

Complementos de Redes

Licenciatura Eng.ª Informática

2º Semestre 2014/15

Prof. José Rogado

jose dot rogado at ulusofona dot pt

Prof. José Faísca

jose dot faisca at gmail dot com

Universidade Lusófona



Sumário Capítulo 3: Redes Multimédia

- ❑ Características das Aplicações Multimédia
 - Características
 - Necessidades
 - Compressão de Dados
- ❑ Fluxo de Áudio e Vídeo
 - Acesso Web
 - Servidores de Streaming
 - RSTP
- ❑ Multimédia em Tempo Real
 - Limitações do “Best Effort”
 - Características e necessidades do VoIP
- ❑ Protocolos para Aplicações Interactivas em Tempo Real
 - RTP
 - RTCP
 - SIP
 - H.323
- ❑ Qualidade de Serviço
 - Necessidades
 - Exemplos
- ❑ Implementação de QoS
 - Mecanismos de Escalonamento
 - Regulação de Serviço
- ❑ Architecturas de QoS
 - Serviços Integrados (Intserv) e Diferenciados (Diffserv)
- ❑ Protocolo de Sinalização: RSVP



Objectivos

Princípios

- ❑ Classificação das aplicações multimédia
- ❑ Identificação dos serviços de rede necessários
- ❑ Limites do “melhor esforço”
- ❑ Mecanismos da Qualidade de Serviço (QoS)

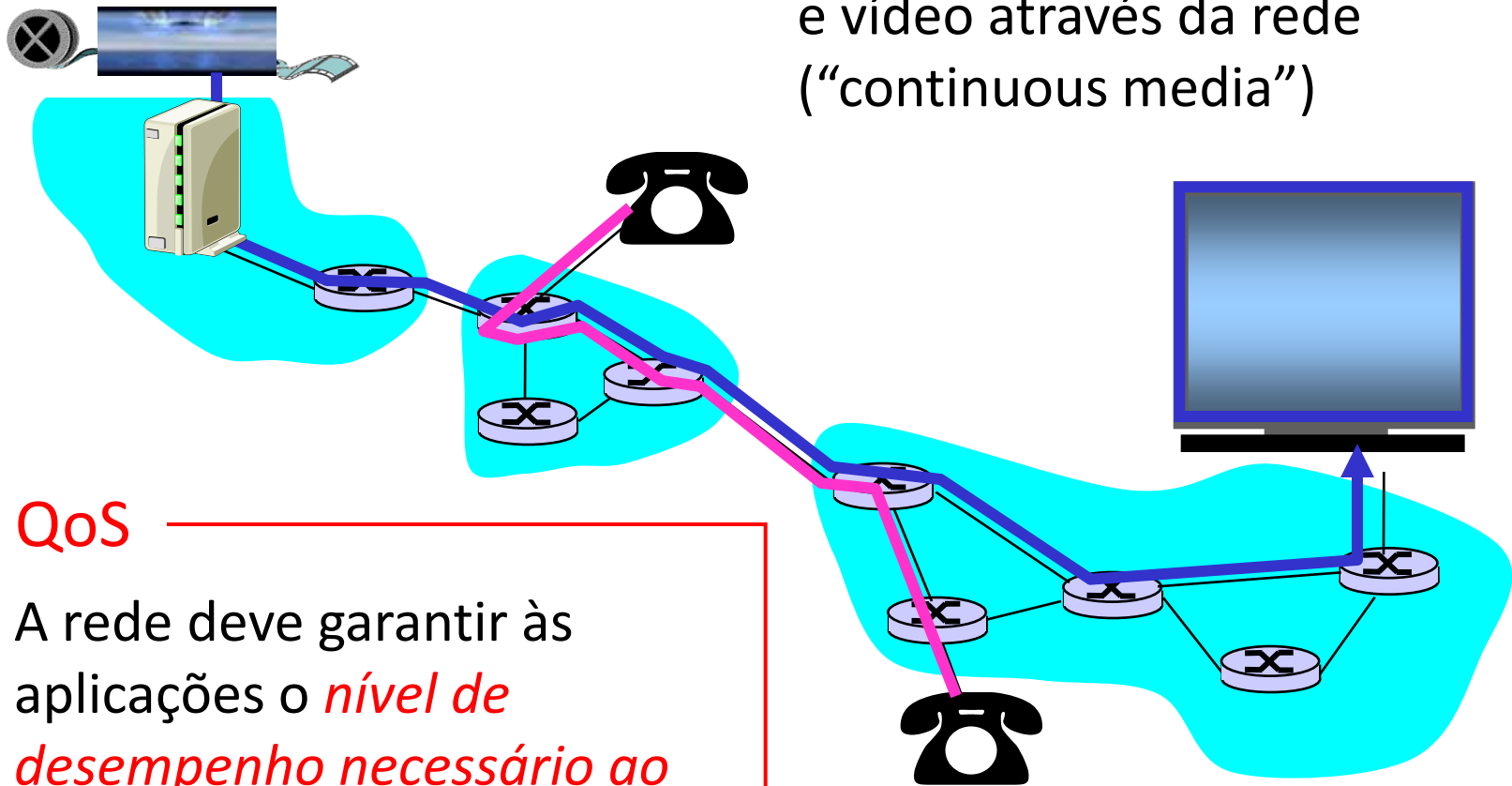
Protocolos e Architecturas

- ❑ Protocolos específicos de “melhor esforço”
- ❑ Architecturas de QoS



Multimédia e Qualidade de Serviço

Aplicações Multimédia : áudio e vídeo através da rede (“continuous media”)



QoS

A rede deve garantir às aplicações o *nível de desempenho necessário ao seu funcionamento*

Aplicações Multimédia

Classes de aplicações:

- 1) Fluxo de áudio e vídeo previamente armazenados
- 2) Fluxo de áudio e vídeo produzido em tempo real (*live*)
- 3) Áudio e vídeo *interactivos* em tempo real (*bidireccional*)

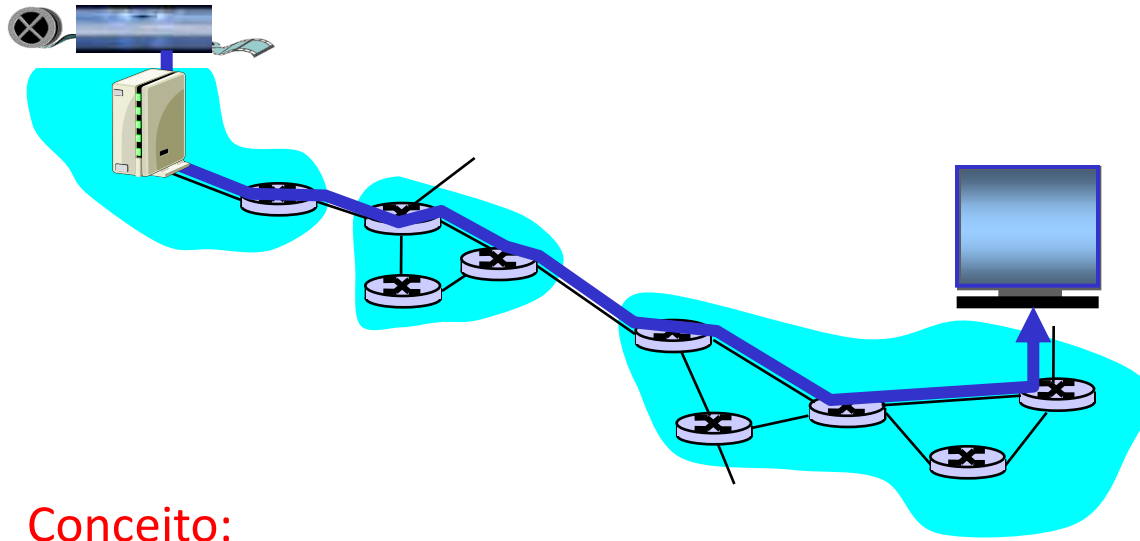
Características fundamentais:

- ❑ Particularmente **sensíveis a atrasos**:
 - Atrasos na entrega
 - Atrasos na cadência (*"jitter"*)
- ❑ Mas **toleram percas**: percas pouco frequentes causam interrupções menores (*glitches*)
- ❑ Diferença fundamental em relação a aplicações de dados
 - Tolerem atrasos
 - Não toleram percas

Jitter designa a variabilidade do atraso entre pacotes dentro do mesmo fluxo de dados



Streaming Multimédia

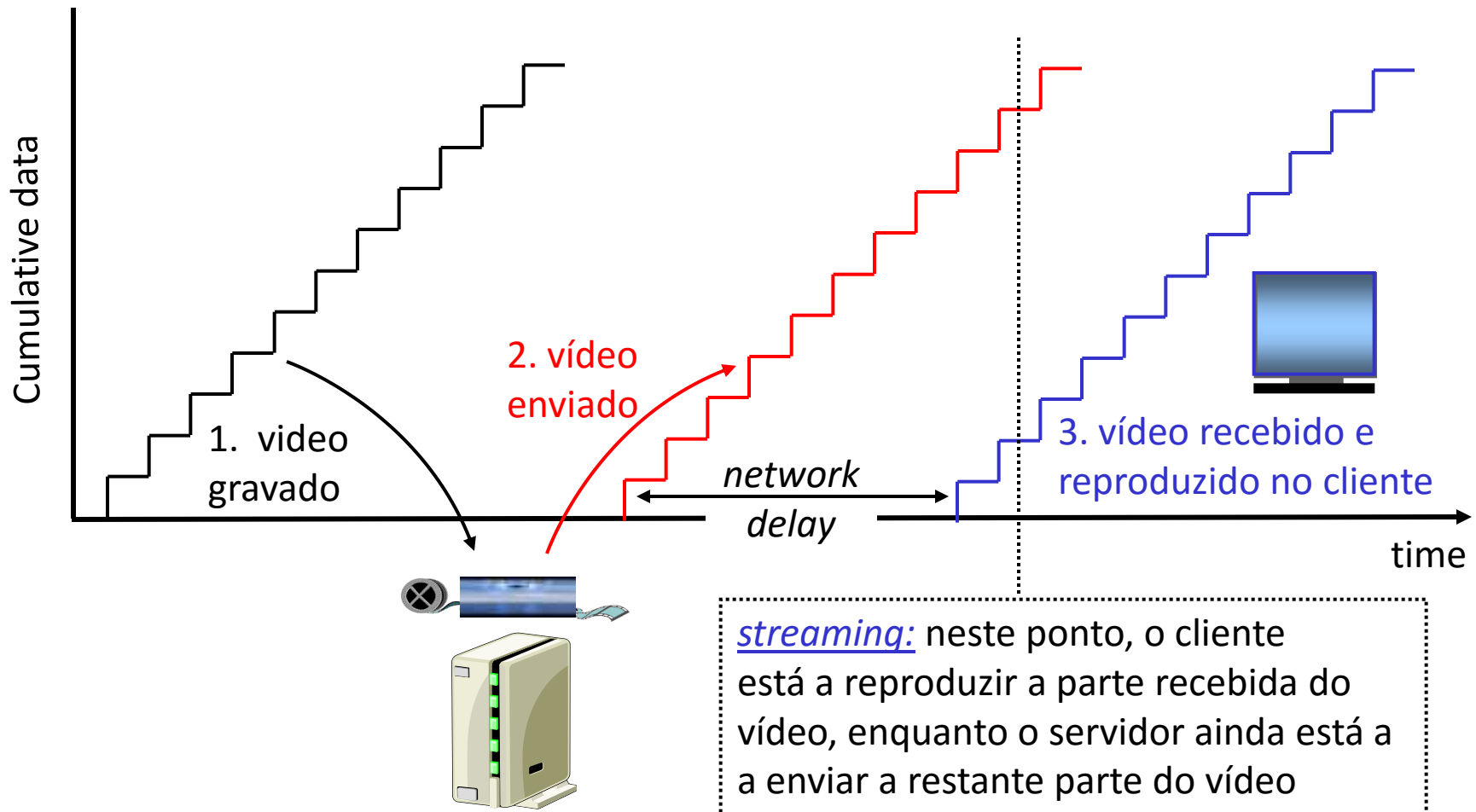


Conceito:

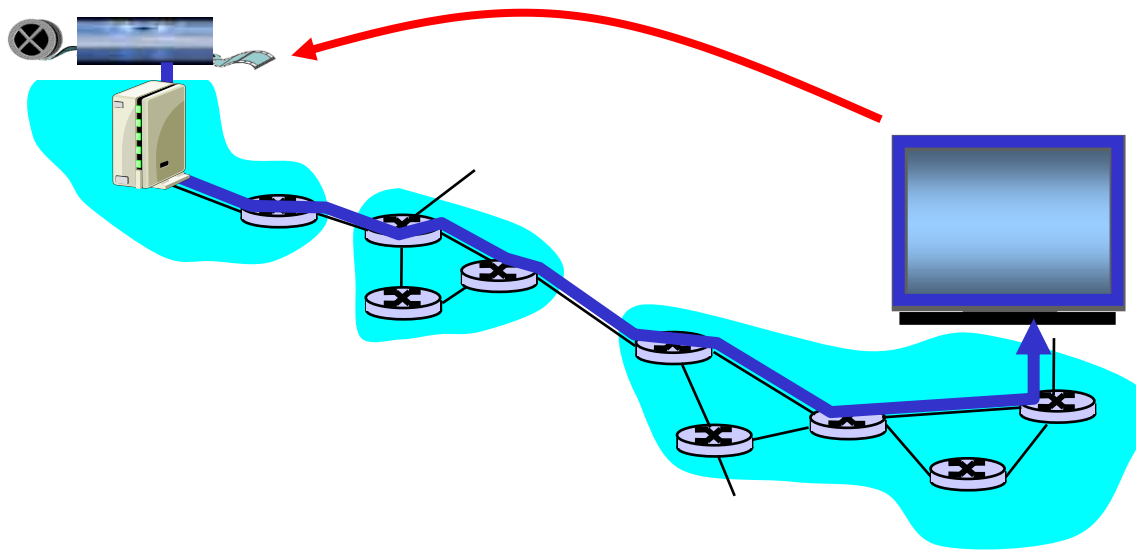
- ☐ Dados armazenados na origem
- ☐ Transmitidos para o cliente
- ☐ *Streaming*: o cliente começa a reprodução antes da totalidade dos dados ter chegado ao destino
- ☐ Constrangimento temporal: os dados em transmissão devem chegar a tempo para poderem ser reproduzidos sem interrupção



Streaming: Aspectos Temporais



Streaming Multimédia Interactivo



Funcionalidades de tipo VCR:

- ☐ A aplicação cliente pode fazer pause, Rewind, FF e reposicionar a reprodução
- ☐ Um atraso inicial de 10 seg. é admissível
- ☐ Atraso de 1-2 seg. no efeito dos comandos
- ☐ Utilização do RTSP (ver depois...)

