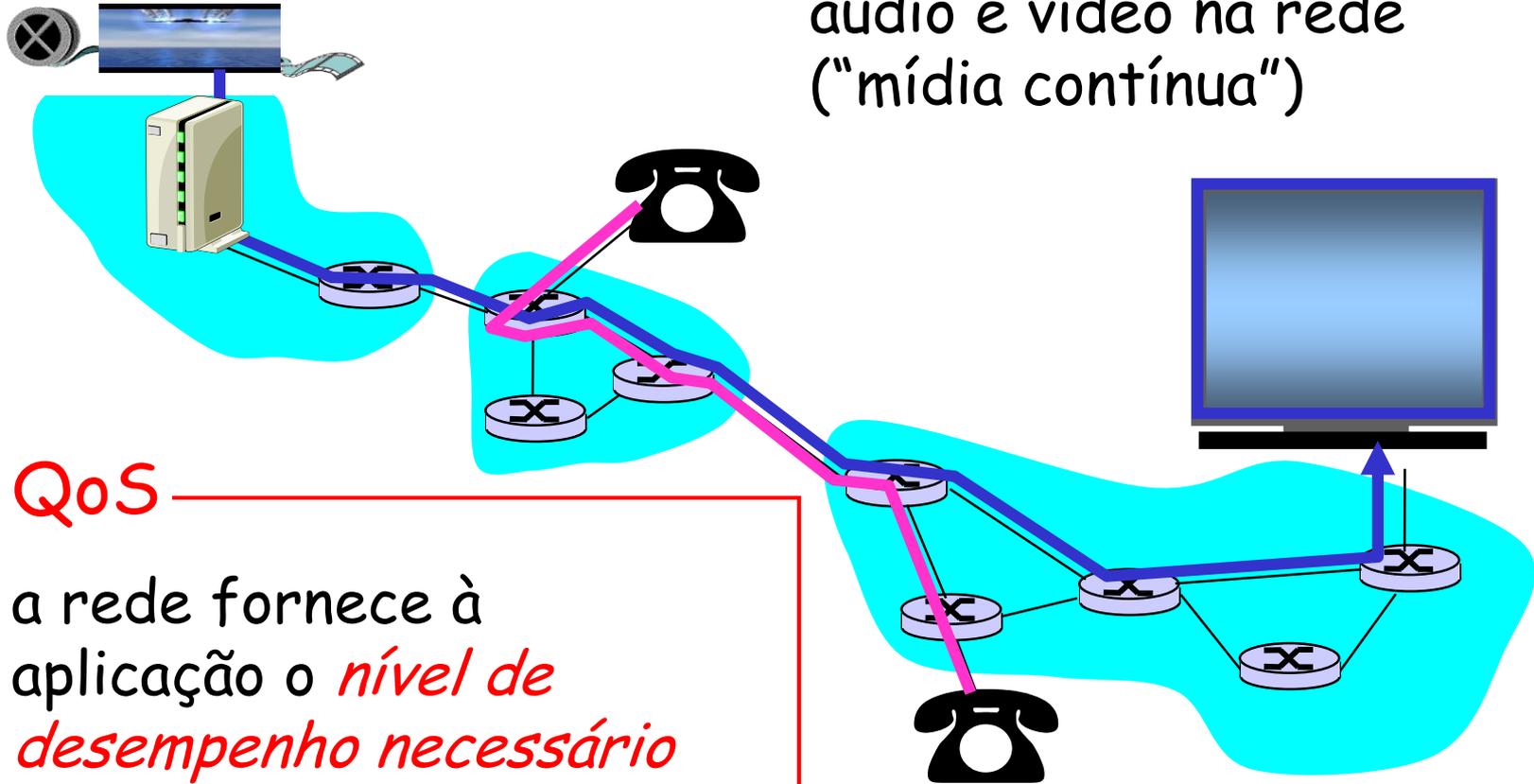


# Capítulo 7: Multimídia em Redes

# O que são Multimídia e Qualidade de Serviço?

Aplicações Multimídia:  
áudio e vídeo na rede  
("mídia contínua")



QoS

a rede fornece à aplicação o *nível de desempenho necessário para que a aplicação funcione como esperado*

# Capítulo 7: Objetivos

## Princípios

- ❑ Classificar as aplicações multimídia
- ❑ Identificar os serviços de rede que as aplicações necessitam
- ❑ Extrair o máximo do serviço de melhor esforço
- ❑ Mecanismos para garantir QoS

## Protocolos e Arquiteturas

- ❑ Protocolos específicos para melhor esforço
- ❑ Arquiteturas para QoS

# Capítulo 7: Roteiro

- ❑ 7.1 Aplicações Multimídia em Redes
- ❑ 7.2 Fluxos contínuos (*Streams*) de áudio e vídeo armazenados
- ❑ 7.3 Multimídia em Tempo Real: estudo do Telefone Internet
- ❑ 7.4 Protocolos para Aplicações Interativas de Tempo Real
  - RTP, RTCP, SIP
- ❑ 7.5 Distribuição de Multimídia: redes de distribuição de conteúdos
- ❑ 7.6 Além do Melhor Esforço
- ❑ 7.7 Mecanismos de Escalonamento e Policiamento
- ❑ 7.8 Serviços Integrados e Serviços Diferenciados
- ❑ 7.9 RSVP

# Aplicações MM em Redes

## Classes de aplicações MM:

- 1) Fluxos contínuos (*Streams*) de áudio e vídeo armazenados
- 2) Fluxos contínuos (*Streams*) de áudio e vídeo ao vivo
- 3) Vídeo interativo de tempo real

*Jitter* é a variabilidade dos atrasos dos pacotes dentro de um mesmo fluxo de pacotes

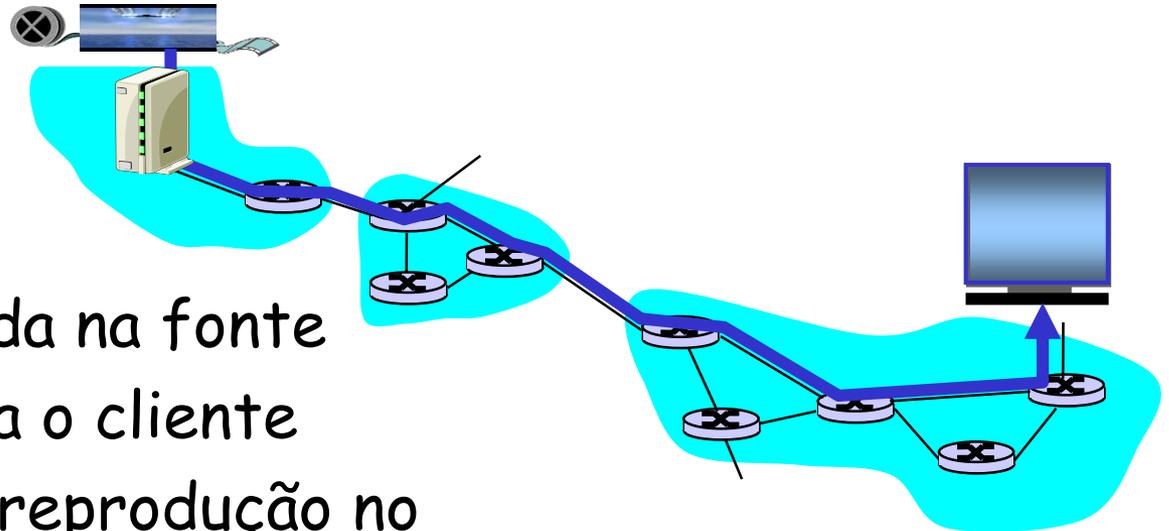
## Características Fundamentais:

- ❑ Tipicamente são **sensíveis a atrasos**
  - atraso fim-a-fim
  - variação do atraso (*jitter*)
- ❑ Mas são **tolerantes a perdas**: perdas não muito frequentes causam apenas pequenos distúrbios
- ❑ Antítese da transferência de dados que é intolerante a perdas mas tolerante a atrasos.

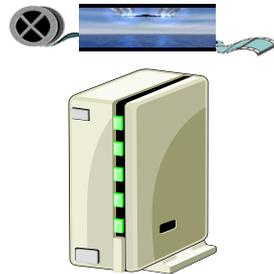
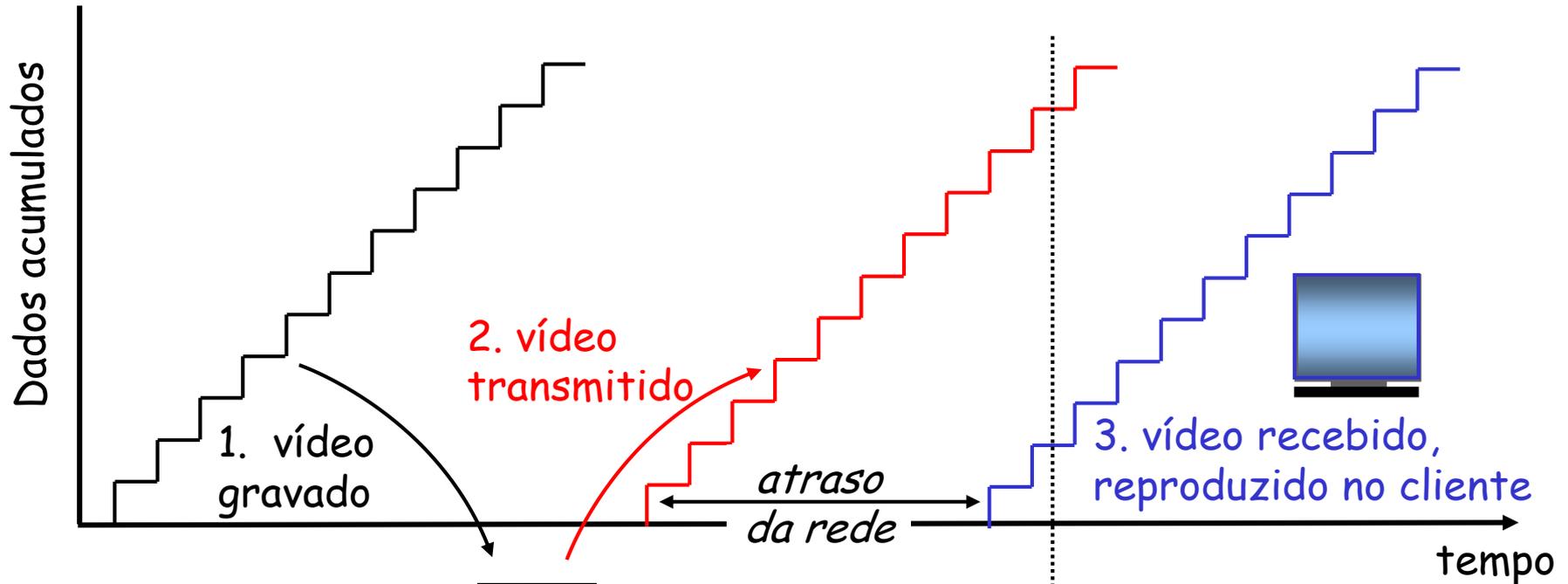
# Fluxo Contínuo de Multimídia Armazenada

## Fluxo Contínuo (Streaming):

- ❑ mídia armazenada na fonte
- ❑ transmitida para o cliente
- ❑ Fluxo contínuo: reprodução no cliente inicia *antes* de que todos os dados tenham sido recebidos
- ❑ restrição de tempo para os dados ainda não transmitidos: devem chegar a tempo de serem reproduzidos

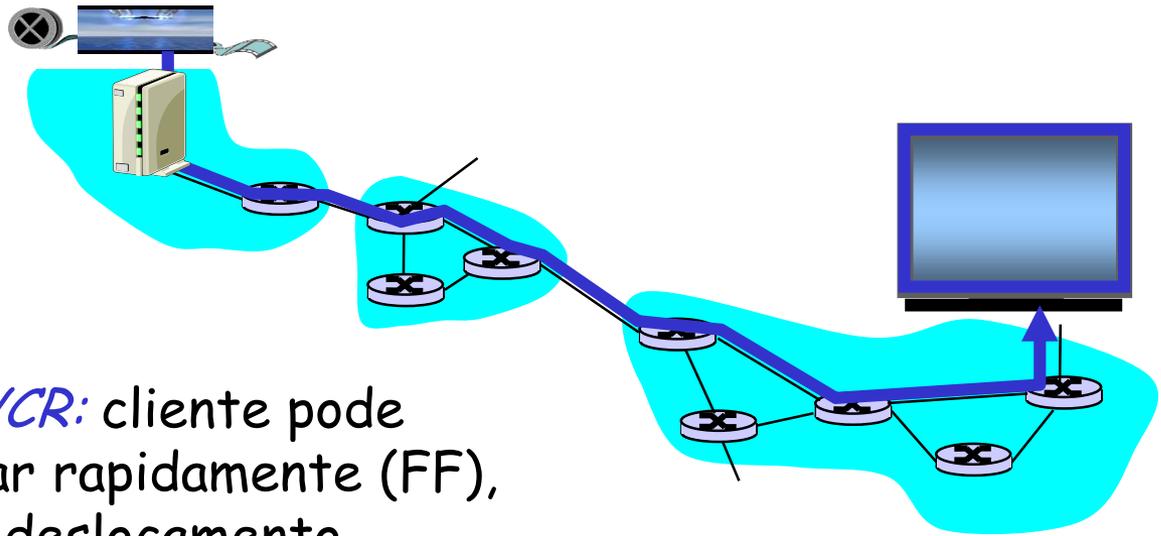


# Fluxo Contínuo Multimídia Armazenada: Como funciona?



Fluxo contínuo: neste instante, o cliente está reproduzindo uma parte anterior do vídeo, enquanto o servidor ainda está transmitindo as partes seguintes

# Fluxo Contínuo Multimídia Armazenada: Interatividade



- *Funcionalidade tipo VCR:* cliente pode pausar, voltar, avançar rapidamente (FF), modificar a barra de deslocamento
  - atraso inicial de 10 seg OK
  - 1-2 seg até que o comando seja executado OK
  - RTSP é freqüentemente usado (mais detalhes posteriormente)
- restrição de tempo para dados ainda não transmitidos: chegar em tempo para reprodução

# Fluxo Contínuo Multimídia ao Vivo

## Exemplos:

- ❑ Programa de bate papo em rádio Internet
- ❑ Evento esportivo ao vivo

## Fluxo Contínuo

- ❑ buffer de reprodução
- ❑ reprodução pode atrasar dezenas de segundos após a transmissão
- ❑ ainda tem restrições de tempo

## Interatividade

- ❑ impossível avançar
- ❑ é possível voltar, pausar!



# Multimídia Sobre a Internet Atual

**TCP/UDP/IP:** "serviço de melhor esforço"

□ *sem* garantias sobre atrasos, perdas



? ? ? ? ? ? ?  
Mas, você disse que as aplicações MM  
necessitam de QoS e nível de desempenho  
para funcionarem!  
? ? ?



As aplicações MM na Internet atual usam técnicas da camada de aplicação para minimizar (da melhor forma) efeitos de atraso, perdas

# Como a Internet deveria evoluir para dar um melhor suporte a aplicações MM?

## Filosofia de serviços integrados:

- ❑ Modificar a Internet de modo que as aplicações possam reservar largura de banda fim-a-fim
- ❑ Requer software novo e complexo nos *hosts* & roteadores

## Filosofia do "deixa como está":

- ❑ sem grandes mudanças
- ❑ aumento da largura de banda quando necessário
- ❑ distribuição de conteúdo, multicast na camada de aplicação

## Filosofia de serviços diferenciados :

- ❑ Menos mudanças na infraestrutura da Internet, mas provendo serviços de 1a. e 2a. classes.



O que você acha?

# Algumas palavras sobre compressão de áudio

- ❑ Sinal analógico amostrado a uma taxa constante
  - telefone: 8.000 amostras/seg
  - CD de música: 44.100 amostras/seg
- ❑ Cada amostra é discretizada (arredondada)
  - ex.,  $2^8=256$  possíveis valores discretos
- ❑ Cada valor discretizado é representado por bits
  - 8 bits para 256 valores
- ❑ Exemplo: 8.000 amostras/seg, 256 valores discretos --> 64.000 bps
- ❑ Receptor converte-o de volta a um sinal analógico:
  - alguma perda de qualidade

## Exemplo de taxas

- ❑ CD: 1,411 Mbps
- ❑ MP3: 96, 128, 160 kbps
- ❑ Telefonia Internet: 5,3 - 13 kbps

# Algumas palavras sobre compressão de vídeo

- ❑ Vídeo é uma seqüência de imagens apresentadas a uma taxa constante
  - ex. 24 imagens/seg
- ❑ Imagem digital é uma matriz de pixels
- ❑ Cada pixel é representado por bits
- ❑ Redundância
  - espacial
  - temporal

## Exemplos:

- ❑ MPEG 1 (CD-ROM) 1,5 Mbps
- ❑ MPEG2 (DVD) 3-6 Mbps
- ❑ MPEG4 (freqüentemente usado na Internet, < 1 Mbps)

## Pesquisa:

- ❑ vídeo em camadas (escalável)
  - adapta as camadas à largura de banda disponível

# Capítulo 7: Roteiro

- 7.1 Aplicações Multimídia em Redes
- 7.2 Fluxos contínuos de áudio e vídeo armazenados
- 7.3 Multimídia em Tempo Real: estudo do Telefone Internet
- 7.4 Protocolos para Aplicações Interativas de Tempo Real
  - RTP, RTCP, SIP
- 7.5 Distribuição de Multimídia: redes de distribuição de conteúdos
- 7.6 Além do Melhor Esforço
- 7.7 Mecanismos de Escalonamento e Policiamento
- 7.8 Serviços Integrados e Serviços Diferenciados
- 7.9 RSVP

# Fluxo Contínuo Multimídia Armazenada

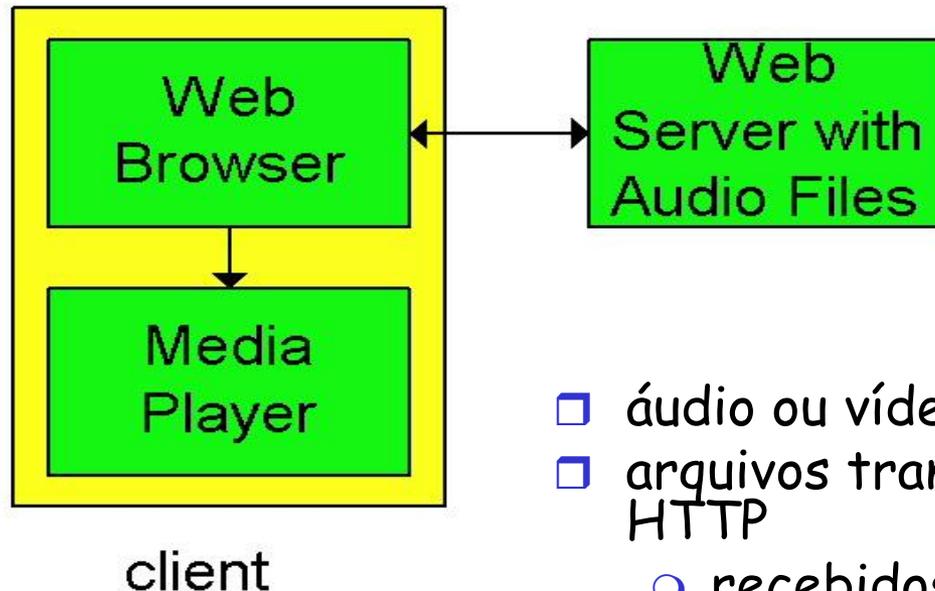
Técnicas de *streaming* da camada de aplicação para extrair o máximo do serviço de melhor esforço

- armazenamento no lado do cliente
- uso do UDP ao invés do TCP
- múltiplas codificações da multimídia

## Tocadores de Mídia

- Remoção de *jitter*
- descompressão
- esconder os erros
- Interface gráfica do usuário com controles para interatividade

# Multimídia Internet: abordagem simplista

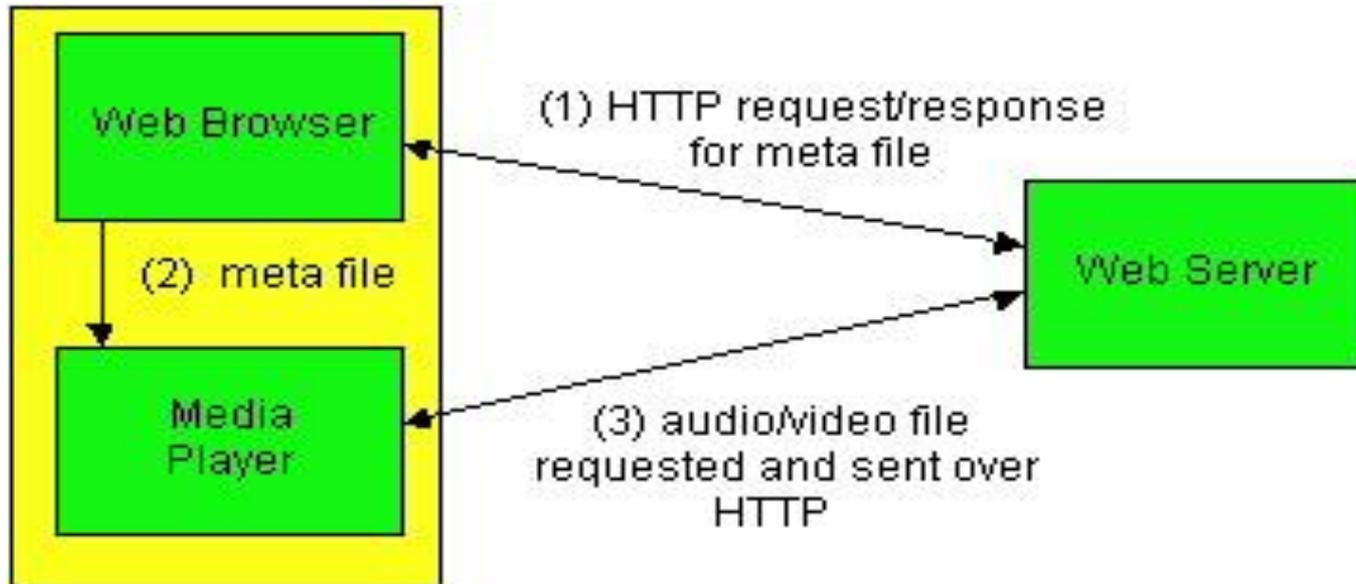


- ❑ áudio ou vídeo armazenado em arquivo
- ❑ arquivos transferidos como objetos HTTP
  - recebidos completamente pelo cliente
  - Depois repassado para o tocador (*player*)

áudio, vídeo não são enviados como fluxo contínuo:

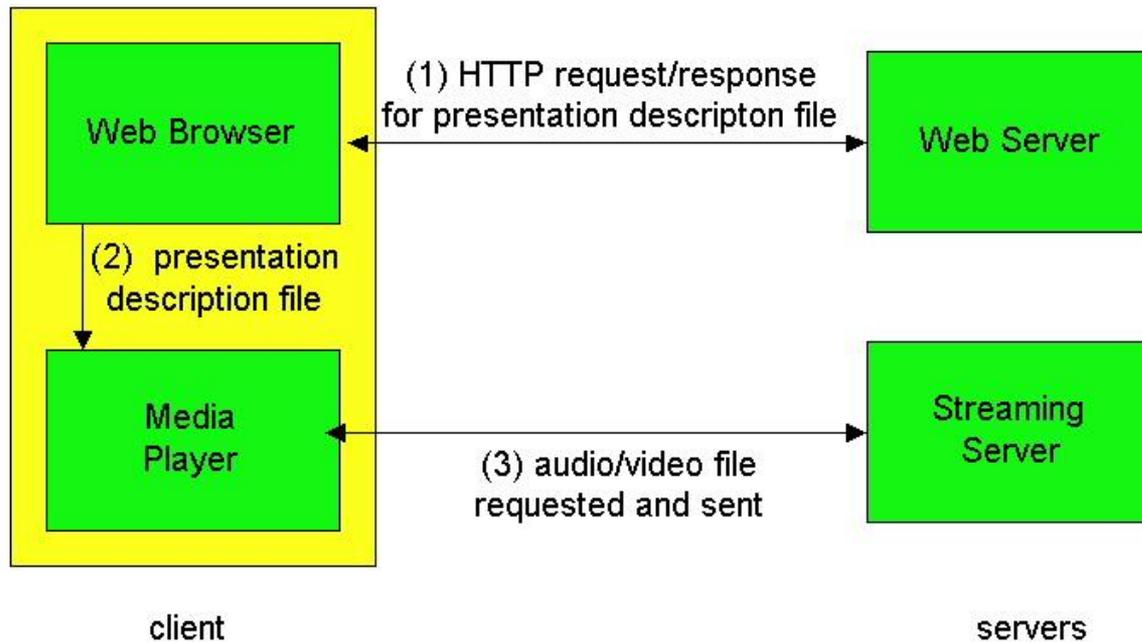
- ❑ não há cadeia de montagem (*pipelining*) longos atrasos até a reprodução!

