Aplicações Multimídia e Protocolos de Streaming

Profa. Débora Christina Muchaluat Saade

debora@midiacom.uff.br

Aplicações Multimídia

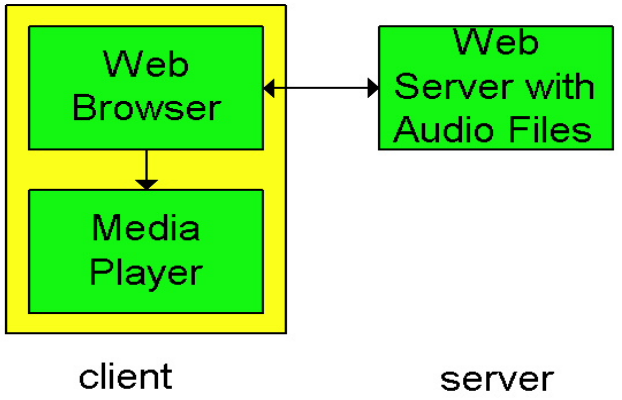
- **Classificação das aplicações multimídia:**
  - ***Transmissão de mídia contínua armazenada***
  - ***Transmissão de mídia contínua ao vivo***
  - ***Transmissão de mídia contínua interativa***

Aúdio e Vídeo Armazenados

- **Mídia Contínua armazenada**
  - *Arquivos de Aúdio e de Vídeo são armazenados em servidores*
  - *Usuários solicitam os arquivos por demanda.*
  - *Aúdio/vídeo são apresentados, digamos, 10s após o pedido.*
  - *Controle da apresentação é permitido.*
- **Executor da mídia (player)**
  - *remove jitter (variação do retardo)*
  - *Decodifica (descomprime) a mídia*
  - *Realiza correção de erros*
  - *Oferece interface gráfica para controle da apresentação*
- **Plug-ins podem ser usados para embutir o executor no browser web**
  - *3 abordagens para a implementação*

Acesso a Mídia Contínua a partir de servidores Web (1a. abordagem)

- **browser cliente solicita o arquivo com uma mensagem HTTP**
- **Servidor Web envia o arquivo na mensagem HTTP de resposta**
- **O cabeçalho "content-type" indica uma codificação apropriada para aúdio e vídeo**
- **browser dispara o executor da mídia e passa o arquivo para ele**
- **executor apresenta o arquivo**

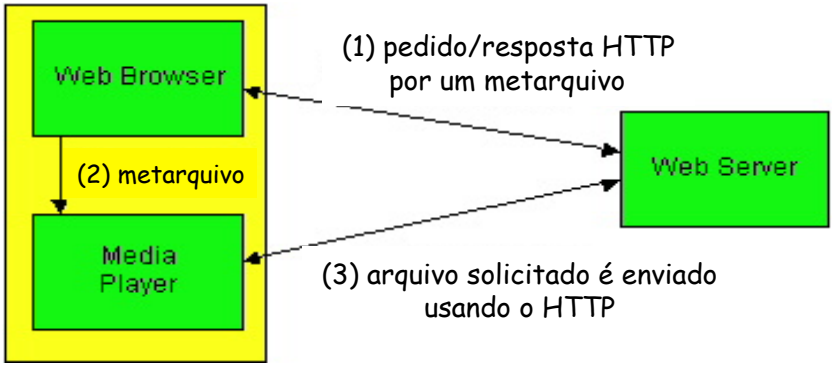


- Maior problema: o executor de mídia interage com o servidor WEB através do Web browser que atua como intermediário.