Streaming Multimedia Live

Exemplos:

- ❑ Radio Internet
- ❑ Webcast
- ❑ ...

Streaming

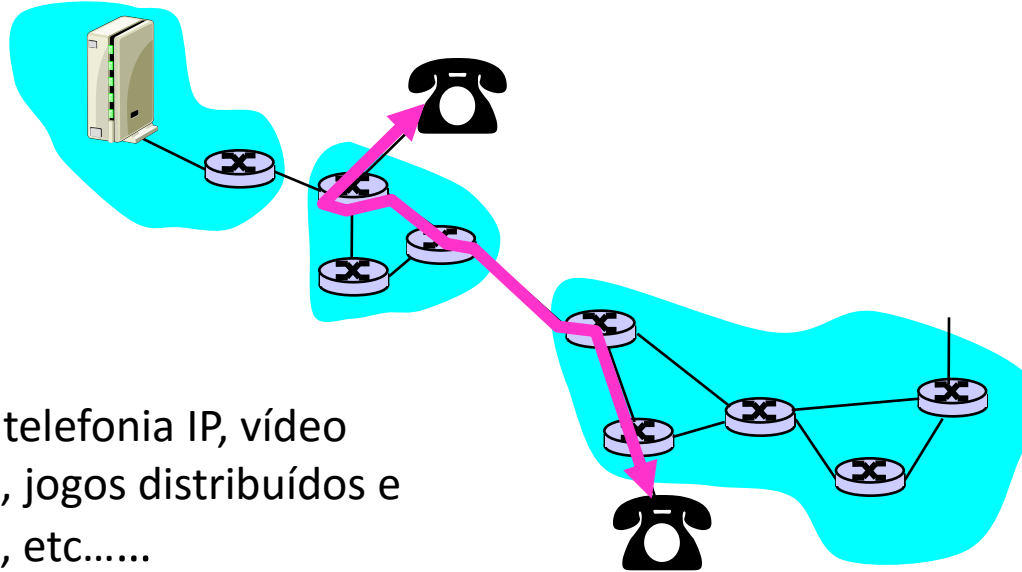
- ❑ Utiliza um *buffer* de reprodução
- ❑ A reprodução pode estar atrasada algumas décimas de segundo em relação à transmissão
- ❑ Maiores constrangimentos temporais
 - Em caso de percas de dados pode não ser possível a retransmissão

Interactividade

- ❑ É impossível avançar (FF)
- ❑ Pause é sempre possível
- ❑ Algumas aplicações permitem Rewind



Interactividade Multimédia em Tempo Real



- ❑ **Aplicações:** telefonia IP, vídeo conferencia, jogos distribuídos e interactivos, etc.....
- ❑ **Requisitos atraso ponto a ponto:**
 - Áudio: < 150 msec bom, < 400 msec aceitável
 - Deve incluir atraso aplicacional e de rede
 - Atrasos maiores tornam-se perceptíveis e impedem interactividade
- ❑ **Necessidade de Inicialização de Sessão**
 - Para o iniciador: indicar o endereço IP do correspondente, número de porto, algoritmos de codificação, ...

Multimédia na Internet actual

TCP/UDP/IP: serviço de “melhor esforço”

❑ **Não** há garantias no atraso e no *jitter* e pode haver perda de pacotes



?

?

?

?

?

Mas as aplicações multimédia requerem QoS e
taxas de transmissão elevadas para serem
utilizáveis!

?

?

?

?

?

?



As aplicações multimédia actuais utilizam técnicas ao nível
aplicacional para mitigar (se possível) os efeitos dos atrasos e
percas de pacotes



Possíveis evoluções da Internet para melhor suportar multimédia

Serviços Integrados:

- ❑ Modificações estruturais nos protocolos da Internet de modo a que as aplic. possam reservar largura de banda ponto a ponto
- ❑ Requer versões de software complexo nos *hosts* e *routers*

Entretanto:

- ❑ Não mudar grande coisa
- ❑ Oferecer mais largura de banda quando necessário
- ❑ Utilizar distribuição de conteúdos e *multicast* ao nível aplicacional
 - Só muda o nível aplicacional

Serviços Diferenciados:

- ❑ Algumas mudanças nas camadas inferiores para distinguir serviços de 1ª e 2ª classe
- ❑ Algoritmos de gestão de filas baseados em prioridades



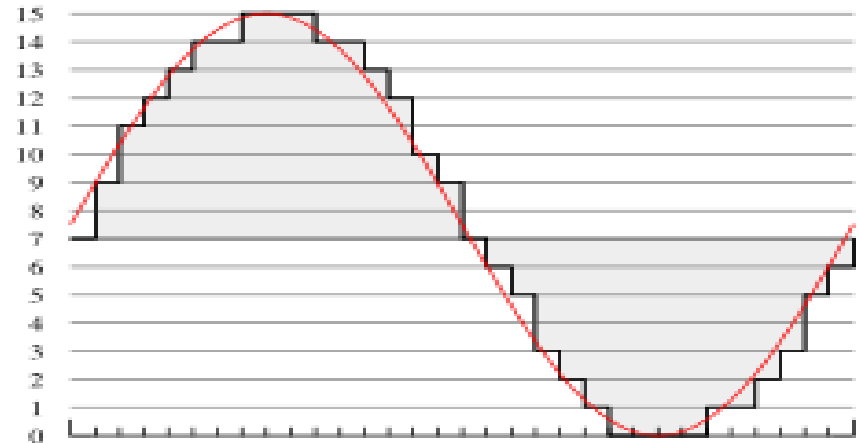
Opiniões?



Codificação de Áudio

PCM - **Pulse Code Modulation**

- ❑ Sinal Analógico amostrado a uma taxa constante
 - telefone: 8.000 amostras/seg
 - CD Musical: 44.100 amostras/seg.
- ❑ Cada amostra é quantificada i.e. digitalizada
 - O número de valores por amostra depende da granularidade
- ❑ Os valores amostrados são armazenados em bits
 - $2^8 = 256$
 - ⇒ 8 bits para cada 256 valores
- ❑ O receptor converte o valor digitalizado para sinal analógico:
 - Alguma perda de qualidade



Exemplo de Taxas de Transmissão

- ❑ Voz: 8.000 amostras/s de 8 bits - 256 valores possíveis
 - => 64 kbps
- ❑ Música HiFi Stereo (CD)
 - $44.100 \times 16 \text{ bits} \times 2 = 1,411 \text{ Mbps}$

=> Transmissão de áudio de qualidade necessita de uma largura de banda adequada



Compressão de Áudio

- ❑ A transmissão de Dados requer uma largura de banda adequada
 - Fala: 64 kbps elevada demais para um modem dial-up
 - Música em Stereo: 1,411 Mbps acessível na maioria do acesso residencial
- ❑ Mesmo assim é utilizada compressão na transmissão
 - Remoção da redundância
 - Remoção de frequências que o ouvido humano não registra
- ❑ Alguns Formatos de Compressão
 - Áudio: MPEG 1 layer 3 (MP3): 96, 128 e 160 kbps
 - Voz: GSM 06.10 (13 kbps), G.729 (8 kbps), G.723.3 (5.3 e 6.4)
 - Ver:
 - <http://www.voip-info.org/wiki-ITU+G.729>



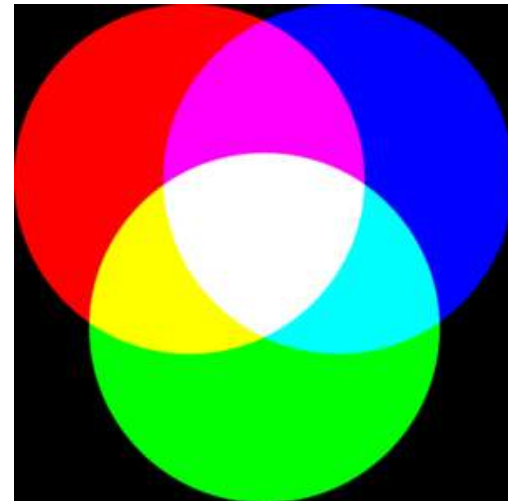
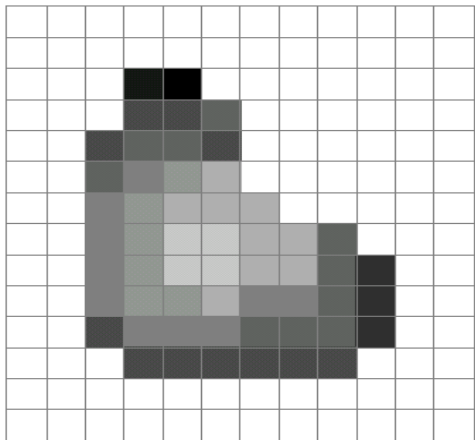
Codificação de Video

❑ Amostragem do Sinal Analógico

- Amostra-se a uma cadência fixa (e.g., 24 ou 30 por seg.)
- Cada amostra é uma imagem

❑ Cada amostra é quantificada

- Uma imagem é representada por uma matriz de elementos de imagem (*picture element: **pixel***)
- Cada elemento é uma mistura de 3 cores fundamentais
- E.g., 24 bits, com 8 bits por cor



Diferenças de Amostragem

A close-up photograph of a person with dark hair, wearing a blue and white striped shirt. They are covering their face with both hands, with their fingers spread. The background is dark and out of focus.

The
2272 x 1704
hand

The
320 x 240
hand

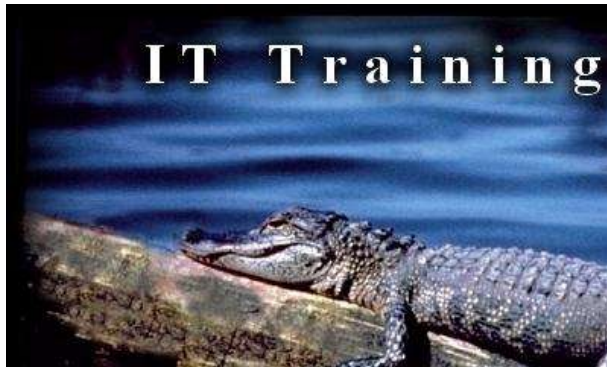
Compressão de Vídeo: em cada imagem

❑ Compressão de Imagem

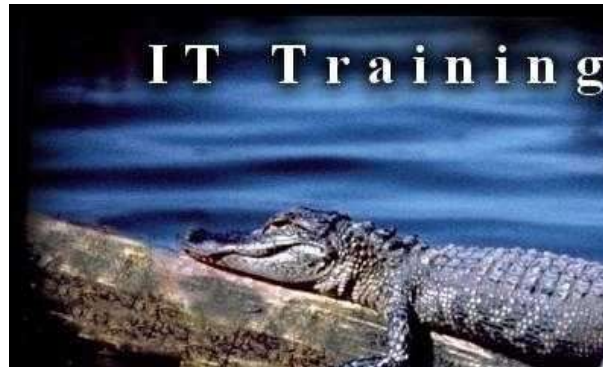
- Tirar partido da redundância espacial (eg: regiões da mesma cor)
- Omitir detalhes que o olho humano não distingue

❑ Formatos de Compressão de Imagem mais comuns

- JPEG - Joint Pictures Expert Group
- GIF - Graphical Interchange Format
- H.264 ou Advanced Video Coding (MPEG-4 AVC)



Uncompressed: 167 KB



Good quality: 46 KB



Poor quality: 9 KB



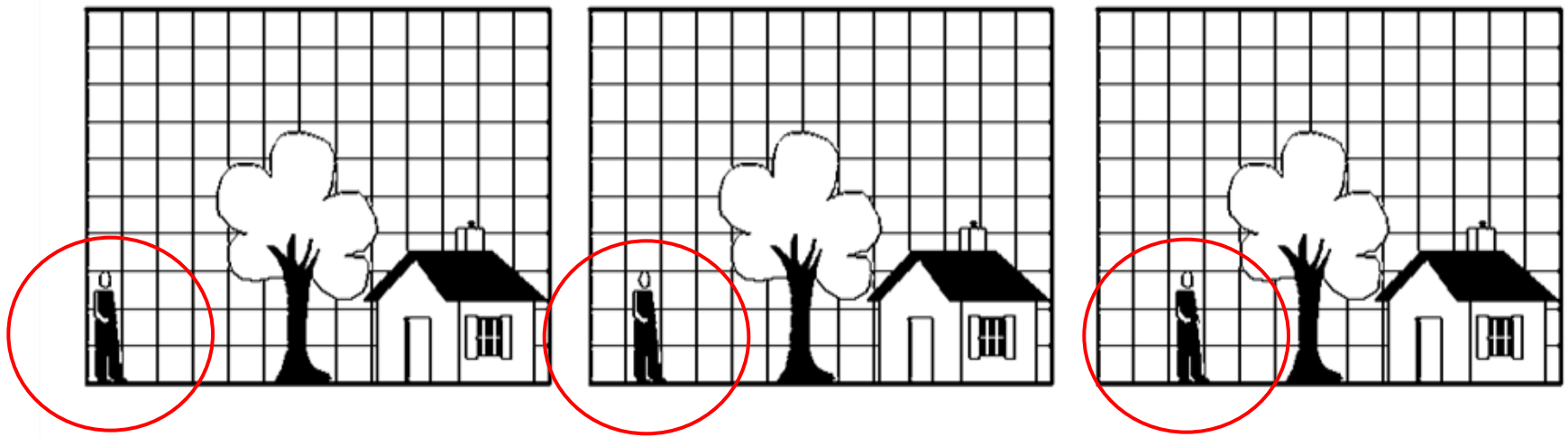
Compressão de Vídeo: entre imagens

❑ Compressão entre imagens

- Tira partido da redundância temporal entre imagens

❑ Formatos de compressão de video mais comuns

- MPEG 1: CD-ROM quality video (1.5 Mbps)
- MPEG 2: high-quality DVD video (3-6 Mbps)
- MPEG 4: muito usada na Internet (< 1 Mbps)
- Protocolos Proprietários como QuickTime and RealNetworks



