

# Protocolos para Serviços Multimídia na Internet

Nelson Fonseca  
IC - UNICAMP

# Multimídia na Internet

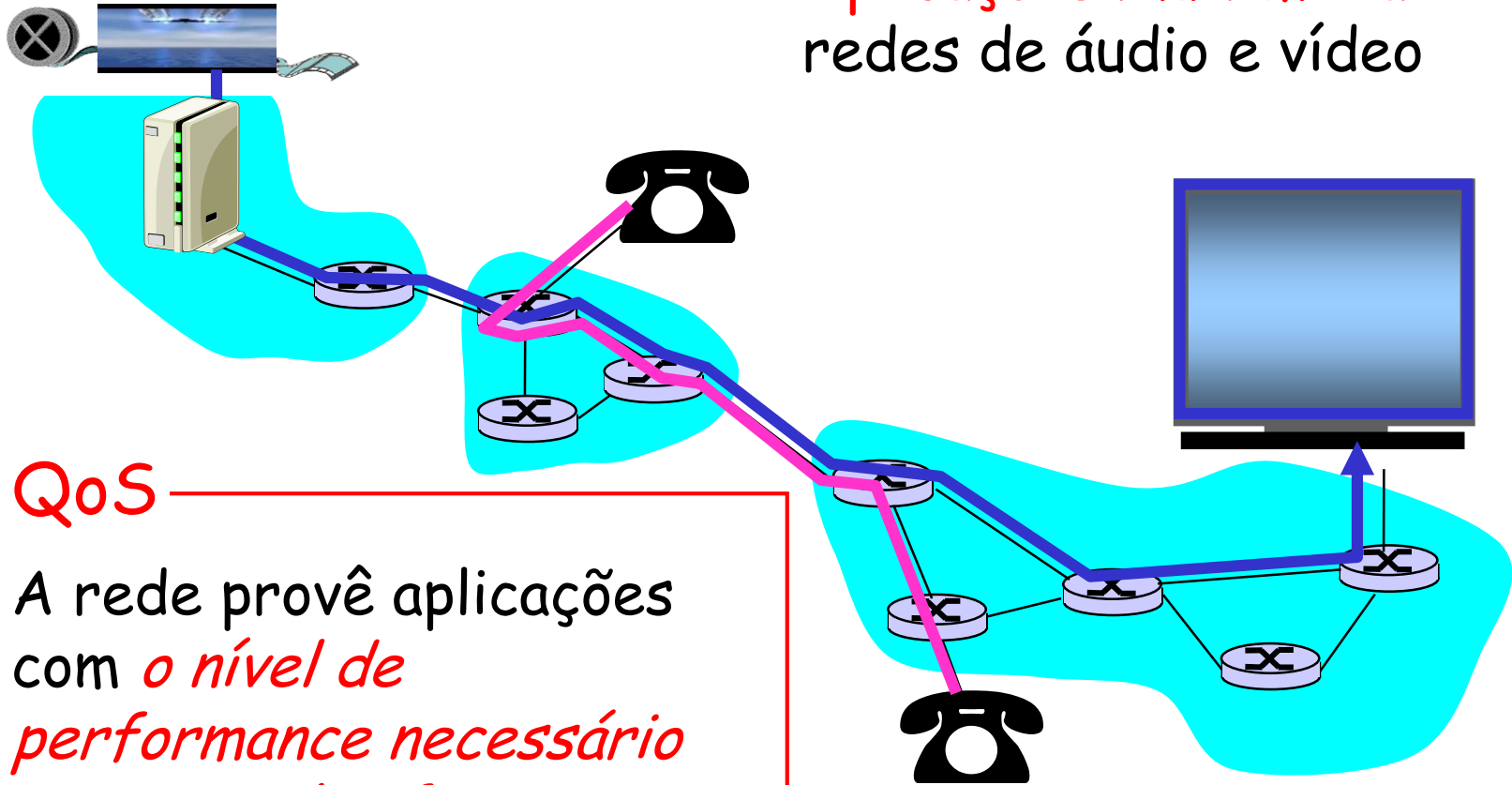
- ❑ Rede Integrada de Serviços
- ❑ Serviços multimídia - apenas um tipo de serviço
- ❑ Reutilização da infra-estrutura
- ❑ Integração com WWW
- ❑ Diversidade de acesso
- ❑ Integração com gravação
- ❑ Disponibilização de grande quantidade de conteúdo

# Aplicações

- Rádio na Internet
- Telefonia IP
- Vídeo sob demanda

# Multimídia, Qualidade de Serviço: O que é isso?

Aplicações Multimídia:  
redes de áudio e vídeo



## QoS

A rede provê aplicações com *o nível de performance necessário para que elas funcionem como esperado.*

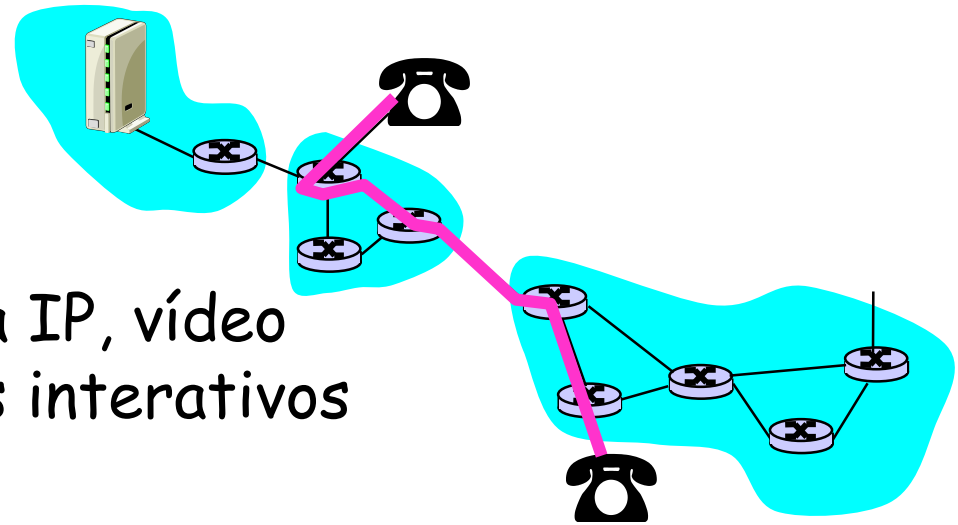
# Requisitos de Performance Multimídia

- **Requisitos:** entregar dados na "hora certa"
- **Multimídia interativa:** pequeno atraso fim-a-fim
  - ex., telefonia IP, teleconf., mundos virtuais, DIS
  - Atraso excessivo prejudica a interação humana
- **Multimídia com streams (não-interativa):**
  - dados devem chegar no tempo certo para uma visualização "suave"
  - Dados atrasados introduzem lacunas em áudio/vídeo renderizado
- confiabilidade: 100% de confiabilidade nem sempre é necessário

# Outros requisitos de streaming

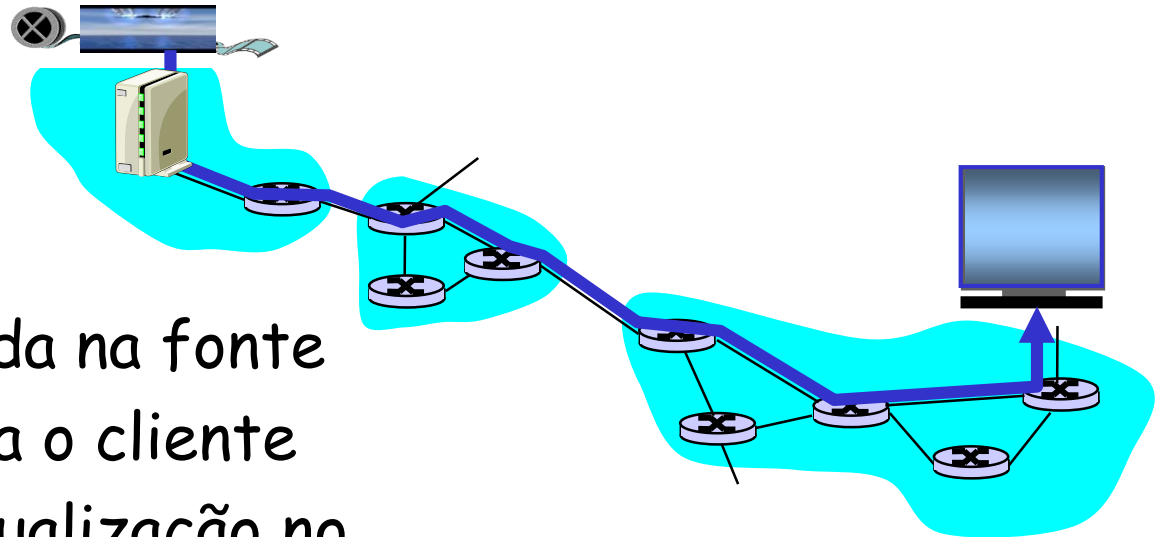
- **Recuperação da mídia no servidor**
  - Vídeo sob demanda (True VoD)
  - Quase vídeo sob demanda (Near VoD)
- **Eventos ao vivo** -> multicast
- **Controle remoto**
- **Editoração remota**