# Multimídia Internet: abordagem com fluxos



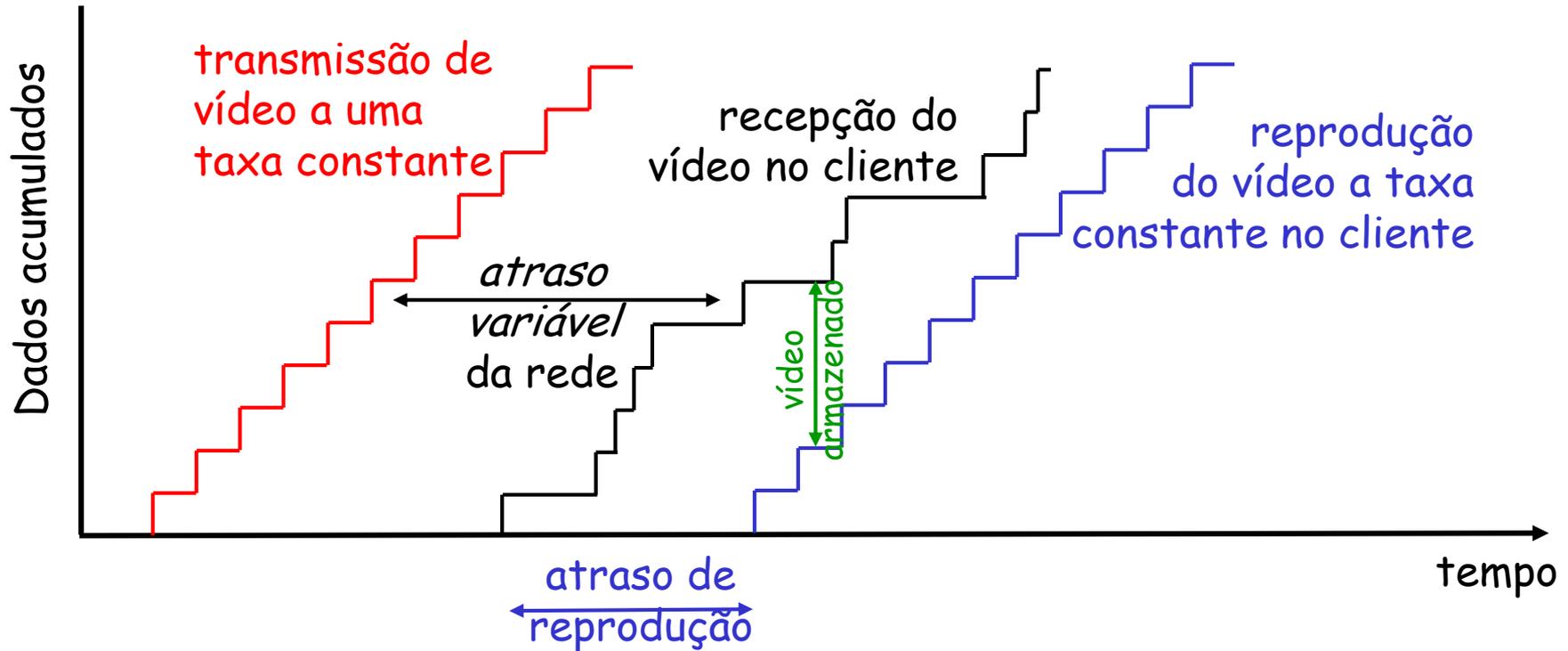
- ❑ *browser* solicita (GETs) **meta arquivo**
- ❑ *browser* inicia o tocador, passando o meta arquivo
- ❑ tocador contacta o servidor
- ❑ servidor cria o **fluxo** de áudio/vídeo até o tocador

# Fluxos a partir de um servidor de fluxos



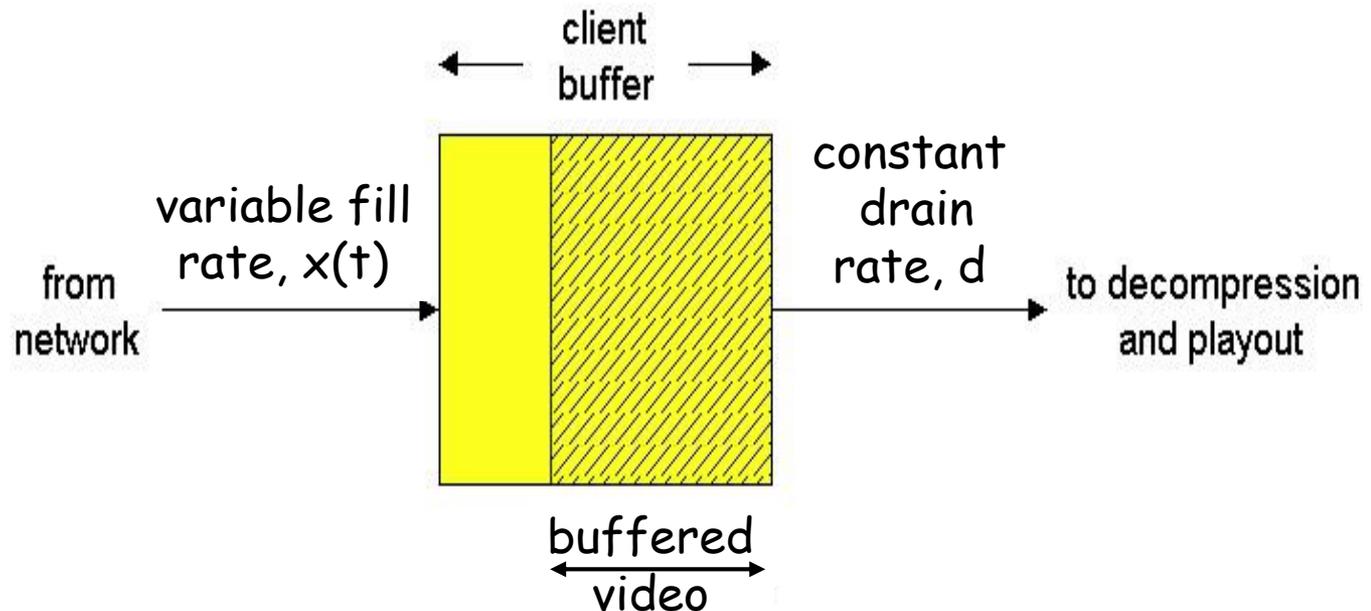
- ❑ Esta arquitetura permite o uso de protocolos não-HTTP entre o servidor e o reprodutor de mídia
- ❑ Também pode usar UDP ao invés do TCP

# Fluxos Multimídia: Armazenamento pelo Cliente



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

# Fluxos Multimídia: Armazenamento pelo Cliente



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

# Fluxo Multimídia: UDP ou TCP?

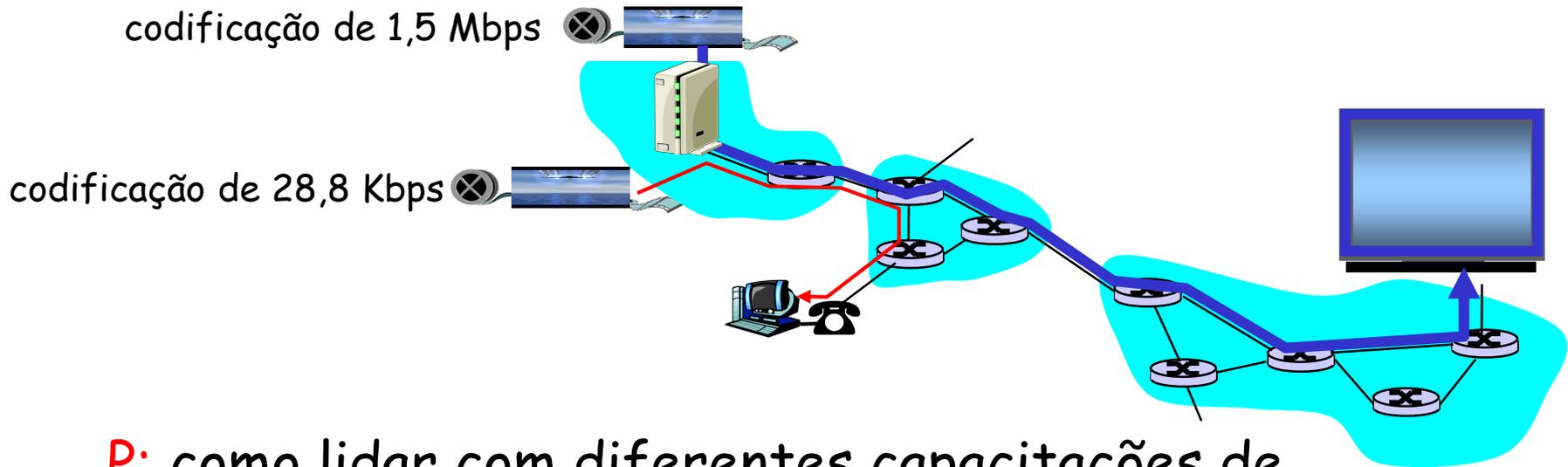
## UDP

- ❑ servidor envia a uma taxa adequada para o cliente (sem se importar com congestionamento da rede!)
  - frequentemente taxa de transmissão = taxa de codificação = constante
  - portanto, taxa de enchimento = taxa constante - perda de pacotes
- ❑ pequeno atraso de reprodução (2-5 segundos) para compensar pela variação do atraso da rede
- ❑ recuperação de erros: se houver tempo

## TCP

- ❑ transmite na taxa máxima permitida pelo TCP
- ❑ taxa de enchimento flutua devido ao controle de congestionamento do TCP
- ❑ maior atraso para reprodução: taxa de entrega do TCP mais suave
- ❑ HTTP/TCP passam mais facilmente através de *firewalls*

# Fluxo Multimídia : taxa(s) do cliente



**P:** como lidar com diferentes capacidades de taxa de recepção do cliente?

- acesso discado de 28,8 Kbps
- Ethernet de 100Mbps

**R:** servidor armazena, transmite múltiplas cópias do vídeo, codificadas em taxas diferentes

# Controle do Usuário de Mídia Contínua: RTSP

## HTTP

- ❑ Não tinha como alvo conteúdo multimídia
- ❑ Não possui comandos para avanço rápido, etc.

## RTSP: RFC 2326

- ❑ Protocolo cliente-servidor da camada de aplicações.
- ❑ O usuário pode controlar a apresentação: retorno, avanço rápido, pausa, retomada, reposicionamento, etc.

## O que ele não faz :

- ❑ Não define como o áudio/vídeo é encapsulado para ser transmitido pela rede
- ❑ Não restringe como a mídia tipo fluxo (*stream*) é transportada; pode ser transportada sobre UDP ou TCP
- ❑ Não especifica como o apresentador da mídia armazena o áudio/vídeo

# RTSP: controle fora da faixa

## FTP usa um canal de controle "fora da faixa":

- ❑ Um arquivo é transferido sobre uma conexão TCP.
- ❑ A informação de controle (mudanças de diretório, remoção de arquivo, renomeação de arquivo, etc.) é enviada numa conexão TCP à parte.
- ❑ Os canais "fora da faixa" e "dentro da faixa" utilizam diferentes números de portas.

## As mensagens RTSP também são enviadas fora da faixa:

- ❑ As mensagens de controle RTSP usam números de porta diferentes do fluxo da mídia, e são, portanto, enviadas fora da faixa
  - Porta 554
- ❑ O fluxo de mídia é considerado "dentro da faixa".

# Exemplo RTSP

## Cenário:

- ❑ meta arquivo enviado para o *browser* web
- ❑ *browser* inicia o tocador
- ❑ tocador estabelece uma conexão de controle RTSP e uma conexão de dados para o servidor de mídia contínua

# Exemplo de Meta arquivo

```
<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>
```

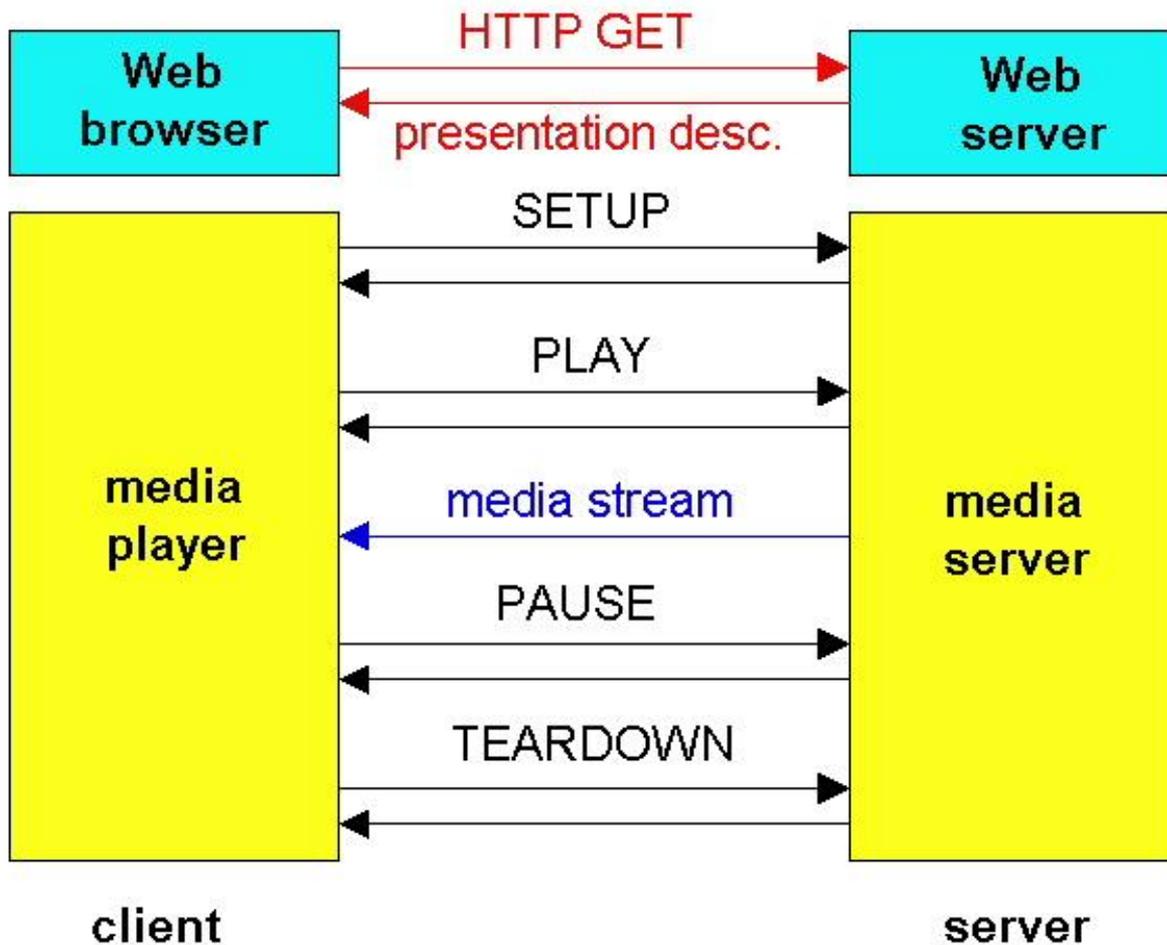
```
      <track type="video/jpeg"
```

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

```
    </group>
```

```
</session>
```

# Operação do RTSP



# RTSP: exemplo de diálogo

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

# Capítulo 7: Roteiro

- 7.1 Aplicações Multimídia em Redes
- 7.2 Fluxos contínuos de áudio e vídeo armazenados
- 7.3 Multimídia em Tempo Real: estudo do Telefone Internet
- 7.4 Protocolos para Aplicações Interativas de Tempo Real
  - RTP, RTCP, SIP
- 7.5 Distribuição de Multimídia: redes de distribuição de conteúdos
- 7.6 Além do Melhor Esforço
- 7.7 Mecanismos de Escalonamento e Policiamento
- 7.8 Serviços Integrados e Serviços Diferenciados
- 7.9 RSVP

# Aplicações interativas de tempo real

- Telefone PC-2-PC
  - serviços de mensagens instantânea estão fornecendo isto
- PC-2-telefone
  - teclado
  - Net2phone
- videoconferência com Webcams

Vamos estudar em detalhes um exemplo de telefone Internet PC-2-PC

# Multimídia Interativa: Telefone Internet

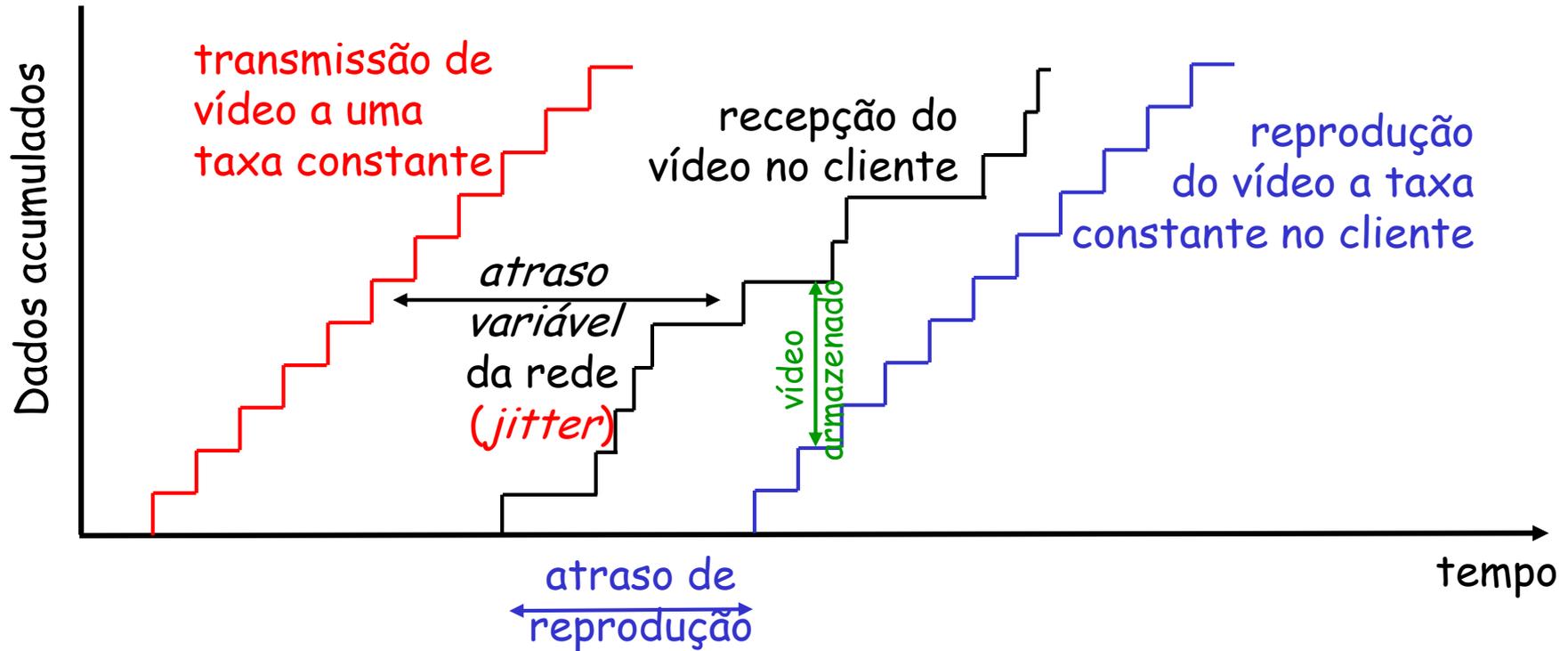
## Introduz o Telefone Internet através de um exemplo

- ❑ áudio do falante: alterna surtos de voz com períodos de silêncio
  - 64 kbps durante surto de voz
- ❑ pcts gerados apenas durante os surtos de voz
  - pedaços de 20 msec a 8 Kbytes/seg: dados de 160 bytes
- ❑ cabeçalho da camada de aplicação é adicionado a cada pedaço.
- ❑ Pedaço+cabeçalho empacotado num segmento UDP
- ❑ Aplicação envia segmentos UDP no socket a cada 20 msec durante um surto de voz.

