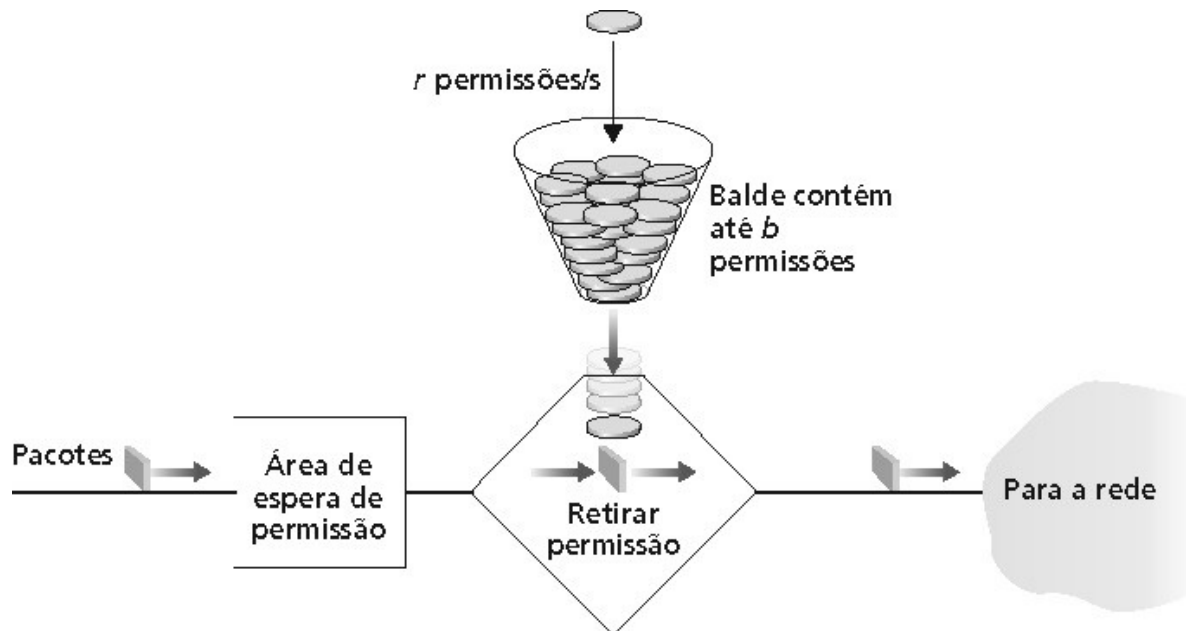
Token Bucket

- Balde de permissões
 - Limita a entrada para o tamanho da rajada e taxa média especificadas



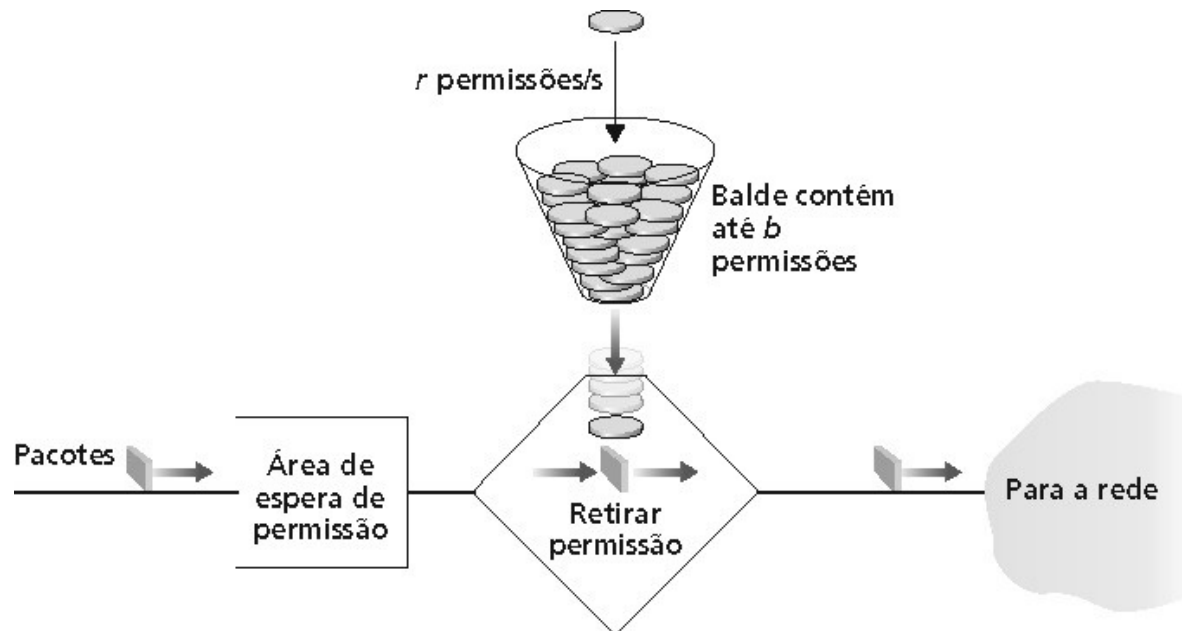
Token Bucket

- Funcionamento
 - Balde pode guardar b permissões (*tokens*)
 - Permissões são geradas a uma taxa de r permissões por segundo a menos que o balde esteja cheio
 - Em um intervalo de comprimento t
 - Número de pacotes admitidos é menor ou igual a $rt + b$