Capítulo 3: Redes Multimédia

- ❑ Características das Aplicações Multimédia
 - Características
 - Necessidades
 - Compressão de Dados
- ❑ Fluxo de Áudio e Vídeo
 - Acesso Web
 - Servidores de Streaming
 - RSTP
- ❑ Multimédia em Tempo Real
 - Limitações do “Best Effort”
 - Características e necessidades do VoIP
- ❑ Protocolos para Aplicações Interactivas em Tempo Real
 - RTP
 - RTCP
 - SIP
 - H.323
- ❑ Qualidade de Serviço
 - Necessidades
 - Exemplos
- ❑ Implementação de QoS
 - Mecanismos de Escalonamento
 - Regulação de Serviço
- ❑ Architecturas de QoS
 - Serviços Integrados (Intserv) e Diferenciados (Diffserv)
- ❑ Protocolo de Sinalização: RSVP



Streaming Multimedia

- ❑ O *streaming* é realizado ao nível aplicacional
- ❑ Utiliza técnicas para tirar o melhor partido do serviço de melhor esforço:
 - Antecipação da parte do cliente (buffering)
 - Utilização de UDP em vez de TCP
 - Utilização de diversos formatos de compressão
- ❑ O cliente é geralmente uma aplicação dedicada à funcionalidade de reprodução
 - Media Player
 - Flash Player (**Real Time Messaging Protocol** - RTMP)

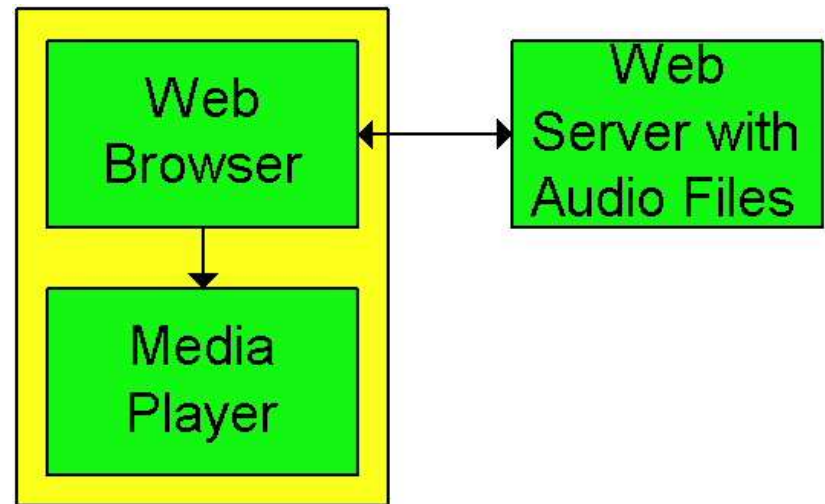
Media Player

- ❑ Supressão do *jitter*
- ❑ Descompressão
- ❑ Isolamento de erros
- ❑ Interface Gráfico para realizar o *rendering* do media e fornecer controle interactivo
- ❑ Pode ser inicializado de várias formas em função do tipo de streaming pretendido

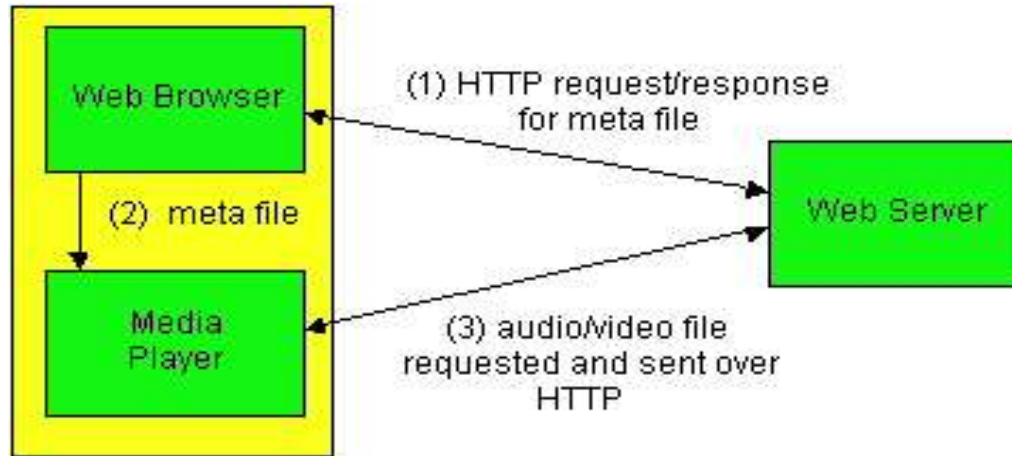


Transferência Ficheiro Multimédia

- ❑ Dados armazenados
 - Ficheiros áudio ou vídeo separados
 - Áudio e vídeo interligados no mesmo ficheiro
- ❑ Pedido-resposta HTTP
 - Ligação TCP entre o cliente e servidor
- ❑ Ficheiros transferidos como um objecto HTTP
 - Recebido totalmente pelo browser
- ❑ O browser analisa o *content-type* e passa o conteúdo ao **media player**
 - O media realiza o *rendering* do conteúdo
- ❑ O *streaming* não é realizado directamente da rede
 - Tempo de espera até à reprodução é longo
 - Mas a qualidade da reprodução pode ser muito melhor
 - Não é necessário gerir aspectos temporais na transferência



Streaming Iniciado por HTTP

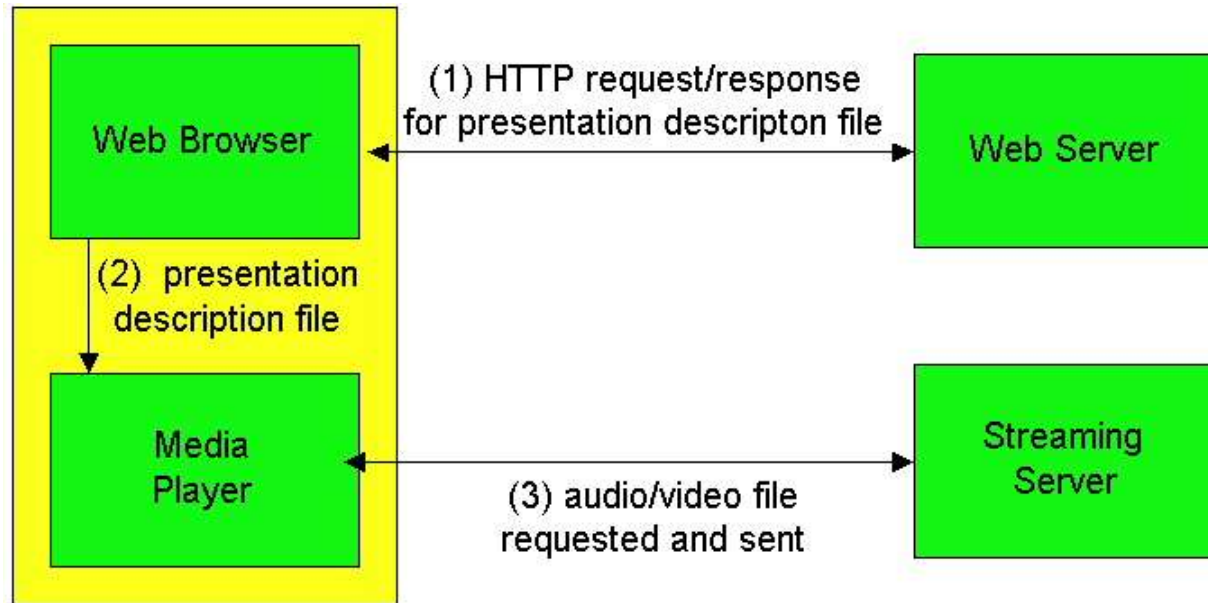


Evita passar os dados pelo Web Browser

- ❑ O Servidor Web retorna um ficheiro com informação (meta file) que descreve o objecto
 - URL, tipo, codificação, etc...
- ❑ O Browser inicia o media player e passa-lhe os essa informação
- ❑ O player estabelece a sua própria ligação com o Web Server
 - O streaming é realizado integralmente pelo player
- ❑ É uma das tecnologias usadas no YouTube (Flash Player – HTTP)



Utilização de Servidores de Streaming



- ❑ O Web Server serve de intermediário na apresentação do objecto
 - O Player estabelece contacto directo com o servidor de media
- ❑ Possibilita a utilização de protocolos diferentes do HTTP entre o servidor de media e o respectivo player
 - Possível utilização de transporte UDP em vez de TCP.
- ❑ CDNs: Content Delivery Networks
 - Exemplos: Akamai, Limelight, Google Video, etc...

No.	Time	Source	Destination	Protocol	Info
5	3.107893	192.168.1.203	209.85.227.101	TCP	2499 > http [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460
6	3.165659	209.85.227.101	192.168.1.203	TCP	http > 2499 [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430
7	3.165716	192.168.1.203	209.85.227.101	TCP	2499 > http [ACK] Seq=1 Ack=1 Win=17160 Len=0
8	3.166004	192.168.1.203	209.85.227.101	HTTP	GET /get_video?video_id=hZ8Jn1JvtdE&t=vjVQa1PpcFMsyAqCLurrD92KKy
9	3.211166	209.85.227.101	192.168.1.203	TCP	http > 2499 [ACK] Seq=1 Ack=883 Win=7056 Len=0
10	3.384040	209.85.227.101	192.168.1.203	HTTP	HTTP/1.1 303 See Other
11	3.385106	192.168.1.203	74.125.170.145	TCP	2500 > http [SYN] Seq=0 Ack=0 Win=16384 Len=0 MSS=1460
12	3.437763	74.125.170.145	192.168.1.203	TCP	http > 2500 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
13	3.437817	192.168.1.203	74.125.170.145	TCP	2500 > http [ACK] Seq=1 Ack=1 Win=17520 Len=0
14	3.438360	192.168.1.203	74.125.170.145	HTTP	GET /videoplayback?ip=0.0.0.0&sparams=id%2Cexpire%2Cip%2Cipbits%
15	3.492402	74.125.170.145	192.168.1.203	TCP	http > 2500 [ACK] Seq=1 Ack=613 Win=6732 Len=0
16	3.518592	74.125.170.145	192.168.1.203	HTTP	HTTP/1.1 200 OK
17	3.518993	74.125.170.145	192.168.1.203	HTTP	Continuation or non-HTTP traffic
18	3.519017	192.168.1.203	74.125.170.145	TCP	2500 > http [ACK] Seq=613 Ack=715 Win=16806 Len=0
19	3.520063	74.125.170.145	192.168.1.203	HTTP	Continuation or non-HTTP traffic
20	3.551717	192.168.1.203	209.85.227.101	TCP	2499 > http [ACK] Seq=883 Ack=614 Win=16547 Len=0
21	3.591017	74.125.170.145	192.168.1.203	HTTP	Continuation or non-HTTP traffic
22	3.591078	192.168.1.203	74.125.170.145	TCP	2500 > http [ACK] Seq=613 Ack=3635 Win=17520 Len=0
23	3.592751	74.125.170.145	192.168.1.203	HTTP	Continuation or non-HTTP traffic
24	3.593537	74.125.170.145	192.168.1.203	HTTP	Continuation or non-HTTP traffic
25	3.593554	192.168.1.203	74.125.170.145	TCP	2500 > http [ACK] Seq=613 Ack=6555 Win=17520 Len=0
26	3.647025	74.125.170.145	192.168.1.203	HTTP	Continuation or non-HTTP traffic
27	3.648314	74.125.170.145	192.168.1.203	HTTP	Continuation or non-HTTP traffic
28	3.648340	192.168.1.203	74.125.170.145	TCP	2500 > http [ACK] Seq=613 Ack=9475 Win=17520 Len=0
29	3.649584	74.125.170.145	192.168.1.203	HTTP	Continuation or non-HTTP traffic
30	3.651035	74.125.170.145	192.168.1.203	HTTP	Continuation or non-HTTP traffic
31	3.651052	192.168.1.203	74.125.170.145	TCP	2500 > http [ACK] Seq=613 Ack=12395 Win=17520 Len=0
32	3.651682	74.125.170.145	192.168.1.203	HTTP	Continuation or non-HTTP traffic
33	3.703307	74.125.170.145	192.168.1.203	HTTP	Continuation or non-HTTP traffic

Redirect

Transmission Control Protocol, Src Port: http (80), Dst Port: 2500 (2500), Seq: 323, Ack: 613, Len: 392

Source port: http (80)
Destination port: 2500 (2500)
Sequence number: 323 (relative sequence number)
[Next sequence number: 715 (relative sequence number)]
Acknowledgement number: 613 (relative ack number)
Header length: 20 bytes
Flags: 0x0018 (PSH, ACK)
Window size: 6732
Checksum: 0x2863

Hypertext Transfer Protocol

Data (392 bytes)

```

0050 0a 6f 6e 4d 65 74 61 44 61 74 61 08 00 00 00 0c .onMetaD ata....
0060 00 08 64 75 72 61 74 69 6f 6e 00 40 74 7a ac 08 ..durati on.@tz..
0070 31 26 e9 00 09 73 74 61 72 74 74 69 6d 65 00 00 l&...sta rttime..
0080 00 00 00 00 00 00 00 00 0d 74 6f 74 61 6c 64 75 .....totaldu
0090 72 61 74 69 6f 6e 00 40 74 7a ac 08 31 26 e9 00 ration.@ tz..l&..
00a0 05 77 69 64 74 68 00 40 74 00 00 00 00 00 00 00 .width.@ t.....
00b0 06 68 65 69 67 68 74 00 40 66 00 00 00 00 00 00 .height.@f.....
00c0 00 0d 76 69 64 65 6f 64 61 74 61 72 61 74 65 00 ..videod atarate.
00d0 40 66 e2 d3 1a 8b 45 04 00 0d 61 75 64 69 6f 64 @f....E. ..audiod
00e0 61 74 61 72 61 74 65 00 40 4f a1 8d 62 fb 6c cc atarate. @0..b.l.
00f0 00 0d 74 6f 74 61 6c 64 61 74 61 72 61 74 65 00 ..totald atarate.
0100 40 6f 52 7d d1 c6 79 bd 00 09 66 72 61 6d 65 72 @oRl...v. ..framer

```


O TCP pode não ser a melhor solução...

- ❑ Entrega fiável
 - Retransmissão dos pacotes perdidos
 - ... mesmo quando não é necessária
- ❑ Adaptação da taxa de emissão
 - Redução da taxa emissão em caso de perda de pacotes
 - ... provocando carência de dados no cliente
- ❑ Overhead do protocolo
 - O Header do TCP é maior do que o do UDP (20 bytes)
 - ... é grande demais para o envio de pacotes de controlo
 - Envia ACKs para todos os pacotes
 - ... que pode ser informação demasiada e redundante
 - Utiliza o slow-start para não saturar a rede
 - ... quando é necessário enviar dados para encher o buffer do cliente
- ❑ Por outro lado o TCP permite
 - Atravessar firewalls mais facilmente
- ❑ A utilização de HTTP tem vindo a generalizar-se...