Acesso a Mídia Contínua a partir de servidores Web (2a. abordagem)

Alternativa: estabelecer conexão entre o servidor e o executor

- browser Web solicita e recebe um metarquivo (um arquivo descrevendo o objeto) ao invés de receber o próprio arquivo;
- O cabeçalho "Content-type" indica uma aplicação específica de aúdio e vídeo
- Browser dispara o executor de mídia e passa o metarquivo para ele
- executor estabelece uma conexão TCP com o servidor e envia a ele a mensagem HTTP

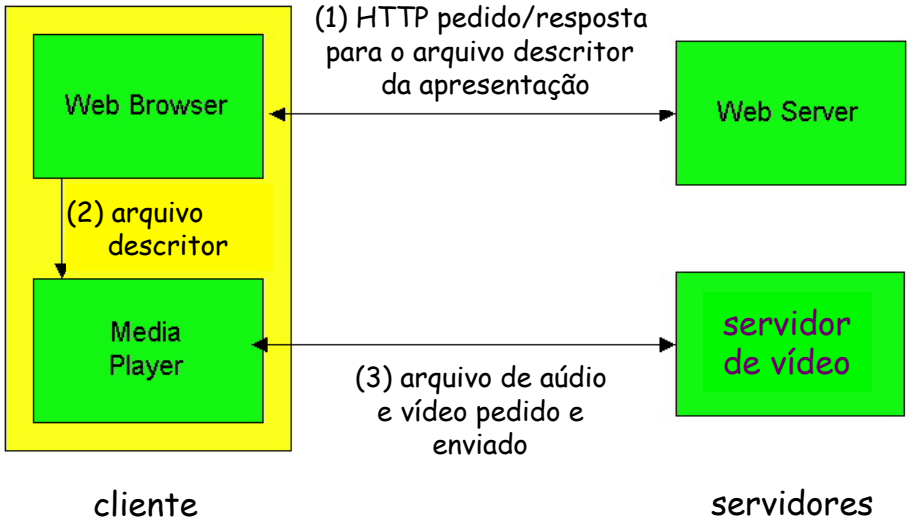


Algumas preocupações:

- O executor de mídia se comunica usando HTTP, que não foi projetado para suportar comandos de controle de apresentação
- Pode desejar enviar o aúdio e o vídeo sobre UDP

Acesso a Mídia Contínua a partir de um servidor específico (3a. abordagem)

- **Esta arquitetura permite o uso de outros protocolos (RTP, RTSP) (além do HTTP) entre o servidor e o executor de mídia (player)**
- **Pode usar também o UDP ao invés do TCP**



Real Time Streaming Protocol: RTSP

RTSP: RFC 2326

- **Protocolo de aplicação do tipo cliente-servidor.**
- **Permite ao usuário controlar apresentações de mídia contínua: voltar ao início, avançar, pausa, continuar, seleção de trilha, etc…**

O que ele não faz:

- **não define como o aúdio e o vídeo é encapsulado para transmissão sobre a rede**
- **não restringe como a mídia contínua é transportada: pode usar UDP ou TCP**
- **não especifica como o receptor armazena o aúdio e o vídeo**

Exemplo de uso: RealNetworks

- **Servidor e cliente usam RTSP para enviar informações de controle de um para o outro**

RTSP: controle fora da banda (out-of-band)

FTP usa um canal de controle "fora-da-banda":

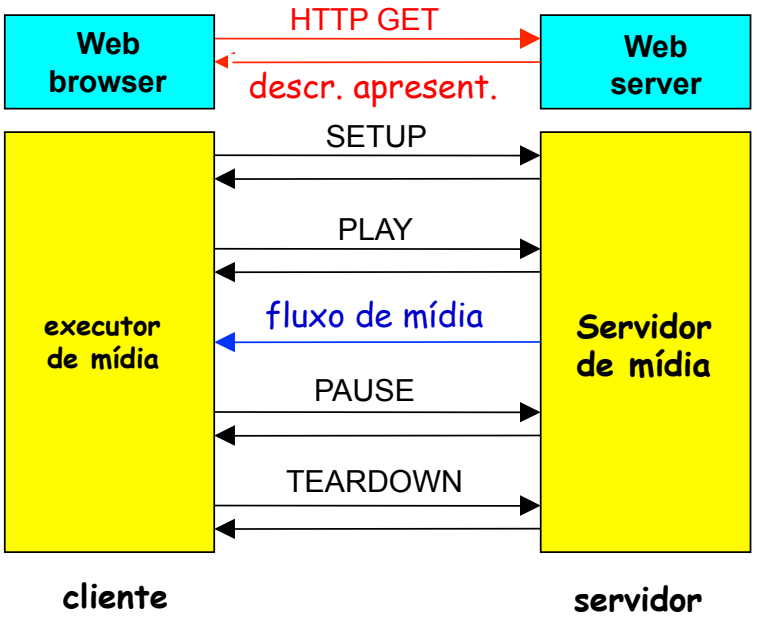
- **Um arquivo é transferido sobre uma conexão TCP.**
- **Informação de controle (mudanças de diretório, remoção de arquivos, trocas de nomes, etc.) é enviada sobre uma conexão TCP separada.**
- **Os canais "dentro-da-banda" e "fora-da-banda" usam diferentes números de portas.**