# Telefone Internet: Perda de Pacotes e Atraso

- ❑ **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
  - atrasos: processamento, enfileiramento na rede; atrasos do sistema terminal (transmissor, receptor)
  - atraso máximo tolerável típico: 400 ms
- ❑ **tolerância a perdas:** a depender da codificação da voz, as perdas podem ser encobertas, taxas de perdas de pacotes entre 1% e 10% podem ser toleradas.

# Variação do atraso (*jitter*)



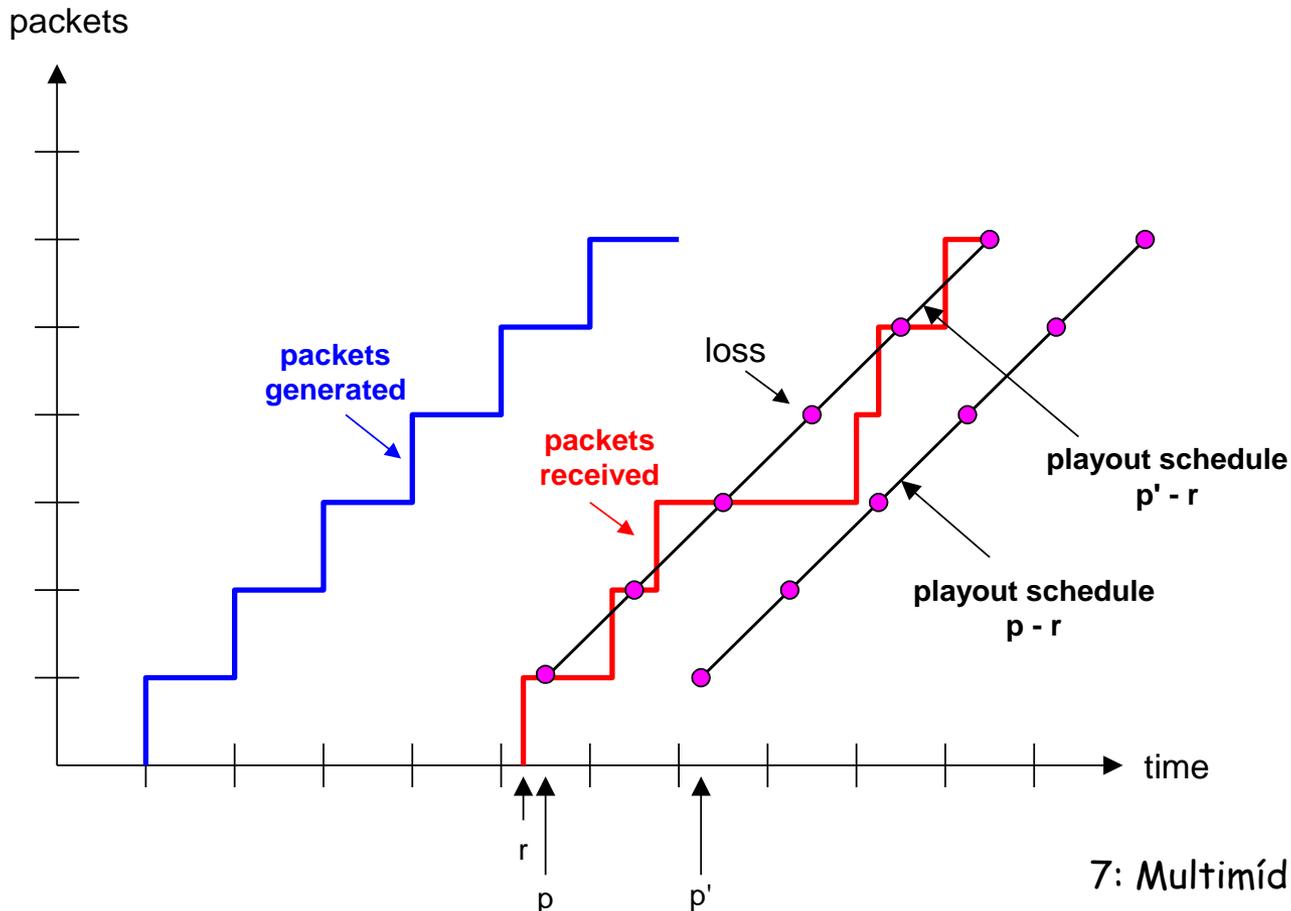
- Considere o atraso fim a fim de dois pacotes consecutivos: a diferença pode ser maior ou menor do que 20 msec

# Telefone Internet: atraso de apresentação fixo

- O receptor tenta reproduzir cada pedaço exatamente  $q$  msecs após a geração do pedaço.
  - Se o pedaço contiver um carimbo 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  $q$ :
  - $q$  longo: menos perda de pacotes
  - $q$  pequeno: melhor experiência interativa

# Atraso de apresentação fixo

- Transmissor gera pacotes a cada 20 mseg durante o surto de voz.
- O primeiro pacote é recebido no instante  $r$
- A primeira reprodução é programada para iniciar no instante  $p$
- A segunda reprodução é programada para iniciar no instante  $p'$



# Atraso de reprodução adaptativo, I

- **Objetivo:** minimizar o atraso de reprodução, mantendo baixa a taxa de perdas
- **Abordagem:** ajuste adaptativo do atraso de reprodução:
  - Estima o atraso da rede e ajusta o atraso de reprodução no início de cada surto de voz.
  - Períodos de silêncio são comprimidos e alongados.
  - Os pedaços ainda são reproduzidos a cada 20 mseg durante um surto de voz.

$t_i$  = carimbo 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

Estimativa dinâmica do atraso médio no receptor:

$$d_i = (1 - u)d_{i-1} + u(r_i - t_i)$$

onde  $u$  é uma constante (ex.,  $u = 0,01$ ).

# Atraso de reprodução adaptativo II

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 usados apenas no início de um surto de voz.

Para o primeiro pacote de um surto de voz, o tempo de apresentação é:

$$p_i = t_i + d_i + Kv_i$$

onde  $K$  é um constante positiva.

Os pacotes restantes em um surto de voz são reproduzidos periodicamente

# Reprodução adaptativa, III

- P:** Como o receptor determina se um pacote é o primeiro de um surto de voz?
- Se nunca houvesse perdas, o receptor poderia simplesmente olhar os carimbos de tempo sucessivos.
    - Diferença entre carimbos sucessivos  $> 20$  mseg, início do surto de voz.
  - Mas, dado que perdas são possíveis, o receptor deve olhar tanto para os carimbos de tempo quanto para os números de seqüência.
    - Diferença entre carimbos sucessivos  $> 20$  mseg e números de seqüência sem falhas, início do surto de voz.

# Recuperação da perda de pacotes (1)

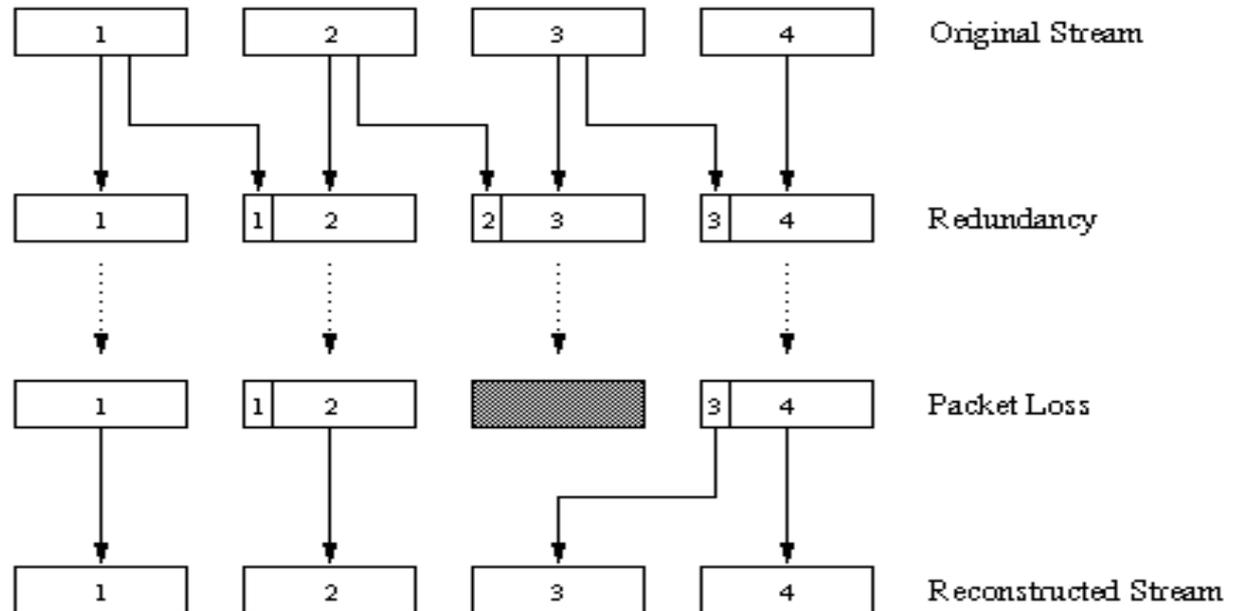
## forward error correction (FEC): esquema simples

- ❑ para cada grupo de  $n$  pedaços criar um pedaço redundante efetuando o OU-exclusivo dos  $n$  pedaços originais
- ❑ transmite  $n+1$  pedaços, aumentando a largura de banda por um fator de  $1/n$ .
- ❑ pode reconstruir os  $n$  pedaços originais se houver no máximo um pedaço perdido dentre os  $n+1$  pedaços.
- ❑ Atraso de reprodução deve ser fixado para o instante de recepção de todos os  $n+1$  pacotes
- ❑ Compromissos:
  - aumento de  $n$ , menos desperdício de banda
  - aumento de  $n$ , atraso de reprodução mais longo
  - aumento de  $n$ , maior probabilidade de que 2 ou mais pedaços sejam perdidos

# Recuperação da perda de pacotes (2)

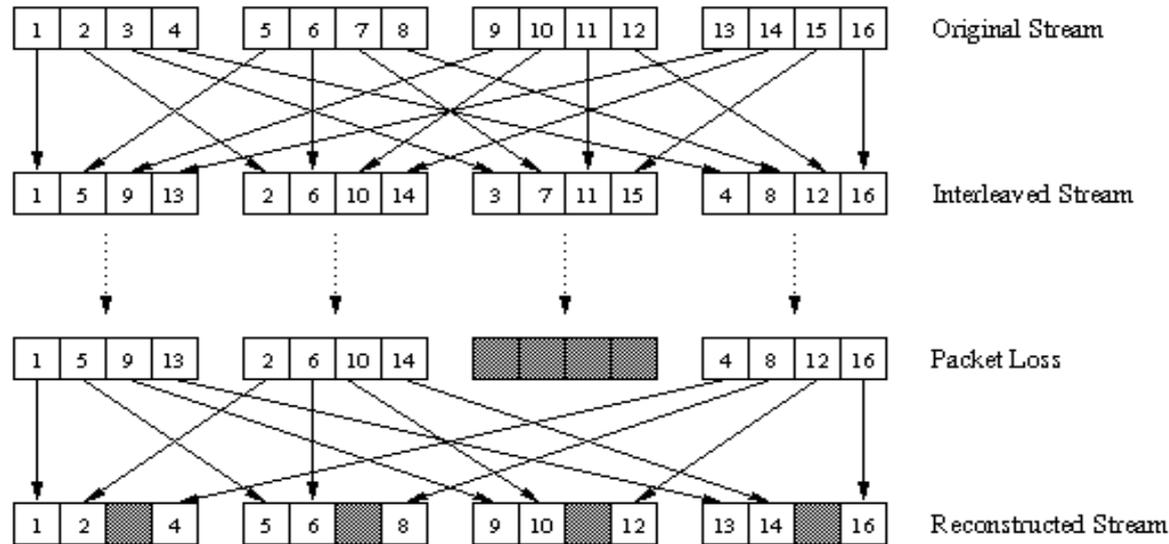
## 2o. Esquema de FEC

- transmissão de "carona" de um fluxo de menor qualidade
- envia fluxo de áudio de baixa resolução como informação redundante
- por exemplo, fluxo nominal PCM a 64 kbps e fluxo redundante GSM a 13 kbps.



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

# Recuperação da perda de pacotes (3)



## Entrelaçamento

- ❑ os pedaços são quebrados em unidades menores
- ❑ por exemplo, quatro unidades de 5 mseg por pedaço
- ❑ pacote agora contém pequenas unidades de pedaços diferentes
- ❑ se o pacote se perder, ainda temos muito de cada pedaço
- ❑ não tem overhead de redundância
- ❑ mas aumenta o atraso de reprodução

# Resumo: Multimídia Internet: truques

- ❑ **use UDP** para evitar o(s) (atrasos) do controle de congestionamento do TCP para tráfego sensível a tempo
- ❑ **atraso de reprodução adaptativo** para o lado do cliente: para compensar o atraso
- ❑ o lado do servidor **casa a largura de banda do fluxo** à largura de banda disponível no caminho cliente-ao-servidor
  - escolha entre taxas de fluxos pré-codificadas
  - taxa dinâmica de codificação do servidor
- ❑ recuperação de erros (acima do UDP)
  - FEC, entrelaçamento
  - retransmissões, se houver tempo
  - encobrimento de erros: repetir dados próximos

# Capítulo 7: Roteiro

- 7.1 Aplicações Multimídia em Redes
- 7.2 Fluxos contínuos de áudio e vídeo armazenados
- 7.3 Multimídia em Tempo Real: estudo do Telefone Internet
- 7.4 Protocolos para Aplicações Interativas de Tempo Real
  - RTP, RTCP, SIP
- 7.5 Distribuição de Multimídia: redes de distribuição de conteúdos
- 7.6 Além do Melhor Esforço
- 7.7 Mecanismos de Escalonamento e Policiamento
- 7.8 Serviços Integrados e Serviços Diferenciados
- 7.9 RSVP

# Protocolo de Tempo Real (RTP)

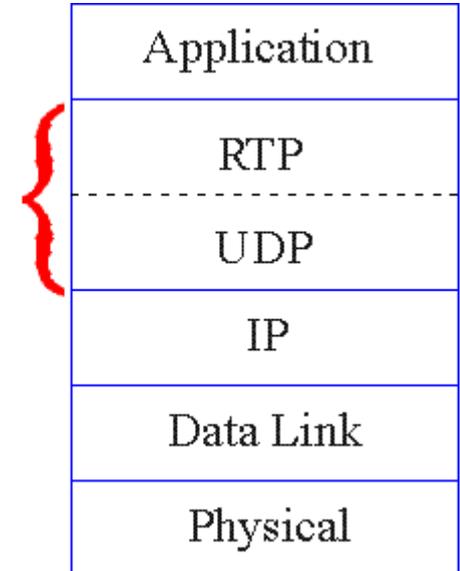
- ❑ RTP = *Real Time Protocol*
- ❑ RTP especifica uma estrutura de pacote para pacotes que transportam dados de áudio e de vídeo
- ❑ RFC 1889.
- ❑ Pacote RTP provê
  - Identificação do tipo da carga
  - Numeração da seqüência de pacotes
  - Carimbo de tempo
- ❑ RTP roda nos sistemas terminais.
- ❑ Pacotes RTP são encapsulados em segmentos UDP
- ❑ Interoperabilidade: Se duas aplicações de telefone Internet rodarem RTP então elas poderão trabalhar em conjunto

# RTP roda sobre UDP

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
- carimbo de tempo

transport  
layer



# Exemplo RTP

- ❑ Considere o envio de voz codificada em PCM de 64 kbps sobre RTP.
- ❑ Aplicação coleta os dados codificados em pedaços, ex., a cada 20 mseg = 160 bytes num pedaço.
- ❑ O pedaço de áudio junto com o cabeçalho RTP formam um pacote RTP, que é encapsulado num segmento UDP.
- ❑ O cabeçalho RTP indica o tipo da codificação de áudio em cada pacote:
  - os transmissores podem mudar a codificação durante a conferência.
- ❑ O cabeçalho RTP também contém números de seqüência e carimbos de tempo.

# RTP e QoS

- ❑ RTP não provê nenhum mecanismo para garantir a entrega em tempo dos dados nem nenhuma outra garantia de qualidade de serviço.
- ❑ O encapsulamento RTP é visto apenas nos sistemas terminais - não é visto por roteadores intermediários.
  - Roteadores provendo o serviço tradicional Internet de melhor esforço não faz nenhum esforço adicional para garantir que os pacotes RTP cheguem ao destino em tempo.

# Cabeçalho RTP



## RTP Header

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

- Tipo de carga 0: PCM mu-law, 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

**Número de Seqüência (16 bits):** O número de seqüência é incrementado de um para cada pacote RTP enviado; pode ser usado para detectar a perda de pacotes e para restaurar a seqüência de pacotes.

# Cabeçalho RTP (2)



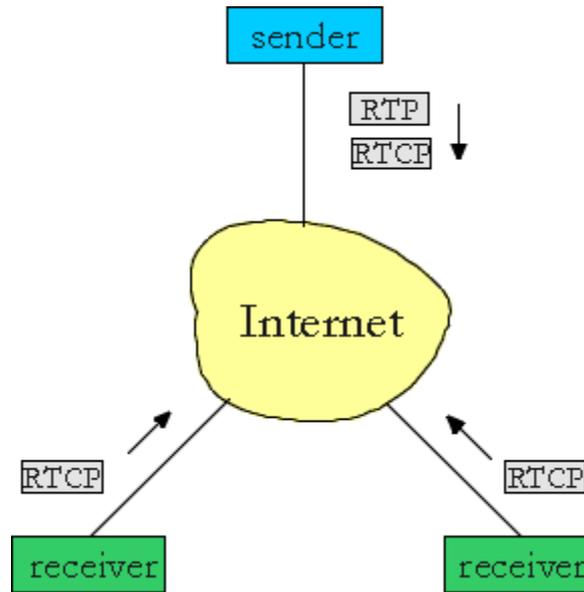
## RTP Header

- ❑ **Campo de carimbo de tempo (32 bits).** Reflete o instante de amostragem do primeiro byte no pacote de dados RTP.
  - Para áudio o relógio de carimbo de tempo incrementa de um para cada período de amostragem (por exemplo, a cada 125  $\mu$ seg para um relógio de amostragem de 8kHz)
  - se a aplicação de áudio gerar pedaços de 160 amostras codificadas, então o carimbo de tempo aumenta de 160 para cada pacote RTP quando a fonte estiver ativa. O relógio de carimbo de tempo continua a aumentar a uma taxa constante mesmo quando a fonte estiver inativa.
  
- ❑ **Campo de SSRC (32 bits).** Identifica a fonte de um fluxo RTP. Cada fluxo numa sessão RTP deve possuir um SSRC distinto.

# Protocolo de Controle de Tempo Real (RTCP)

- ❑ *Real-Time Control Protocol*
- ❑ Trabalha em conjunto com o RTP.
- ❑ Cada participante numa sessão RTP periodicamente transmite pacotes de controle RTCP para todos os demais participantes.
- ❑ Cada pacote RTCP contém relatos do transmissor e/ou receptor
  - relatam estatísticas úteis para as aplicações.
- ❑ Estas estatísticas incluem o número de pacotes enviados, o número de pacotes perdidos, *jitter* entre chegadas, etc.
- ❑ Esta realimentação de informação para as aplicações pode ser usada para controlar o desempenho
  - O transmissor pode modificar as suas transmissões baseadas na realimentação.

# RTCP - Continuação



- Para uma sessão RTP há tipicamente um único endereço multicast; todos os pacotes RTP e RTCP pertencentes à sessão usam o endereço multicast.
- Pacotes RTP e RTCP são diferenciados uns dos outros através do uso de números de portas distintos.
- Para limitar o tráfego, cada participante reduz o seu tráfego RTCP à medida que cresce o número de participantes da conferência.

# Pacotes RTCP

## Pacotes de relato do receptor:

- ❑ Fração dos pacotes perdidos, último número de seqüência, *jitter* entre chegadas médio.

## Pacotes de relato do transmissor:

- ❑ SSRC do fluxo RTP, tempo atual, número de pacotes enviados e número de bytes enviados.

## Pacotes de descrição da origem:

- ❑ Endereço de e-mail do transmissor, nome do transmissor, o SSRC do fluxo RTP associado.
- ❑ Os pacotes provêm um mapeamento entre o SSRC e o nome do usuário/host.

# Sincronização de Fluxos

- ❑ O RTCP pode ser usado para sincronizar diferentes fluxos de mídia dentro de uma sessão RTP.
- ❑ Considere uma aplicação de videoconferência para a qual cada transmissor gera um fluxo RTP para vídeo e outro para áudio.
- ❑ Os carimbos de tempo nestes pacotes RTP estão vinculados aos relógios de amostragem de vídeo e de áudio, e não estão vinculadas ao relógio de tempo real.
- ❑ Cada pacote de relato do transmissor contém, para o pacote gerado mais recentemente no fluxo RTP associado,
  - o carimbo de tempo do pacote RTP
  - e instante num relógio de tempo real em que o pacote foi criado.
- ❑ Os receptores pode usar esta associação para sincronizar a reprodução de áudio e de vídeo.