Os protocolos de streaming podem utilizar transporte UDP ou TCP indiferentemente, dependendo das condições e configuração da rede de acesso



... mas UDP implica mais trabalho para a aplicação

- ❑ A utilização de UDP implica mais funcionalidade a nível aplicacional
 - Determinação dos timings de transmissão
 - Encapsulamento dos dados com informações de controlo
 - Retransmissão dos pacotes realmente perdidos
 - Adaptação da taxa de emissão
 - Qualidade dos dados transmitidos (compressão...)
- ❑ Necessidade de redefinir a noção de perda
 - O pacote chega a tempo para a reprodução, ou não
 - Não há necessidade de retransmissão se já passou o deadline
- ❑ Utilização da noção de **Forward Error Correction (FEC)**
 - Introdução de informação adicional no stream de dados
 - Envio simultâneo de trechos de mais baixa qualidade
 - Scrambling do conteúdo dos pacotes para que no caso de percas não seja de um bloco contínuo



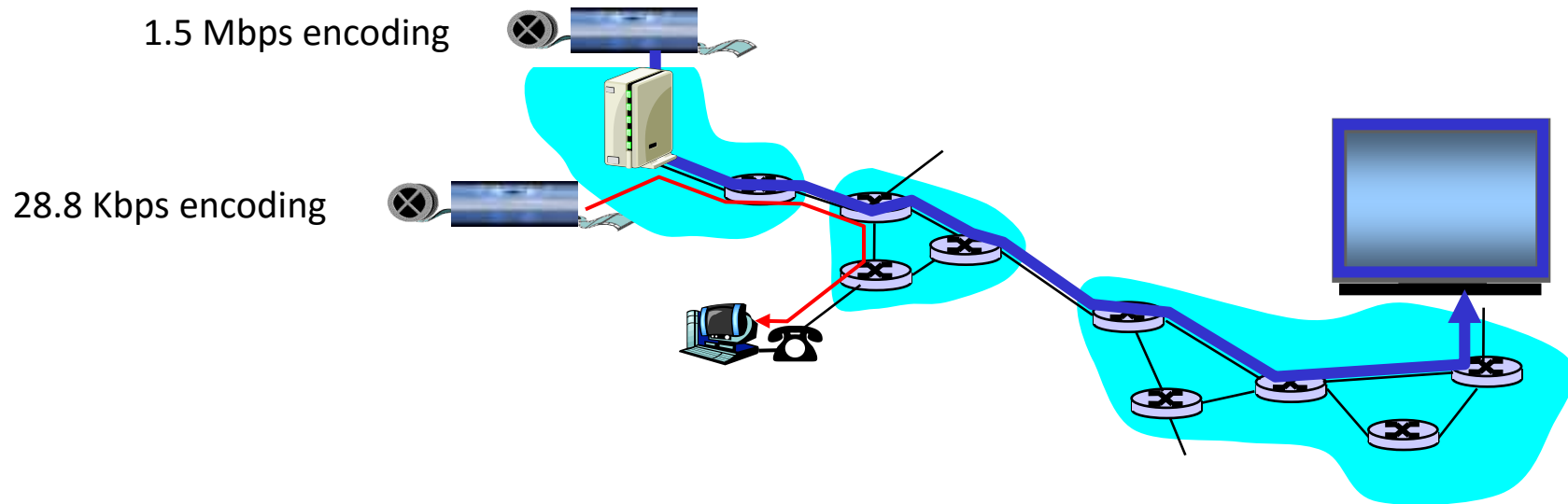
Protocolos de Streaming na Web

- ❑ Adobe
 - HDS (HTTP Dynamic Streaming)
 - Adobe RTMP (Adobe Flash Player)
- ❑ Apple HLS (HTTP Live Streaming)
 - iPhone, iPad, iPod touch, QuickTime, ...
- ❑ Microsoft Smooth Streaming
 - Silverlight, Windows Phone , ...
- ❑ MPEG-DASH (Dynamic Adaptive Streaming over HTTP)
 - Standard independente, em vias de adoção pela comunidade
- ❑ RTSP/RTP
 - QuickTime, VLC, 3GPP devices, set-top boxes, ...

<http://www.wowza.com/forums/content.php?621-Understanding-streaming-protocols-and-output-file-format>



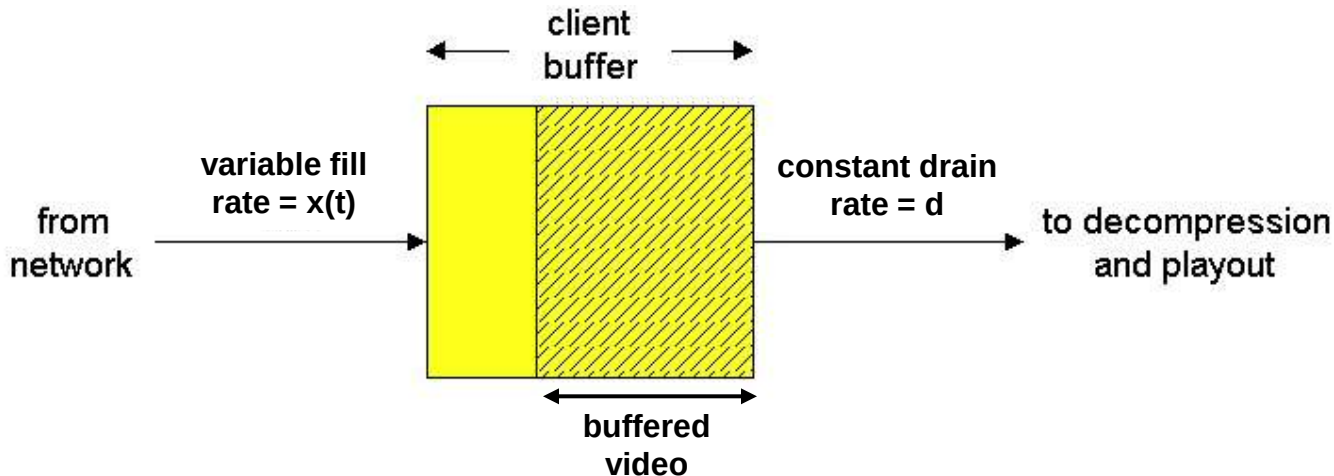
Adaptação à taxa de recepção do cliente



- ❑ Para gerir diferentes taxas de recepção do cliente
 - 1Mbps UMTS
 - 100Mbps Ethernet
- ❑ O servidor armazena e disponibiliza múltiplas cópias de áudio/vídeo



Antecipação no Cliente



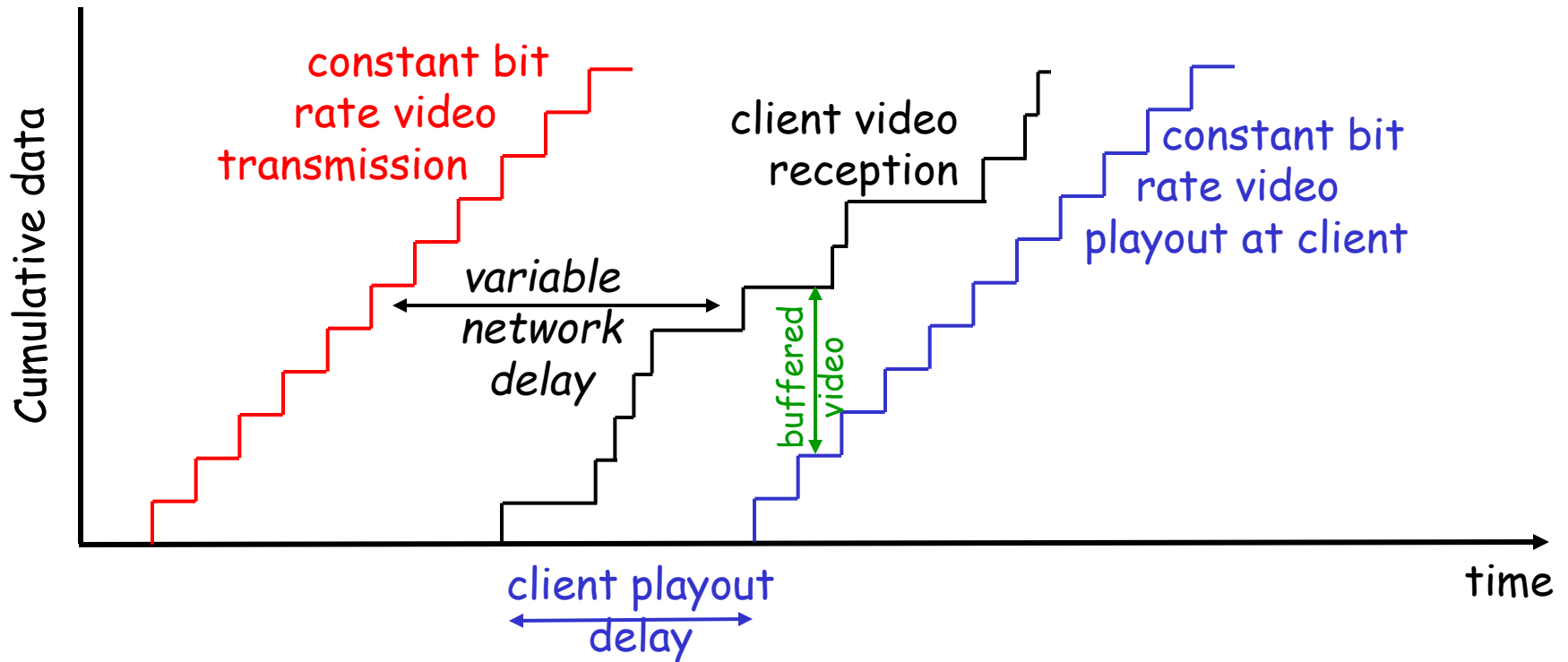
❑ Buffer no Cliente

- Armazena os dados à recepção
- Só começa a reprodução com parte do buffer cheio

❑ Permite margem de manobra adicional para compensar atrasos, percas e *jitter*

- Fornece uma taxa constante de recepção à aplicação
- Limitada à capacidade de armazenamento do buffer

Efeito do Buffer de Antecipação



- ❑ Transforma a taxa de recepção variável $x(t)$ numa taxa de entrega constante
- ❑ Dentro dos limites da capacidade do buffer



Protocolo de Controle de Streaming: RTSP

Real Time Streaming Protocol

HTTP

- ❑ Não foi previsto para transferir conteúdo multimédia
 - Mas de facto tem sido...
- ❑ Não tem comandos de controle
- ❑ Utiliza TCP por defeito

RTSP: RFC 2326 (1998)

- ❑ Protocolo de nível aplicativo
- ❑ Permite ao utilizador controlar o fluxo de dados multimedia
 - rewind, fast forward, pause, resume, repositioning, etc...

O que o RTSP não faz:

- ❑ Não define a forma como são encapsulados os dados de streaming
- ❑ Não restringe o protocolo de transporte
 - UDP ou TCP
- ❑ Não especifica como o media player gere o buffer de antecipação

<http://www.zvon.org/tmRFC/RFC2326/Output/index.html>



RTSP: controlo “out of band”

O FTP utiliza um canal de controlo “out-of-band”:

- ❑ Os ficheiros são transferidos numa conexão TCP
- ❑ Os comandos e as informações de controlo (directory changes, file deletion, file renaming, etc.) são transmitidas numa conexão separada
- ❑ Os canais “out-of-band” e “in-band” utilizam portos separados

As mensagens RTSP são enviadas “out-of-band”:

- ❑ As mensagens de controlo RTSP utilizam um número de porto diferente do *stream* de media
 - Porto 554
- ❑ O *stream* de media é considerado “in-band”.
- ❑ Geralmente é utilizado RTP para o *stream* de media.
 - Ver mais tarde.



Exemplo RTSP

Cenário:

- ❑ O browser recebe um Metafile com a descrição do *binding* do fluxo de media
- ❑ O browser lança o player e fornece-lhe o Metafile
- ❑ O player estabelece uma ligação RTSP de controlo e uma ligação de dados com o *streaming* server



Exemplo de Metafile

<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"

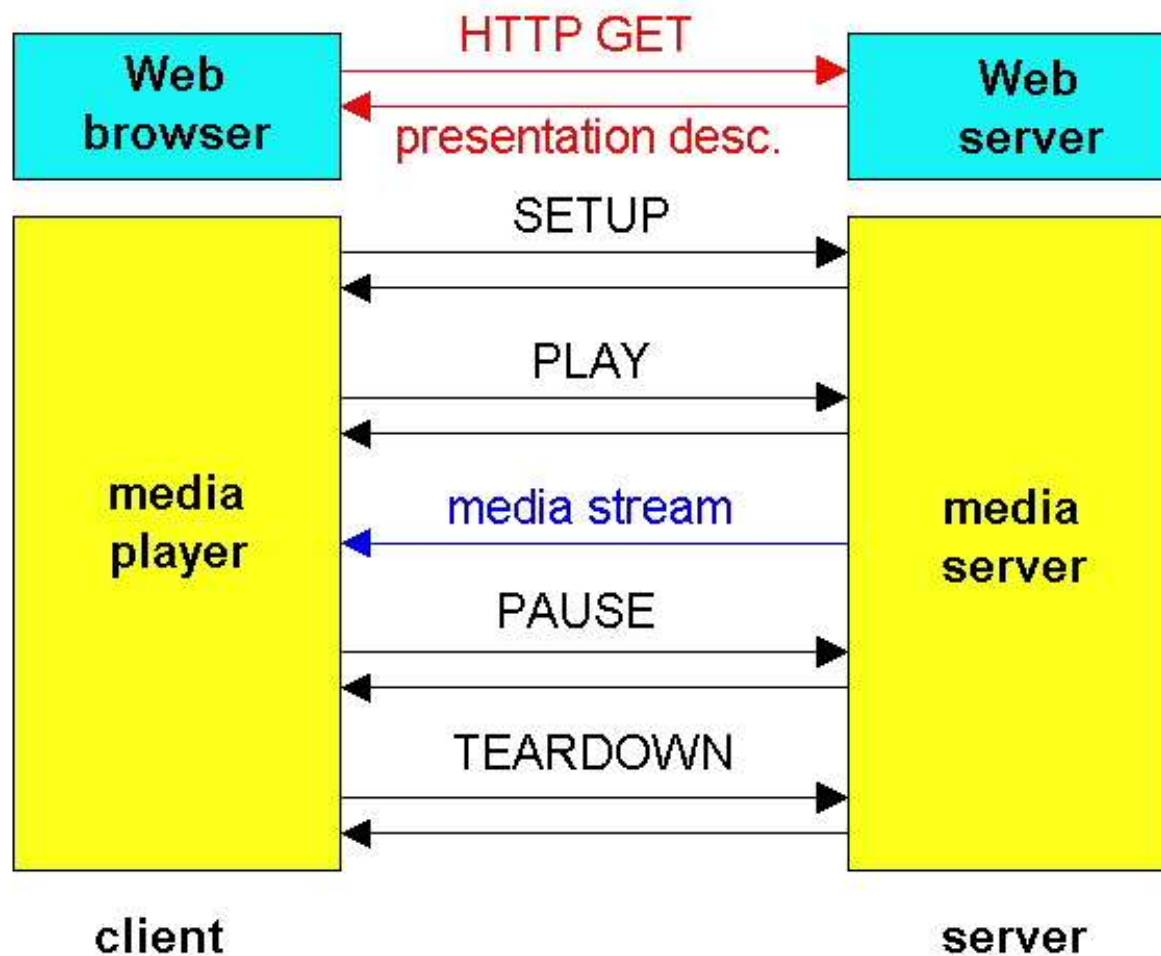
src="rtsp://video.example.com/twister/video">