# Multimídia em Tempo Real, Interativa



- **aplicações:** telefonia IP, vídeo conferência, mundos interativos e distribuídos
- **Requisitos de atraso fim-a-fim:**
  - vídeo: < 150 msec aceitável
  - áudio: < 150 msec bom, < 400 msec OK
  - Inclui atrasos no nível da aplicação (empacotamento) e na rede
  - Atrasos maiores prejudicam a interatividade

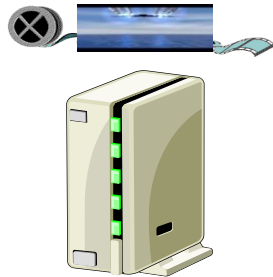
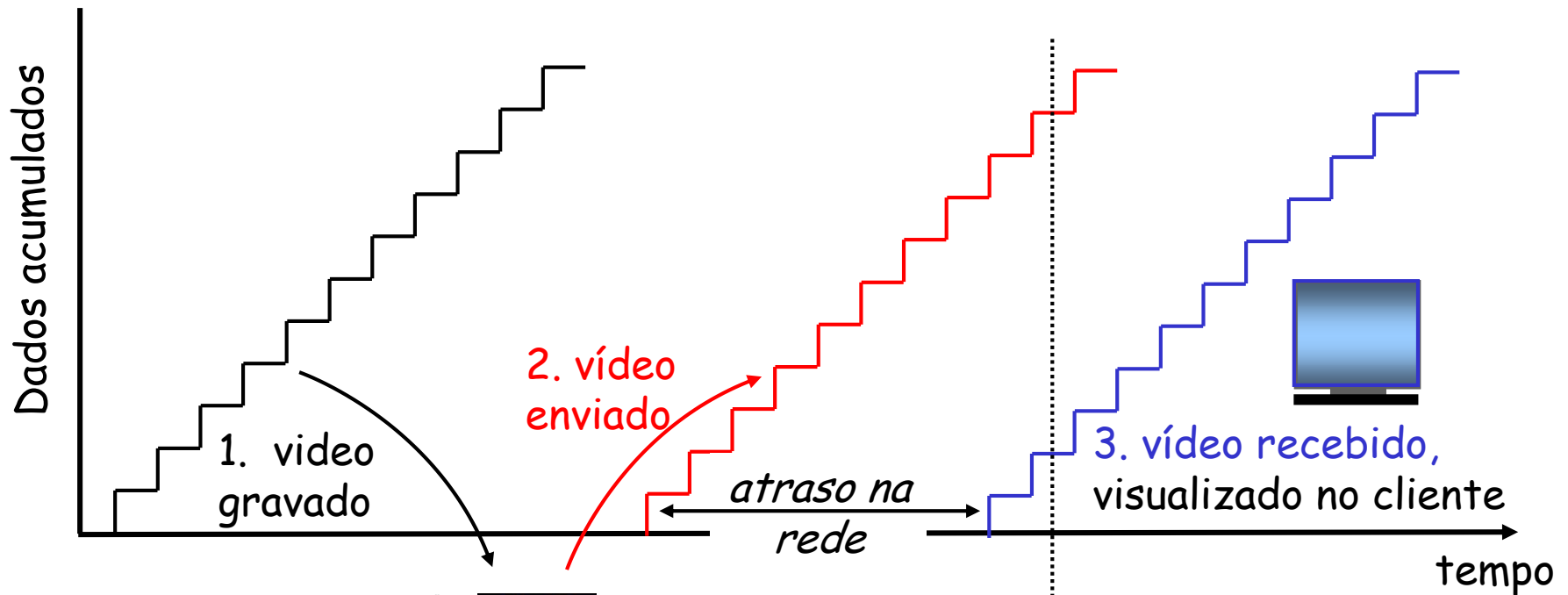
# Multimídia de Streams



- **Streaming:**
- mídia armazenada na fonte
- transmitida para o cliente
- streaming: a visualização no cliente inicia antes que todos os dados tenham chegado
- Restrição de tempo para dados que ainda serão transmitidos: deve chegar a tempo para visualização

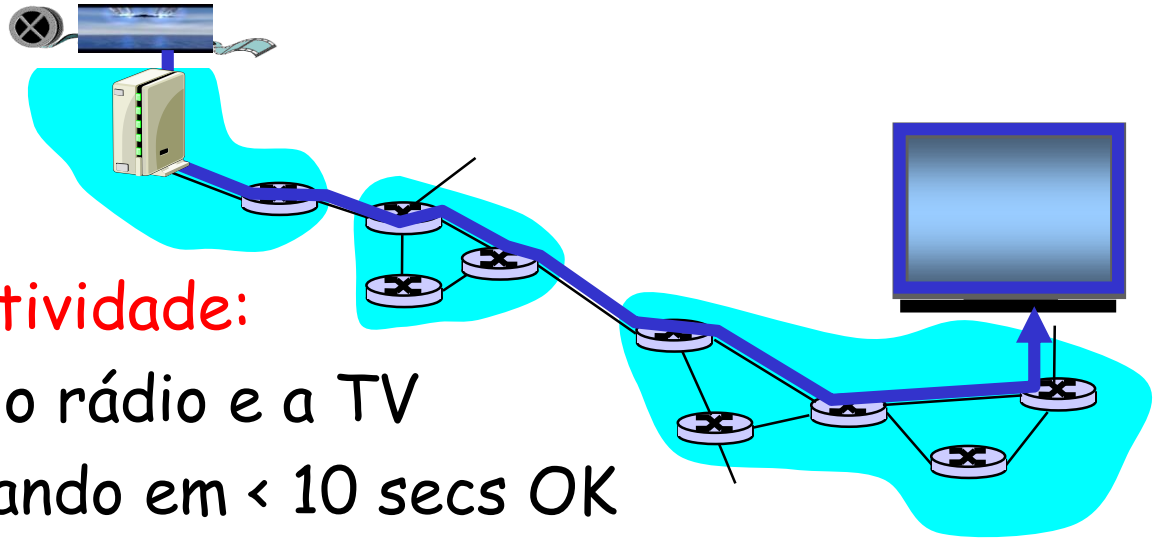


# Streaming: o que é isso?



streaming: nesse ponto, o cliente visualiza a parte inicial do vídeo, enquanto o servidor ainda está enviando parte do vídeo.

# Multimídia de Streams (mais)



- **Tipos de interatividade:**
- *nenhuma:* como o rádio e a TV
  - atrasos iniciando em < 10 secs OK
- *funcionalidade VCR:* o cliente pode congelar, voltar, avanço rápido
  - 1-2 sec até que o comando tenha efeito OK
  - restrição de tempo para dados que ainda serão transmitidos: em tempo para visualização

# Multimídia Sobre a Internet de Hoje

**TCP/UDP/IP: serviço "best-effort"**

□ *sem* garantias no atraso, perda



? ? ? ? ? ?  
Mas você disse que aplicações ?  
multimídia precisam de QoS e nível de ?  
performance para serem eficientes!  
? ? ?