Token Bucket

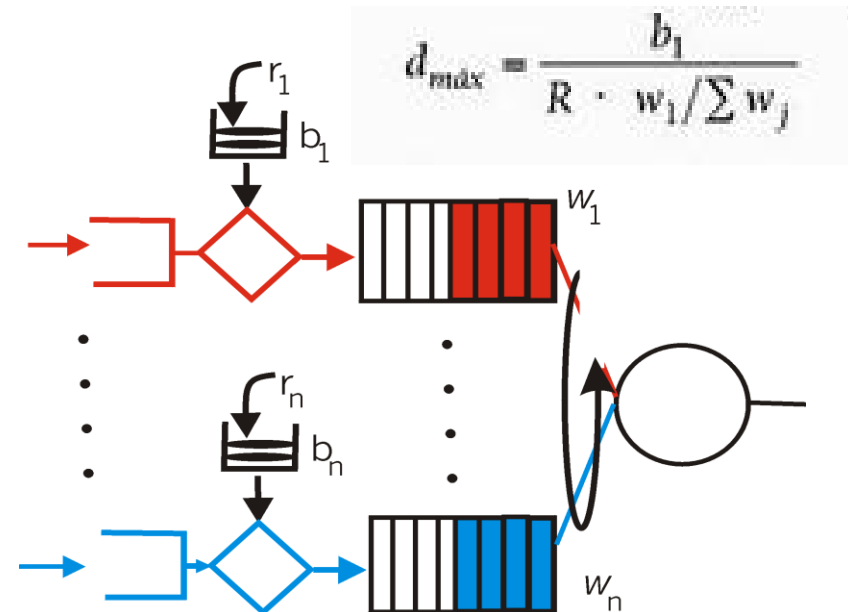
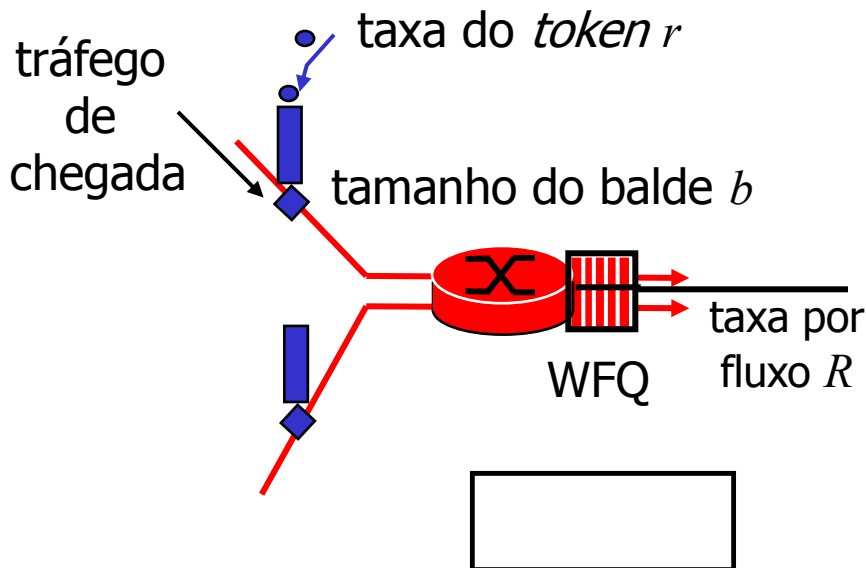
- Resultado
 - Taxa média de r permissões/s
 - Rajada de até b permissões
 - Taxa de pico de até $rt + b$ permissões/s



Balde furado	Balde de permissões
Não depende de permissões	Depende de permissões
Se o balde está cheio, pacotes são descartados	Se o balde está cheio, permissões são descartadas, mas não os pacotes (dados)
Pacotes são transmitidos continuamente	Pacotes só são transmitidos se existem permissões disponíveis
Não armazena permissões	Armazena permissões para permitir o envio de rajadas
Pacotes são enviados em taxa constante	Rajadas de pacotes podem ser enviadas em uma taxa maior do que a média

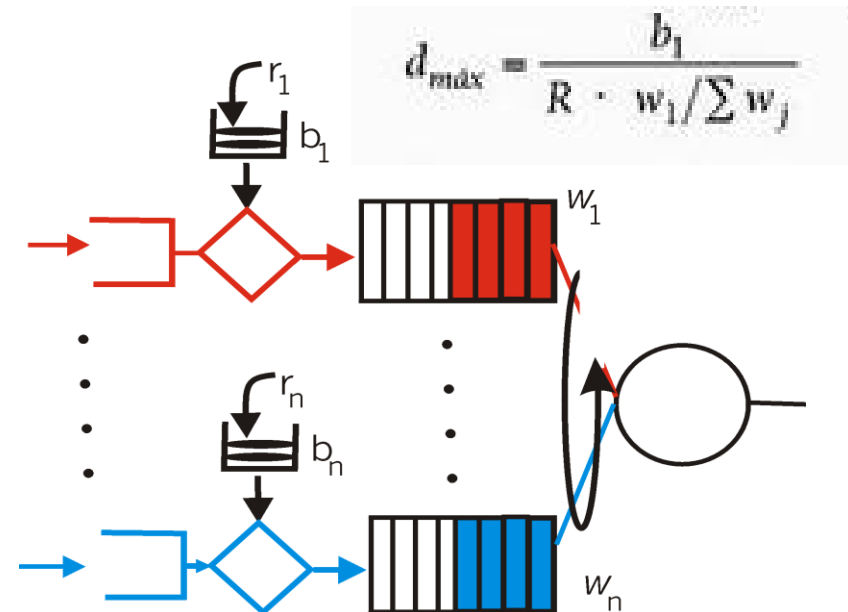
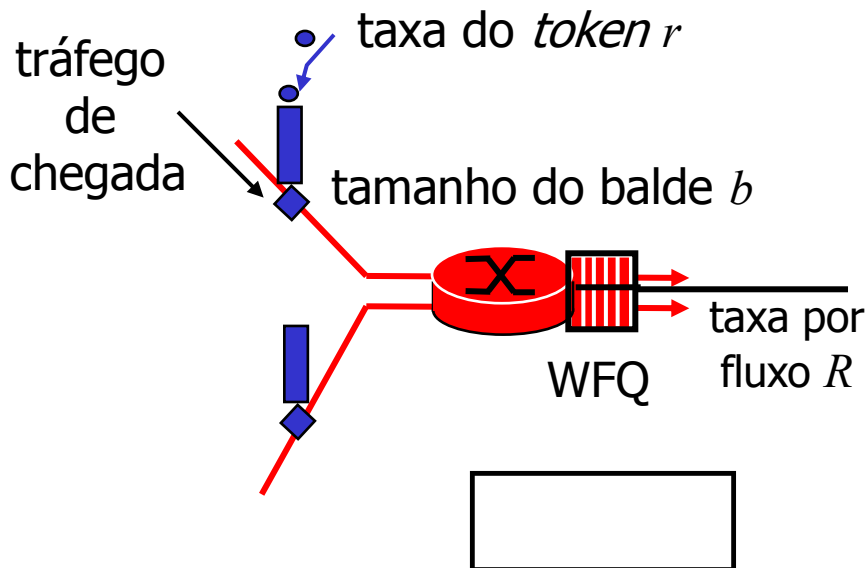
Token Bucket + WFQ

- Combinação do *token bucket* com o WFQ
 - Qual o tamanho máximo de uma rajada?
 - Qual a taxa mínima de atendimento dos pacotes de uma classe?