Mensagens RTSP também são enviadas "fora-da-banda":

- **As mensagens de controle RTSP usam diferentes números de portas em relação ao fluxo de dados de mídia contínua e, portanto, são enviadas "fora-da-banda".**
  - ***RTSP usa porta 544 do TCP ou UDP***
- **O fluxo de dados de mídia contínua, cuja estrutura de pacotes não é definida pelo RTSP, é considerada "dentro-da-banda".**

Iniciação do RTSP e controles de entrega

- **Cliente obtém uma descrição da apresentação multimídia, que pode consistir de vários fluxos de dados (através do HTTP).**
- **O browser chama o executor de mídia (aplicação auxiliar) com base no tipo de conteúdo da descrição da apresentação.**
- **A descrição da apresentação inclui referências aos fluxos de mídia (URLs) usando o esquema "rtsp://…"**
  1. *executor envia o comando RTSP SETUP; servidor envia a resposta RTSP SETUP.*
  2. *executor envia o comando RTSP PLAY; servidor envia a resposta RTSP PLAY.*
  3. *O servidor de mídia descarrega o fluxo de mídia.*
  4. *executor envia o comando RTSP PAUSE; o servidor envia a resposta RTSP PAUSE.*
  5. *executor envia o comando RTSP TEARDOWN; servidor envia a resposta RTSP TEARDOWN.*