As aplicações multimídia na Internet de hoje usam técnicas no nível de aplicação para aliviar (ao máximo possível) os efeitos do atraso e perda

# Multimídia na Internet com Streaming

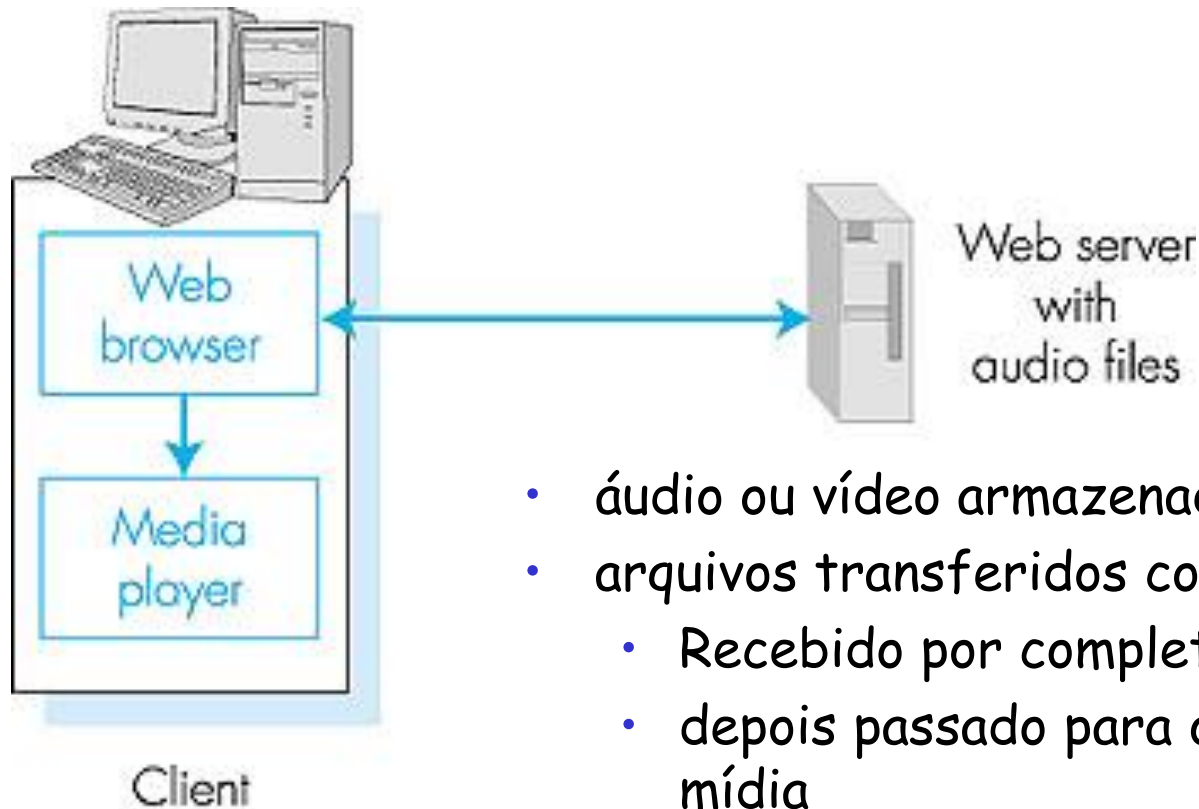


- Técnicas de streaming no nível da aplicação para obter o máximo do serviço "best effort":

- armazenamento (buffering) no cliente
- uso do UDP vs TCP
- multimídia com codificações de taxa múltipla

..... Vamos dar uma olhada em algumas .....

## Multimídia na Internet: abordagem mais simples

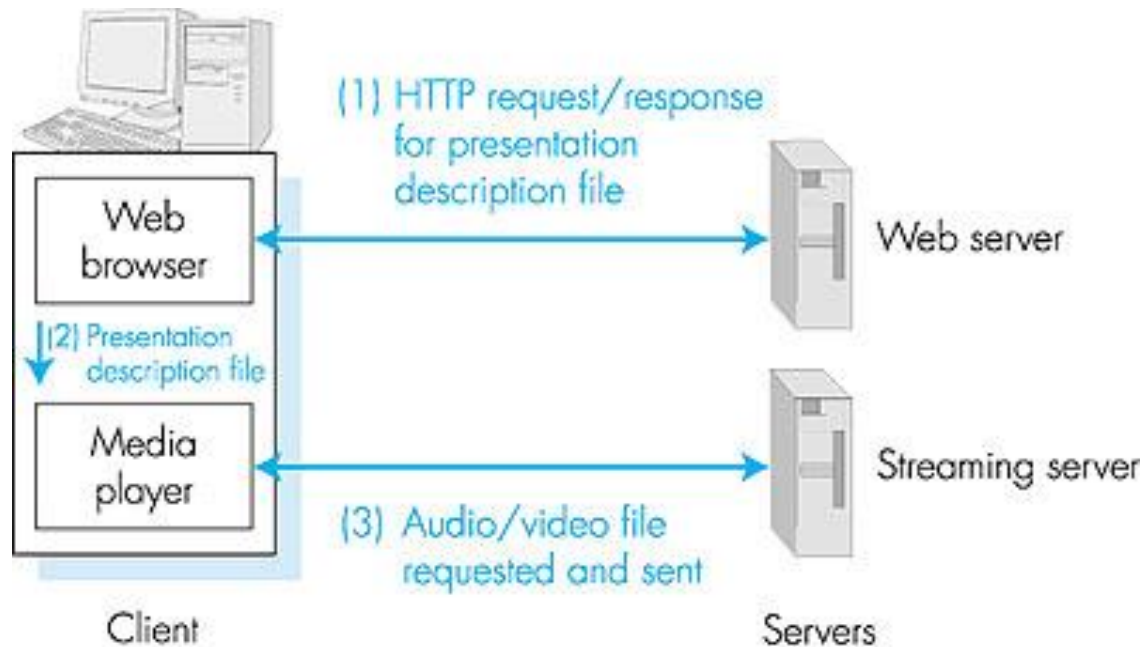


- áudio ou vídeo armazenados em arquivo
- arquivos transferidos como objetos HTTP
  - Recebido por completo no cliente
  - depois passado para o tocador de mídia

### áudio, vídeo não streamed:

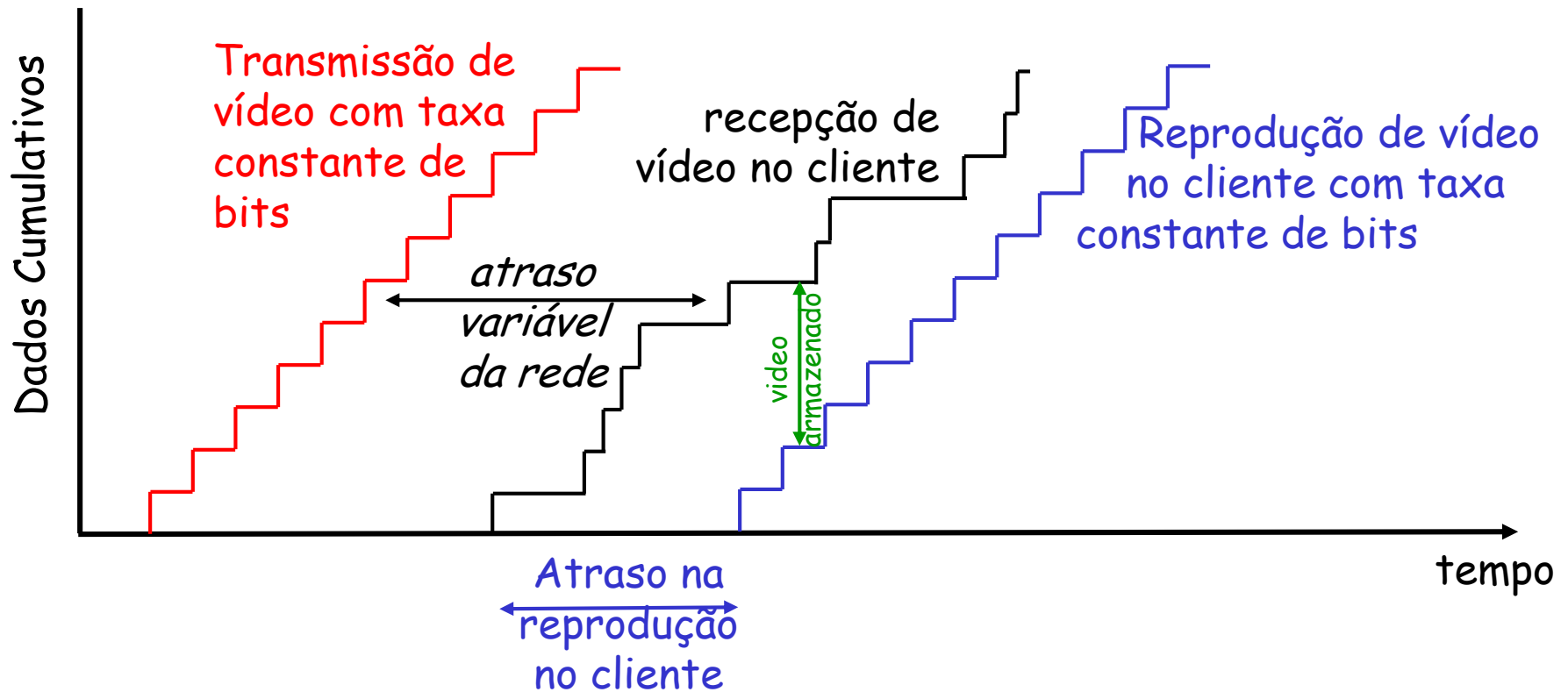
- sem, "pipelining," longos atrasos antes de tocar!