Token Bucket + WFQ

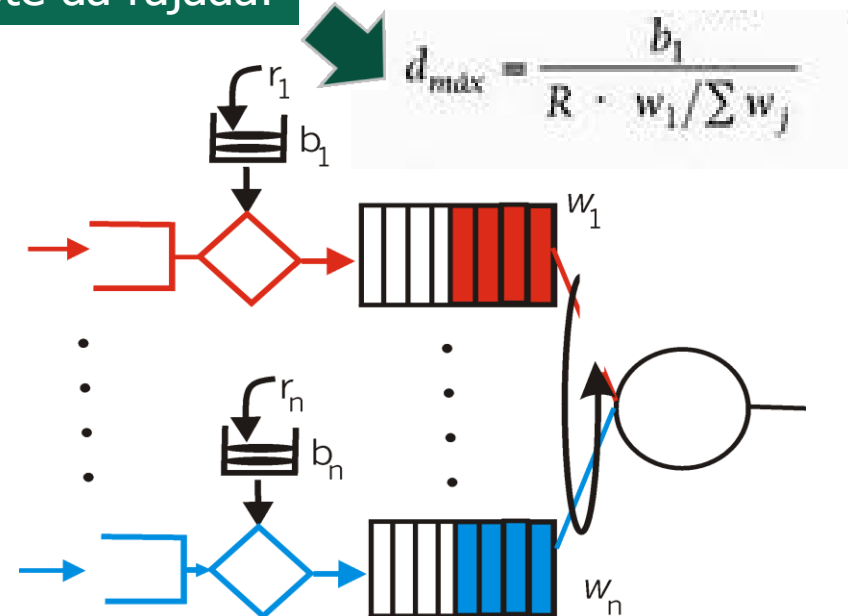
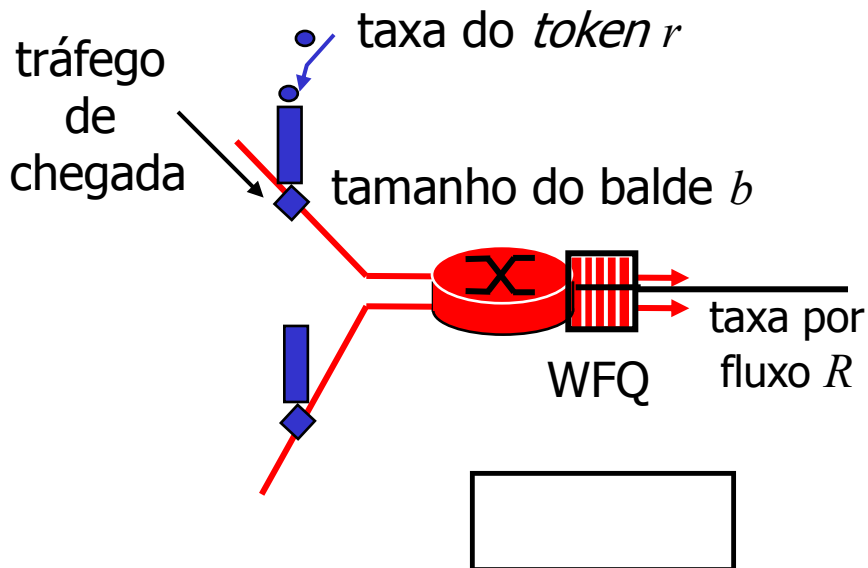
- Combinação do *token bucket* com o WFQ
 - Qual o tamanho máximo de uma rajada de uma classe? b_i
 - Qual a taxa mínima de atendimento dos pacotes de uma classe? $R \cdot w_i / \sum w_j$



Token Bucket + WFQ

- Combinação do *token bucket* com o WFQ
 - Provê um limite superior garantido no atraso → **garantia de QoS!**

Tempo de espera em fila
do último pacote da rajada!



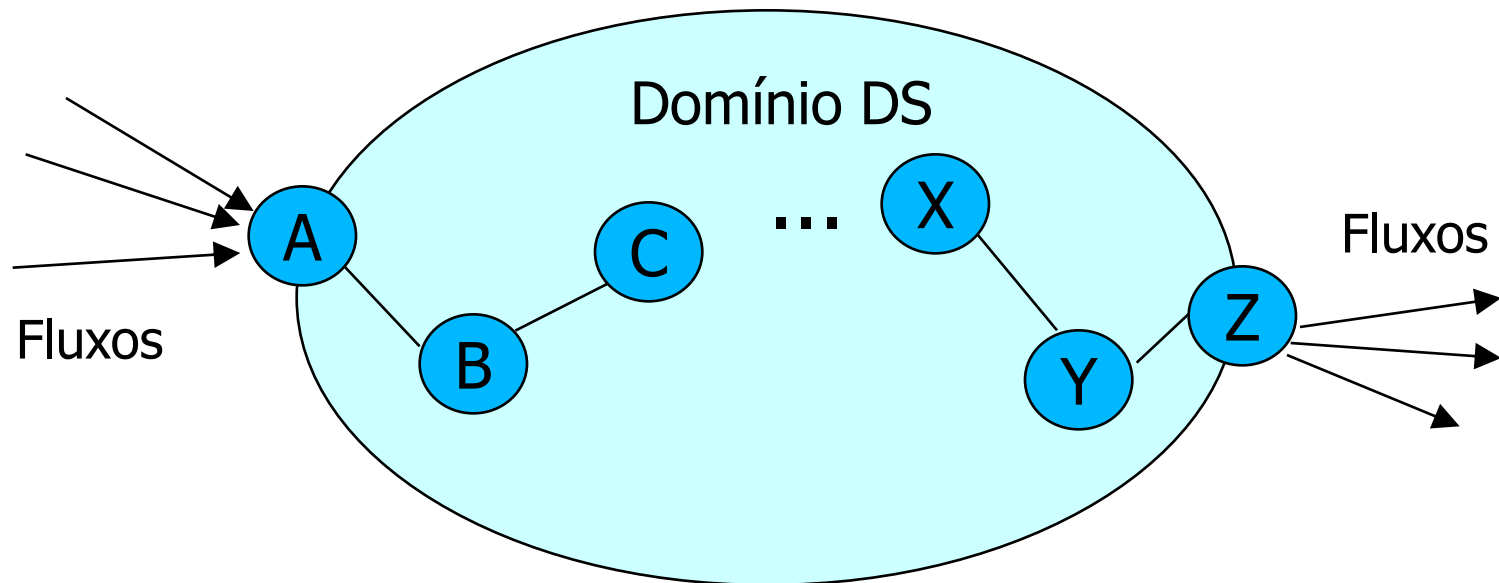
Arquitetura de Serviços Diferenciados (DiffServ)