# Escalonamento da Largura de Banda do RTCP

- O RTCP tenta limitar o seu tráfego a 5% da largura de banda da sessão.

## Exemplo

- Suponha que haja um transmissor enviando vídeo a uma taxa de 2 Mbps. Então o RTCP tenta limitar o seu tráfego a 100 Kbps.
  - O protocolo atribui 75% desta taxa, ou 75 kbps, para os receptores; e atribui os restantes 25% da taxa, ou 25 kbps, para o transmissor.
- Os 75 kbps alocados são compartilhados igualmente entre os receptores:
    - se houver  $R$  receptores, então cada receptor pode transmitir tráfego RTCP a uma taxa de  $75/R$  kbps
  - Transmissor pode transmitir tráfego RTCP a uma taxa de 25 kbps.
  - Um participante (um transmissor ou receptor) determina o período de transmissão dos pacotes RTCP através do cálculo dinâmico do tamanho médio de um pacote RTCP (ao longo de toda a sessão) e dividindo o tamanho médio do pacote RTCP pela sua taxa alocada.

# SIP

- ❑ *Session Initiation Protocol* (Protocolo de Iniciação de Sessões)
- ❑ Vem da IETF

## Visão de longo prazo do SIP

- ❑ Todas as chamadas telefônicas e de vídeo conferência 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 chamado, não importa onde ele esteja, em qualquer dispositivo IP que o chamado esteja usando no momento

# Serviços SIP

- ❑ Estabelecendo uma chamada
  - Provê mecanismos para que o chamador informe ao chamado que ele deseja estabelecer uma chamada
  - Provê mecanismos para que o chamador e o chamado concordem no tipo de mídia e na codificação
  - Provê mecanismos para encerrar a chamada
- ❑ Determina o endereço IP atual do chamado.
  - Mapeia identificador mnemônico para o endereço IP atual
- ❑ Gerenciamento de chamadas
  - Adiciona novos fluxos de mídia durante a chamada
  - Altera a codificação durante a chamada
  - Convida outros
  - Transfere e mantém (*hold*) chamadas

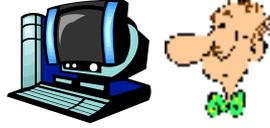
# Estabelecendo uma chamada para um endereço IP conhecido

Alice

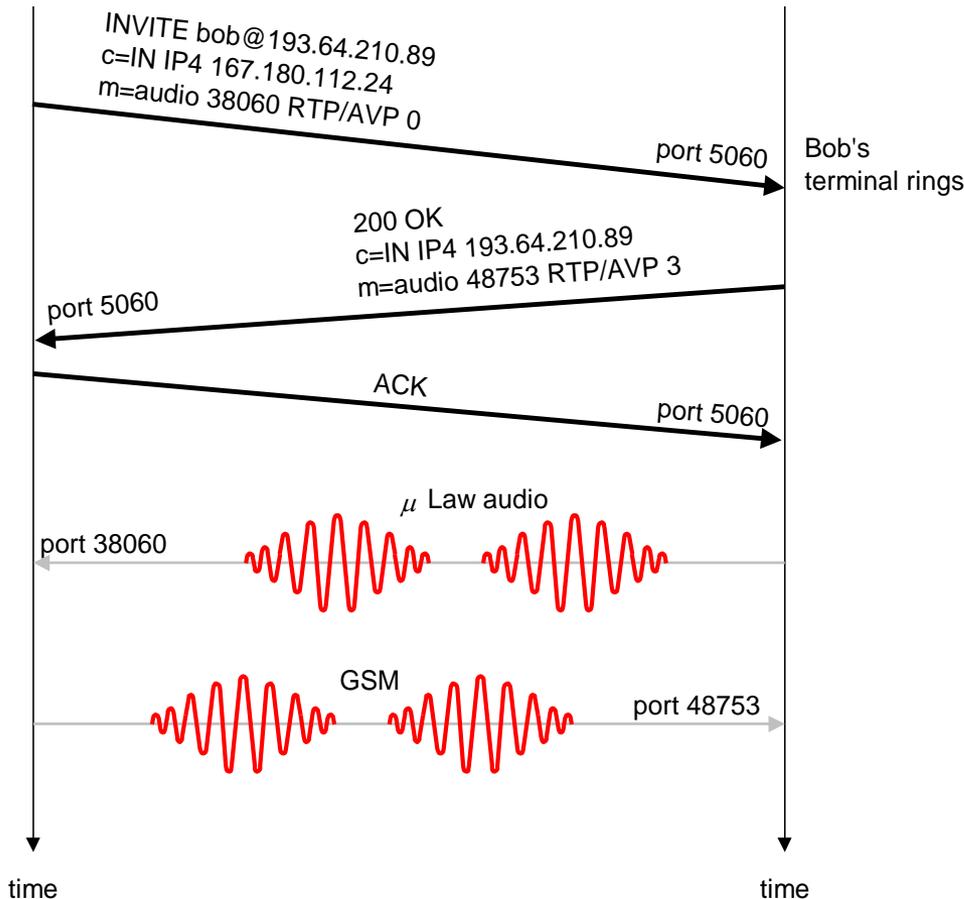


167.180.112.24

Bob



193.64.210.89



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

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

- As mensagens SIP podem ser transmitidas sobre TCP ou UDP; aqui está sendo enviada sobre RTP/UDP

- O número de porta Default do SIP é a 5060.

# Estabelecendo uma chamada (mais)

- Negociação Codec:
  - Suponha que Bob não possui um codificador PCM ulaw.
  - Bob responderá então 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
- Rejeitando a chamada
  - Bob pode rejeitar com respostas "busy" (ocupado), "gone" (fora), "payment required" (necessário pagamento), "forbidden" (proibido).
- A mídia pode ser enviada sobre RTP ou algum outro protocolo

# Exemplo de mensagem SIP

```
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 mensagem HTTP
- ❑ sdp = *session description protocol* (protocolo de descrição da sessão)
- ❑ Identificador (Call-ID) único para cada chamada.

- Aqui não conhecemos o endereço IP de Bob. Serão necessários servidores SIP intermediários.

- Alice envia e recebe mensagens SIP usando o número de porta default do SIP 5060.

- Alice especifica em Via: cabeçalho que o cliente SIP envia e recebe mensagens SIP sobre UDP

# Tradução de Nome e localização do usuário

- ❑ Chamador deseja chamar o chamado, mas possui apenas o nome ou o endereço de e-mail do chamado.
  - ❑ Precisa obter o endereço IP do host atual do chamado:
    - usuário se desloca
    - protocolo DHCP
    - usuário possui diferentes dispositivos IP (PC, PDA, dispositivo no automóvel)
  - ❑ Resultado pode depender de:
    - hora do dia (trabalho, casa)
    - chamador (não deseja que o chefe lhe chame em casa)
    - status do chamado (chamadas enviadas para correio de voz quando o chamado já estiver falando com alguém)
- Serviço provido por servidores SIP:
- ❑ Servidor de registro SIP
  - ❑ Servidor proxy do SIP

# Registro SIP

- Quando Bob inicia cliente SIP, o cliente envia uma mensagem SIP de registro (REGISTER) para o servidor de registros de Bob  
(função semelhante é necessária para serviço 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
```

# Proxy SIP

- ❑ Alice envia mensagem de convite para o seu servidor proxy
  - contém endereço sip:bob@domain.com
- ❑ O proxy é responsável por rotear mensagens SIP para o chamado
  - possivelmente através de múltiplos proxies
- ❑ O chamado envia resposta através do mesmo conjunto de proxies.
- ❑ O proxy retorna a mensagem de resposta SIP para Alice
  - contendo o endereço IP de Bob
- ❑ Nota: proxy é análogo aum servidor DNS local

# Exemplo

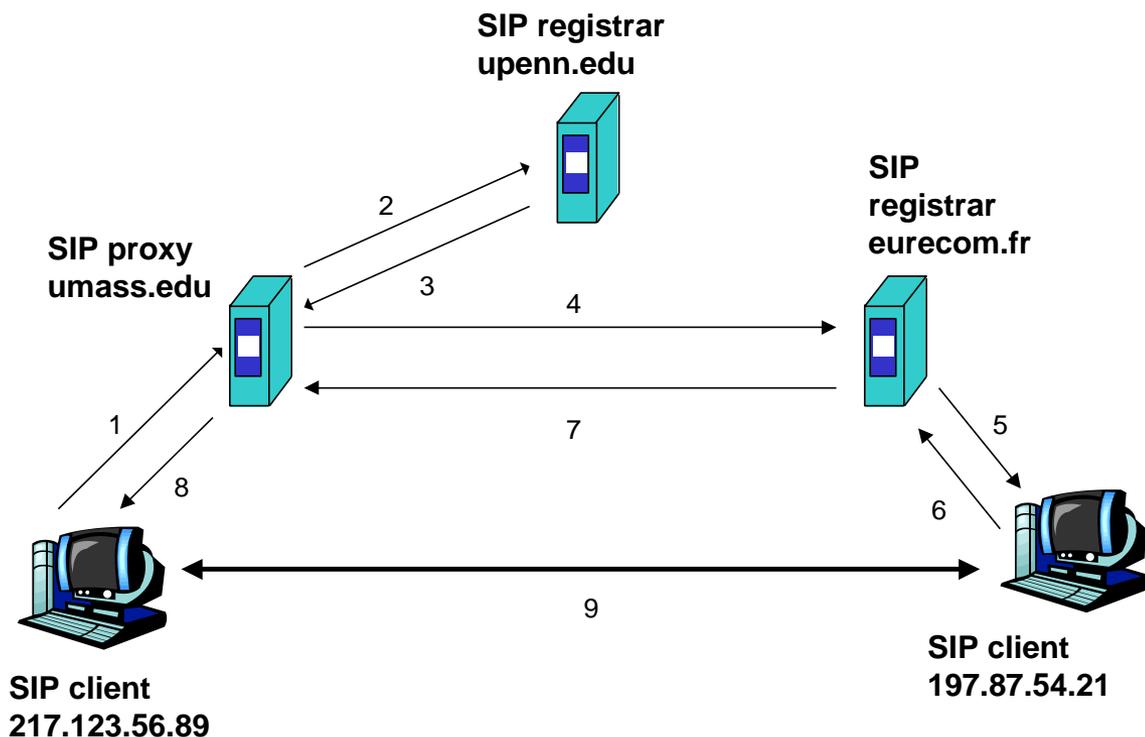
Chamador `jim@umass.edu`  
coloca uma chamada para  
`keith@upenn.edu`

(1) Jim envia mensagem INVITE para o proxy SIP da UMass. (2) Proxy encaminha o pedido para o servidor de registro da UPenn.

(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. (5) Registro da eurecom encaminha o INVITE para 197.87.54.21, que está rodando o cliente SIP de Keith. (6-8) retorno da resposta SIP. (9) mídia enviada diretamente entre clientes.

**Nota:** mensagens ack do SIP, não apresentadas.



# Comparação com o H.323

- ❑ H.323 é um outro protocolo de sinalização para tempo-real, interativo
- ❑ H.323 é um conjunto integrado de protocolos para conferência multimídia: sinalização, registro, controle de admissão, transporte e codecs.
- ❑ SIP é um único componente. Trabalha com RTP, mas não é obrigatório. Pode ser combinado com outros protocolos e serviços.
- ❑ H.323 vem do ITU (telefonia).
- ❑ SIP vem do IETF: Empresta muitos dos seus conceitos do HTTP. SIP tem um sabor Web, enquanto que o H.323 tem um sabor da telefonia.
- ❑ SIP usa o princípio KISS: *Keep it simple stupid.*

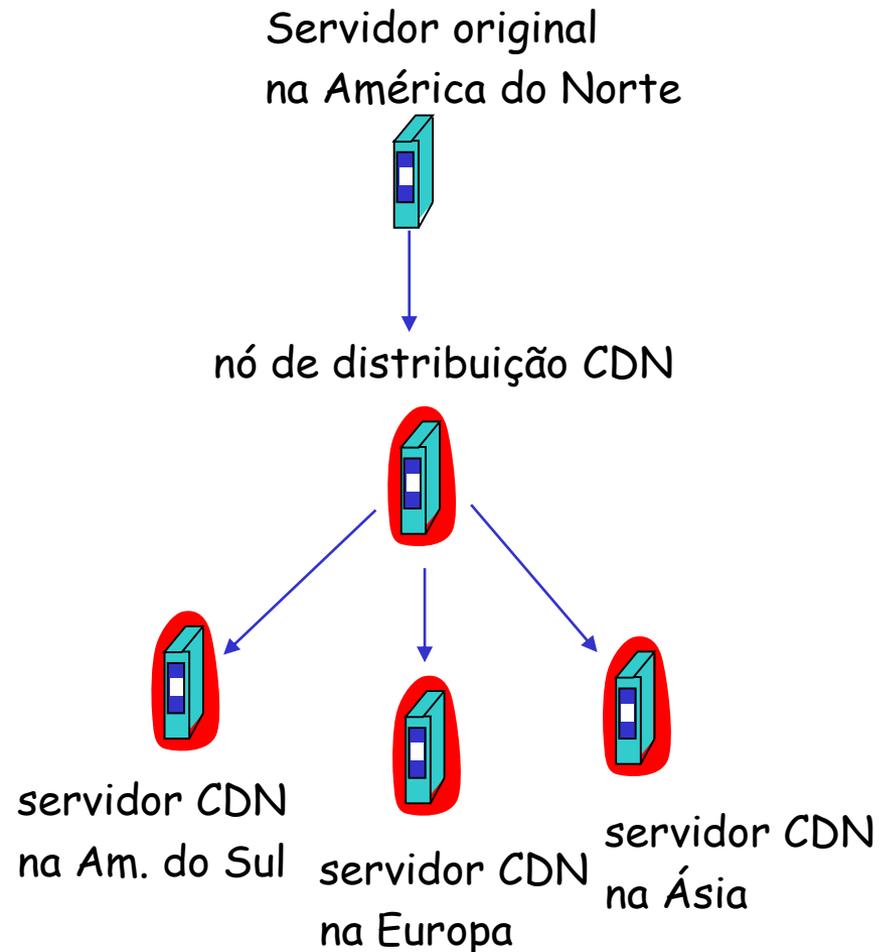
# Capítulo 7: Roteiro

- 7.1 Aplicações Multimídia em Redes
- 7.2 Fluxos contínuos de áudio e vídeo armazenados
- 7.3 Multimídia em Tempo Real: estudo do Telefone Internet
- 7.4 Protocolos para Aplicações Interativas de Tempo Real
  - RTP, RTCP, SIP
- 7.5 Distribuição de Multimídia: redes de distribuição de conteúdos
- 7.6 Além do Melhor Esforço
- 7.7 Mecanismos de Escalonamento e Policiamento
- 7.8 Serviços Integrados e Serviços Diferenciados
- 7.9 RSVP

# Redes de distribuição de conteúdos (CDNs - *Content distribution networks*)

## Replicação de conteúdo

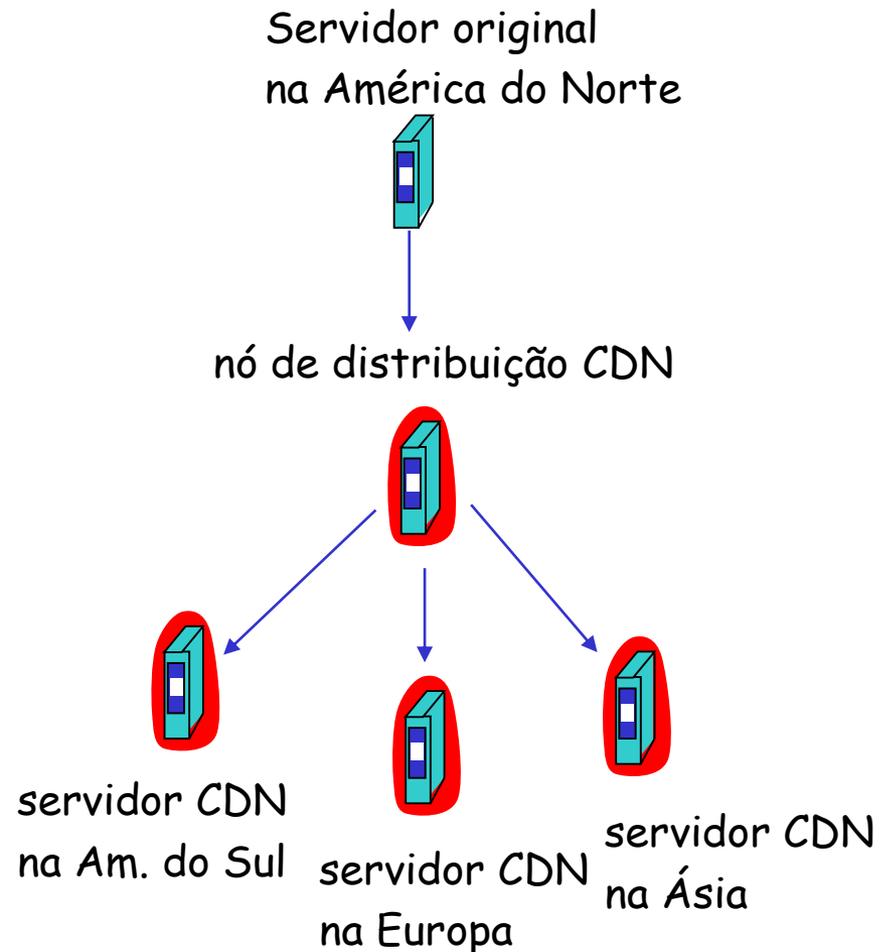
- Desafio transmitir fluxo de grandes arquivos (ex, vídeo) de um único servidor origem em tempo real
- Solução: replicar o conteúdo em centenas de servidores através da Internet
  - conteúdo carregado antecipadamente nos servidores CDN
  - Colocando o conteúdo "perto" do usuário evita impedimentos (perda,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



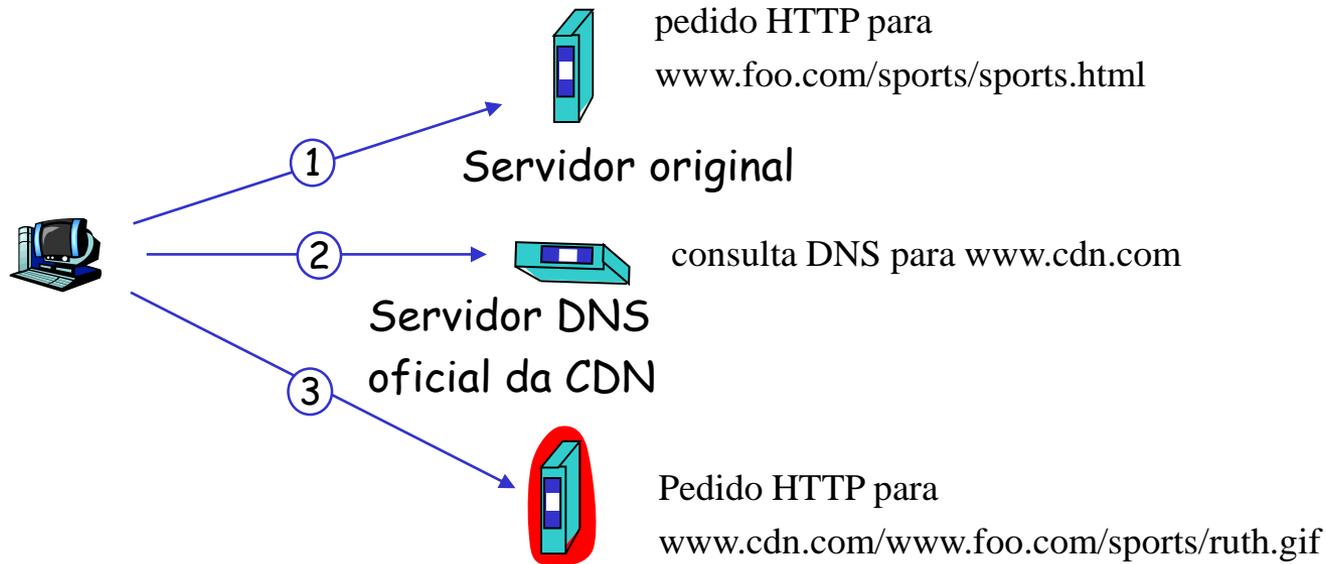
# Redes de distribuição de conteúdos (CDNs - *Content distribution networks*)

## 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>

Servidor  
CDN próximo

## Empresa CDN

- ❑ cdn.com
- ❑ distribui arquivos gif
- ❑ usa o seu servidor DNS oficial para redirecionar os pedidos

# Mais sobre CDNs

## roteamento de pedidos

- ❑ A CDN cria um "mapa", indicando as distâncias entre os ISPs folhas e os nós CDN
- ❑ quando a solicitação chega num servidor DNS oficial:
  - o servidor determina qual é o ISP de onde provém o pedido
  - usa o "mapa" para determinar qual o melhor servidor CDN
- ❑ nós CDN criam uma rede sobreposta na camada de aplicação

# Capítulo 7: Roteiro

- 7.1 Aplicações Multimídia em Redes
- 7.2 Fluxos contínuos de áudio e vídeo armazenados
- 7.3 Multimídia em Tempo Real: estudo do Telefone Internet
- 7.4 Protocolos para Aplicações Interativas de Tempo Real
  - RTP, RTCP, SIP
- 7.5 Distribuição de Multimídia: redes de distribuição de conteúdos
- 7.6 Além do Melhor Esforço
- 7.7 Mecanismos de Escalonamento e Policiamento
- 7.8 Serviços Integrados e Serviços Diferenciados
- 7.9 RSVP