## Multimídia na Internet: abordagem de streaming



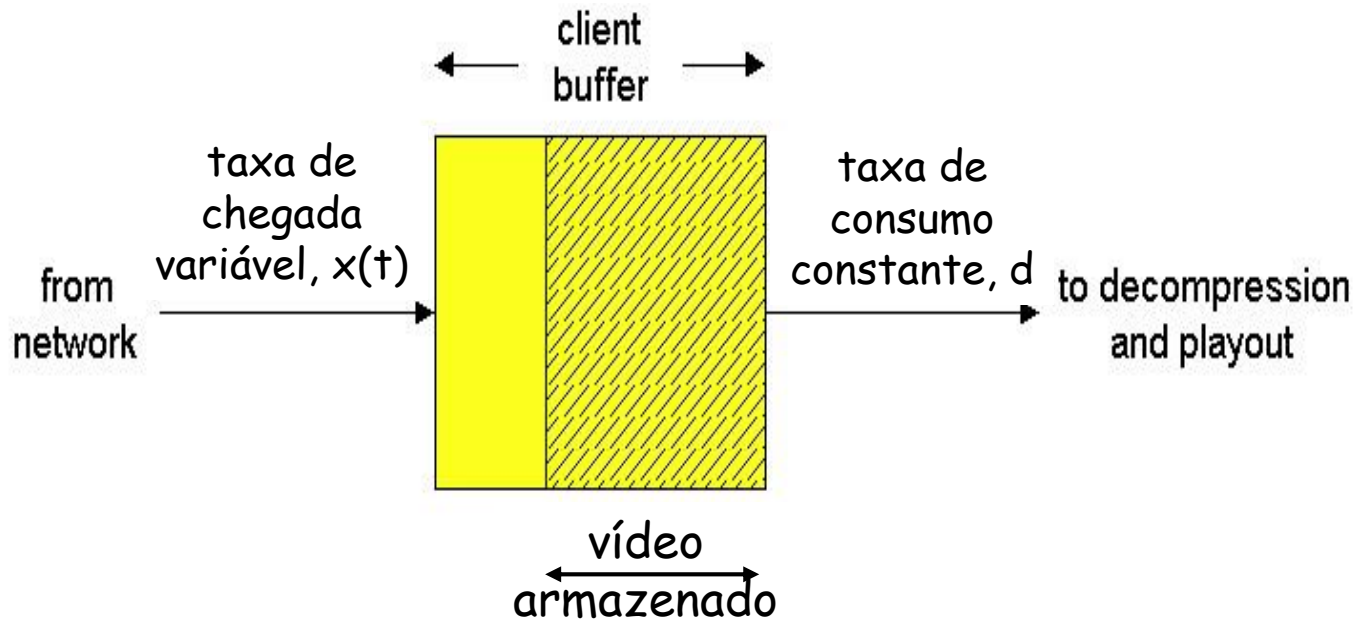
- browser GETs **meta-arquivo**
- browser executa tocador, passando o meta-arquivo
- tocador contata o servidor
- servidor fornece o **stream** de áudio/vídeo para o tocador

# Multimídia com Streaming : Armazenamento no Cliente



- Armazenamento no cliente, atraso na reprodução, para compensar o atraso adicional da rede, jitter

# Multimídia com Streaming : Armazenamento no Cliente



- Armazenamento no cliente, atraso na reprodução, para compensar o atraso adicional da rede, jitter



# Multimídia Interativa: Internet Phone

Introduzindo o Internet Phone na forma de um exemplo (nota: ainda não existe um "padrão"):

- áudio do locutor: alternando rajadas de fala, períodos de silêncio.
- pkts são gerados apenas durante as rajadas de fala
  - Ex., pedaços de 20 msec a 8 Kbytes/sec: 160 bytes de dados
- um cabeçalho da camada de aplicação é adicionado a cada pedaço.
- pedaço+cabeçalho são encapsulados dentro de um segmento UDP.
- a aplicação envia segmentos UDP pelos sockets a cada 20 msec durante a rajada de fala.

## Internet Phone: Perda e Atraso de Pacotes

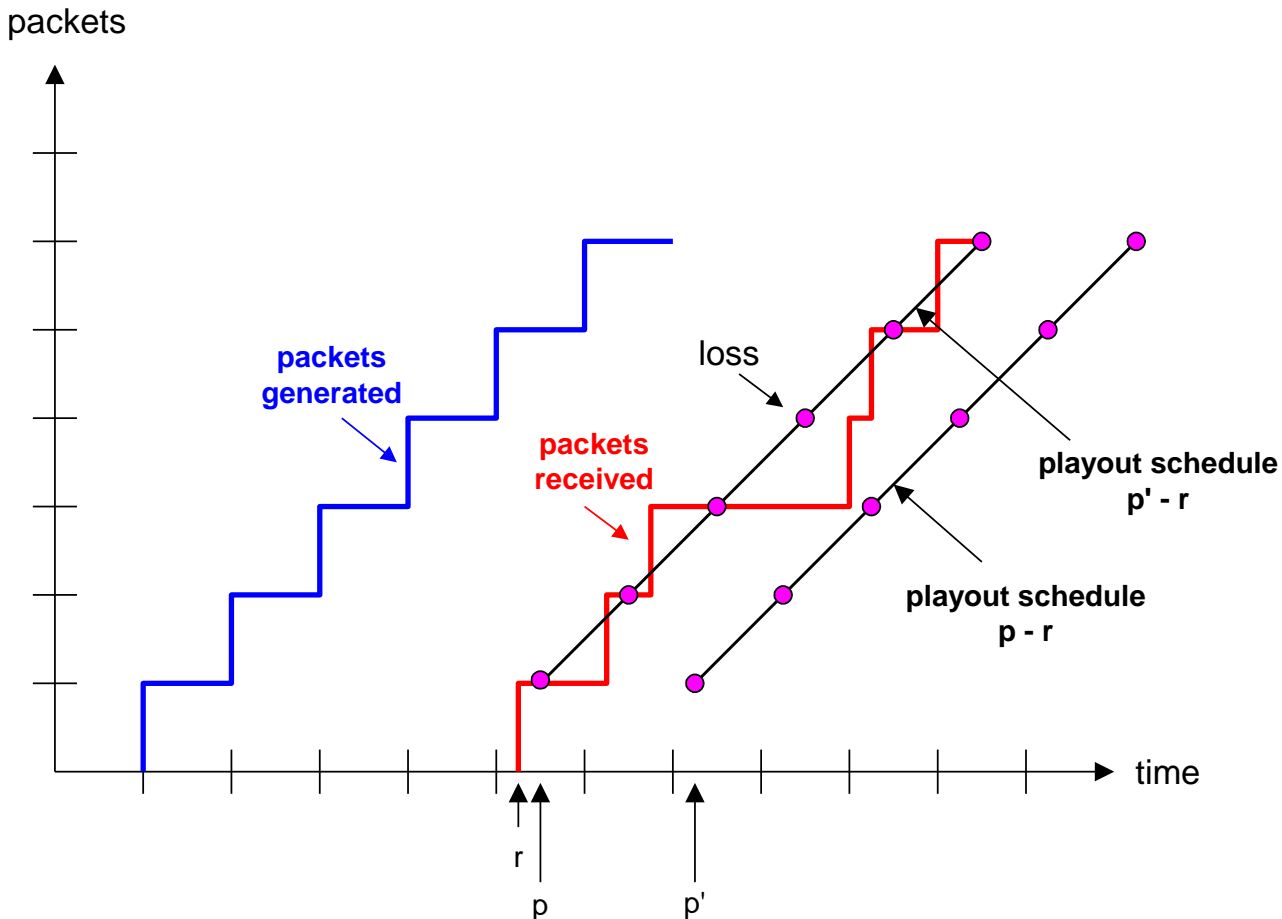
- **perda na rede:** perda de datagramas IP devido ao congestionamento na rede (transbordo de buffer no roteador)
- **atraso na rede:** datagramas IP chegam atrasados para reprodução no receptor
  - atrasos: processamento, enfileiramento na rede; atraso no end-system (transmissor, receptor)
  - atraso máximo tolerável típico: 400 ms
- tolerância de atraso: depende da codificação de voz, perdas consecutivas, taxa de perda de pacotes entre 1% e 10% podem ser toleradas.