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>
```

Sessão RTSP

- Cada sessão RTSP tem um identificador de sessão, que é escolhido pelo servidor.
- O cliente inicia a sessão com o comando SETUP, e o servidor responde ao comando com um identificador.
- O cliente repete o identificador de sessão em cada comando, até que o cliente encerra a sessão com o comando TEARDOWN.
- O número de porta do RTSP é 544.
- RTSP pode ser usado sobre UDP ou TCP. Cada mensagem RTSP pode ser enviada numa conexão TCP separada.

RTSP: exemplo de mensagens

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**

Aplicações interativas em tempo-real

- **Exemplo:**
  - ***Telefonia IP***

Telefonia Internet sobre melhor-esforço

Serviço de Melhor esforço

- **Acarreta atraso de pacotes, perdas e variação de retardo (jitter)**

Exemplo de telefone Internet

- **As aplicações de telefonia na Internet geram pacotes durante momentos de atividade da voz**
  - ***Rajadas de voz alternadas com períodos de silêncio***
- **taxa de bits é 64 kbps nos intervalos de atividade (G.711)**
- **durante os intervalos de atividade a aplicação produz um bloco de 160 bytes a cada 20 ms (8 kbytes/s x 20 ms)**
- **cabeçalho é acrescentado ao bloco; então bloco mais cabeçalho são encapsulados num pacote UDP e enviados**
- **alguns pacotes podem ser perdidos e o retardo de um pacote irá flutuar.**
- **receptor deve determinar quando reproduzir um bloco e determinar o que fazer com um bloco perdido**

Telefonia Internet sobre melhor-esforço

perda de pacotes

- **O segmento UDP é encapsulado num datagrama IP**
- **datagrama pode ser descartado por falta de espaço num roteador**
- **TCP pode eliminar perdas, mas**
  - ***retransmissões aumentam o atraso***
  - ***O controle de congestionamento do TCP limita a taxa de transmissão***
- **Taxas de perda entre 1 e 20% podem ser toleradas**
- **Pacotes redundantes podem ajudar**
  - ***Mecanismos de FEC (forward error control) ajudam a ocultar as perdas***

Telefonia Internet sobre melhor-esforço

retardo fim-a-fim

- **acúmulo dos retardos de transmissão, propagação, atrasos nas filas dos roteadores e retardos de processamento nos sistemas finais**
- **mais que 400 ms de retardo fim-a-fim compromete a interatividade; quanto menor o retardo melhor**
  - ***Normalmente o receptor descarta pacotes com retardo maior que um patamar***

Telefonia Internet sobre melhor-esforço

variação de atraso (jitter)

- **Retardo nas filas dos roteadores é aleatório**
- **considere dois pacotes consecutivos num intervalo de atividade**