Serviços Diferenciados do IETF



- Provê qualidade de serviço **qualitativa**
 - Diferenciação **relativa** entre as classes de serviço
- Principal vantagem: **escalabilidade**
 - Funções simples executadas no núcleo da rede
 - Funções mais complexas nos roteadores de borda
 - Ou até nos sistemas finais
 - Sinalização, manutenção do estado do roteador por fluxo é difícil para um grande número de fluxos
- Não define as classes de serviço
 - Provê componentes funcionais para construir as classes de serviço

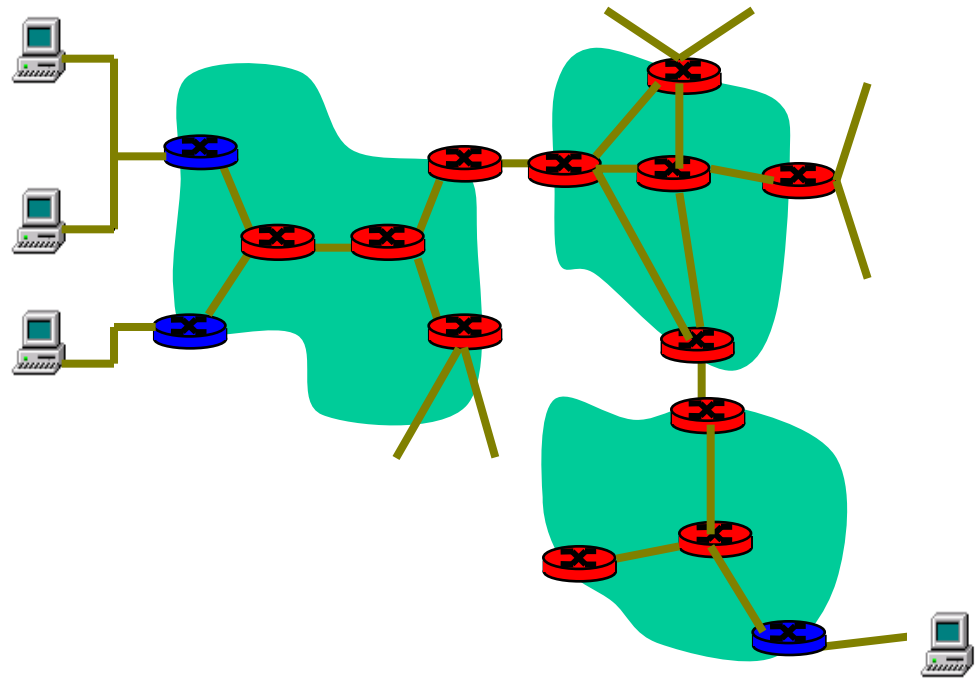
Arquitetura DiffServ

- Domínio DiffServ
 - Serviços diferenciados oferecidos dentro de um domínio
 - Agregação de fluxos
 - Mais um exemplo de complexidade nas bordas





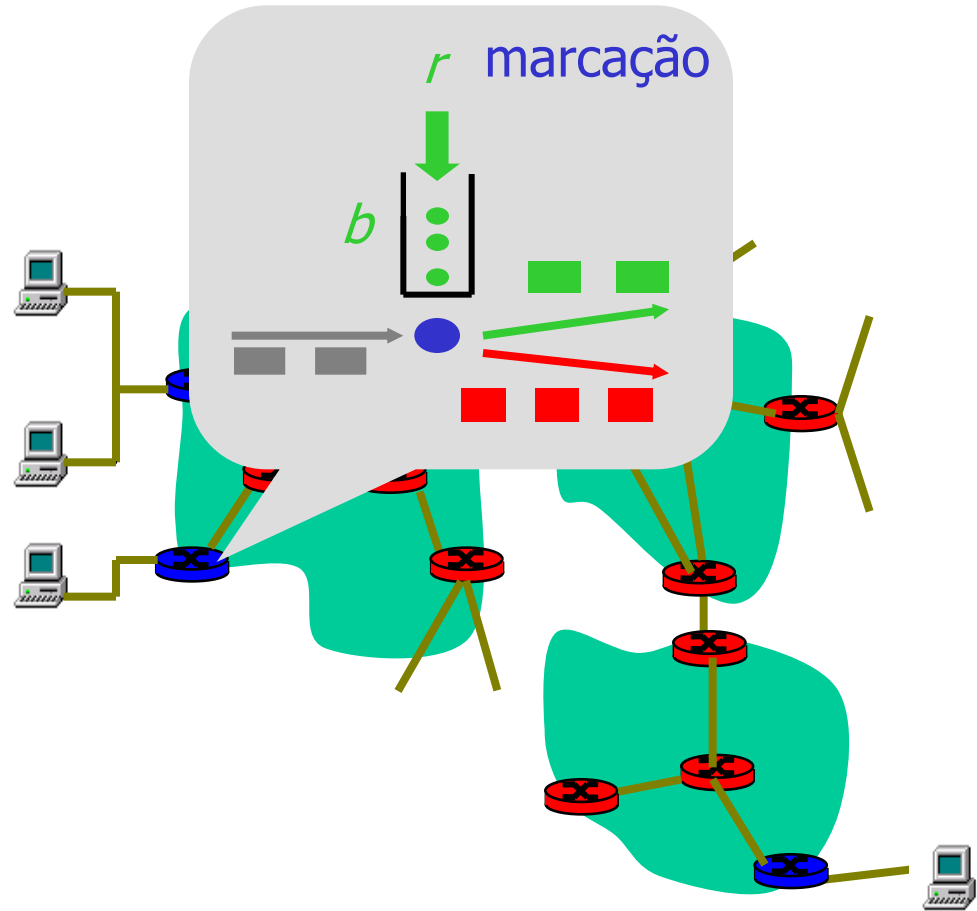
Arquitetura DiffServ

- Roteador de **borda** 
 - Gerenciamento de tráfego **por fluxo**
 - Marca os pacotes com **dentro** e **fora** do perfil
- Roteador de **núcleo** 
 - Gerenciamento de tráfego **por classe**
 - Enfileiramento e escalonamento baseados nas **marcações** da borda
 - Preferência para os pacotes **dentro** do perfil