## Internet Phone: Atraso Fixo da Reprodução

- O receptor tenta reproduzir cada pedaço exatamente  $q$  msec depois que o pedaço foi gerado.
  - cada pedaço tem um "time stamp"  $t$ : reproduza o pedaço em  $t+q$ .
  - cada pedaço chega após  $t+q$ : dados chegaram tarde demais para reprodução, "perda" de dados
- Definição do  $q$ :
  - $q$  grande: menor perda de pacotes
  - $q$  pequeno: melhor experiência interativa

# Atraso Fixo da Reprodução

- O Transmissor gera pacotes a cada 20 msec durante a fala.
- Primeiro pacote é recebido no tempo  $r$
- Primeira reprodução é escalonada: inicia em  $p$
- Segunda reprodução é escalonada: inicia em  $p'$



# Atraso Adaptativo da Reprodução, I

- **Objetivo:** minimizar o atraso da reprodução, mantendo a taxa de perda tardia baixa
- **Abordagem:** ajustes do atraso adaptativo da reprodução:
  - Estima o atraso na rede, ajusta o atraso da reprodução no início de cada rajada de fala.
  - Períodos de silêncio são comprimidos e alongados.
  - Os pedaços continuam a ser reproduzidos a cada 20 msec durante a fala.

$t_i$  = timestamp of the  $i$ th packet

$r_i$  = the time packet  $i$  is received by receiver

$p_i$  = the time packet  $i$  is played at receiver

$r_i - t_i$  = network delay for  $i$ th packet

$d_i$  = estimate of average network delay after receiving  $i$ th packet

Estimativa dinâmica do atraso médio no receptor:

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

onde  $u$  é uma constante fixa (ex.,  $u = 0.01$ ).

# Atraso Adaptativo da Reprodução, II

Também útil para estimar o desvio padrão do atraso,  $v_i$ :

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

$d_i$ ,  $v_i$  calculado para todo pkt, mas usado apenas no início da rajada

Para o primeiro pacote na rajada de fala, o momento da reprodução é:

$$p_i = t_i + d_i + Kv_i$$

Atraso da reprodução para o 1 pkt na rajada de fala:

$$q_i = p_i - t_i$$

Para o pacote  $j$  em alguma rajada, reproduza o pacote em

$$p_j = t_j + q_j$$

# Reprodução adaptiva, III

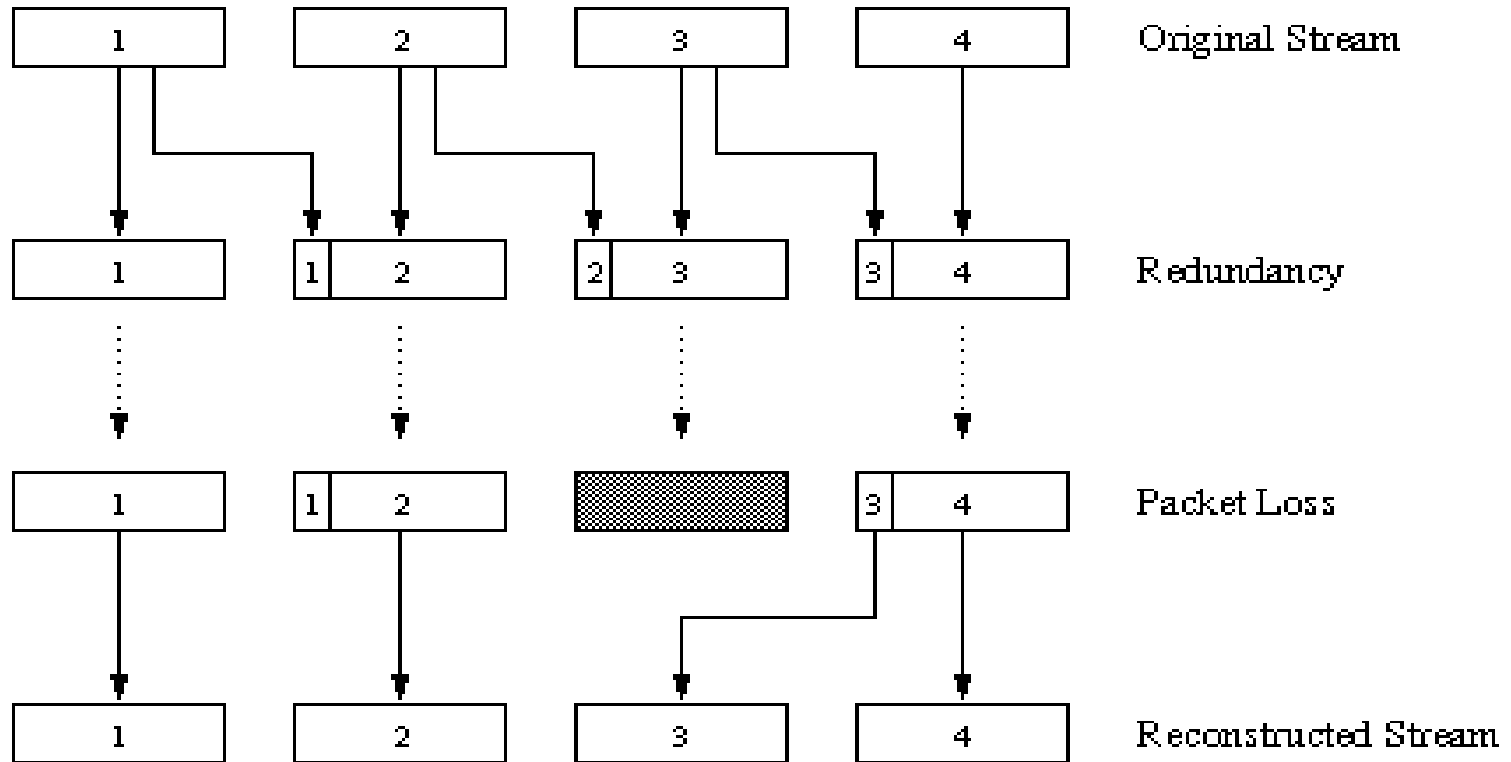
- P:** Como o receptor determina se um pacote é o primeiro de uma rajada de fala?
- Se não há perda, o receptor olha para "timestamps" sucessivos.
    - diferença de pacotes sucessivos  $> 20$  msec  $\rightarrow$  começou uma rajada.
  - Com a possibilidade de perda, o receptor deve olhar tanto para o "timestamp" quanto para o número de seqüência.
    - diferença de pacotes sucessivos  $> 20$  msec e número de seqüência sem intervalos, começou uma rajada.

# Recuperação Após Perda de Pacotes

- perda: pkt nunca chega ou chega tarde demais
- Restrição de tempo-real: pequeno (nenhum) tempo para retransmissão!
  - O que fazer?
- **Forward Error Correction (FEC)**: adiciona bits de correção de erros (relembrar paridade de 2 dimensões)
  - ex.;; adicionar pedaço redundante feito do OR exclusivo de  $n$  pedaços; a redundância é  $1/n$ ; pode reconstruir até um pedaço perdido
- **Interleaving**: espalha a perda igualmente entre os dados recebidos para minimizar o impacto da perda