- **Os retardos fim-a-fim desses dois pacotes podem ser diferentes**
- **espaçamento inicial é de 20 ms, mas o espaçamento no receptor pode ser maior ou menor que 20 ms**

o jitter pode ser removido:

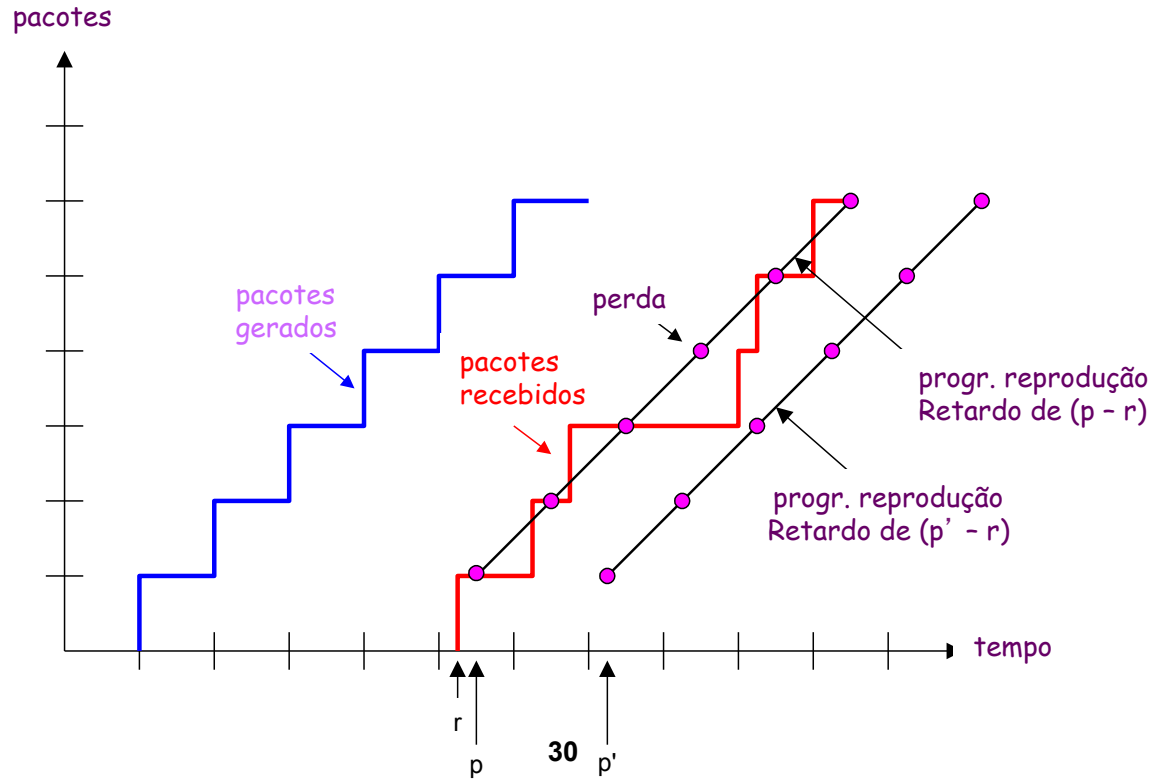
- **Precedendo cada bloco com um número de seqüência**
  - *transmissor incrementa esse número para cada novo pacote*
- **Precedendo cada bloco com uma marca de tempo**
  - *transmissor marca cada bloco com o tempo em que foi gerado*
- **Atrasando a reprodução**
  - *O atraso na reprodução deve ser suficiente para que a maioria dos pacotes seja recebida antes do seu tempo de reprodução programado*
    - Atraso de reprodução fixo ou adaptativo

Atraso de Reprodução Fixo

- **Receptor tenta reproduzir cada bloco exatamente *q* ms depois que o bloco é gerado.**
  - ***Se o bloco tem marca de tempo t, receptor usa o bloco no instante t+q .***
  - ***Se o bloco chega após o instante t+q, receptor o descarta.***
- **Escolha do valor de *q*:**
  - ***q grande: menos perda de pacotes***
  - ***q pequeno: melhor controle da interatividade***

Atraso de Reprodução Fixo

- Transmissor gera pacotes a cada 20 ms durante os intervalos de atividade.
- Primeiro pacote é recebido no instante r
- Primeira programação de reprodução: começa em p
- Segunda programação de reprodução: começa em p'



Atraso de Reprodução Adaptativo

• para serviços com interatividade, atrasos fixos longos podem se tornar incômodos ou intoleráveis

•Estima o retardo da rede e ajusta o retardo de reprodução no início de cada intervalo de atividade.

• Intervalos de silêncio são comprimidos e alongados.

• Blocos ainda são gerados a cada 20 ms nos intervalos de atividade.

t_i = marca de tempo do i – ésimo pacote

r_i = instante no qual o pacote i é recebido pelo receptor

p_i = instante no qual o pacote i é reproduzido no receptor

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

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

Estimativa dinâmica do retardo 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 de Reprodução Adaptativo

É também usual estimar a variância média do atraso, v_i:

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

As estimativas de d_i e v_i são calculadas para cada pacote recebido, embora elas sejam usadas apenas no início de um intervalo de atividade.

Para o primeiro pacote de um intervalo de atividade, o instante de reprodução é:

$$p_i = t_i + d_i + Kv_i$$

onde K é uma constante positiva. Para este mesmo pacote, o retardo de reprodução é:

$$q_i = p_i - t_i$$