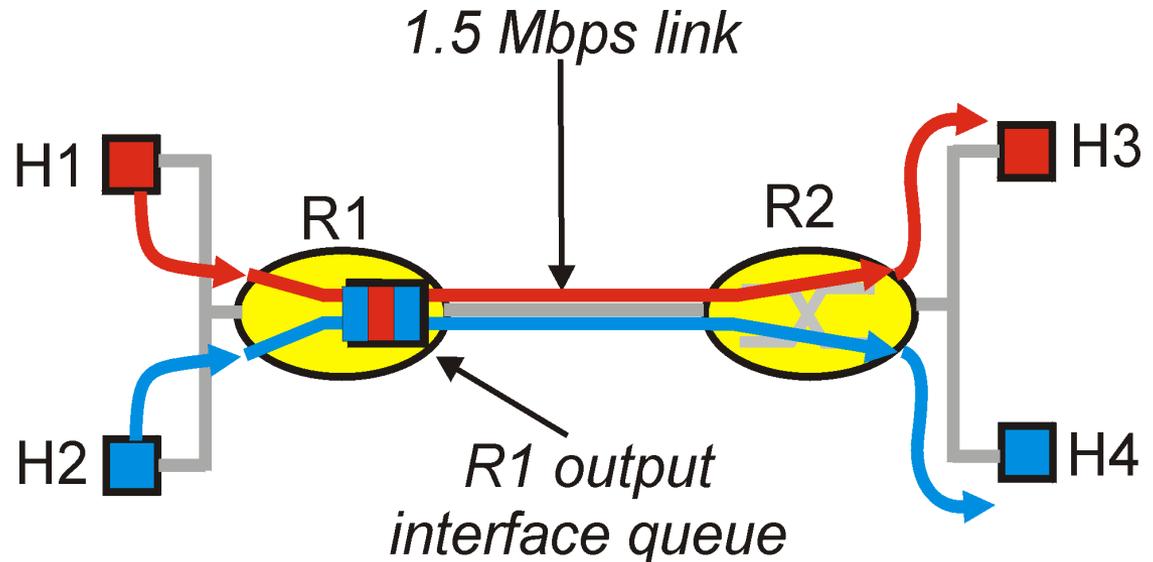
# Melhorando a QoS em Redes IP

**Até o momento:** "extraímos o máximo do melhor-esforço"

**Futuro:** próxima geração da Internet com garantias de QoS

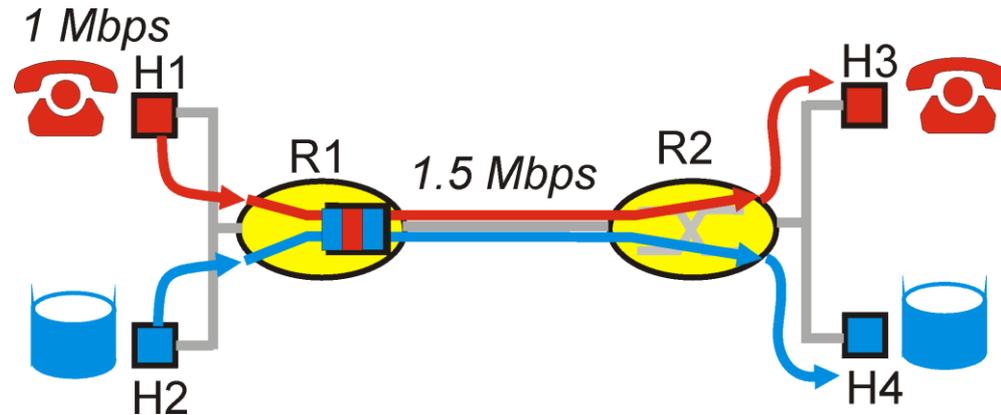
- **RSVP:** sinalização para reservas de recursos
- **Serviços Diferenciados:** garantias diferenciadas
- **Serviços Integrados:** garantias firmes

- Modelo simples para estudos de compartilhamento e congestionamento



# Princípios para a Garantia da QoS

- Exemplo: Telefone IP de 1Mbps, FTP compartilhando enlace de 1,5 Mbps.
  - Surtos de FTP podem congestionar o roteador e causar a perda de pacotes de áudio.
  - Gostaríamos de dar prioridade ao tráfego de áudio sobre o de FTP

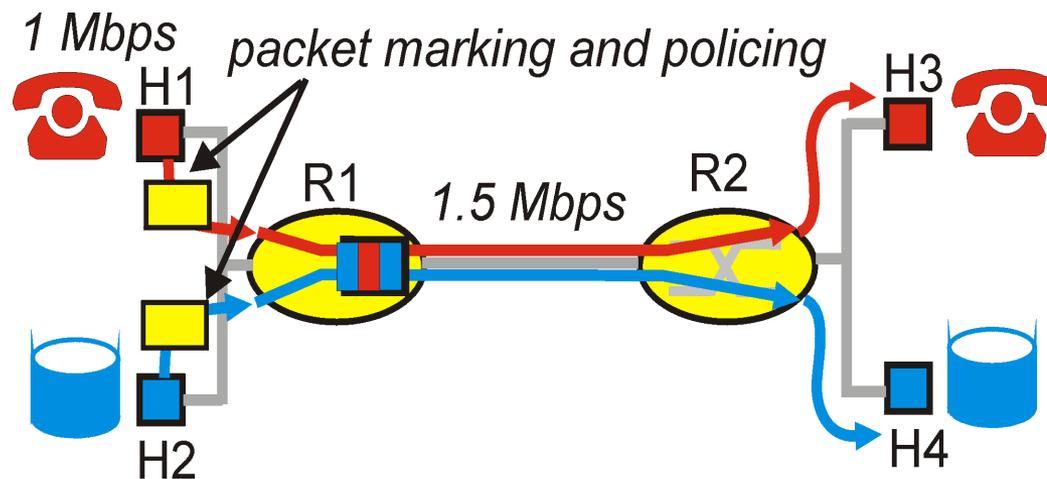


## Princípio 1

É preciso marcar os pacotes para que o roteador faça uma distinção entre as classes diferentes; e uma nova política no roteador para tratar os pacotes de forma diferenciada

## Princípios para a Garantia da QOS (cont.)

- ❑ E se as aplicações se comportarem mal (áudio envia pacotes a uma taxa mais elevada do que a declarada)?
  - Policiamento: força que as fontes respeitem as alocações de banda
- ❑ marcação e policiamento nas bordas da rede:
  - semelhante à UNI (*User Network Interface*) do ATM

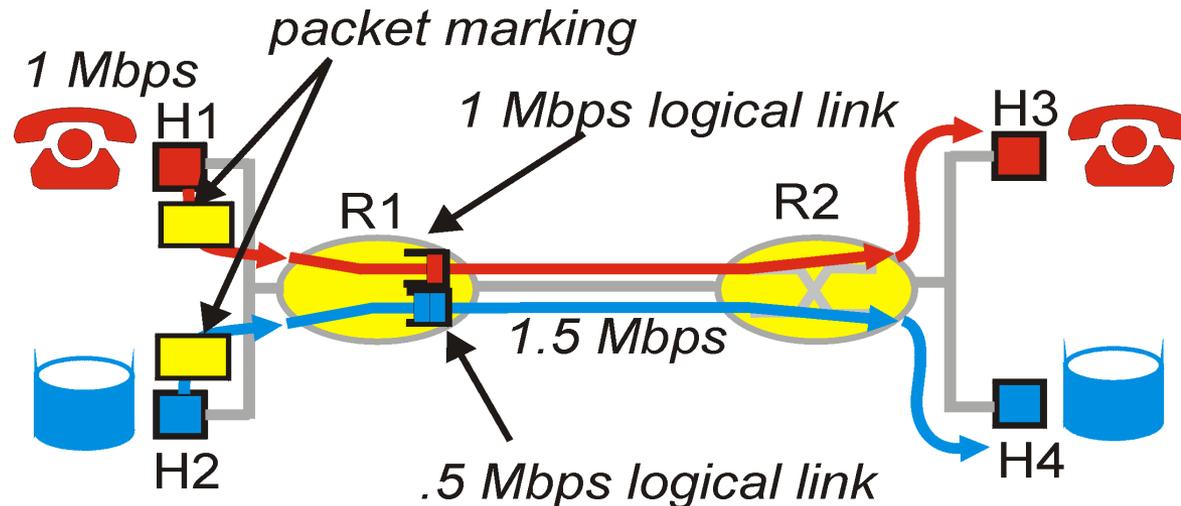


### Princípio 2

prover proteção (*isolamento*) de uma classe sobre as demais

## Princípios para a Garantia da QOS (cont.)

- ❑ Alocar uma largura de banda *fixa* (não-compartilhável) para o fluxo: uso *ineficiente* da banda se os fluxos não usarem suas alocações

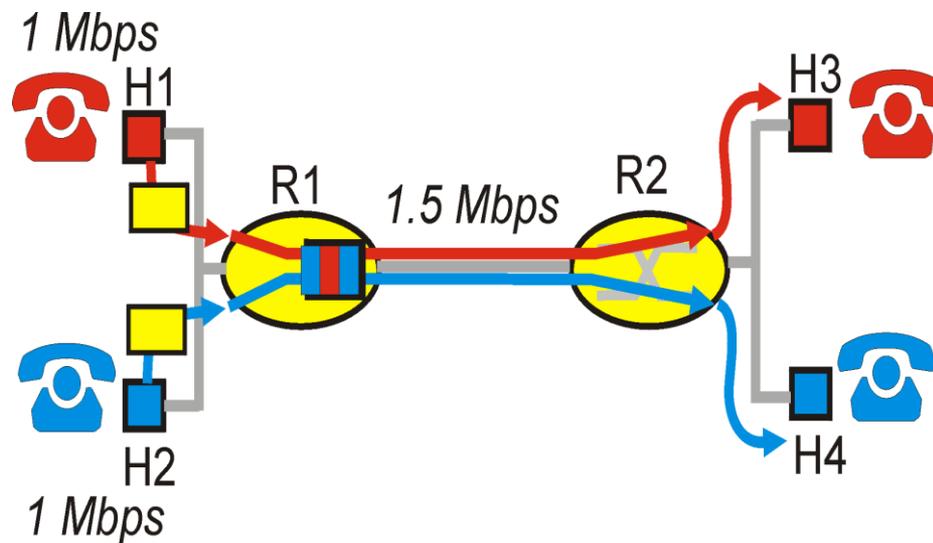


### Princípio 3

Enquanto provê isolamento, é desejável utilizar os recursos o mais eficientemente possível

## Princípios para a Garantia da QOS (cont.)

- *Fato básico da vida: não é possível atender a um tráfego superior à capacidade do enlace*

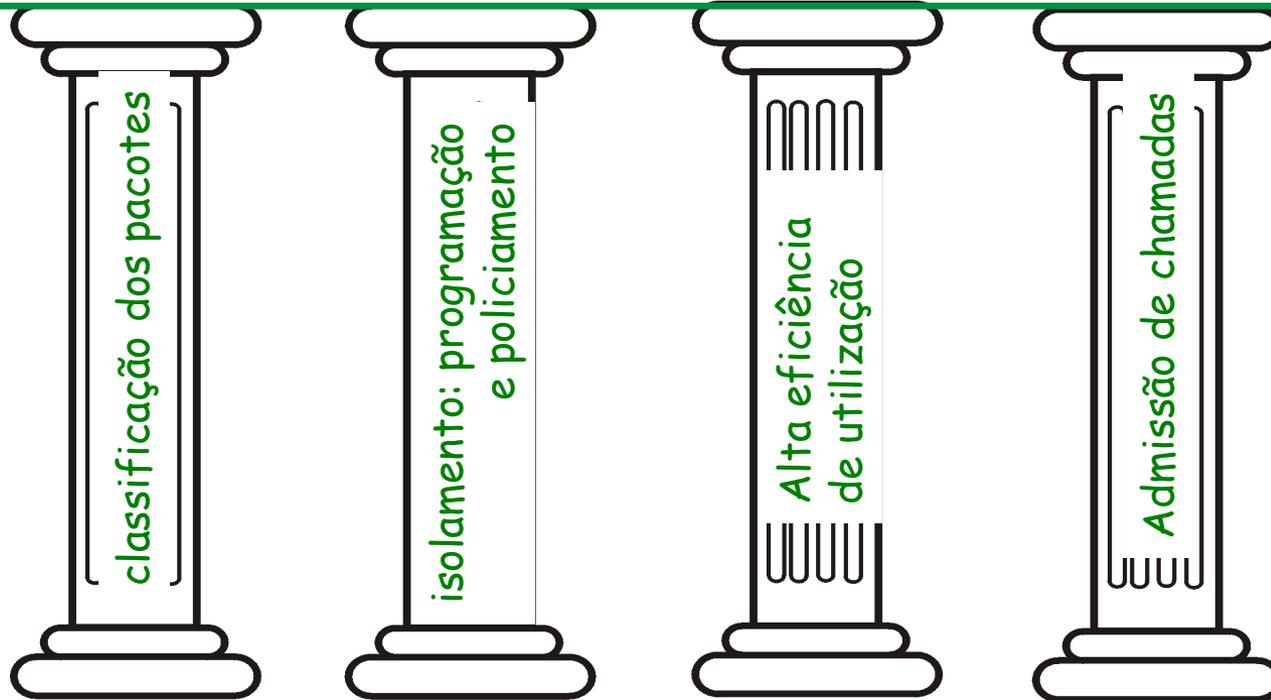


### Princípio 4

Admissão de Chamadas: o fluxo da aplicação declara as suas necessidades, a rede pode bloquear a chamada se não puder atender a estas necessidades

# Resumo dos Princípios de QoS

## QoS para Aplicações em Rede



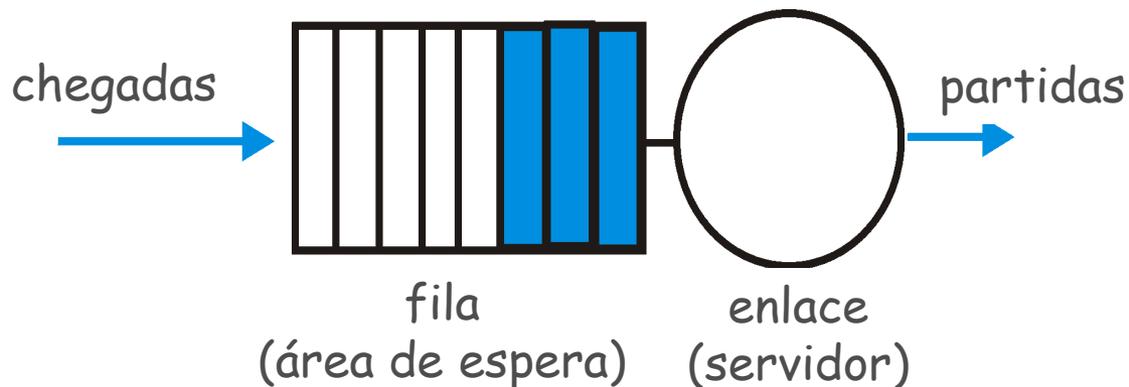
Olhemos a seguir para os mecanismos que obtêm isto...

# Capítulo 7: Roteiro

- ❑ 7.1 Aplicações Multimídia em Redes
- ❑ 7.2 Fluxos contínuos de áudio e vídeo armazenados
- ❑ 7.3 Multimídia em Tempo Real: estudo do Telefone Internet
- ❑ 7.4 Protocolos para Aplicações Interativas de Tempo Real
  - RTP, RTCP, SIP
- ❑ 7.5 Distribuição de Multimídia: redes de distribuição de conteúdos
- ❑ 7.6 Além do Melhor Esforço
- ❑ 7.7 Mecanismos de Escalonamento e Policiamento
- ❑ 7.8 Serviços Integrados e Serviços Diferenciados
- ❑ 7.9 RSVP

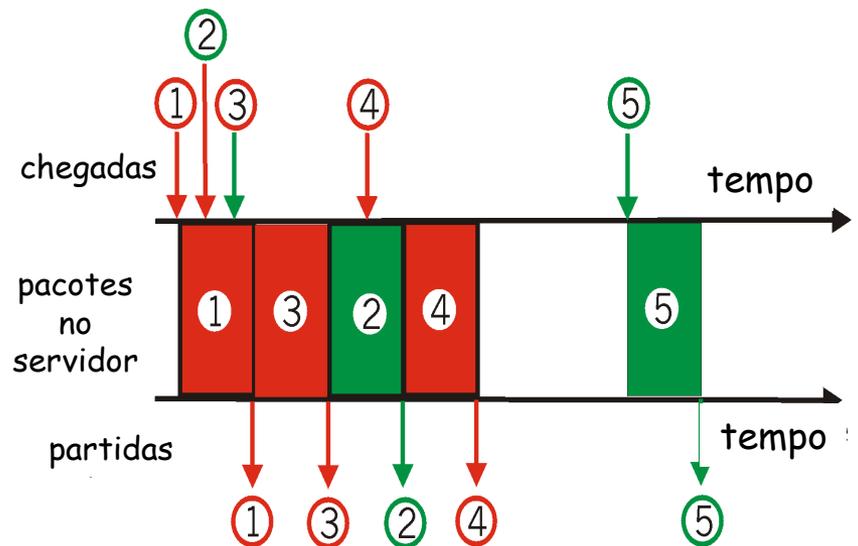
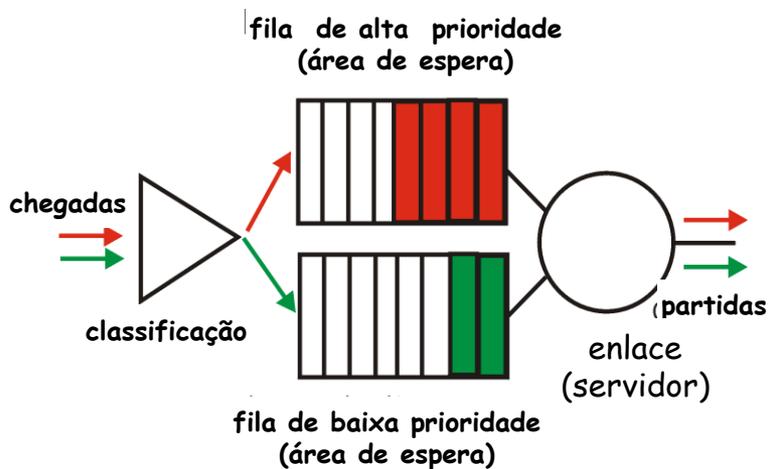
# Mecanismos de Escalonamento e Policiamento

- **Escalonamento**: escolha do próximo pacote para transmissão num canal
- **FIFO (*first in first out*)**: transmite na ordem de chegada à fila
  - Exemplos da vida real?
  - **política de descarte**: se os pacotes ao chegarem encontrarem a fila cheia: quem deve ser descartado?
    - Descarta o último (cauda): descarta o pacote que acabou de chegar
    - Prioridade: descarta/remove baseado na prioridade
    - Aleatório: descarta/remove aleatoriamente



# Disciplinas de Escalonamento: mais

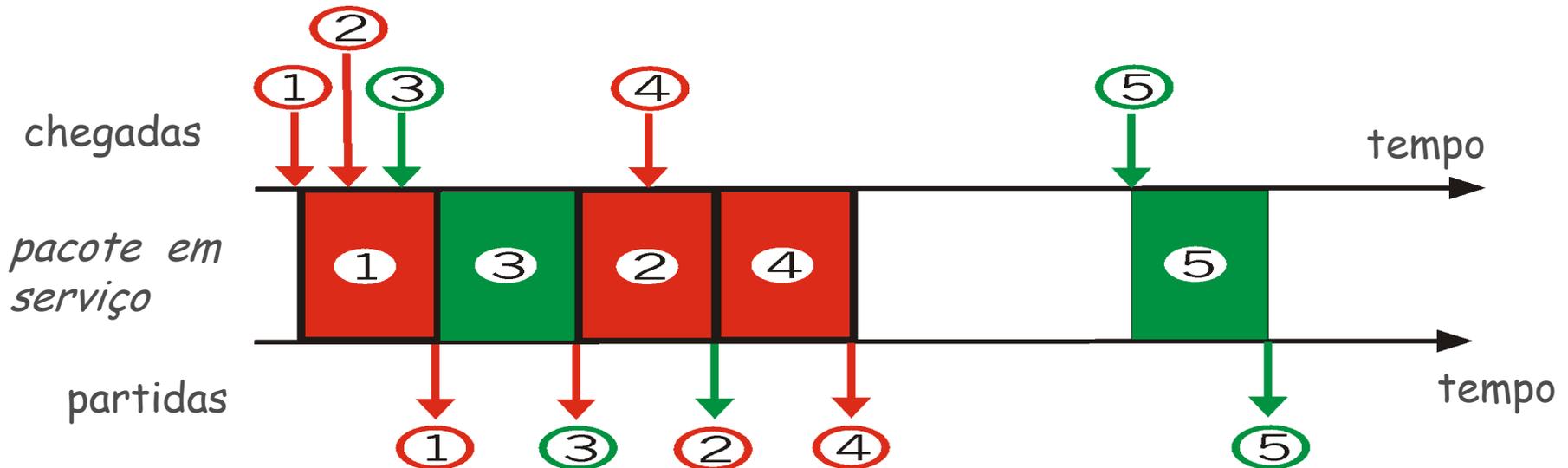
- ❑ **Escalonamento com Prioridades:** transmite enfileirado de mais alta prioridade
- ❑ múltiplas classes, com diferentes prioridades
  - classe pode depender da marcação ou outra informação do cabeçalho (ex. IP origem/destino, números de portas, etc.)
  - Exemplo da vida real?