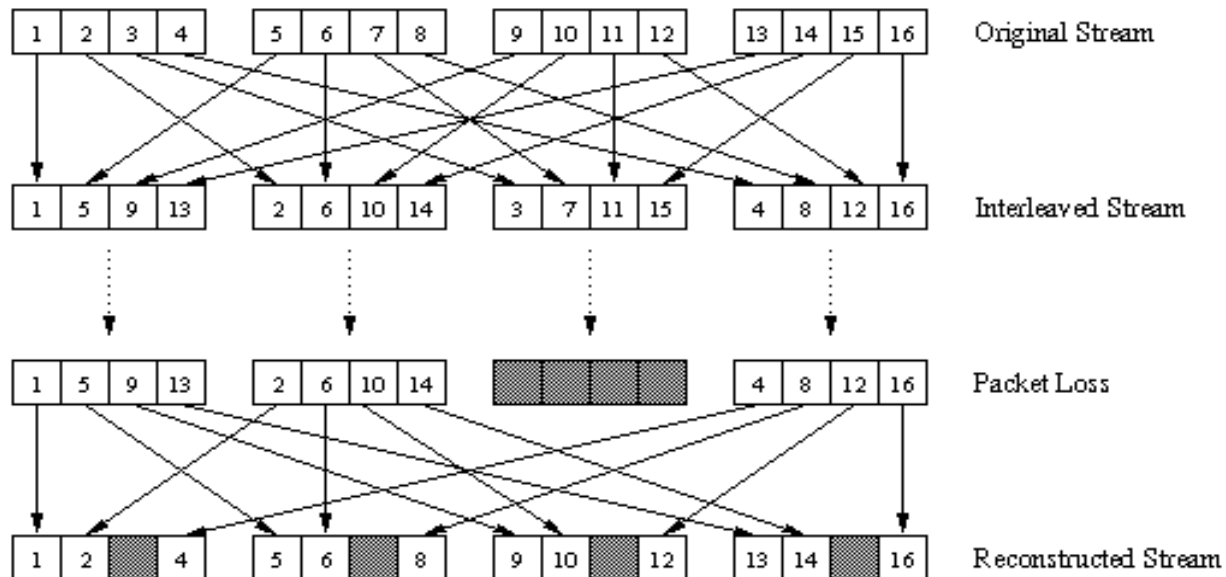


# Piggybacking Streams de Baixa Qualidade



# Interleaving


- Não apresenta redundância, mas pode causar um atraso na reprodução maior que o dos requisitos de tempo-real
- Divide 20 msec de dados de áudio em unidades menores de 5 msec cada e os rearranja
- Havendo a perda, haverá um conjunto de pedaços parcialmente completos

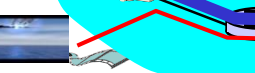


# Sumario: Multimídia na Internet: truques

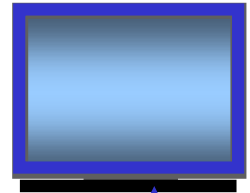
- use **UDP** para evitar o controle de congestionamento  
TCP (atrasos) para tráfego sensível ao tempo
- Lado do cliente **atraso adaptativo da reprodução**:  
para compensar o atraso
- Lado do servidor **igual a banda passante do stream** com a banda passante do caminho cliente-servidor
  - Escolhe entre taxas de streams pré-codificadas
  - Servidor faz codificação de taxa dinamicamente
- Recuperação de erros (sobre o UDP)
  - FEC
  - retransmissões, subordinadas ao tempo
  - Erros mascarados: repete os dados próximos

# Multimídia com Streaming : taxa(s) no cliente

codificado para 1.5 Mbps 



codificado para 28.8 Kbps 



**P:** como tratar clientes com diferentes taxas de recebimento?

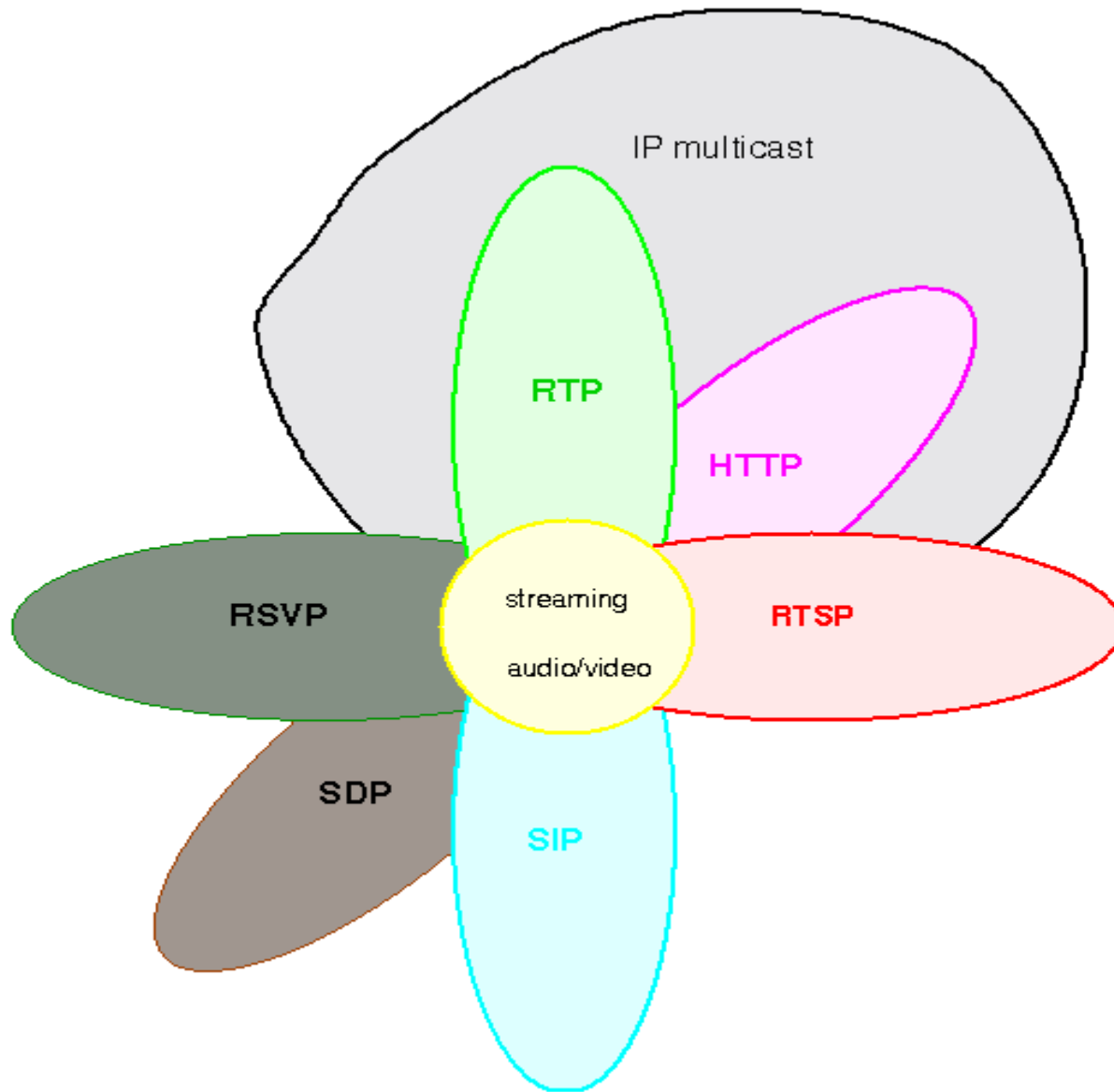
- 28.8 Kbps linha discada
- 100Mbps Ethernet

**R:** o servidor armazena e transmite múltiplas copias do vídeo codificado em taxa diferentes

# Protocolos - Aplicações multimídia

- Reserva de recursos: RSVP, YESSIR
- Transporte de mídia: RTP
- Controle de stream: RTSP
- Descrição de conteúdo: SDP, SMIL, RTSL