Para o pacote j no mesmo intervalo de atividade, o pacote deve ser reproduzido em:

$$p_j = t_j + q_i$$

Atraso de Reprodução Adaptativo

Como saber se um pacote é o primeiro de um intervalo de atividade:

- **Se nunca houvesse perdas o receptor poderia simplesmente olhar nas marcas de tempo sucessivas.**
  - ***Se a diferença de marcas de tempo sucessivas for maior que 20 ms, então temos o início de um intervalo de atividade.***
- **Mas porque as perdas são possíveis, o receptor deve olhar tanto as marcas de tempo como os números de sequência dos pacotes.**
  - ***Se a diferença de marcas de tempo sucessivas for maior que 20 ms e não há pulos nos números de seqüência então tem-se o início de um intervalo de atividade.***

Recuperação de Perdas de Pacotes

- **Perdas: pacote nunca chega ou chega depois do seu tempo de reprodução programado**
- **Correção por FEC ou intercalamento**

Forward error correction (FEC): esquema simples

- **para cada grupo de n blocos, cria um bloco redundante realizando uma operação OU exclusivo (XOR) entre os n blocos originais**
- **envia os n+1 blocos, aumentando a banda passante por um fator de 1/n.**
- **pode reconstruir os n blocos originais se houver no máximo um bloco perdido nos n+1 blocos enviados**
- **retardo de reprodução precisa ser definido para receber todos os n+1 pacotes**
- **Compromisso:**
  - ***aumentar n, menor desperdício de banda***
  - ***aumentar n, maior retardo de reprodução***
  - ***aumentar n, maior a probabilidade que dois ou mais blocos sejam perdidos***

Operação XOR (n=2)

- **Transmissão:**

- **1o. Pacote: 1 0 1**
- **2o. Pacote: 1 0 0**
- **Pacote 1 XOR Pacote 2**
- **Pacote FEC: 0 0 1**

- **Recepção:**

- **1o. Pacote: 1 0 1**
- **2o. Pacote: PERDIDO**
- **Pacote FEC: 0 0 1**

Corrige o erro:

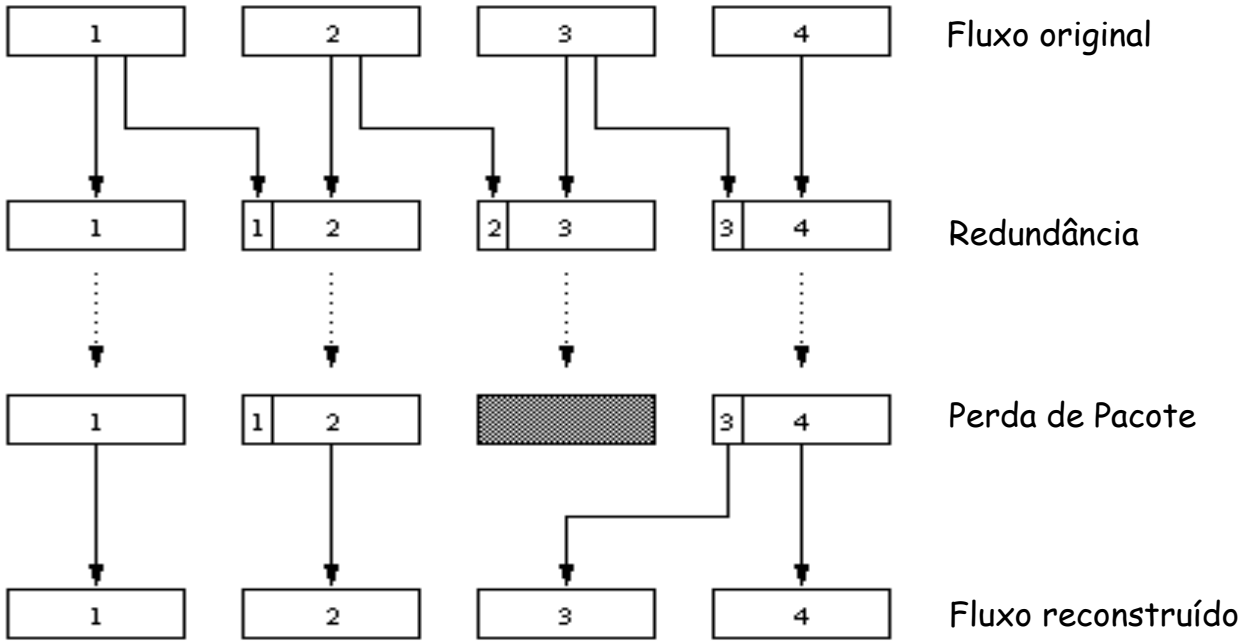
- **Pacote 1 XOR Pacote FEC: 1 0 0**
- **2o. Pacote: 1 0 0**

Recuperação de Perdas de Pacotes

2o. esquema FEC

- enviar um fluxo de menor qualidade como "carona"
- envia fluxo de aúdio de menor resolução como a informação redundante
- por exemplo, um fluxo PCM nominal a 64 kbps e um fluxo redundante a 13 kbps.
- Transmissor cria pacote tomando o bloco n do fluxo nominal e anexando a ele o bloco (n-1) do fluxo redundante.

Recuperação de Perdas de Pacotes



- **Sempre que ocorre perda não-consecutiva, o receptor pode esconder a perda.**
- **Apenas dois pacotes precisam ser recebidos antes do início da reprodução (aumento no retardo de reprodução é pequeno)**
- **Pode também anexar os blocos (n-1) e (n-2) do fluxo de baixa qualidade**