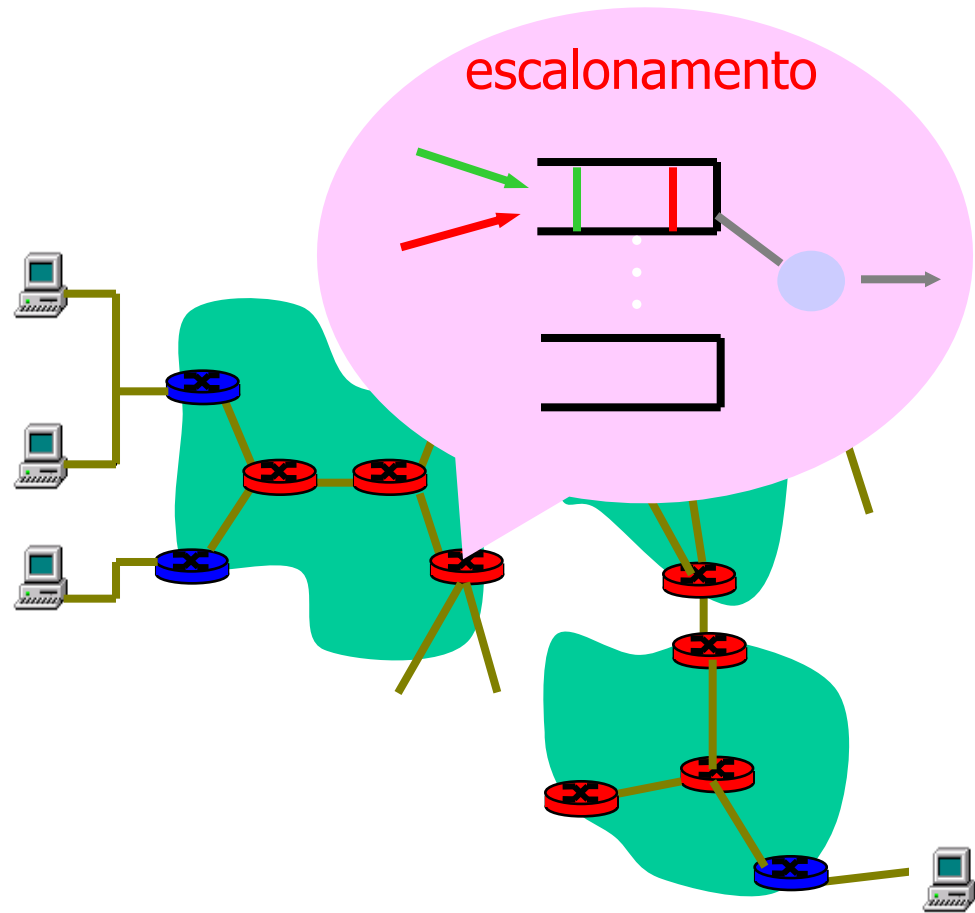
Arquitetura DiffServ

- Roteador de **borda** 
 - Gerenciamento de tráfego **por fluxo**
 - Marca os pacotes com **dentro** e **fora** do perfil
- Roteador de **núcleo** 
 - Gerenciamento de tráfego **por classe**
 - Enfileiramento e escalonamento baseados nas **marcações** da borda
 - Preferência para os pacotes **dentro** do perfil



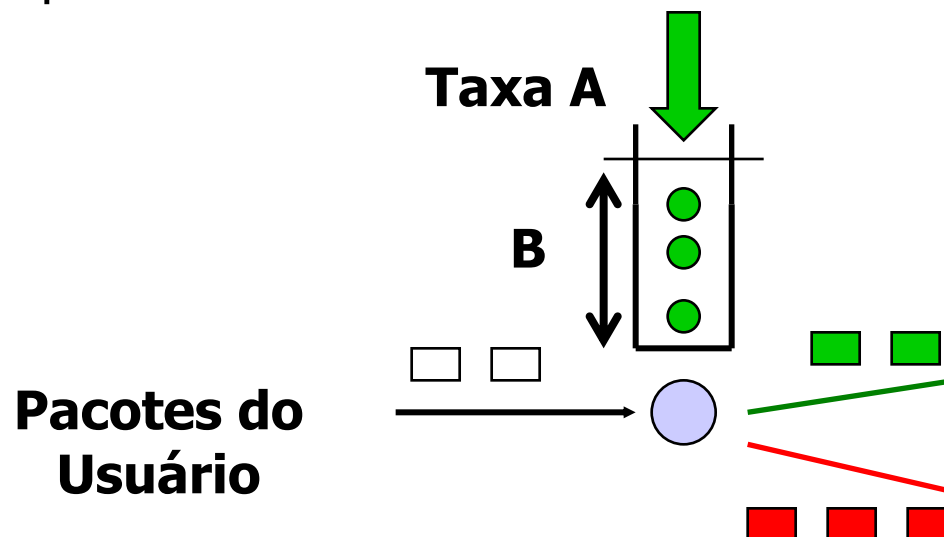
Arquitetura DiffServ

- Roteador de **borda** 
 - Gerenciamento de tráfego **por fluxo**
 - Marca os pacotes com **dentro** e **fora** do perfil
- Roteador de **núcleo** 
 - Gerenciamento de tráfego **por classe**
 - Enfileiramento e escalonamento baseados nas **marcações** da borda
 - Preferência para os pacotes **dentro** do perfil



Marcação nos Roteadores de Borda

- Possíveis usos da marcação
 - Marcação baseada em classes
 - Pacotes de classes diferentes recebem marcas diferentes
 - Marcação intraclasse
 - Porção do fluxo bem comportado marcado diferentemente do mal comportado