# Protocolos - Aplicações Multimídia

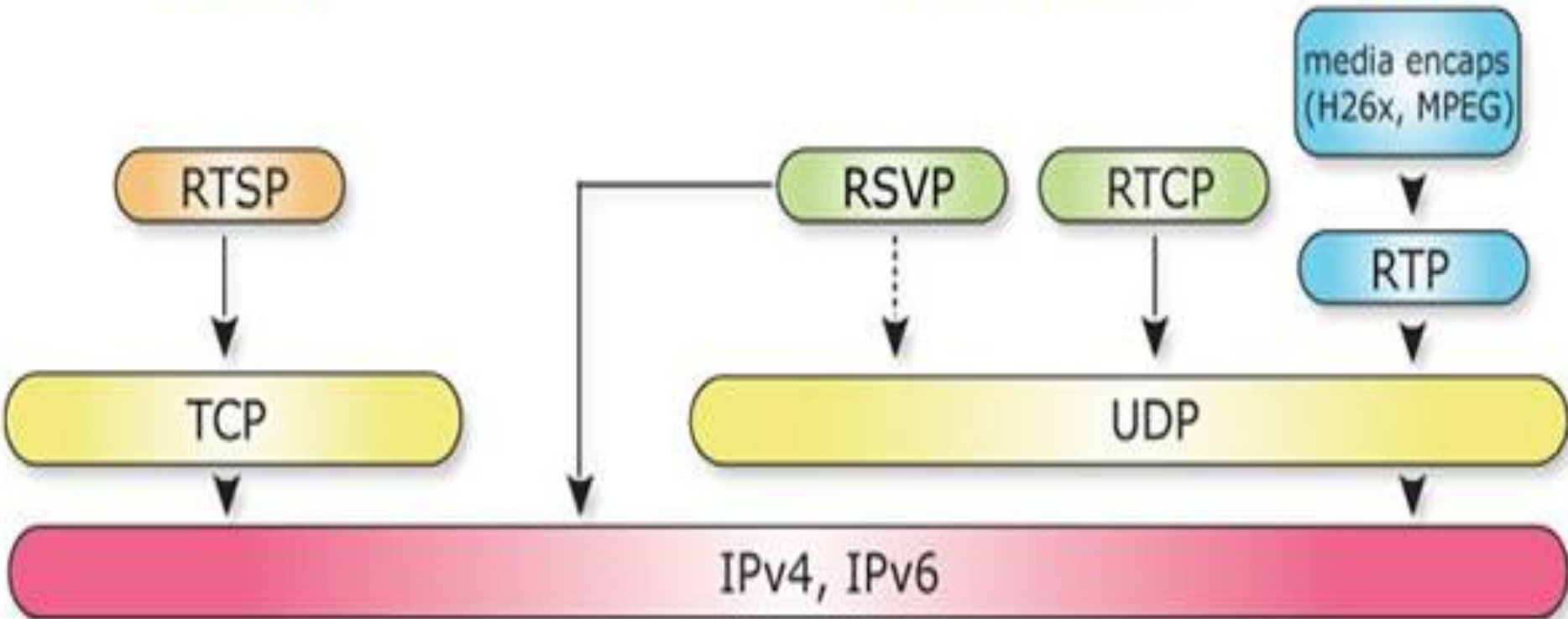


# Protocolos - Aplicações Multimídia

signaling

quality of service

media transport



## RTP - Características

- *Lightweight* - implementação e especificação
- *Flexível* - provê mecanismos e não algoritmos
- *Pode ser usado por vários protocolos* - UDP, IPX, ...
- *Escalável* - unicast e multicast  $O(10^7)$
- Separação de dados e controle
- Suporte a criptografia, possibilidade de autenticação



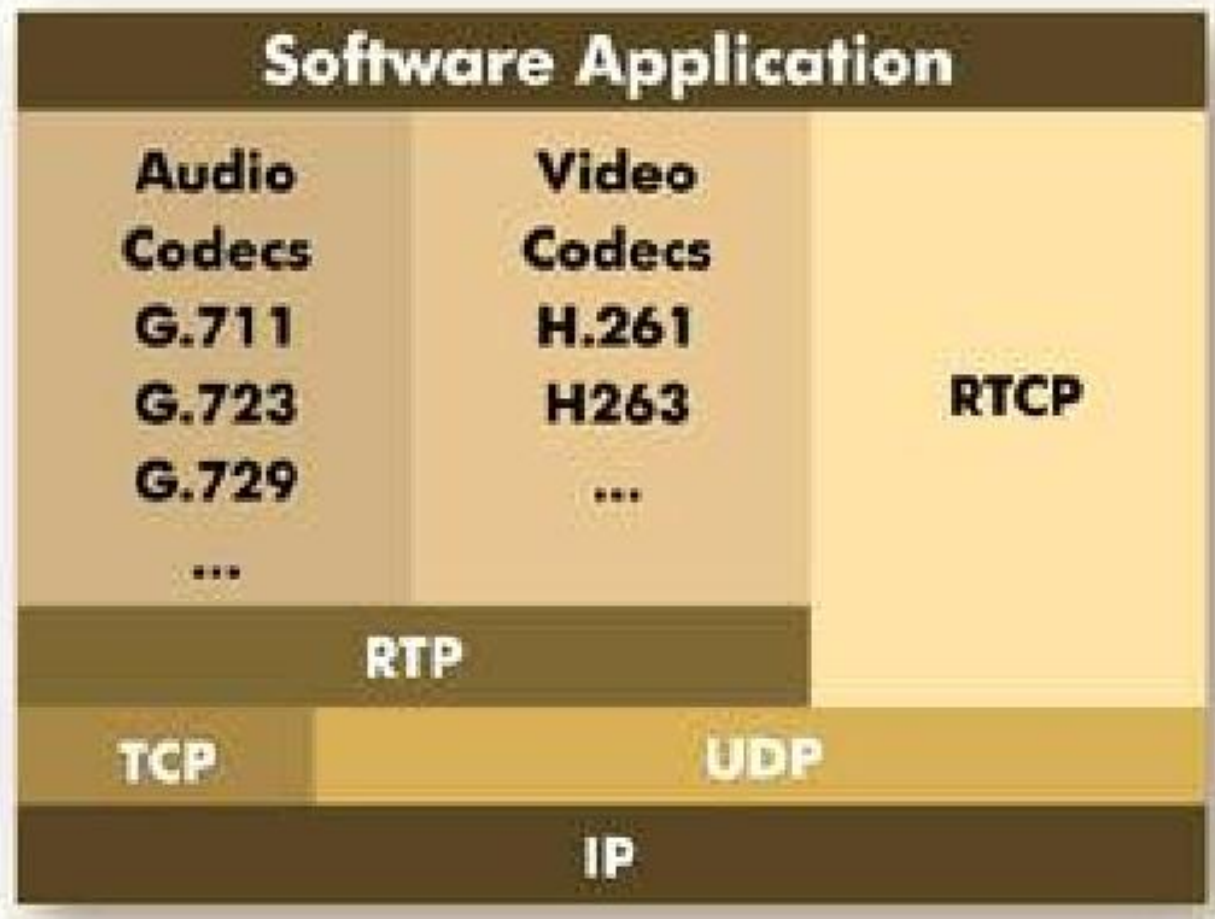
## RTP - Funcionalidades

- Segmentação e montagem feitas pelo UDP
- Resequenciamento (se necessário)
- Detecção de perda para estimação da qualidade
- Sincronização *intra-media*
- Sincronização *inter-media*
- *Feedback* sobre Qualidade de Serviço
- Adaptação da taxa
- Identificação da fonte

# RTP

- Tipicamente UDP, sem porta fixa
- Tipicamente uma *media* por porta

# RTP



# Real-Time Protocol (RTP)

- Formato do pacote padrão da Internet para aplicações de tempo-real
- tipicamente executa sobre o UDP



RTP Header

- **Tipo de Payload:** 7 bits, provendo 128 tipos diferentes de codificação; ex. PCM, MPEG2 vídeo, etc.
- **Número de Sequencia:** 16 bits; usado para detectar a perda de pacotes

# Real-Time Protocol (RTP)



RTP Header

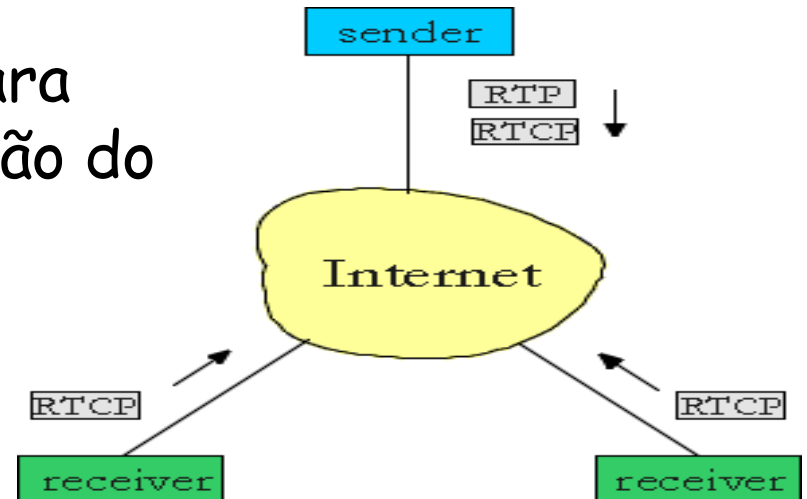
- **Timestamp:** 32 bits
  - contém o instante da amostragem do primeiro byte de áudio/vídeo no payload do pacote
- **Synchronization Source identifier (SSRC):** 32 bits
  - id da fonte do stream; definido aleatoriamente pela fonte