# Disciplinas de Escalonamento (cont.)

## escalonamento *Round Robin* (circular):

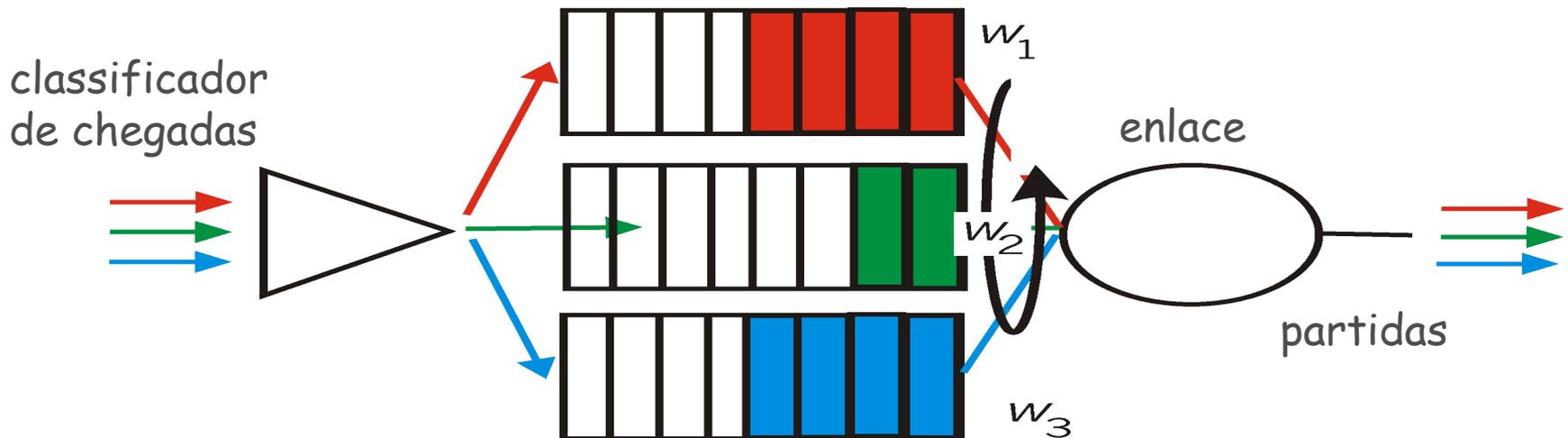
- ❑ múltiplas classes
- ❑ varre as filas das classes transmitindo um pacote de cada classe cuja fila não estiver vazia
- ❑ exemplo da vida real?



# Disciplinas de Escalonamento (cont.)

## *Weighted Fair Queuing (Fila justa ponderada):*

- ❑ *Round Robin* generalizado
- ❑ cada classe recebe um tempo de serviço diferenciado em cada ciclo
- ❑ exemplo do mundo real?



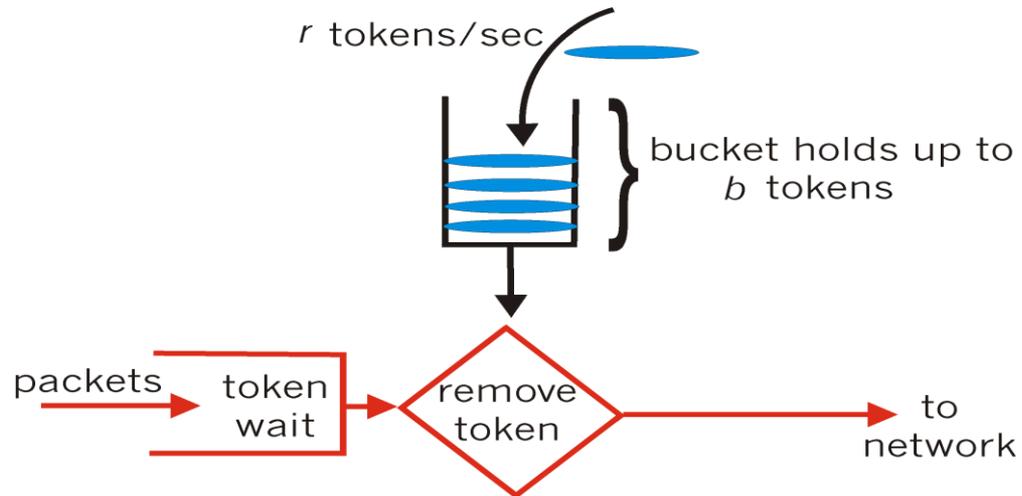
# Mecanismos de Policiamento

**Objetivo:** limita o tráfego para que não exceda os parâmetros declarados

- ❑ Três critérios comumente usados:
- ❑ *Taxa Média (de Longo prazo):* quantos pacotes podem ser enviados por unidade de tempo (em longo prazo)
  - questão crucial: qual é o comprimento do intervalo: 100 pacotes por seg ou 6000 pacotes por min têm a mesma média!
- ❑ *Taxa de Pico:* ex., 6000 pctes por minuto (ppm) em média e taxa de pico de 1500 ppm
- ❑ *Comprimento (Máx.) do Surto:* número máximo de pacotes enviados consecutivamente (sem intervalo ocioso)

# Mecanismos de Policiamento

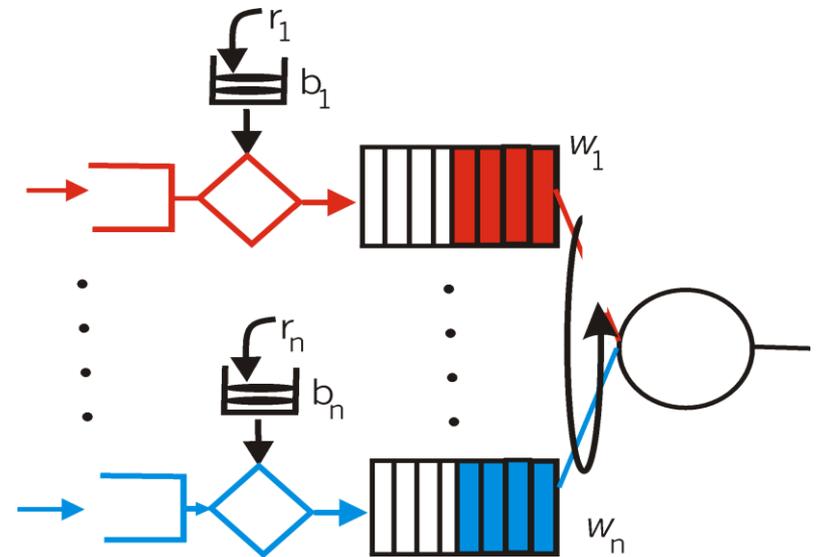
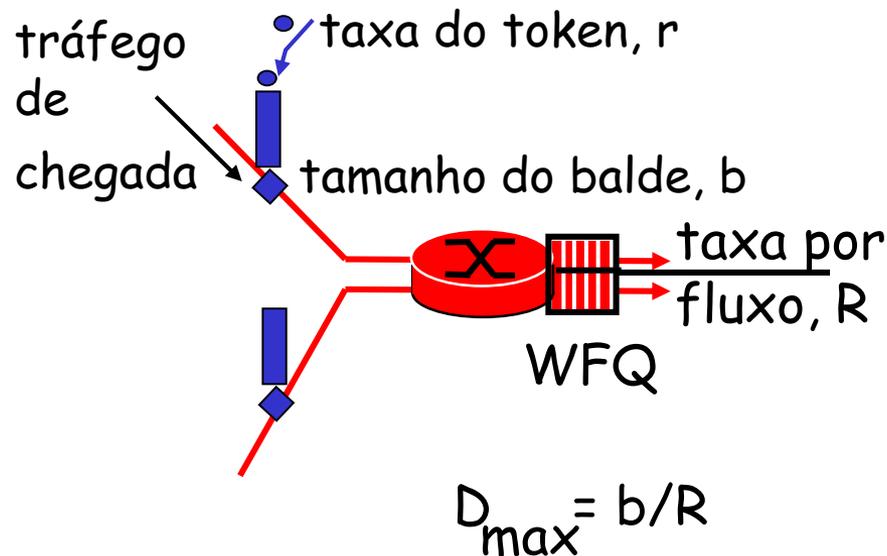
*Token Bucket (Balde de Permissões)*: limita a entrada para Tamanho do Surto e Taxa Média especificadas.



- ❑ balde pode guardar  $b$  permissões
- ❑ tokens são gerados a uma taxa de  $r$  token/seg a menos que o balde esteja cheio
- ❑ *num intervalo de comprimento  $t$ : número de pacotes admitidos é menor ou igual a  $(r t + b)$ .*

# Mecanismos de Policiamento (mais)

- combinação de *token bucket*, WFQ provê limite superior garantido no atraso, i.e., *garantia de QoS!*



# Capítulo 7: Roteiro

- 7.1 Aplicações Multimídia em Redes
- 7.2 Fluxos contínuos de áudio e vídeo armazenados
- 7.3 Multimídia em Tempo Real: estudo do Telefone Internet
- 7.4 Protocolos para Aplicações Interativas de Tempo Real
  - RTP, RTCP, SIP
- 7.5 Distribuição de Multimídia: redes de distribuição de conteúdos
- 7.6 Além do Melhor Esforço
- 7.7 Mecanismos de Escalonamento e Policiamento
- 7.8 Serviços Integrados e Serviços Diferenciados
- 7.9 RSVP

# Serviços Integrados do IETF

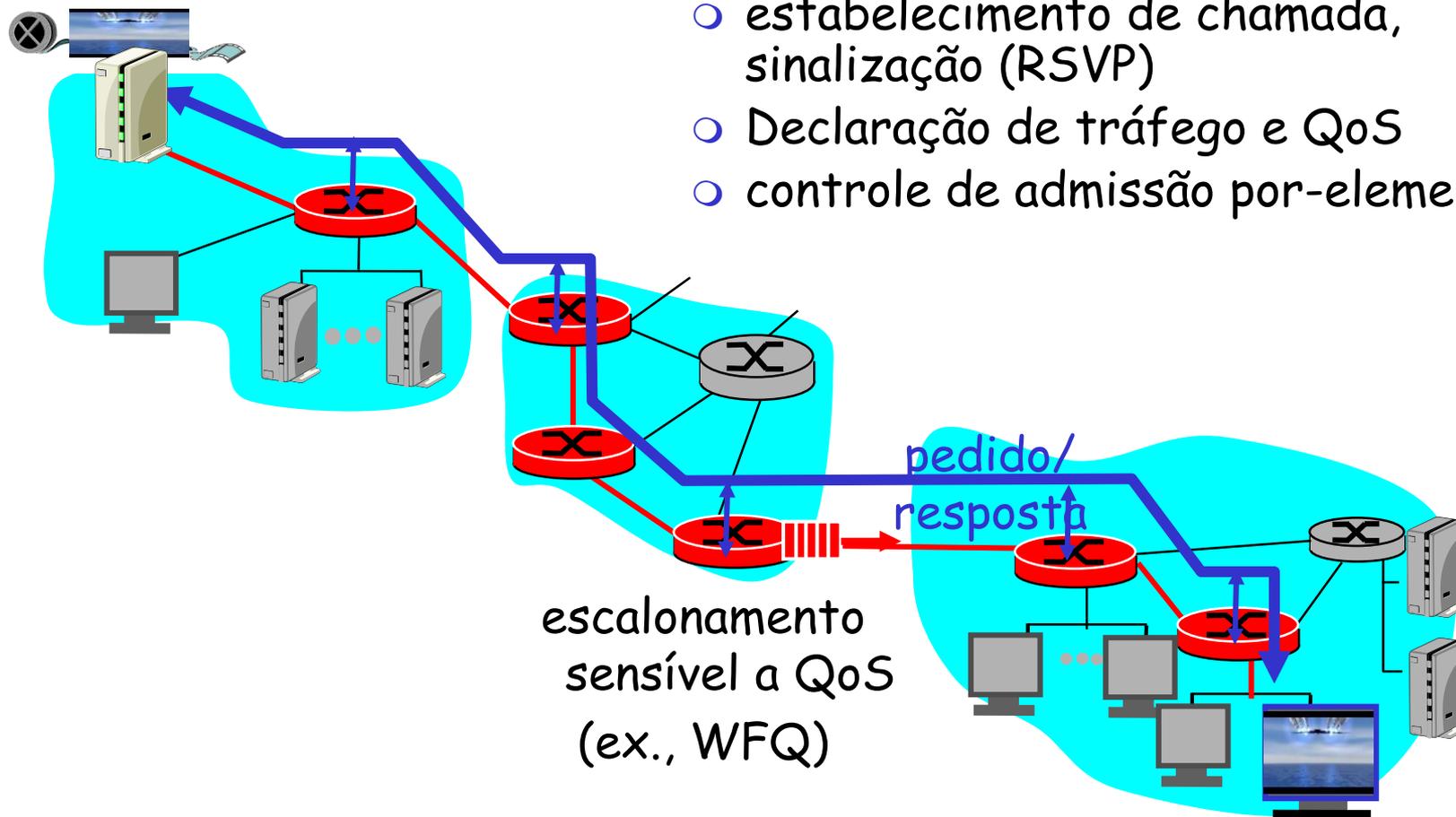
- ❑ Uma arquitetura para prover garantias de QoS em redes IP para sessões individuais de aplicações
- ❑ reserva de recursos: roteadores devem manter info de estado (Circuito Virtual??), manter registros dos recursos alocados, requisitos de QoS
- ❑ admitir/rejeitar novos pedidos de chamadas:

Pergunta: os novos fluxos que chegam podem ser admitidos com garantias de desempenho e ainda assim não violar as garantias de QoS prometidas a fluxos já admitidos?

# Intserv: cenário de garantia de QoS

## □ Reserva de recursos

- estabelecimento de chamada, sinalização (RSVP)
- Declaração de tráfego e QoS
- controle de admissão por-elemento



# Admissão de Chamadas

A sessão entrante 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
- ❑ protocolo de sinalização: necessário para levar a R-spec e T-spec aos roteadores (onde a reserva é necessária)
  - **RSVP**

# Intserv QoS: Modelos de serviço

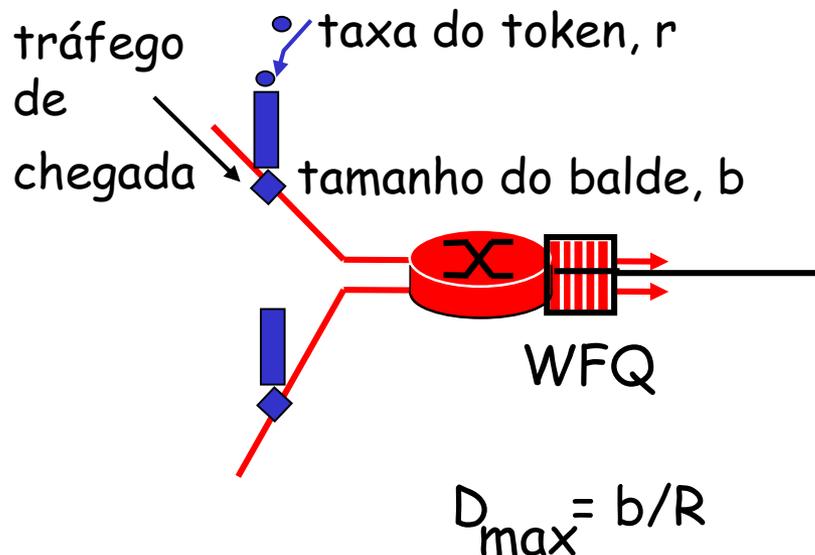
[rfc2211, rfc 2212]

## serviço Garantido:

- ❑ chegada de tráfego de pior caso: fonte policiada por um balde furado.
- ❑ *limite* simples (provável matematicamente) para o atraso [Parekh 1992, Cruz 1988]

## Serviço de Carga controlada:

- ❑ “uma qualidade de serviço que muito se aproxima da QoS que o mesmo fluxo receberia de um elemento de rede em carga leve”



# Serviços Diferenciados do IETF

## Preocupações com o Intserv:

- ❑ **Escalabilidade:** sinalização, manutenção do estado do roteador por fluxo é difícil para um grande número de fluxos
- ❑ **Modelos de Serviço Flexíveis:** Intserv tem apenas duas classes. Também deseja classes de serviço "qualitativas"
  - "comportamento como um fio"
  - distinção relativa entre serviços: Platina, Ouro, Prata

## Abordagem Diffserv :

- ❑ funções simples no núcleo da rede, funções relativamente complexas nos roteadores de borda (ou hosts)
- ❑ Não define classes de serviço, provê componentes funcionais para construir as classes de serviço

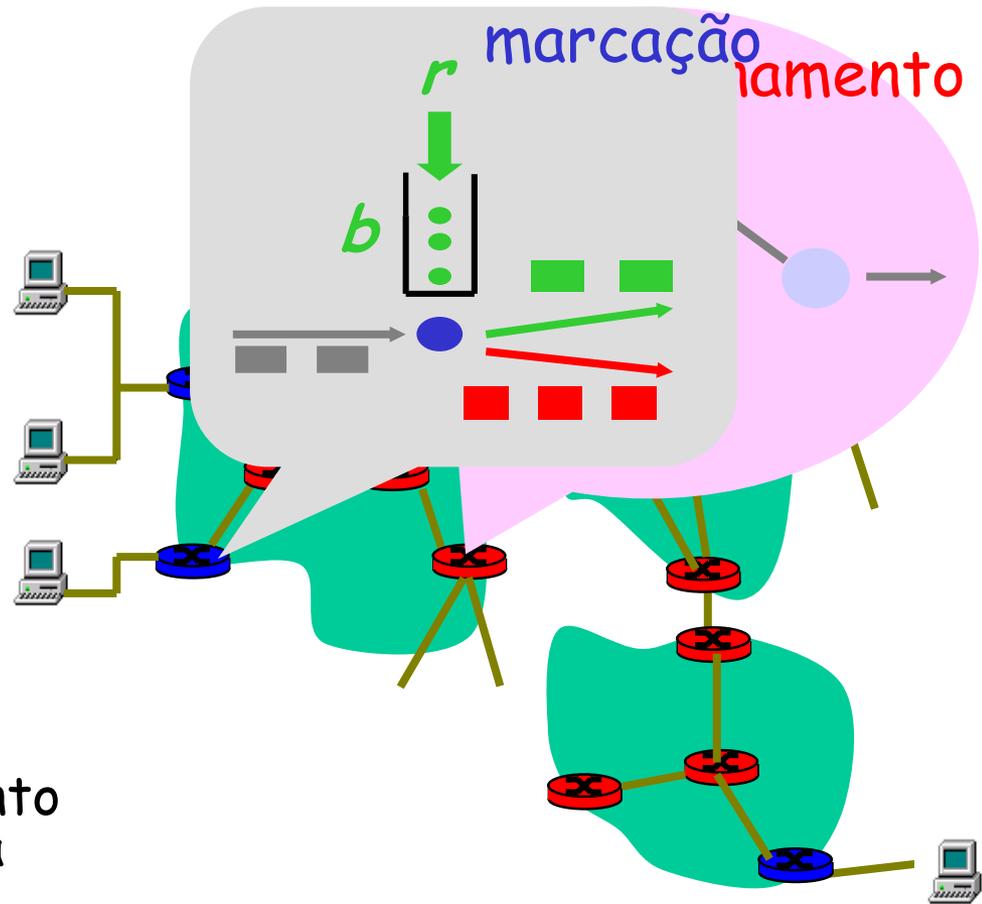
# Arquitetura Diffserv

## Roteador de borda:

- ❑ gerenciamento de tráfego por-fluxo
- ❑ marca os pacotes como dentro-perfil e fora-perfil

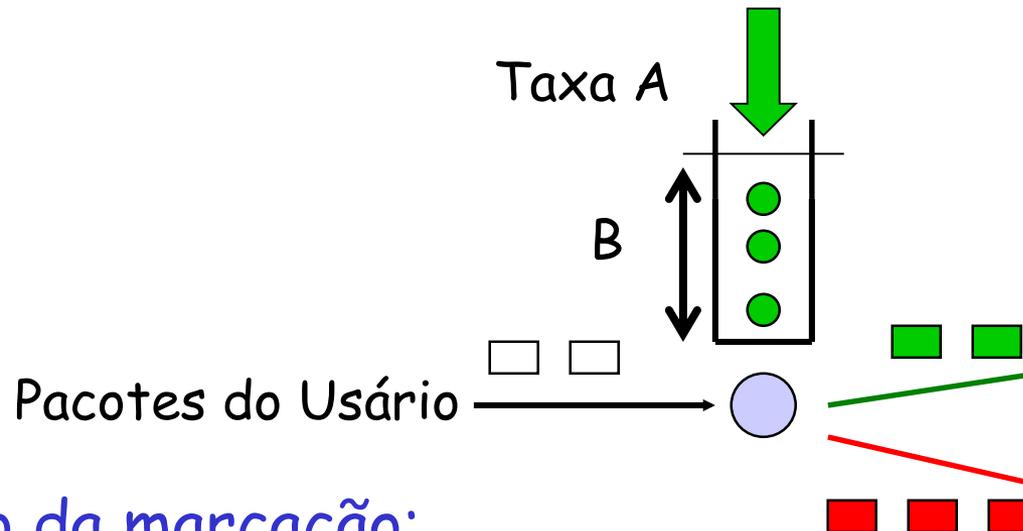
## Roteador de Núcleo:

- ❑ gerenciamento de tráfego por classe
- ❑ armazenamento e escalonamento baseado na marcação na borda
- ❑ Preferência para os pacotes dentro-perfil
- ❑ Encaminhamento assegurado



# Marcação de Pacotes no Roteador de Borda

- ❑ perfil: taxa A, comprimento do balde B pré-negociados
- ❑ marcação de pacote na borda baseada no perfil **por-fluxo**



## Possível uso da marcação:

- ❑ marcação baseada em classes: pacotes de classes diferentes são marcados diferentemente
- ❑ Marcação dentro da mesma classe: porção do fluxo bem comportado marcado diferentemente do mau comportado

# Classificação e Condicionamento

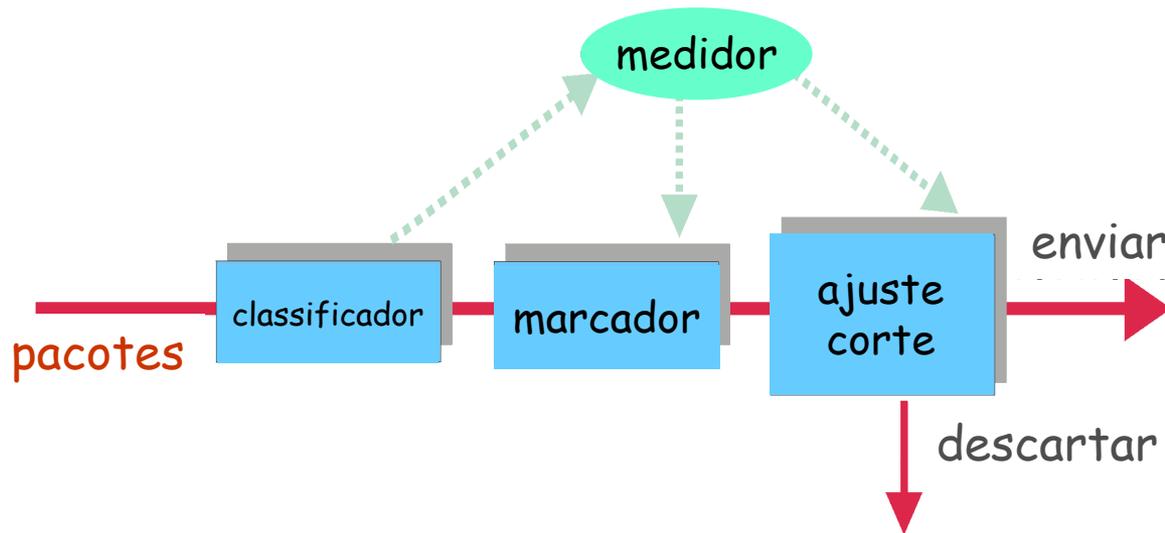
- ❑ O Pacote é marcado no campo de Tipo de Serviço (TOS) no IPv4 e, Classe de Tráfego no IPv6
- ❑ São usados 6 bits para fornecer a codificação dos Serviços Diferenciados e determinar a PHB que o pacote receberá
- ❑ No momento há 2 bits que não estão sendo usados.