</group>

</session>



Funcionamento do RTSP



Exemplo de diálogo 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



Captura de Diálogo RTSP (i)

No.	Time	Source	Destination	Protocol	Info
183	7.615906	10.0.0.1	194.65.5.146	RTSP	SETUP rtsp://a923.v77617.c7761.g.vm.akamaistream.net/7/923/7761/v0001/hollywood.download.ak
184	7.675696	194.65.5.146	10.0.0.1	RTSP	Reply: RTSP/1.0 200 OK
185	7.676310	10.0.0.1	194.65.5.146	RTSP	PLAY rtsp://a923.v77617.c7761.g.vm.akamaistream.net/7/923/7761/v0001/hollywood.download.aka
186	7.872287	194.65.5.146	10.0.0.1	RTSP	Reply: RTSP/1.0 200 OK
187	7.872819	10.0.0.1	194.65.5.146	RTSP	SET_PARAMETER rtsp://a923.v77617.c7761.g.vm.akamaistream.net/7/923/7761/v0001/hollywood.dow
188	7.888047	194.65.5.146	10.0.0.1	TCP	Interleaved channel 0x02, 1458 bytes[Unreassembled Packet]
189	7.901339	194.65.5.146	10.0.0.1	RTSP	Continuation
190	7.901391	10.0.0.1	194.65.5.146	TCP	2892 > 554 [ACK] Seq=3933 Ack=14687 Win=17424 Len=0
191	7.914655	194.65.5.146	10.0.0.1	RTSP	Continuation
192	7.927963	194.65.5.146	10.0.0.1	RTSP	Continuation
193	7.928012	10.0.0.1	194.65.5.146	TCP	2892 > 554 [ACK] Seq=3933 Ack=17591 Win=17424 Len=0
194	7.941058	194.65.5.146	10.0.0.1	RTSP	Continuation
195	7.954285	194.65.5.146	10.0.0.1	RTSP	Continuation
196	7.954330	10.0.0.1	194.65.5.146	TCP	2892 > 554 [ACK] Seq=3933 Ack=20495 Win=17424 Len=0
197	7.967644	194.65.5.146	10.0.0.1	RTSP	Continuation
198	7.980935	194.65.5.146	10.0.0.1	RTSP	Continuation
199	7.980977	10.0.0.1	194.65.5.146	TCP	2892 > 554 [ACK] Seq=3933 Ack=23399 Win=17424 Len=0
200	7.994848	194.65.5.146	10.0.0.1	RTSP	Continuation
201	8.007287	194.65.5.146	10.0.0.1	RTSP	Continuation
202	8.007336	10.0.0.1	194.65.5.146	TCP	2892 > 554 [ACK] Seq=3933 Ack=26303 Win=17424 Len=0
203	8.020584	194.65.5.146	10.0.0.1	RTSP	Continuation
204	8.023867	194.65.5.146	10.0.0.1	RTSP	Continuation

```

# Frame 181 (675 bytes on wire, 675 bytes captured)
# Ethernet II, Src: IntelCor_0c:98:50 (00:12:f0:0c:98:50), Dst: ThomsonT_28:0a:94 (00:0e:50:28:0a:94)
# Internet Protocol, Src: 10.0.0.1 (10.0.0.1), Dst: 194.65.5.146 (194.65.5.146)
# Transmission Control Protocol, Src Port: 2892 (2892), Dst Port: 554 (554), Seq: 1563, Ack: 9625, Len: 621
# Real Time Streaming Protocol
# SETUP rtsp://a923.v77617.c7761.g.vm.akamaistream.net/7/923/7761/v0001/hollywood.download.akamai.com/7287/windows/piratescaribbean3_dis_v9_t
  Method: SETUP
    URL: rtsp://a923.v77617.c7761.g.vm.akamaistream.net/7/923/7761/v0001/hollywood.download.akamai.com/7287/windows/piratescaribbean3_dis_v9_t
    User-Agent: WMPlayer/10.0.0.364 guid/3300AD50-2C39-46C0-AE0A-F8EAB16AFC77\r\n
    Accept-Charset: UTF-8, *,q=0.1\r\n
    X-Accept-Authentication: Negotiate, NTLM, Digest, Basic\r\n
    Accept-Language: pt, *,q=0.1\r\n
# Session: 11871978393235161326\r\n
  Session: 11871978393235161326
  CSeq: 4\r\n
  X-Playlist-Gen-Id: 363413\r\n
  Supported: com.microsoft.wm.srvppair, com.microsoft.wm.sswitch, com.microsoft.wm.eosmsg, com.microsoft.wm.startupprofile\r\n
  Transport: RTP/AVP/TCP;unicast;interleaved=2-3;ssrc=2cf5a9a9;mode=PLAY\r\n
\r\n

```

Captura de Diálogo RTSP (ii)

No.	Time	Source	Destination	Protocol	Info
187	7.872819	10.0.0.1	194.65.5.146	RTSP	SET_PARAMETER rtsp://a923.v77617.c7761.g.vm.akamaistream.net/7/923/7761/v0001/hollywood.dow
188	7.888047	194.65.5.146	10.0.0.1	TCP	Interleaved channel 0x02, 1458 bytes[Unreassembled Packet]
189	7.901339	194.65.5.146	10.0.0.1	RTSP	Continuation
190	7.901391	10.0.0.1	194.65.5.146	TCP	2892 > 554 [ACK] Seq=3933 Ack=14687 Win=17424 Len=0
191	7.914655	194.65.5.146	10.0.0.1	RTSP	Continuation
192	7.927963	194.65.5.146	10.0.0.1	RTSP	Continuation
193	7.928012	10.0.0.1	194.65.5.146	TCP	2892 > 554 [ACK] Seq=3933 Ack=17591 Win=17424 Len=0
194	7.941058	194.65.5.146	10.0.0.1	RTSP	Continuation
195	7.954285	194.65.5.146	10.0.0.1	RTSP	Continuation

Frame 187 (760 bytes on wire, 760 bytes captured)

Ethernet II, Src: IntelCor_0c:98:50 (00:12:f0:0c:98:50), Dst: ThomsonT_28:0a:94 (00:0e:50:28:0a:94)

Internet Protocol, Src: 10.0.0.1 (10.0.0.1), Dst: 194.65.5.146 (194.65.5.146)

Transmission Control Protocol, Src Port: 2892 (2892), Dst Port: 554 (554), Seq: 3227, Ack: 11783, Len: 706

Real Time Streaming Protocol

- SET_PARAMETER rtsp://a923.v77617.c7761.g.vm.akamaistream.net/7/923/7761/v0001/hollywood.download.akamai.com/7287/windows/piratescaribbean3_
Method: SET_PARAMETER
URL: rtsp://a923.v77617.c7761.g.vm.akamaistream.net/7/923/7761/v0001/hollywood.download.akamai.com/7287/windows/piratescaribbean3_dis_v9_
Content-Length: 220\r\nUser-Agent: WMPlayer/10.0.0.364 guid/3300AD50-2C39-46C0-AE0A-F8EAB16AFC77\r\nAccept-Charset: UTF-8, *;q=0.1\r\nX-Accept-Authentication: Negotiate, NTLM, Digest, Basic\r\nAccept-Language: pt, *;q=0.1\r\nSession: 11871978393235161326\r\nSession: 11871978393235161326
Content-Type: application/x-wms-Logconnectstats;charset=UTF-8\r\nCSeq: 7\r\n\r\nData (220 bytes)

01f0	73	2d	4c	6f	67	63	6f	6e	6e	65	63	74	73	74	61	74	...
0200	73	3b	63	68	61	72	73	65	74	3d	55	54	46	2d	38	0d	...
0210	0a	43	53	65	71	3a	20	37	0d	0a	0d	0a	3c	58	4d	4c	...
0220	3e	3c	53	75	6d	6d	61	72	79	3e	3c	2f	53	75	6d	6d	...
0230	61	72	79	3e	3c	63	2d	64	6e	73	3e	2d	3c	2f	63	2d	...
0240	64	6e	73	3e	3c	63	2d	69	70	3e	30	2e	30	2e	30	2e	...
0250	30	3c	2f	63	2d	69	70	3e	3c	63	2d	6f	73	3e	57	69	...
0260	6e	64	6f	77	73	5f	58	50	3c	2f	63	2d	6f	73	3e	3c	...
0270	63	2d	6f	73	76	65	72	73	69	6f	6e	3e	35	2e	31	2e	...
0280	30	2e	32	36	30	30	3c	2f	63	2d	6f	73	76	65	72	73	...
0290	69	6f	6e	3e	3c	64	61	74	65	3e	32	30	30	37	2d	30	...
02a0	35	2d	32	30	3c	2f	64	61	74	65	3e	3c	74	69	6d	65	...
02b0	3e	31	39	3a	30	34	3a	34	74	3c	2f	74	69	6d	65	3e	...
02c0	3c	63	2d	63	70	75	3e	50	65	6e	74	69	75	6d	3c	2f	...
02d0	63	2d	63	70	75	3e	3c	74	72	61	6e	73	70	6f	72	74	...
02e0	3e	54	43	50	3c	2f	74	72	61	6e	73	70	6f	72	74	3e	...
02f0	3c	2f	58	4d	4c	3e	0d	0a	...	...	...	...	...	...	...	...	...

```
application/x-wms-Logconnectstats
charset=UTF-8
CSeq: 7 ...<XML>
<Summary></Summary>
<cdns></cdns>
<ip>0.0.0.0</ip>
<os>Windows_XP</os>
<osversion>5.1.0.2600</osversion>
<date>2007-05-20</date>
<time>19:04:44</time>
<cpu>Pentium</cpu>
<transport>TCP</transport>
</XML>...
```

Metafile de Controle

Capítulo 3: Redes Multimédia

- ❑ Características das Aplicações Multimédia
 - Características
 - Necessidades
 - Compressão de Dados
- ❑ Fluxo de Áudio e Vídeo
 - Acesso Web
 - Servidores de Streaming
 - RSTP
- ❑ Multimédia em Tempo Real
 - Limitações do “Best Effort”
 - Características e necessidades do VoIP
- ❑ Protocolos para Aplicações Interactivas em Tempo Real
 - RTP / RTCP
 - SIP / H.323
- ❑ Qualidade de Serviço
 - Necessidades
 - Exemplos
- ❑ Implementação de QoS
 - Mecanismos de Escalonamento
 - Regulação de Serviço
- ❑ Arquitecturas de QoS
 - Serviços Integrados (Intserv) e Diferenciados (Diffserv)
- ❑ Protocolo de Sinalização: RSVP



Multimédia em Tempo Real

- ❑ Requisitos de uma aplicação multimédia interactiva
 - Caso da voz na Internet -> **Voice over IP (VoIP)**
- ❑ O tráfego é caracterizado pela **alternância** de **silêncios** e de **sequências** de palavras
 - A voz é codificada à cadência de 64 kbits/seg (8 KBytes/s)
- ❑ A aplicação VoIP só gera pacotes quando há som
 - Transmite pacotes de **160 bytes** todos os **20 ms** (50 p/s)
- ❑ Cada pacote leva um *header* aplicacional e é entregue ao protocolo de transporte
- ❑ Necessidade de minimizar o *overhead* de cada pacote
 - Utilização de um protocolo com header e latência mínimos
 - UDP é geralmente utilizado

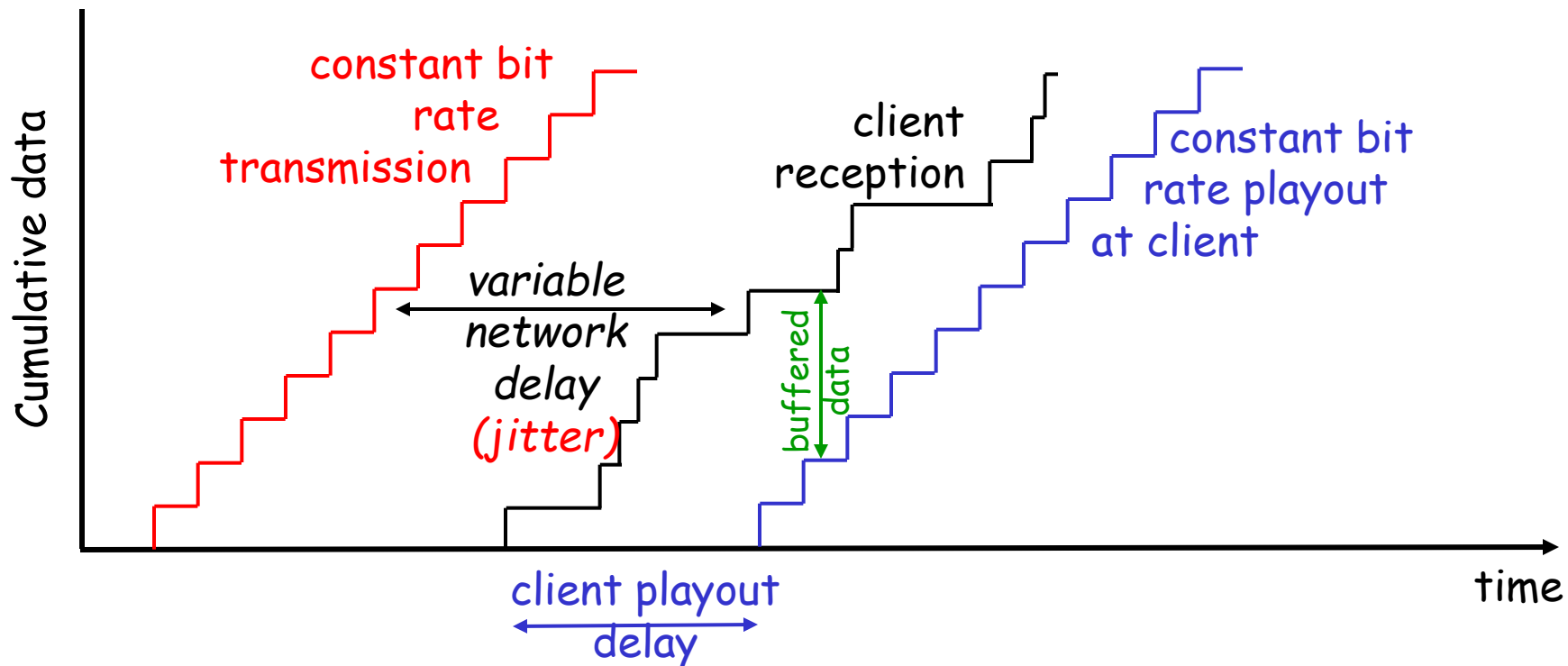


Limitações do “Best-Effort”: Percas

- ❑ **Percas na rede:** os datagramas podem ser descartados devido a problemas de congestão na rede (overflow dos buffers dos routers)
- ❑ **Percas devido a atraso:** os datagramas podem não se perder, mas chegar tarde demais para serem reproduzidos
 - Atraso máximo tolerado: 400 ms
- ❑ **Tolerância a percas**
 - Em função do tipo de codificação e do algoritmo de recuperação de percas, podem ser toleradas percas da ordem de 1 a 10%