# Implementações RTP

<i>Ferramenta</i>	<i>Quem</i>	<i>Media</i>	<i>RSVP</i>
NeVoT	GMD Fokus	Áudio	Sim
vic	LBNL	Vídeo	Não
vat	LBNL	Áudio	Não
rat	UCL	Áudio	Não
Rendezvous	INRIA	A/V	Não
NetMeeting	Microsoft	A/V	Não
IP/TV	Cisco	A/V	Não
RM G2	Real	A/V	Não

# RTP Control Protocol (RTCP)

- parcela de controle (contrário de dados) do RTP
- define o protocolo/controla os dados trocados entre fonte, destinos
- três tipos de boletins (reports):
  - recepção no receptor, transmissor, e descrição da fonte
  - contém estatísticas como número de pacotes enviados, número de pacotes perdidos, jitter entre chegadas
- usado para diagnosticar, e para modificar a taxa de transmissão do transmissor



## Balanceamento de Banda Passante do RTCP

- Se cada receptor envia (multicasts) pacotes RTCP para todos os receptores, resulta em alta carga
- RTCP:
  - ajusta os intervalos entre boletins baseado no número de receptores
  - regra prática: limitar a banda passante do RTCP em 5% da banda passante da sessão, dividida entre boletins do transmissor (25%), boletins do receptor (75%)



## Controle do Usuário em Multimídia com streaming

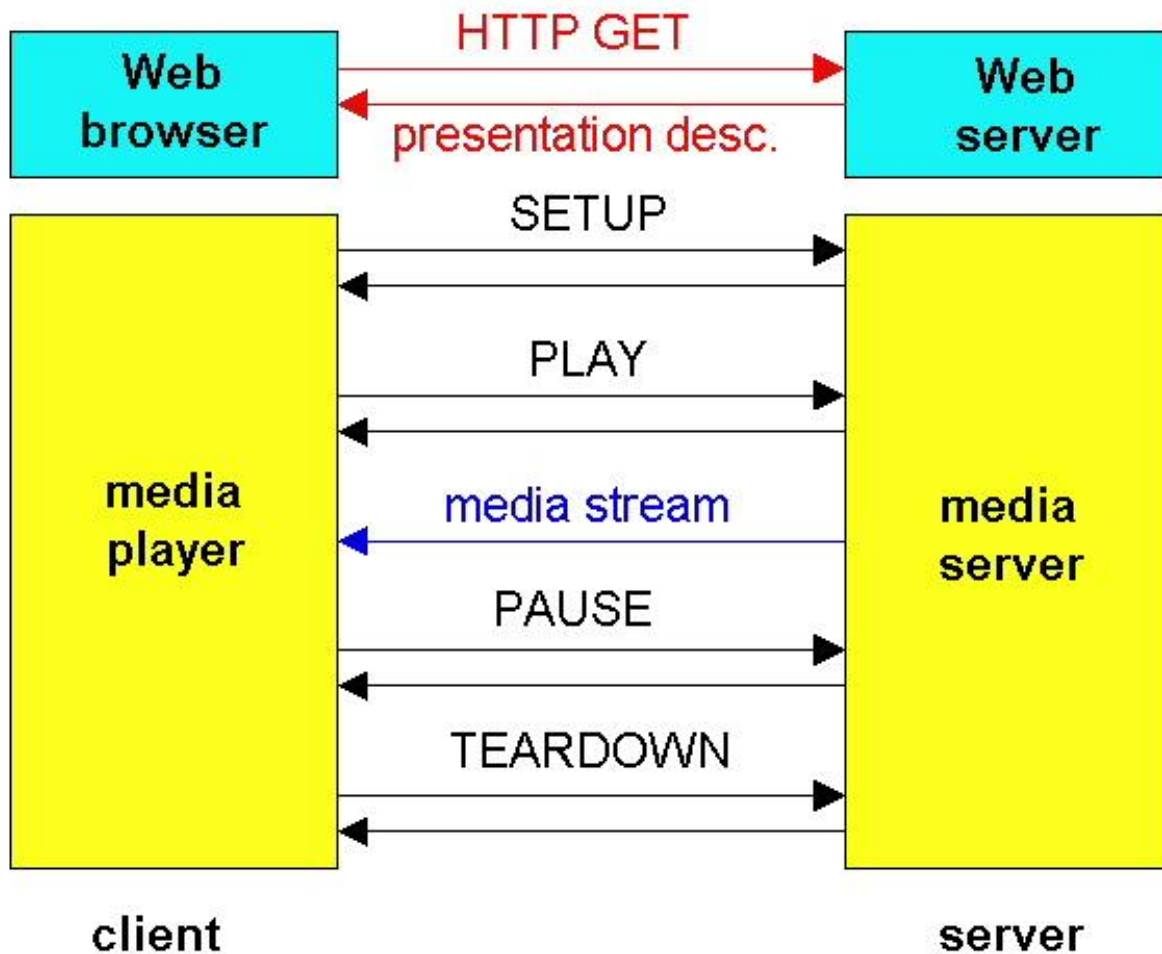
### Real Time Streaming Protocol (RTSP): RFC 2326

- controle do usuário: voltar, avanço rápido, congelar, continuar, etc...
- protocolo "out-of-band":
  - uma porta (544) para msgs de controle
  - uma porta para o stream de mídia
- TCP ou UDP para controlar a conexão de msg

### Cenário:

- o meta-arquivo é comunicado para o browser web
- browser inicia o tocador de mídia
- o tocador inicializa uma conexão de controle e outra de dados RTSP com o servidor

# Operação do RTSP



# RTSP - Características

- Sincronização de diversos servidores
- Balanceamento de carga
- Controle de dispositivos
- Caching

# Funcionalidades do RTSP

- **Recuperação de mídia contínua**
  - Recupera descritor da apresentação
  - Unicast e Multicast
  - Independente dos formatos de arquivos
- **Habilidade de adicionar mídia**
- **Controle agregado ou por fluxo** -> Um comando para vários fluxos

# RTSP: Integração com a Web

- Página da web com "guia de programas"
- Contém ponteiro para a descrição do conteúdo (por exemplo em SMIL)

```
<session>
  <group>
    <track src="rtsp://audio.mtv.com/movie">
    <track src="rtsp://video.mtv.com/movie">
  </group>
</session>
```

- RTSP configura e controla a entrega
- RSVP reserva os recursos
- RTP entrega os dados

# Exemplo de Troca 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

# 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>
```

# RTSP e HTTP

- **Mesmo formato:** texto, cabeçalho MIME
- **Pedido/resposta** -> Pedido+cabeçalho+corpo
- Formato da URL
- Negociação do conteúdo
- Mecanismos de segurança
- Código de status



# RTSP e HTTP

- Necessário conhecimento do estado do servidor
- Métodos diferentes
- Sinalização out-of-band

# Métodos RTSP

- **OPTIONS** Recebe os métodos disponíveis
- **SETUP** Estabelece o transporte
- **ANNOUNCE** Muda a descrição de um objeto de mídia
- **DESCRIBE** Recebe a descrição (em baixo nível) de um objeto de mídia
- **PLAY** Inicia a reprodução ou reposição
- **RECORD** Inicia a gravação
- **REDIRECT** Redireciona cliente a um novo servidor
- **PAUSE** Interrompe a entrega, mas mantém o estado
- **SET\_PARAMETER** Controla a codificação ou o dispositivo
- **TEARDOWN** Remove o estado

# Cabeçalhos de requisição

- **Accept** formatos de descrição da mídia
- **Accept-encoding** Codificação do formato da mídia
- **Accept-language** Linguagem humana
- **Authorization** Autenticação básica e "digest"
- **Bandwidth** Largura de banda disponível no cliente
- **Conference** Identificador da conferência
- **From** Nome do requisitante
- **If-Modified-Since** Recuperação condicional
- **Range** Intervalo de tempo que será reproduzido
- **Referer** Como nós chegamos aqui?
- **Scale**  $(\text{Tempo de reprodução}) / (\text{Tempo real})$
- **Speed** Acelerar a entrega
- **User-Agent** O programa utilizado no cliente

# Cabeçalhos de resposta

- **Location** Redirecionamento
- **Proxy-Authenticate** Autenticar em um proxy
- **Public** Métodos suportados
- **Retry-After** Ocupado; Volte mais tarde
- **Server** Software do servidor
- **Vary** Rótulo de cache
- **WWW-Authenticate** Requisita autorização