# Classificação e Condicionamento

Pode ser desejável limitar a taxa de injeção de tráfego para alguma classe;

- ❑ o usuário declara o seu perfil de tráfego (ex. taxa e comprimento das rajadas);
- ❑ o tráfego é medido e moldado se não estiver de acordo com o seu perfil



# Encaminhamento (PHB)

- ❑ PHB resulta num comportamento de desempenho de encaminhamento diferente observável (mensurável)
- ❑ O PHB não especifica quais os mecanismos a serem usados para garantir o comportamento de desempenho PHB requisitado
- ❑ Exemplos:
  - Classe A recebe  $x\%$  da taxa de transmissão do enlace de saída dentro de intervalos de tempo de comprimento especificado
  - Pacotes da classe A deixam os buffers antes dos pacotes da classe B

# Encaminhamento (PHB)

## PHBs em desenvolvimento:

- ❑ *Expedited Forwarding (Encaminhamento Expresso)*: taxa de partida dos pacotes de uma classe é maior ou igual a uma taxa especificada
  - enlace lógico com uma taxa mínima garantida
- ❑ *Assured Forwarding (Encaminhamento Assegurado)*: 4 classes de tráfego
  - a cada uma é garantida uma quantidade mínima de largura de banda
  - cada uma com três partições de preferência para o descarte

# Capítulo 7: Roteiro

- 7.1 Aplicações Multimídia em Redes
- 7.2 Fluxos contínuos de áudio e vídeo armazenados
- 7.3 Multimídia em Tempo Real: estudo do Telefone Internet
- 7.4 Protocolos para Aplicações Interativas de Tempo Real
  - RTP, RTCP, SIP
- 7.5 Distribuição de Multimídia: redes de distribuição de conteúdos
- 7.6 Além do Melhor Esforço
- 7.7 Mecanismos de Escalonamento e Policiamento
- 7.8 Serviços Integrados e Serviços Diferenciados
- 7.9 RSVP

# Signaling in the Internet

connectionless  
(stateless)  
forwarding by IP  
routers + best effort  
service = no network  
signaling protocols  
in initial IP  
design

- ❑ **New requirement:** reserve resources along end-to-end path (end system, routers) for QoS for multimedia applications
- ❑ **RSVP:** Resource Reservation Protocol [RFC 2205]
  - “ ... allow users to communicate requirements to network in robust and efficient way.” i.e., signaling !
- ❑ earlier Internet Signaling protocol: ST-II [RFC 1819]

# RSVP Design Goals

1. accommodate **heterogeneous receivers** (different bandwidth along paths)
2. accommodate different applications **with different resource requirements**
3. make **multicast a first class service**, with adaptation to multicast group membership
4. **leverage existing multicast/unicast routing**, with adaptation to changes in underlying unicast, multicast routes
5. **control protocol overhead** to grow (at worst) linear in # receivers
6. **modular design** for heterogeneous underlying technologies

## RSVP: does not...

- ❑ specify how resources are to be reserved
  - ❑ rather: a mechanism for communicating needs
- ❑ determine routes packets will take
  - ❑ that's the job of routing protocols
  - ❑ signaling decoupled from routing
- ❑ interact with forwarding of packets
  - ❑ separation of control (signaling) and data (forwarding) planes

# RSVP: overview of operation

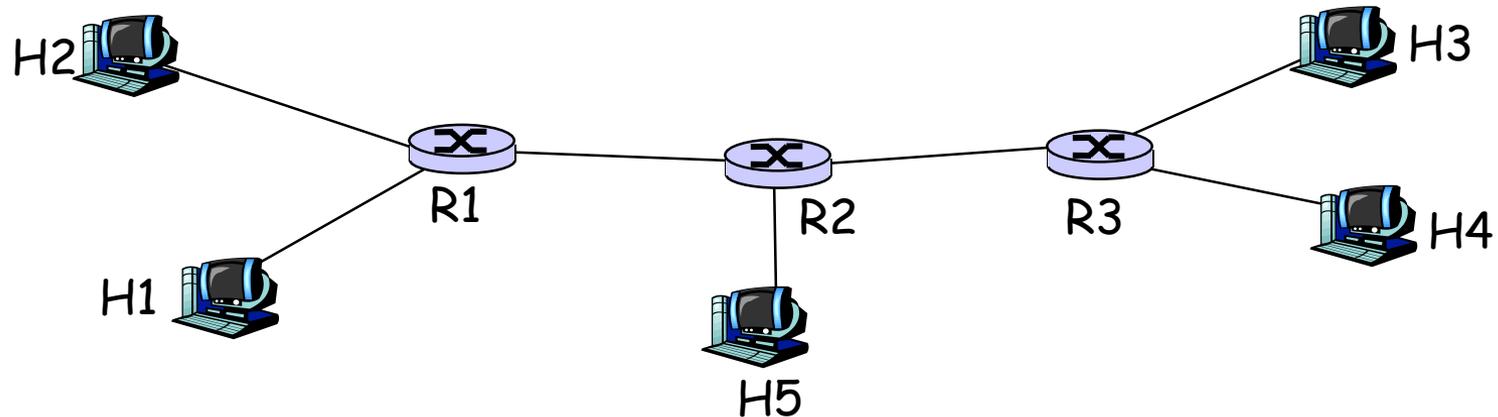
- ❑ **senders, receiver join a multicast group**
  - done outside of RSVP
  - senders need not join group
- ❑ **sender-to-network signaling**
  - *path message*: make sender presence known to routers
  - path teardown: delete sender's path state from routers
- ❑ **receiver-to-network signaling**
  - *reservation message*: reserve resources from sender(s) to receiver
  - reservation teardown: remove receiver reservations
- ❑ **network-to-end-system signaling**
  - path error
  - reservation error

# Path msgs: RSVP *sender-to-network* signaling

- **path message** contents:
  - *address*: unicast destination, or multicast group
  - *flowspec*: bandwidth requirements spec.
  - *filter flag*: if yes, record identities of upstream senders (to allow packets filtering by source)
  - *previous hop*: upstream router/host ID
  - *refresh time*: time until this info times out
- path message: communicates sender info, and reverse-path-to-sender routing info
  - later upstream forwarding of receiver reservations

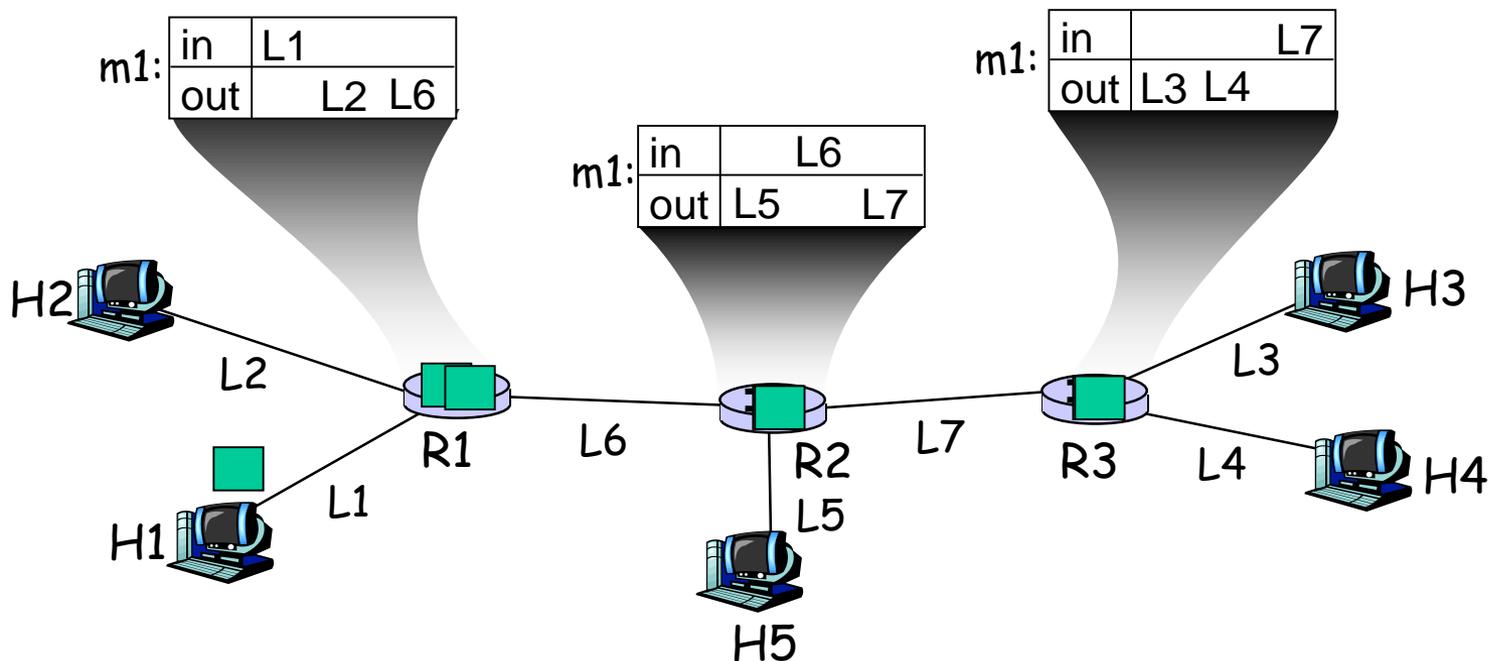
# RSVP: simple audio conference

- H1, H2, H3, H4, H5 both senders and receivers
- multicast group m1
- no filtering: packets from any sender forwarded
- audio rate:  $b$
- only one multicast routing tree possible



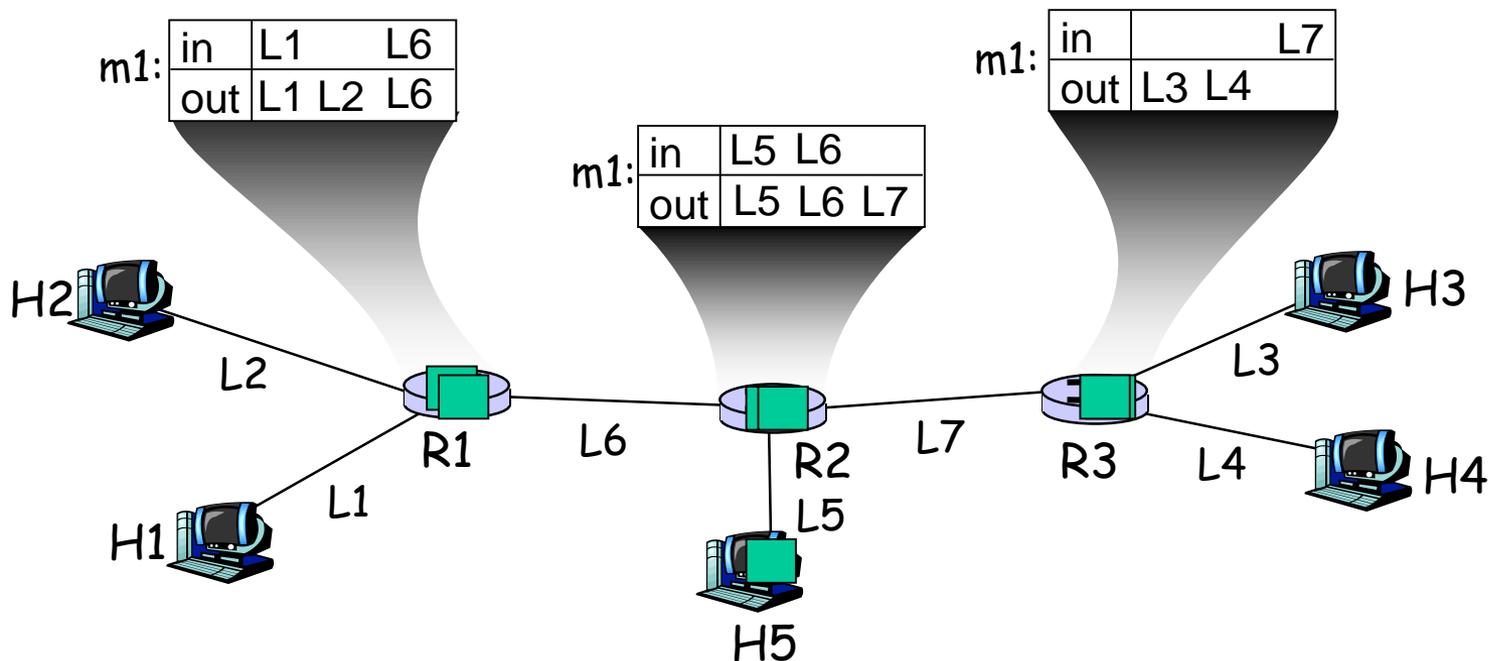
# RSVP: building up path state

- H1, ..., H5 all send path messages on *m1*:  
(address=*m1*, Tspec=*b*, filter-spec=no-filter,refresh=100)
- Suppose H1 sends first path message



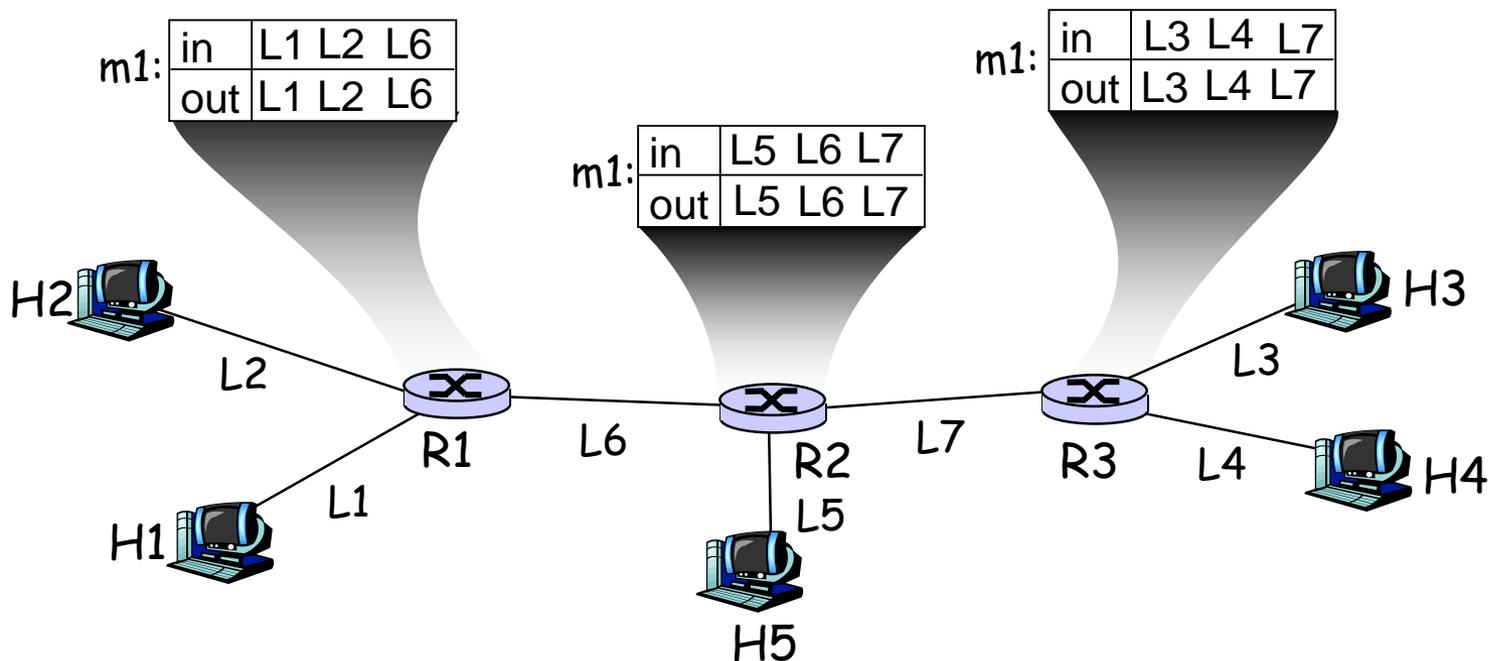
# RSVP: building up path state

- next, H5 sends path message, creating more state in routers



# RSVP: building up path state

- H2, H3, H5 send path msgs, completing path state tables



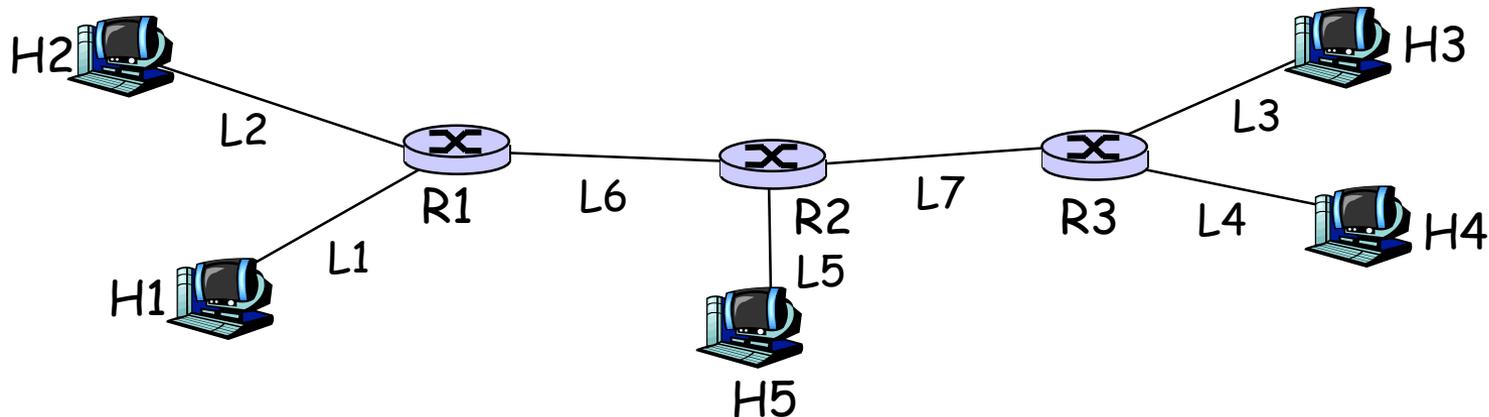
## reservation msgs: *receiver-to-network* signaling

- reservation message contents:
  - *desired bandwidth:*
  - *filter type:*
    - no filter: any packets address to multicast group can use reservation
    - fixed filter: only packets from specific set of senders can use reservation
    - dynamic filter: senders who's p[ackets can be forwarded across link will change (by receiver choce) over time.
  - *filter spec*
- reservations flow upstream from receiver-to-senders, reserving resources, creating additional, *receiver-related* state at routers

# RSVP: *receiver* reservation example 1

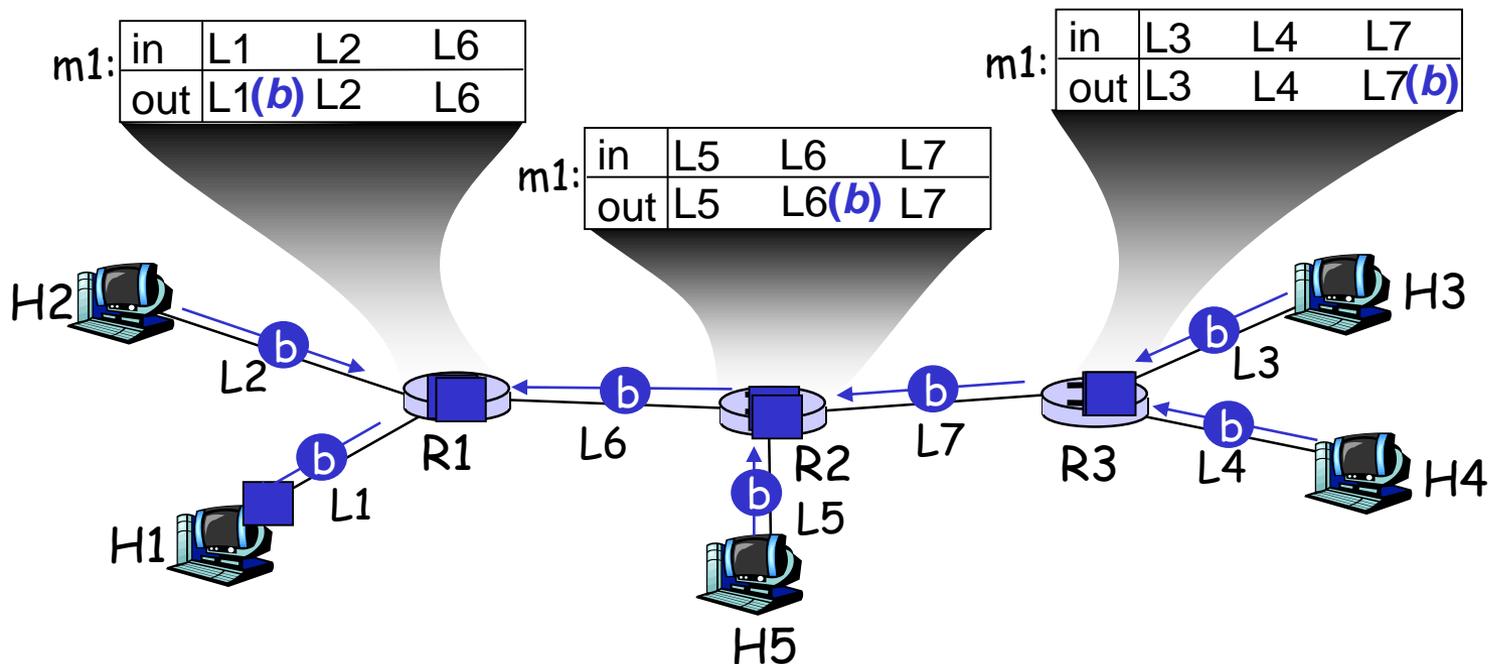
H1 wants to receive audio from all other senders