Limitações do “Best-Effort”: Jitter



- ❑ A cadência de recepção de uma sequência de pacotes pode ter valores muito variáveis, maiores ou menores que 20 msec

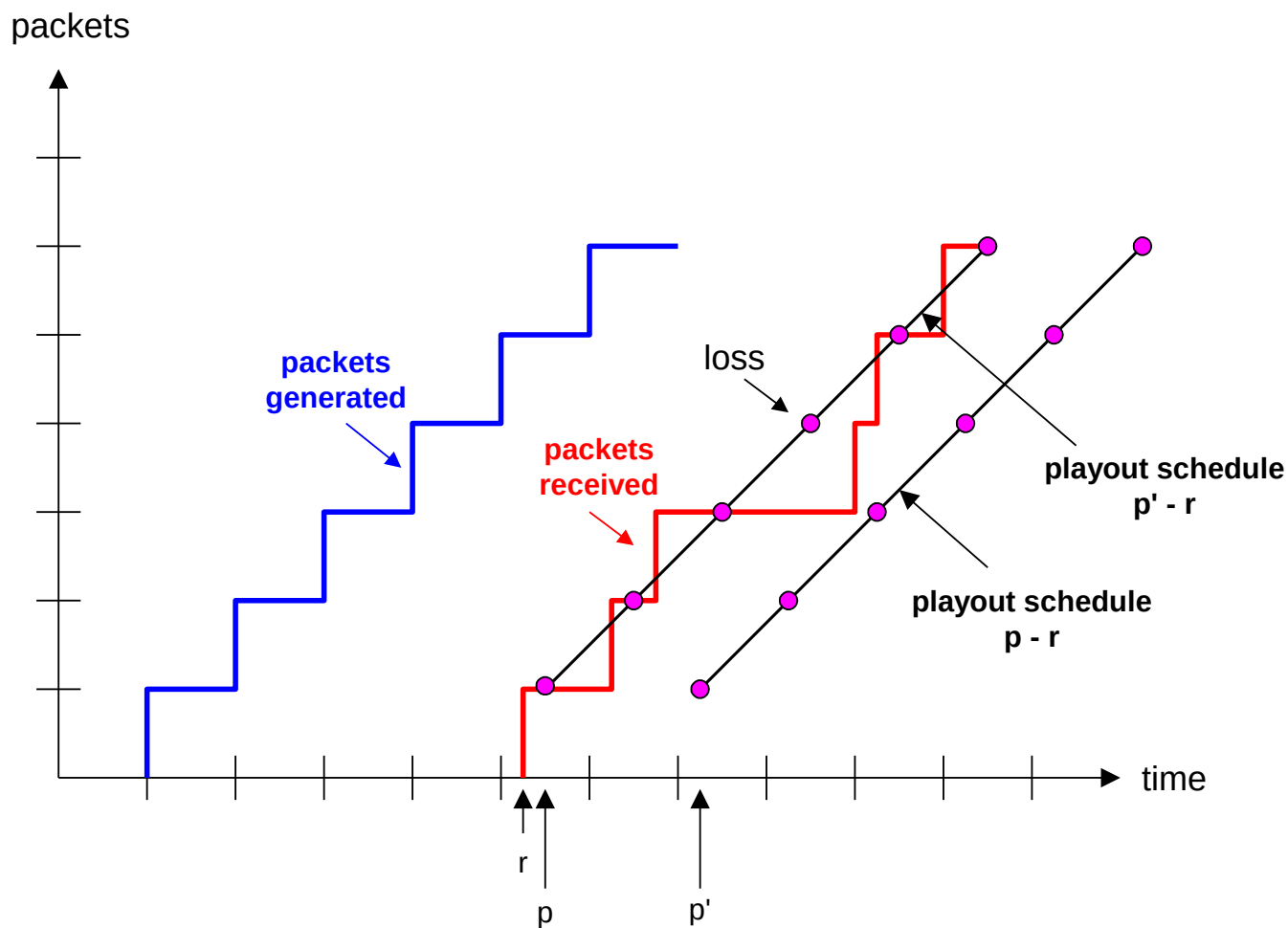


Supressão do *Jitter*: caso do VoIP

- ❑ Para que o som seja reproduzido fielmente na recepção, deve-se tentar manter uma taxa de reprodução fixa mesmo em presença de *jitter*
- ❑ São utilizados os seguintes mecanismos:
 - Cada pacote é numerado com um número de sequência que é incrementado para cada pacote
 - Cada pacote é marcado com uma estampilha temporal que reflecte o momento em que foi gerado
 - O receptor atrasa a reprodução de pacotes de forma a armazenar uma quantidade de pacotes suficiente para manter a cadência
 - O atraso de reprodução pode ser **fixo** ou **variável**



Diferentes Atrasos de Reprodução



- ❑ Primeiro pacote chega no instante r
- ❑ Reprodução no instante p : perca de um pacote
- ❑ Reprodução no instante p' : não há percas

Atraso Fixo

- ❑ Atraso Fixo: é estipulado um atraso de n milissegundos, que depende das condições da rede
- ❑ O receptor vai tentar reproduzir o pacote recebido exactamente n ms depois de ter sido gerado
 - Se o pacote foi gerado no instante t , é reproduzido em $t + n$
 - Se o pacote chegar depois de $t + n$, é considerado perdido
- ❑ Valores de n :
 - n grande ($\sim 400\text{ms}$): menos perda de pacotes
 - n pequeno ($\sim 150\text{ms}$): melhor interactividade



Adaptação do Atraso (i)

O atraso pode ser adaptado às condições da rede:

- ❑ Objectivo: minimizar o atraso de reprodução, mantendo a taxa de rejeição baixa
- ❑ Método: mecanismo de ajustamento de atraso:
 - Estimar o atraso da rede e ajustar o atraso de reprodução no início de cada sequência de pacotes de voz
 - Alongar ou encurtar os períodos de silêncio para mascarar o atraso
- ❑ Para isso, definem-se as variáveis seguintes:

t_i = estampilha temporal do pacote de ordem i

r_i = instante de recepção do pacote de ordem i

p_i = instante de reprodução do pacote de ordem i

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

d_i = estimativa da média do atraso da rede depois da recepção do pacote i



Adaptação do Atraso (ii)

- Estimativa dinâmica da média do atraso no receptor (cf. TTL no TCP):

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

- u é uma constante positiva inferior a 1

- Desvio médio do atraso:

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

- Os valores d_i e v_i são calculados para cada pacote recebido mas só são aplicados para a reprodução da próxima sequência.
- Para o 1º pacote dessa sequência o atraso estimado será assim:

$$p_i = t_i + d_i + Kv_i$$

- Onde K é uma constante positiva, que garante um intervalo de guarda
- Os restantes pacotes da sequência são reproduzidos com o mesmo atraso



Adaptação do Atraso (iii)

O algoritmo é **válido** desde que se consiga determinar correctamente o **primeiro pacote** de uma sequência

- ❑ A determinação do início de uma sequência pode ser feita examinando as estampilhas e os números dos pacotes sucessivos
- ❑ Se a diferença for superior a 20 ms (50 pct/s):
 - **Se** a sequência de numeração não tiver falhas
=> começo de uma nova sequência, adapta-se o valor do atraso
 - **Se** a sequência apresentar falhas de numeração
=> mesma sequência da qual alguns pacotes foram perdidos, o atraso mantém-se



Recuperação de Perdas de Pacotes

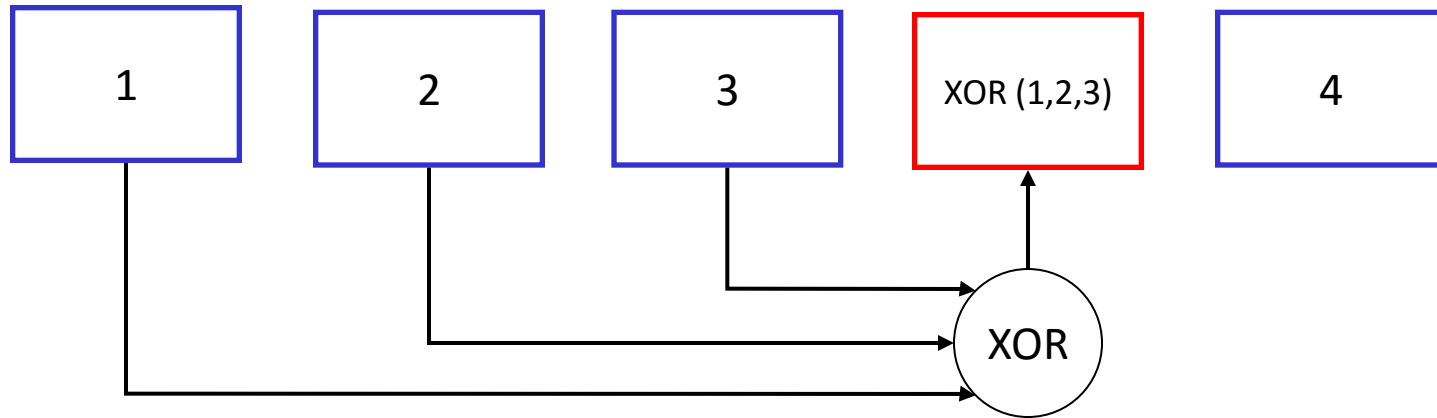
Forward Error Correction (FEC):

- ❑ Acrescentar informação redundante de forma a poder regenerar pacotes em caso de percas
- ❑ **Primeiro mecanismo:**
 - Para cada grupo de n pacotes, envia-se um pacote redundante que é o OU exclusivo (xor) dos pacotes dessa sequência
 - Em caso de perda de **um** pacote dos n , é possível reconstruí-lo
 - Um pacote adicional aumenta a taxa de transmissão de $1/n$
 - Se n for pequeno, aumenta-se a probabilidade de se conseguir reconstruir um pacote perdido
 - Mas maior é o aumento relativo da taxa de transmissão
 - Ex: $n = 3$, aumento de 0.333
 - Ex: $n = 10$, aumento de 0.1
 - O atraso de reprodução deve ser ajustado para permitir a recepção dos $n+1$ pacotes