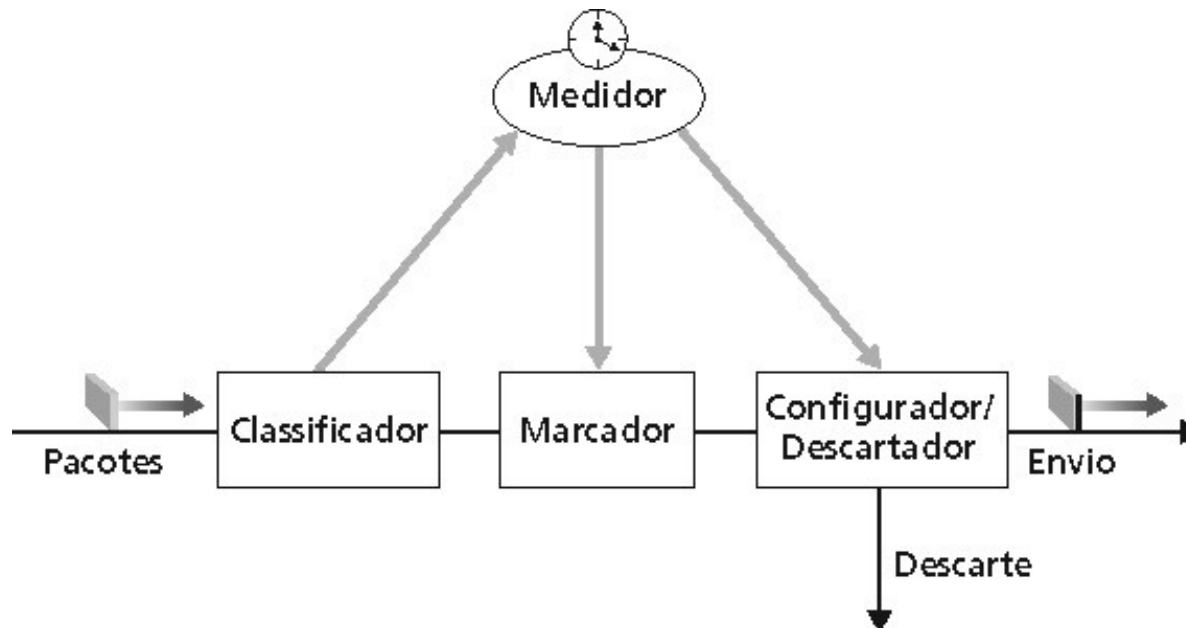


Classificação

- Campos do cabeçalho IP usados para marcar os pacotes
 - Tipo de Serviços (TOS) no IPv4
 - Classe de tráfego, no IPv6 } Ambos possuem 8 bits
- São mapeados em um novos campos específicos para a arquitetura DiffServ
 - *Differentiated Service Code Point* (DSCP): 6 bits
 - Determina o tratamento salto-a-salto (PHB) que o pacote irá receber dos roteadores intermediários
 - 2 bits atualmente não são usados



- Pode ser necessário limitar a taxa de entrada de uma dada classe de serviço
 - Usuário deve especificar o **perfil de tráfego** acordado
 - Ex.: taxa, tamanho da rajada, etc.
 - Tráfego é medido e moldado se está fora de conformidade



Encaminhamento (PHB)

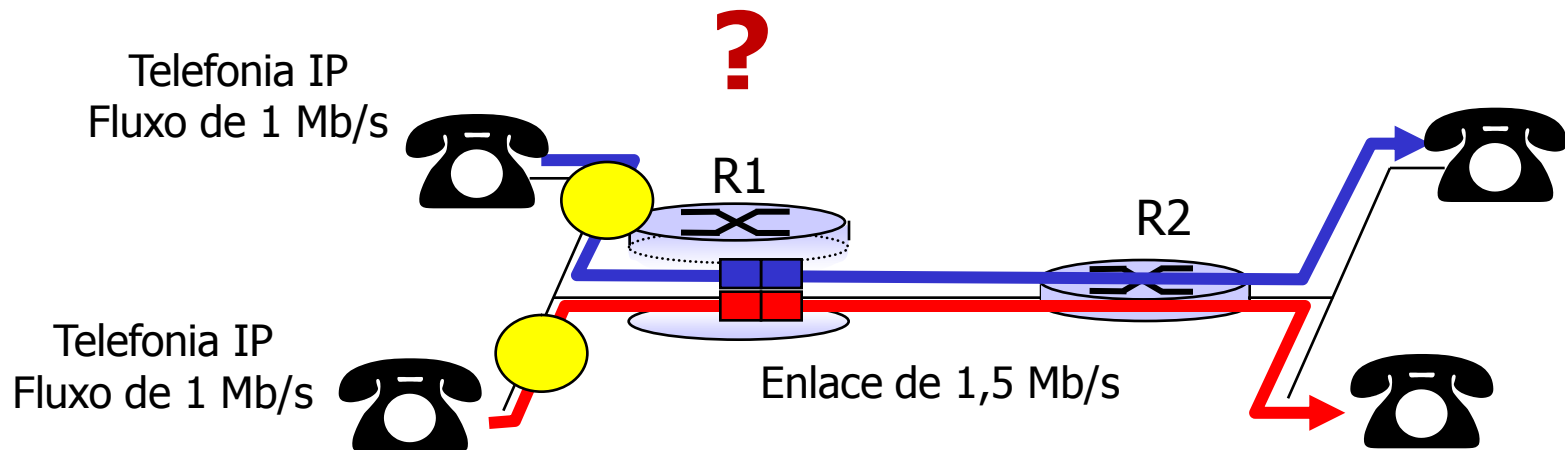
- Chamado de *Per-Hop Behavior* (PHB) na arquitetura Diffserv
 - Definem o tratamento dos pacotes a cada roteador intermediário
- Cada PHB possui um desempenho mensurável de encaminhamento diferente
- Um PHB não especifica quais os mecanismos a serem usados para garantir o desempenho desejado para o PHB
- Exemplos de PHBs
 - Classe A usa $x\%$ da banda do enlace de saída durante intervalos de tempo de duração especificada
 - Pacotes da Classe A são encaminhados antes de pacotes da Classe B

- PHBs já definidos
 - Encaminhamento Expresso (*Expedited Forwarding* - EF)
 - Taxa de envio dos pacotes de uma classe é maior ou igual a uma taxa especificada
 - Enlace lógico com uma taxa mínima garantida
 - Encaminhamento assegurado (*Assured Forwarding* - AF)
 - Quatro classes de tráfego
 - Para cada uma é garantida uma quantidade mínima de largura de banda
 - Cada uma com três partições de preferência para o descarte

Arquitetura de Serviços Integrados (IntServ)