- H1 reservation msg flows uptree to sources
- H1 only reserves enough bandwidth for 1 audio stream
- reservation is of type "no filter" - any sender can use reserved bandwidth



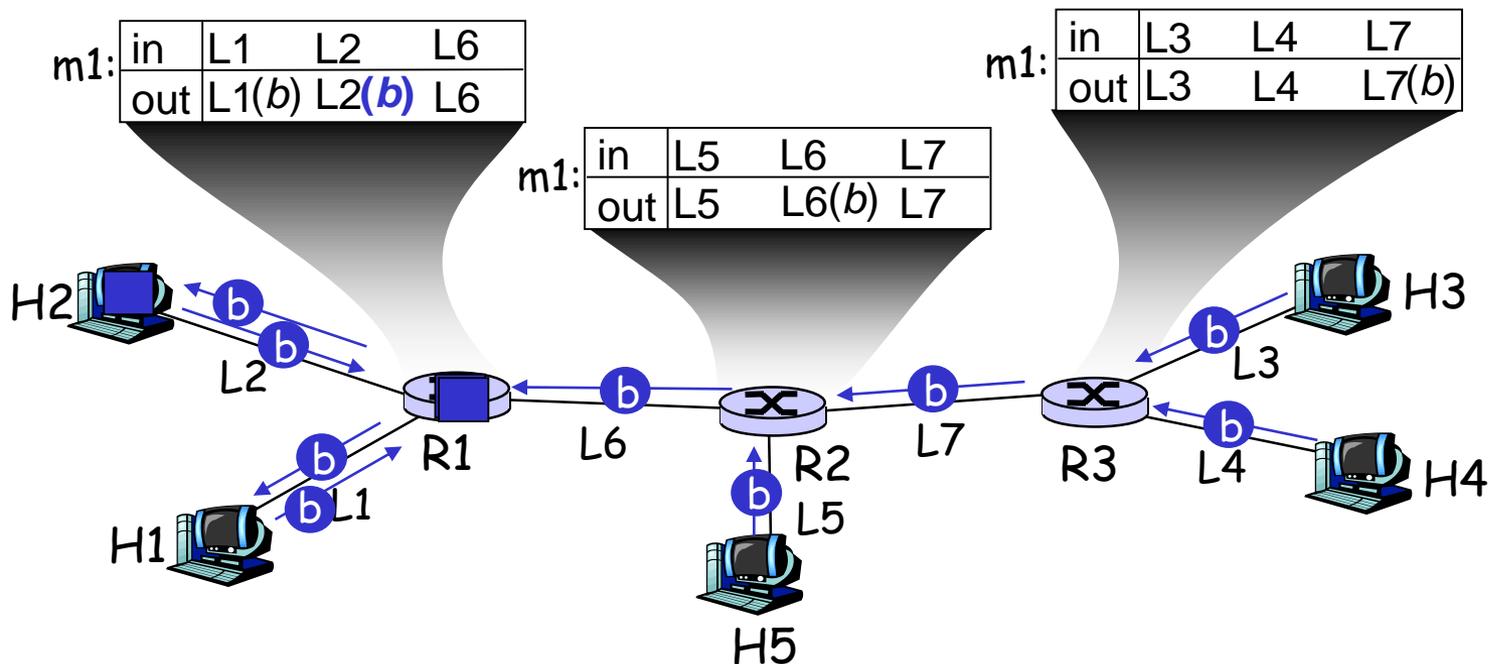
# RSVP: *receiver* reservation example 1

- H1 reservation msgs flows uptree to sources
- routers, hosts reserve bandwidth  $b$  needed on downstream links towards H1



## RSVP: *receiver* reservation example 1 (more)

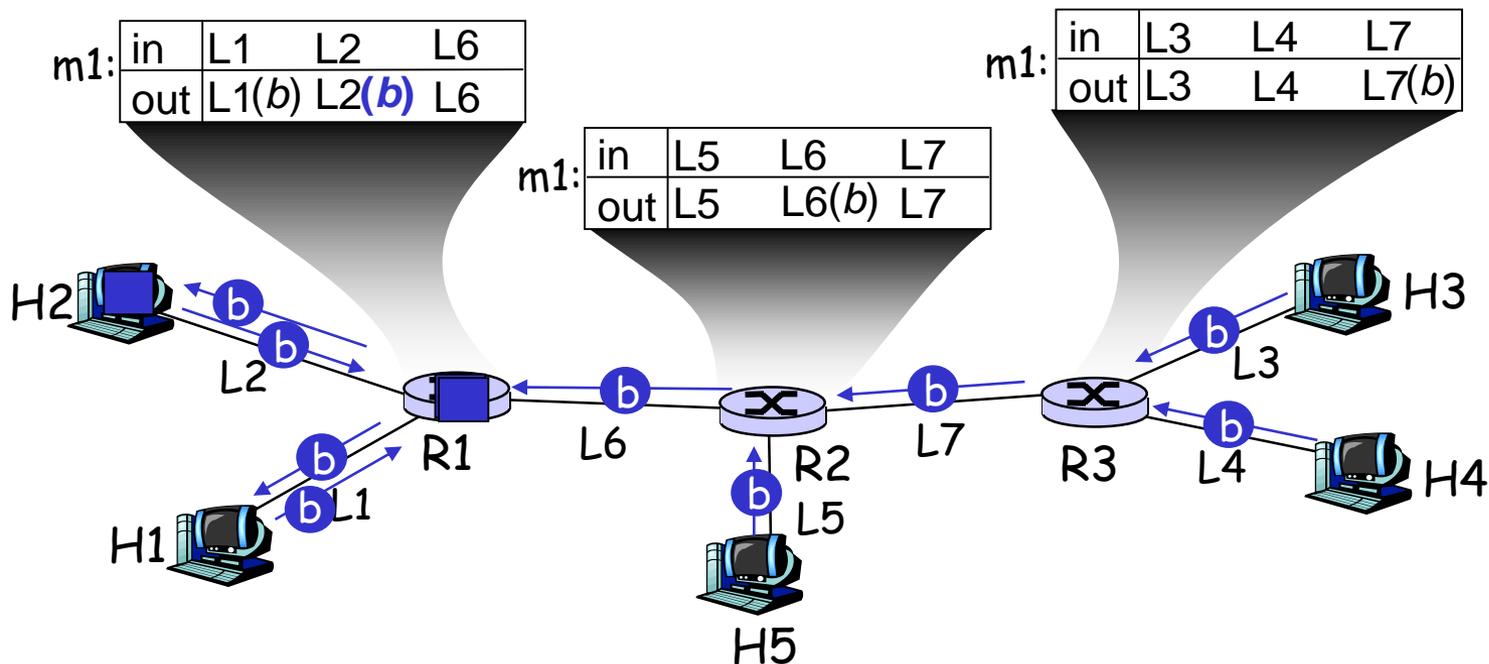
- next, H2 makes no-filter reservation for bandwidth  $b$
- H2 forwards to R1, R1 forwards to H1 and R2 (?)
- R2 takes no action, since  $b$  already reserved on L6



# RSVP: *receiver* reservation: issues

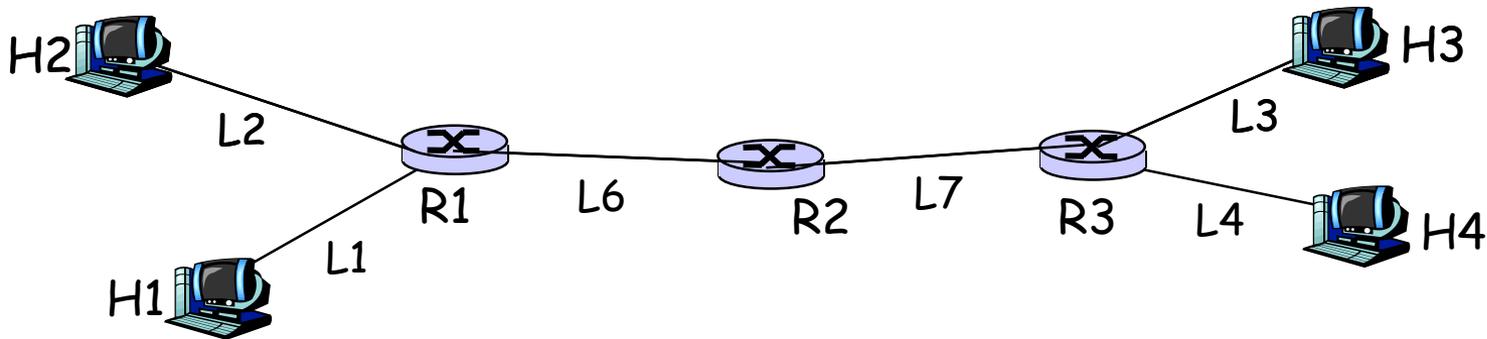
What if multiple senders (e.g., H3, H4, H5) over link (e.g., L6)?

- arbitrary interleaving of packets
- L6 flow policed by leaky bucket: if H3+H4+H5 sending rate exceeds  $b$ , packet loss will occur



# RSVP: example 2

- H1, H4 are only senders
  - send *path messages* as before, indicating filtered reservation
  - Routers store upstream senders for each upstream link
- H2 will want to receive from H4 (only)



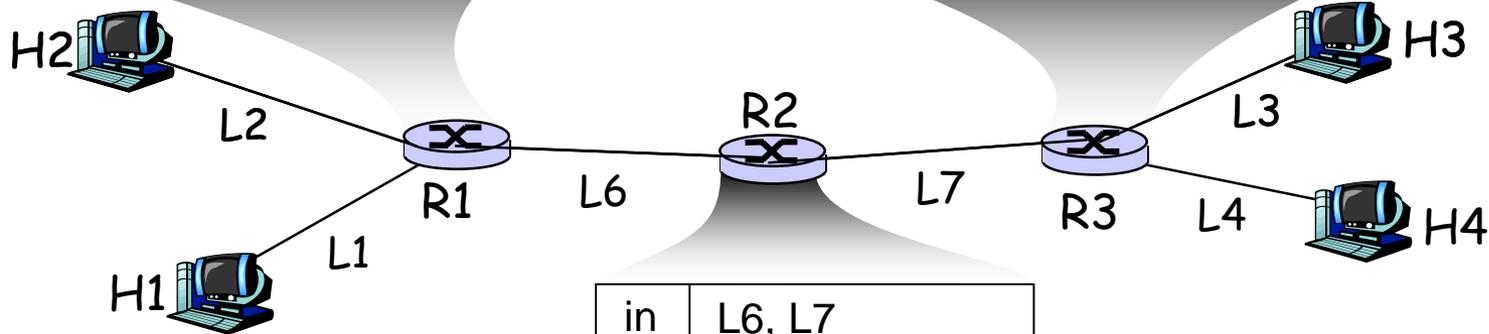
# RSVP: example 2

□ H1, H4 are only senders

○ send *path messages* as before, indicating filtered reservation

in	L1, L6
out	L2(H1-via-H1 ; H4-via-R2 ) L6(H1-via-H1 ) L1(H4-via-R2 )

in	L4, L7
out	L3(H4-via-H4 ; H1-via-R3 ) L4(H1-via-R2 ) L7(H4-via-H4 )



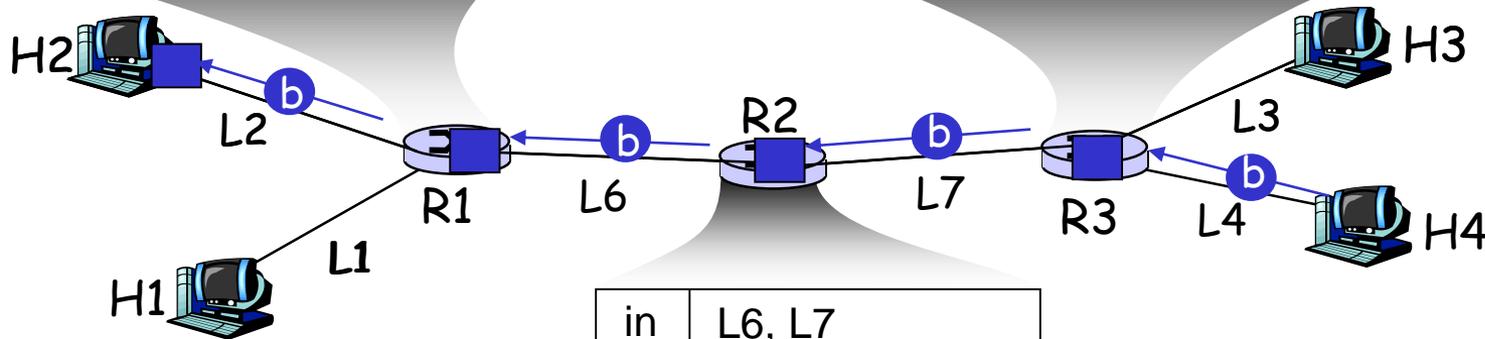
in	L6, L7
out	L6(H4-via-R3 ) L7(H1-via-R1 )

# RSVP: example 2

- receiver H2 sends reservation message for source H4 at bandwidth  $b$ 
  - propagated upstream towards H4, reserving  $b$

in	L1, L6
out	L2(H1-via-H1 ; H4-via-R2 ( <b>b</b> )) L6(H1-via-H1 ) L1(H4-via-R2 )

in	L4, L7
out	L3(H4-via-H4 ; H1-via-R2 ) L4(H1-via-R2 ) L7(H4-via-H4 ( <b>b</b> ))



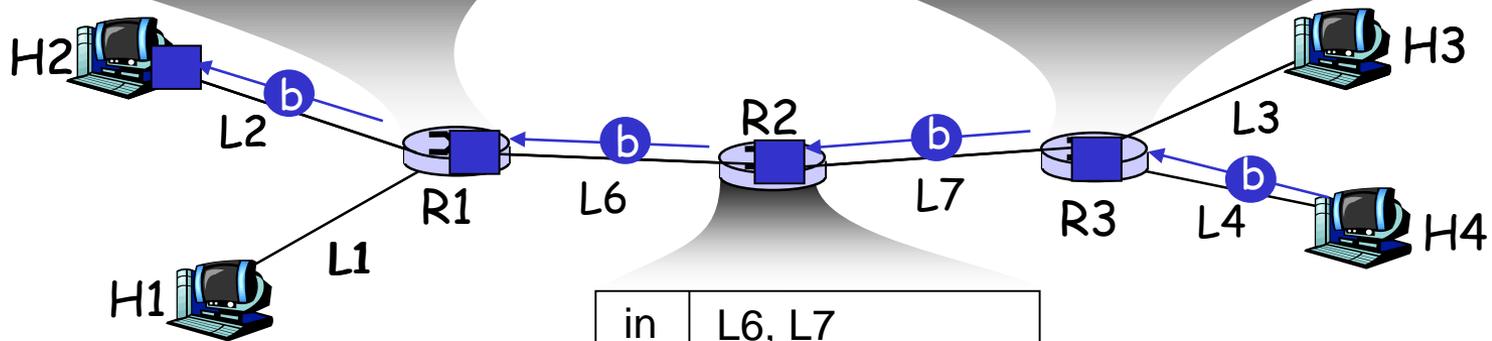
in	L6, L7
out	L6(H4-via-R3 ( <b>b</b> )) L7(H1-via-R1 )

# RSVP: *soft-state*

- senders periodically resend path msgs to refresh (maintain) state
- receivers periodically resend resv msgs to refresh (maintain) state
- path and resv msgs have TTL field, specifying refresh interval

in	L1, L6
out	L2(H1-via-H1 ; H4-via-R2 <b>(b)</b> ) L6(H1-via-H1 ) L1(H4-via-R2 )

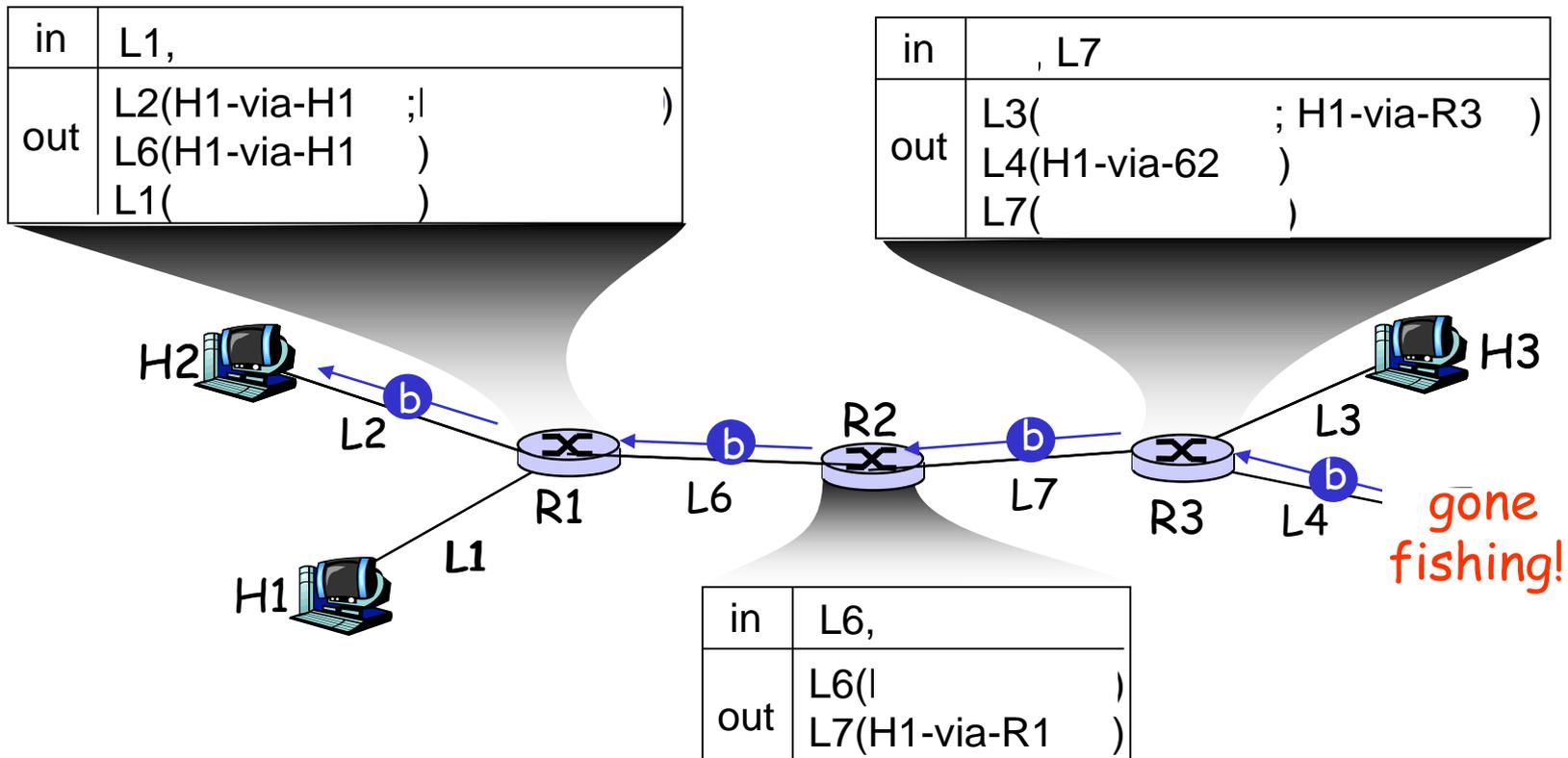
in	L4, L7
out	L3(H4-via-H4 ; H1-via-R3 ) L4(H1-via-62 ) L7(H4-via-H4 <b>(b)</b> )



in	L6, L7
out	L6(H4-via-R3 <b>(b)</b> ) L7(H1-via-R1 )

# RSVP: *soft-state*

- suppose H4 (sender) leaves without performing teardown
- eventually state in routers will timeout and disappear!



# The many uses of reservation/path refresh

- ❑ recover from an earlier lost refresh message
  - expected time until refresh received must be longer than timeout interval! (short timer interval desired)
- ❑ Handle receiver/sender that goes away without teardown
  - Sender/receiver state will timeout and disappear
- ❑ Reservation refreshes will cause new reservations to be made to a receiver from a sender who has joined since receivers last reservation refresh
  - E.g., in previous example, H1 is only receiver, H3 only sender. Path/reservation messages complete, data flows
  - H4 joins as sender, nothing happens until H3 refreshes reservation, causing R3 to forward reservation to H4, which allocates bandwidth

# RSVP: reflections

- ❑ multicast as a “first class” service
- ❑ receiver-oriented reservations
- ❑ use of soft-state

# Multimídia em Rede: Resumo

- ❑ aplicações multimídia e requisitos
- ❑ extraindo o máximo do serviço atual de melhor esforço
- ❑ mecanismos de escalonamento e policiamento
- ❑ próxima geração da Internet: Intserv, RSVP, Diffserv