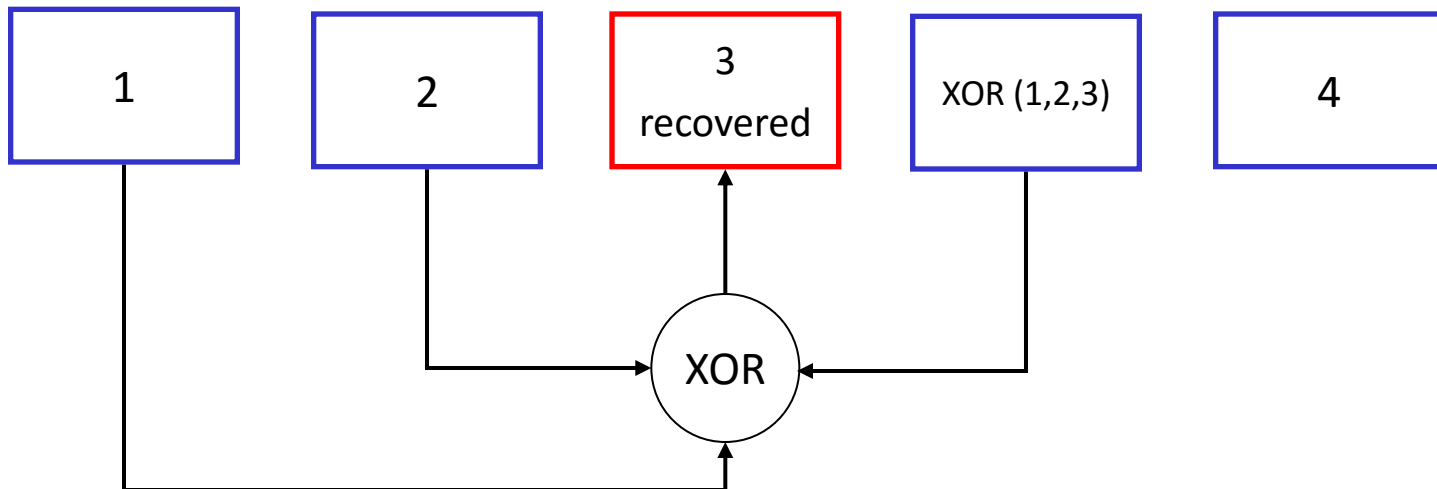


Recuperação por OU Exclusivo

Emissão



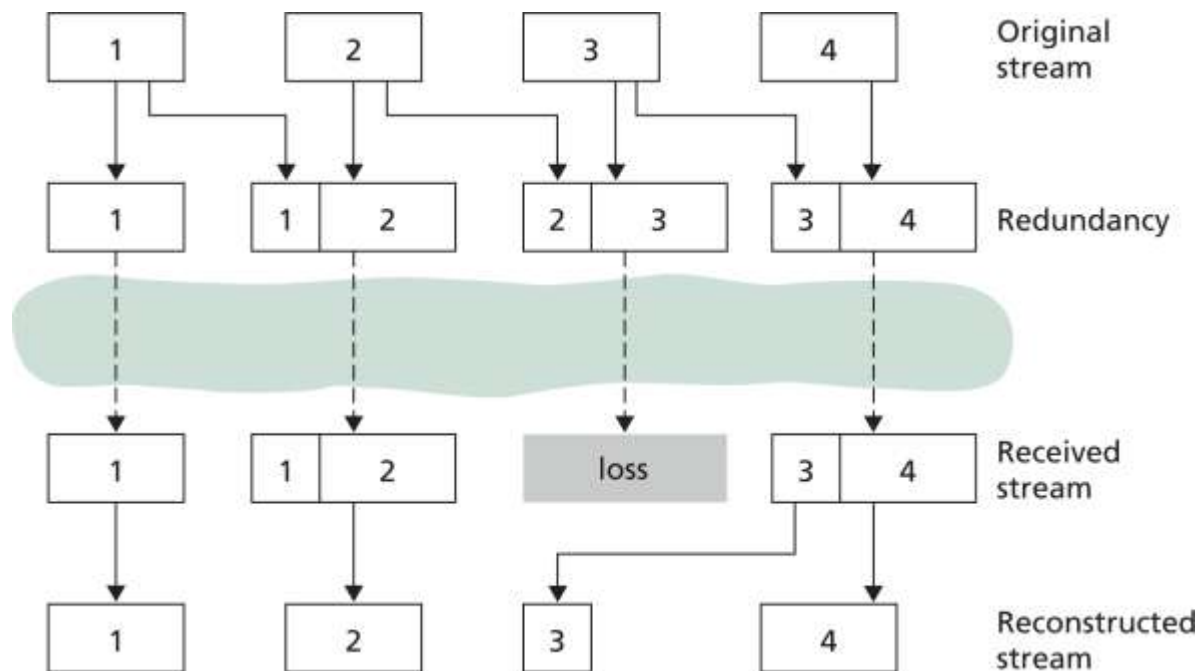
Recepção



Recuperação por Entrelaçamento

Segundo Mecanismo

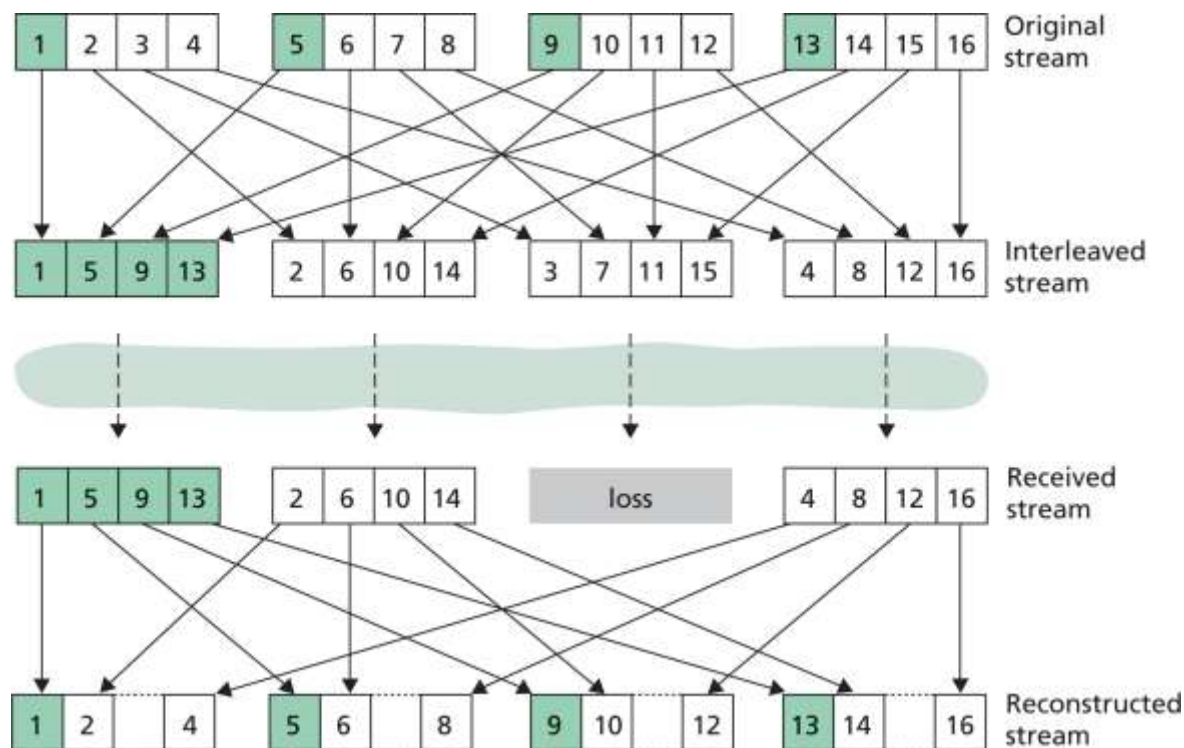
- ❑ Entrelaçar pacotes redundantes de mais baixa resolução
 - Pacote $n-1$ com o pacote n
- ❑ Por exemplo se o fluxo nominal for PCM de 64 kbs enviam-se pacotes com um fluxo redundante de GSM a 13 kbps
- ❑ Em caso de percas de pacotes não consecutivos é possível recuperar a sequência com os pacotes de baixa qualidade
- ❑ É possível adicionar também o pacote de ordem $n-2$



Recuperação por Dispersão

Terceiro Mecanismo

- ❑ Os pacotes são divididos em fragmentos mais pequenos
 - Por exemplo 4 pacotes de 5 ms ($4 \times 5 = 20$)
- ❑ Os fragmentos são “baralhados” e inseridos em pacotes diferentes
- ❑ Se um pacote se perde, pode-se reconstruir a partir dos outros fragmentos
- ❑ Não introduz *overhead* de dados, mas o atraso de reprodução tem de ser adaptado para permitir reconstruir a ordem original



Resumo: Técnicas usadas no VoIP

- ❑ Utilização de UDP para evitar controle de congestão do protocolo em tráfego com requisitos temporais
- ❑ Atraso de reprodução adaptável para compensar os atrasos
- ❑ Recuperação de Perdas
 - Redundância
 - Pacotes entrelaçados e dispersos
- ❑ Utilização de técnicas de recuperação de pacotes baseadas em interpolação e substituição



Capítulo 3: Redes Multimédia

- ❑ Características das Aplicações Multimédia
 - Características
 - Necessidades
 - Compressão de Dados
- ❑ Fluxo de Áudio e Vídeo
 - Acesso Web
 - Servidores de Streaming
 - RSTP
- ❑ Multimédia em Tempo Real
 - Limitações do “Best Effort”
 - Características e necessidades do VoIP
- ❑ Protocolos para Aplicações Interactivas em Tempo Real
 - RTP / RTCP
 - SIP / H.323
- ❑ Qualidade de Serviço
 - Necessidades
 - Exemplos
- ❑ Implementação de QoS
 - Mecanismos de Escalonamento
 - Regulação de Serviço
- ❑ Arquitecturas de QoS
 - Serviços Integrados (Intserv) e Diferenciados (Diffserv)
- ❑ Protocolo de Sinalização: RSVP



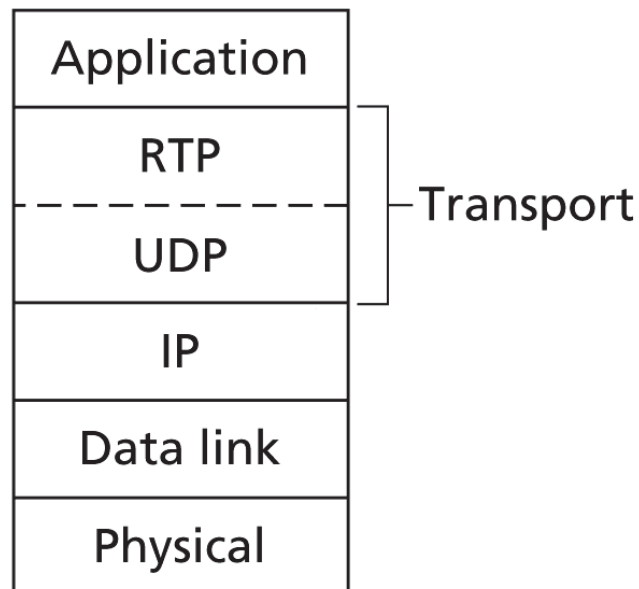
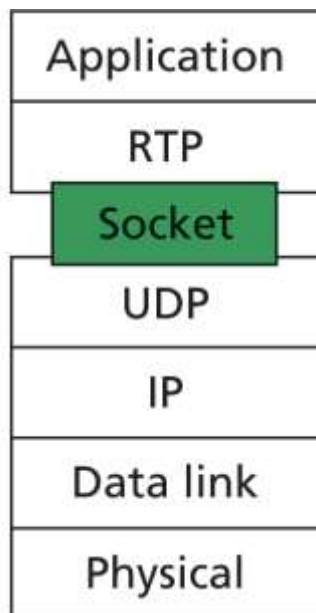
Real-Time Protocol (RTP)

- ❑ Permite a **interoperabilidade** entre aplicações multimédia, fornecendo mecanismos standard para implementar as funcionalidades vistas no ponto anterior
- ❑ Especificado pelos RFCs 1889 de 1996 e 3550 de 2003
 - <http://rfc-ref.org/RFC-TEXTS/3550/index.html>
- ❑ O RTP especifica a **estrutura dos pacotes** para transporte de áudio e vídeo. Um pacote RTP fornece:
 - Identificação dos dados
 - Numeração da sequência de pacotes
 - Estampilhas temporais
- ❑ O RTP é implementado no nível aplicacional e encapsulado em pacotes UDP
- ❑ Existe uma extensão denominada **Secure RTP**, que garante confidencialidade e defesa contra repetições (*replay*)
 - <http://rfc-ref.org/RFC-TEXTS/3711/index.html>



O RTP e a Pilha Protocolar

- ❑ O RTP fornece uma camada funcional que estende a camada de transporte, neste caso UDP
- ❑ O acesso ao transporte é feito dentro da aplicação através da API socket
- ❑ Conceptualmente, pode-se considerar o RTP como um nível adicional da camada de transporte
- ❑ Existem bibliotecas em C e Java que implementam RTP
 - librtsp e Java Media Framework



Exemplo de Pacote RTP de tráfego VoIP

No. -	Time	Source	Destination	Protocol	Info
16	2.106345	10.0.0.2	10.0.0.4	SIP/SD	Request: INVITE sip:jrogado@10.0.0.4, with session descrip
17	2.110152	10.0.0.4	10.0.0.2	SIP	Status: 100 trying -- your call is important to us
18	2.119801	10.0.0.4	10.0.0.2	SIP	Status: 180 Ringing
340	8.621400	10.0.0.4	10.0.0.2	SIP/SD	Status: 200 OK, with session description
342	8.627389	10.0.0.2	10.0.0.4	RTCP	Receiver Report [Malformed Packet]
343	8.641867	10.0.0.4	10.0.0.2	RTCP	Receiver Report
344	8.649690	10.0.0.4	10.0.0.2	RTP	Payload type=ITU-T G.711 PCMU, SSRC=1295595096, Seq=2568,
345	8.663553	10.0.0.2	10.0.0.4	RTP	Payload type=ITU-T G.711 PCMU, SSRC=1352170915, Seq=7280,
353	8.692382	10.0.0.2	10.0.0.4	RTP	Payload type=ITU-T G.711 PCMU, SSRC=1352170915, Seq=7281,
354	8.712919	10.0.0.2	10.0.0.4	RTP	Payload type=ITU-T G.711 PCMU, SSRC=1352170915, Seq=7282,
357	8.732419	10.0.0.2	10.0.0.4	RTP	Payload type=ITU-T G.711 PCMU, SSRC=1352170915, Seq=7283,
358	8.752896	10.0.0.2	10.0.0.4	RTP	Payload type=ITU-T G.711 PCMU, SSRC=1352170915, Seq=7284,
359	8.772407	10.0.0.2	10.0.0.4	RTP	Payload type=ITU-T G.711 PCMU, SSRC=1352170915, Seq=7285,
360	8.792908	10.0.0.2	10.0.0.4	RTP	Payload type=ITU-T G.711 PCMU, SSRC=1352170915, Seq=7286,

Frame 344 (214 bytes on wire, 214 bytes captured)

- Ethernet II, Src: D-Link_52:3c:4f (00:0d:88:52:3c:4f), Dst: IntelCor_0c:98:50 (00:12:f0:0c:98:50)
- Internet Protocol, Src: 10.0.0.4 (10.0.0.4), Dst: 10.0.0.2 (10.0.0.2)
- User Datagram Protocol, Src Port: 8000 (8000), Dst Port: 60294 (60294)
 - Source port: 8000 (8000)
 - Destination port: 60294 (60294)
 - Length: 180
 - Checksum: 0xbda2 [correct]
- Real-Time Transport Protocol
 - [Stream setup by SDP (frame 16)]
 - 10.. = Version: RFC 1889 Version (2)
 - ..0. = Padding: False
 - ...0 = Extension: False
 - 0000 = Contributing source identifiers count: 0
 - 1... = Marker: True
 - Payload type: ITU-T G.711 PCMU (0)
 - Sequence number: 2568
 - Timestamp: 1585755512
 - Synchronization Source identifier: 1295595096
 - Payload: FF...

Annotations:

- A aplicação encapsula os dados num pacote RTP que é transportado por UDP
- Os dados são enviados em pacotes de 160 bytes todos os 20 ms
- O pacote RTP contém informação sobre o tipo de codificação utilizado e parâmetros de sincronização



RTP e Qualidade de Serviço

- ❑ O RTP **não** fornece nenhum mecanismo para garantir a entrega dentro de limites temporais nem garantias de qualidade de serviço
- ❑ O encapsulamento RTP só é detectado pelos sistemas finais e não é visto pelos *routers* intermédios
 - Estes fornecem serviço de melhor esforço e não fazem nada de especial para garantir que os pacotes RTP chegam ao destino dentro de limites temporais

Ver o FAQ: <http://www.cs.columbia.edu/~hgs/rtp/faq.html>



Header RTP (i)



RTP Header

- ❑ **Payload Type (7 bits):** Indica o tipo de codificação em uso nos dados transportados. Se o emissor muda a codificação a meio do fluxo, informa o receptor através do conteúdo do payload type.
 - Payload type 0: PCM mu-law, 64 kbps
 - Payload type 3, GSM, 13 kbps
 - Payload type 7, LPC, 2.4 kbps
 - Payload type 26, Motion JPEG
 - Payload type 31. H.261
 - Payload type 33, MPEG2 video
- ❑ **Sequence Number (16 bits):** É incrementado de um em cada pacote RTP enviado e é utilizado para detectar pacotes perdidos e restaurar a sequência original.