Princípios para Garantia de QoS

- Fato
 - Não é possível atender uma demanda de tráfego maior do que a capacidade do enlace



Princípios para Garantia de QoS

- Fato
 - Não é possível atender uma demanda de tráfego maior do que a capacidade do enlace

Princípio 4:

- É necessário ter um **controle de admissão**
 - Um fluxo declara o que precisa e requisita sua admissão
 - A rede pode bloquear uma requisição se ela não possui os recursos necessários para atendê-la
 - Ex. Envia um sinal de ocupado

Fluxo de 1 Mb/s

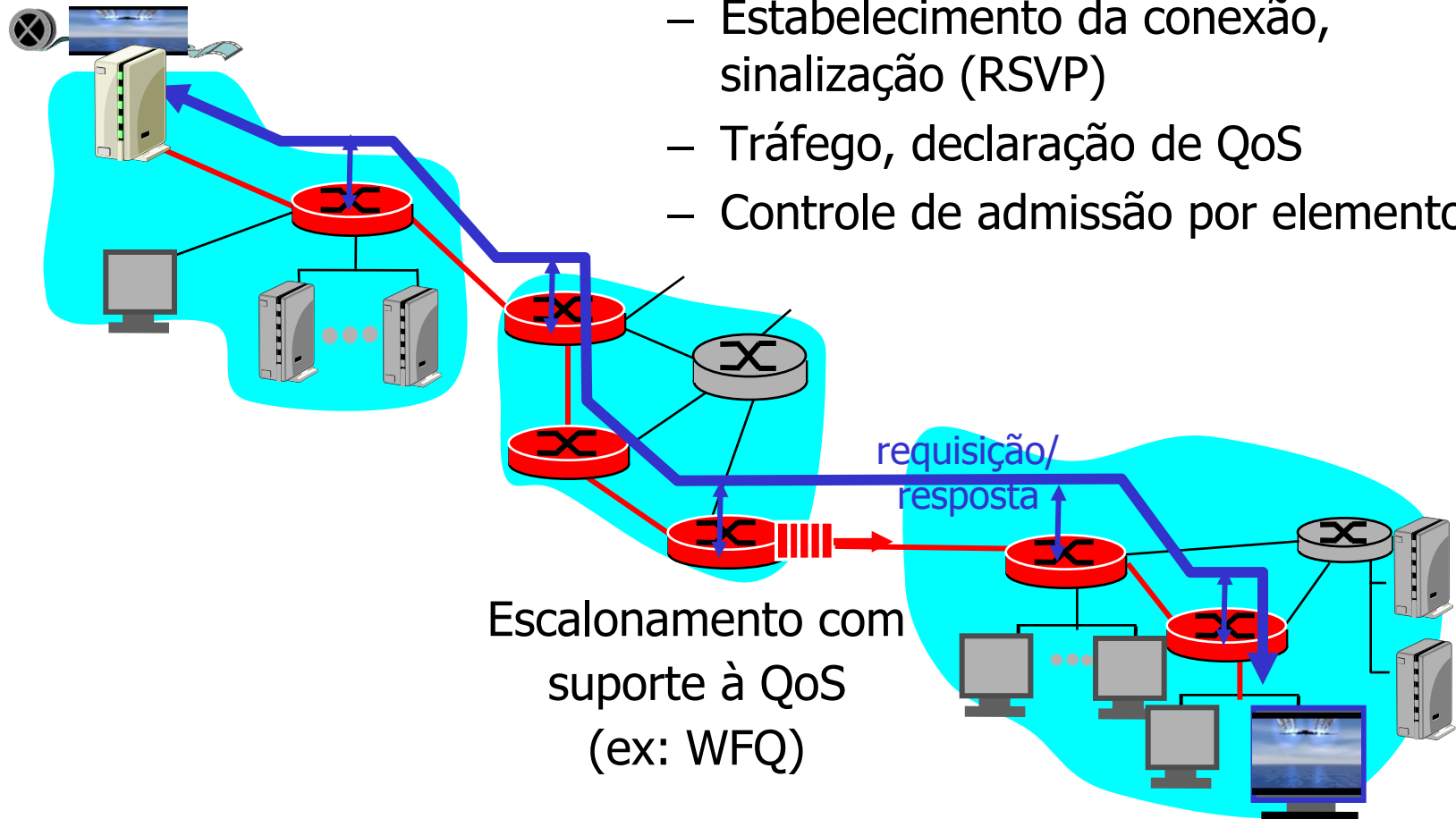


Enlace de 1 Mb/s



Cenário com QoS Garantida

- Reserva de recursos
 - Estabelecimento da conexão, sinalização (RSVP)
 - Tráfego, declaração de QoS
 - Controle de admissão por elemento



Serviços Integrados do IETF

- Provê qualidade de serviço **garantida**
 - Garante os requisitos de fluxos individuais
- Baseada na **reserva de recursos**
 - Roteadores devem manter
 - Informações de estado
 - Registros dos recursos alocados
 - Requisitos de QoS
- Possui **controle de admissão**
 - Admite/rejeita novos pedidos de “chamadas”
 - A admissão de novos fluxos pode violar as garantias de QoS de fluxos já admitidos?

Serviços Integrados do IETF

- Principal desvantagem: **não é escalável**
 - Sinalização, manutenção do estado do roteador por fluxo é difícil para um grande número de fluxos

- Um novo fluxo deve
 - Declarar os seus requisitos de QoS
 - R-spec: define a QoS que está sendo solicitada
 - Caracterizar o tráfego que injetará na rede
 - T-spec: define as características do tráfego
- É necessário um protocolo de sinalização
 - Informar a R-spec e T-spec aos roteadores que precisam reservar os recursos

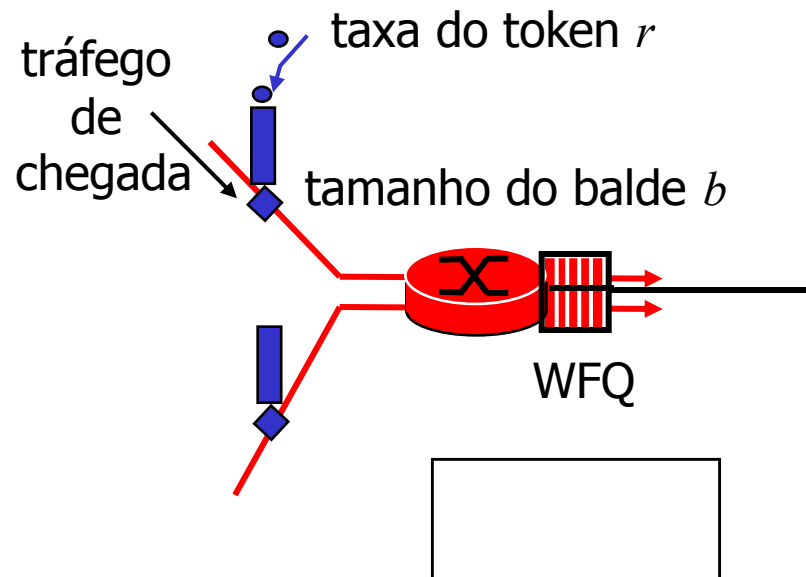
- Um novo fluxo deve
 - Declarar os seus requisitos de QoS
 - R-spec: define a QoS que está sendo solicitada
 - Caracterizar o tráfego que injetará na rede
 - T-spec: define as características do tráfego
- É necessário um protocolo de sinalização
 - Informar a R-spec e T-spec aos roteadores que precisam reservar os recursos



Resource Reservation Protocol (RSVP)

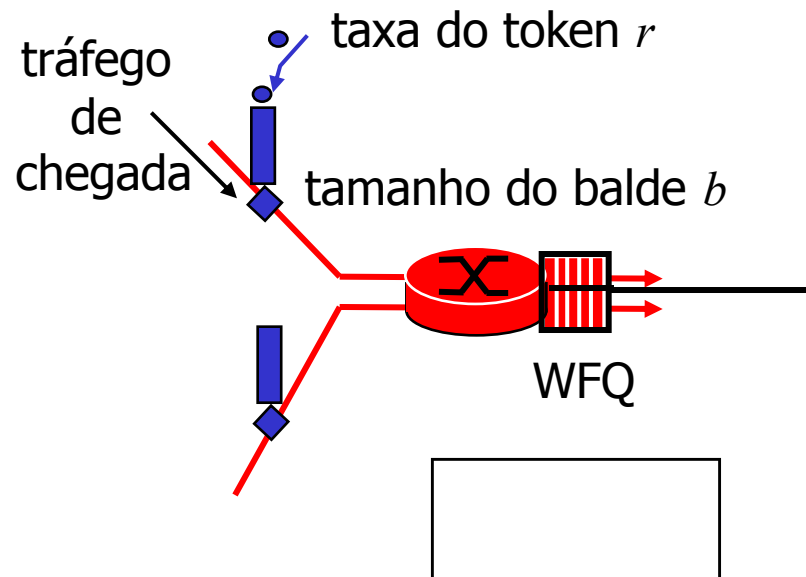
Modelos de Serviço IntServ

- Serviço Garantido (*Guaranteed Service*)
 - Chegada de tráfego de pior caso
 - Fonte policiada por um balde furado (*leaky bucket*)
 - Limite simples para o atraso
 - Prova matemática



Modelos de Serviço IntServ

- Serviço de Carga Controlada (*Controlled Load Service*)
 - Fluxo é atendido com uma QoS próxima da QoS que o mesmo fluxo receberia de um elemento de rede com baixa carga



Sinalização na Internet

**Encaminhamento
sem conexões (sem
estado) pelos
roteadores IP**



**Serviço de
melhor esforço
do IP**



**Nenhum protocolo
de sinalização de
rede no projeto
inicial da Internet**

- IntServ introduz um novo requisito
 - Reserva de recursos fim-a-fim para garantir QoS para aplicações multimídias
 - Tanto em roteadores quanto em sistemas finais

Sinalização na Internet

- *Internet Signaling protocol (ST-II)*
 - Definido pela RFC 1819
 - O primeiro protocolo de sinalização para a Internet
- *Resource Reservation Protocol (RSVP)*
 - Definido pela RFC 2205
 - Permite que os usuários comuniquem à rede os recursos que precisam
 - “Disfarce” para o termo sinalização

RSVP: Requisitos de Projeto

1. Suportar receptores heterogêneos
 - Larguras de banda diferentes ao longo dos caminhos
2. Suportar diferentes aplicações com diferentes requisitos de recursos
3. Tornar o *multicast* um serviço de primeira classe, com adaptação para participação em grupo *multicast*
4. Aproveitamento do roteamento *multicast/unicast* existente, com adaptação a mudanças nas rotas *unicast* e *multicast*

RSVP: Requisitos de Projeto

5. Sobrecarga de controle deve crescer linearmente (no pior caso) em função do número de receptores
6. Projeto modular para atender tecnologias heterogêneas das camadas inferiores

RSVP: Observações

- **NÃO** especifica como os recursos devem ser reservados
 - Ele é um mecanismo para comunicar as necessidades de cada fluxo
- **NÃO** determina as rotas seguidas pelos pacotes
 - É tarefa dos protocolos de roteamento
 - Sinalização desvinculada do roteamento
- **NÃO** há interação com o encaminhamento de pacotes
 - Separação dos planos de controle (sinalização) e dados (encaminhamento)

RSVP: Funcionamento

1. Transmissores e receptores se associam a um grupo *multicast*
 - Como a associação é feita está fora do escopo do RSVP
 - Transmissores não precisam se associar ao grupo
2. Sinalização do **transmissor** para a **rede**
 - Mensagem de **estabelecimento caminho** (*path message*)
 - Informar os roteadores sobre a presença do transmissor
 - Mensagem de **liberação de caminho** (*path teardown*)
 - Apagar as informações do caminho do transmissor nos roteadores

3. Sinalização do **receptor** para a **rede**

- Mensagem de **reserva** (*reservation message*)
 - Reserva os recursos a partir do(s) transmissor(es) até o receptor
- Mensagem de **liberação de reserva** (*reservation teardown*)
 - Remove as reservas do receptor

4. Sinalização da **rede** para os **sistemas finais**

- Mensagem de **erro de caminho** (*path error*)
 - Não foi possível estabelecer um caminho
- Mensagem de **erro de reserva** (*reservation error*)
 - Não foi possível alocar os recursos

Aulas 12, 13 e 14

Redes Multimídias

Conceitos, aplicações, requisitos e mecanismos; QoS

Igor Monteiro Moraes
Redes de Computadores II