Header RTP (ii)



RTP Header

- ❑ **Timestamp field (32 bits).** Contém o instante de amostragem do primeiro byte do pacote RTP.
 - Em áudio, o timestamp incrementa de uma unidade para cada período de amostragem
 - i.e. cada 125 µsecs numa amostragem a 8 KHz
 - Se a aplicação gera pacotes de 160 amostras codificadas, o timestamp incrementa desse valor em cada pacote RTP
 - O timestamp continua a incrementar mesmo quando o emissor está inactivo, de forma a manter um “conceito” de tempo real
- ❑ **SSRC field (32 bits).** Identifica a origem do fluxo RTP. Cada fluxo numa sessão RTP tem um SSRC distinto
- ❑ Para uma descrição detalhada do header RTP ver
 - <http://www.networksorcery.com/enp/protocol/rtp.htm>



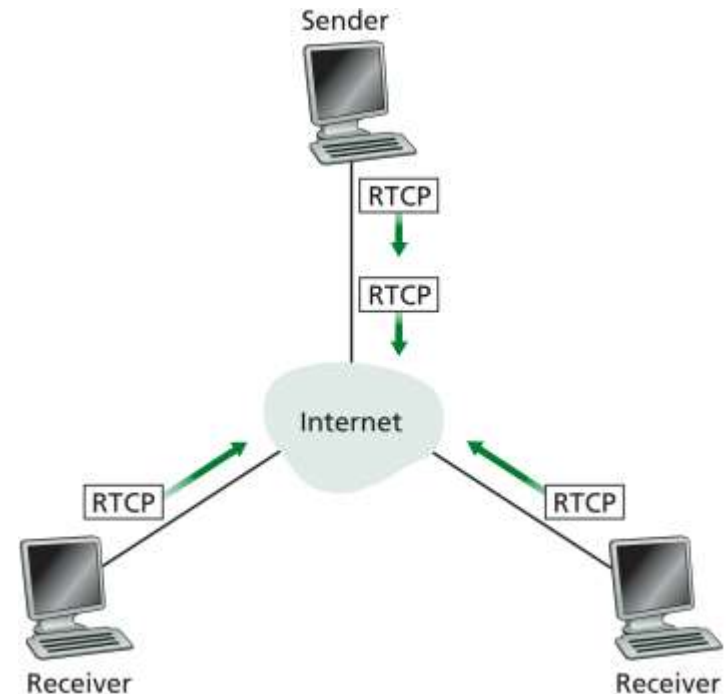
Real-Time Control Protocol (RTCP)

- ❑ Protocolo que permite o controlo do RTP, definido nos mesmos RFCs 1889 e 3550
 - Funciona como complemento do RTP
- ❑ O RTCP permite enviar relatórios de emissão e/ou recepção
 - Reporta informação estatística necessária e/ou útil para o desempenho da aplicação
 - Inclui o número de pacotes enviados, recebidos, perdidos, jitter, delay, etc..
- ❑ O feedback recebido é utilizado para controlar o desempenho do streaming em curso
 - O emissor pode modificar o seu comportamento baseado na informação recebida (p.ex.: mudança de codec)
- ❑ Cada participante de uma sessão RTP transmite periodicamente pacotes de controle RTCP a todos os outros participantes através de multicast



Sessões e Multicast de Pacotes

- ❑ Para uma sessão RTP existe um endereço de multicast (Classe D) para onde são enviados todos os pacotes de RTCP dessa sessão
 - Útil em sessões de Webcast
- ❑ Pacotes RTP e RTCP são distinguidos pelo número de porto (RTCP = RTP+1)
- ❑ Para limitar o tráfego RTCP , cada participante reduz o número de pacotes enviados em função do número de participantes
 - < 5% tráfego total
- ❑ Vários *reports* podem ser agregados num único pacote RTCP



Principais Tipos e Formato dos Pacotes RTCP

Receiver Report:

- ❑ Fracção de pacotes perdidos, último número de sequência, intervalo de jitter médio.

Sender Report:

- ❑ SSRC do stream RTP
- ❑ Timestamps e data corrente
- ❑ Número de pacotes e de bytes enviados no stream

Formato do Pacote RTCP (32 bits):

2	3	8	16 bit
Version	P	RC	Packet type
Length			

- ♦ Version - Identifies the RTP version which is the same in RTCP packets as in RTP data packets. The version defined by this specification is two (2).
- ♦ P - Padding. When set, this RTCP packet contains some additional padding octets at the end which are not part of the control information. The last octet of the padding is a count of how many padding octets should be ignored. Padding may be needed by some encryption algorithms with fixed block sizes. In a compound RTCP packet, padding should only be required on the last individual packet because the compound packet is encrypted as a whole.
- ♦ RC- Reception report count, the number of reception report blocks contained in this packet. A value of zero is valid. Packet type Contains the constant 200 to identify this as an RTCP SR packet.
- ♦ Length - The length of this RTCP packet in 32-bit words minus one, including the header and any padding. (The offset of one makes zero a valid length and avoids a possible infinite loop in scanning a compound RTCP packet, while counting 32-bit words avoids a validity check for a multiple of 4.)

Source Description:

- ❑ De cada vez que um emissor gera um novo fluxo RTP, envia um conjunto de pacotes RTCP com a sua descrição
- ❑ Endereço nome, email do emissor, hostname e SSRC do stream RTP associado
- ❑ Tipo de aplicação que está a gerar o stream
- ❑ Permite criar uma associação entre o SSRC e a fonte do stream

End of Participation:

- ❑ Permite ao emissor enviar uma mensagem (BYE) para terminar o fluxo

Exemplo de Pacote RTCP numa sessão VoIP

No. -	Time	Source	Destination	Protocol	Info
343	0.014478	10.0.0.4	10.0.0.2	RTCP	Receiver Report
344	0.007823	10.0.0.4	10.0.0.2	RTP	Payload type=ITU-T G.711 PCMU, SSRC=1295595
345	0.013863	10.0.0.2	10.0.0.4	RTP	Payload type=ITU-T G.711 PCMU, SSRC=1352170
353	0.028829	10.0.0.2	10.0.0.4	RTP	Payload type=ITU-T G.711 PCMU, SSRC=1352170
354	0.020537	10.0.0.2	10.0.0.4	RTP	Payload type=ITU-T G.711 PCMU, SSRC=1352170
357	0.019500	10.0.0.2	10.0.0.4	RTP	Payload type=ITU-T G.711 PCMU, SSRC=1352170
358	0.020477	10.0.0.2	10.0.0.4	RTP	Payload type=ITU-T G.711 PCMU, SSRC=1352170
359	0.019511	10.0.0.2	10.0.0.4	RTP	Payload type=ITU-T G.711 PCMU, SSRC=1352170
360	0.020501	10.0.0.2	10.0.0.4	RTP	Payload type=ITU-T G.711 PCMU, SSRC=1352170
361	0.019517	10.0.0.2	10.0.0.4	RTP	Payload type=ITU-T G.711 PCMU, SSRC=1352170

< [] >

⊕ Frame 343 (94 bytes on wire, 94 bytes captured)

⊕ Ethernet II, Src: D-Link_52:3c:4f (00:0d:88:52:3c:4f), Dst: IntelCor_0c:98:50 (00:12:f0:0c:98:50)

⊕ Internet Protocol, Src: 10.0.0.4 (10.0.0.4), Dst: 10.0.0.2 (10.0.0.2)

⊕ User Datagram Protocol, Src Port: 8001 (8001), Dst Port: 60295 (60295)

⊖ Real-time Transport Control Protocol

- ⊖ [Stream setup by SDP (frame 16)]
 - [Setup frame: 16]
 - [Setup Method: SDP]
 - 10.. = Version: RFC 1889 Version (2)
 - ..0. = Padding: False
 - ...0 0001 = Reception report count: 1
 - Packet type: Receiver Report (201)
 - Length: 7
 - Sender SSRC: 1295595096
- ⊖ Source 1
 - Identifier: 1352170915
- ⊖ SSRC contents
 - Fraction lost: 0 / 256
 - Cumulative number of packets lost: 0
- ⊖ Extended highest sequence number received: 0
 - Sequence number cycles count: 0
 - Highest sequence number received: 0
 - Interarrival jitter: 0
 - Last SR timestamp: 0
 - Delay since last SR timestamp: 0

Sincronização de Streams

O RTCP permite sincronizar diferentes streams de media numa sessão RTP

- ❑ No caso de uma aplicação de videoconferência na qual cada emissor gera fluxos separados de áudio e vídeo
- ❑ Os timestamps nos pacotes RTP são valores relativos relacionados com as cadências de amostragem de áudio e vídeo
 - Não estão correlacionadas com o tempo real (data)
- ❑ Os pacotes RTCP com reports do emissor contêm, para cada fluxo RTP :
 - Timestamps de amostragem do último pacote gerado
 - Data tempo real do instante de geração do pacote
- ❑ A partir desta associação, os receptores podem sincronizar a reprodução de áudio e vídeo
 - lip sync



Limitação de Largura de Banda RTCP

O RTCP tenta manter o seu tráfego dentro do limite de 5% da largura de banda da sessão RTP associada

Exemplo

- ❑ Supondo que um emissor gera um fluxo de vídeo a 2 Mbps:
 - o RTCP tenta manter o seu tráfego abaixo dos 100 Kbps
- ❑ O RTCP atribui 75% da sua banda a todos os receptores e os restantes 25% ao emissor
- ❑ Os 75% são igualmente distribuídos por todos os receptores
 - Com R receptores, cada um limita a sua taxa a $75/R$ kbps
 - O emissor pode utilizar até 25%
- ❑ Cada participante determina o período de emissão de pacotes RTCP calculando o tamanho médio dos pacotes enviados na sessão e dividindo pela taxa que lhe está alocada



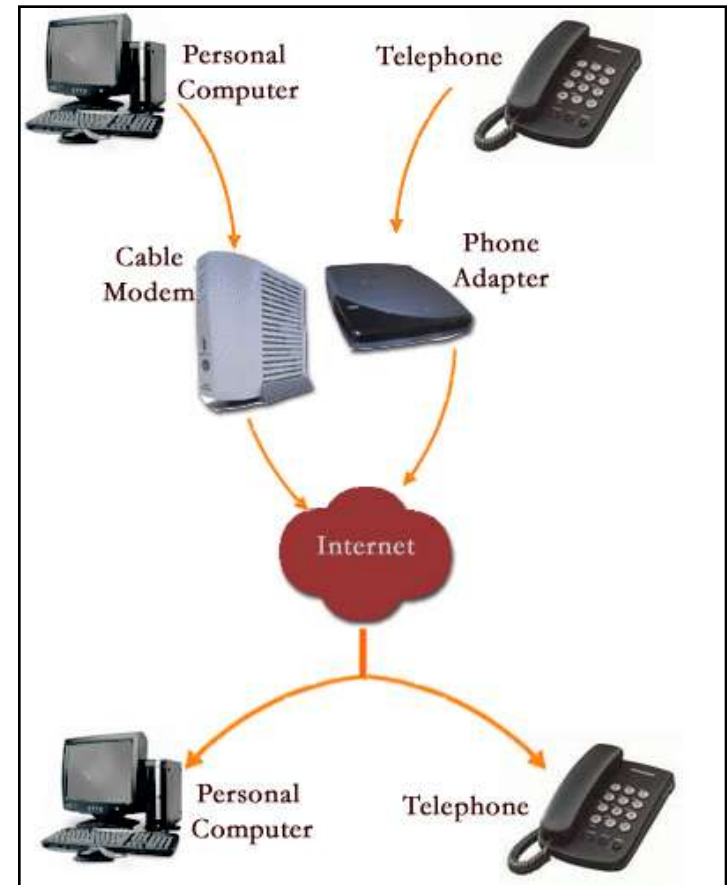
Voice Over IP (VoIP)

❑ Realização de chamadas telefônicas sobre IP

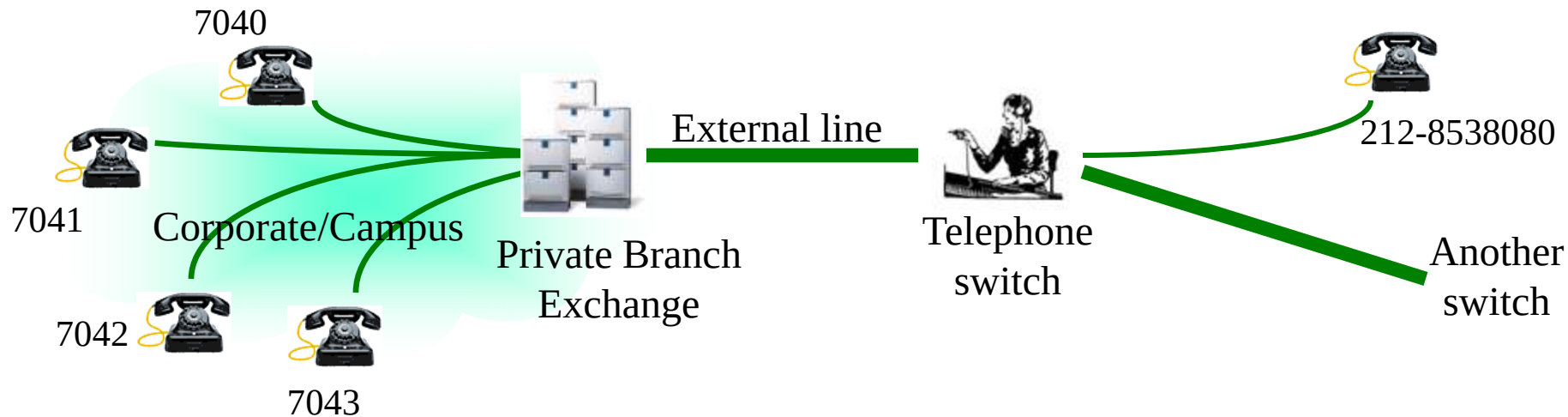
- Computador a computador
- Telefone a computador
- Telefone a telefone

❑ Motivações para o VoIP

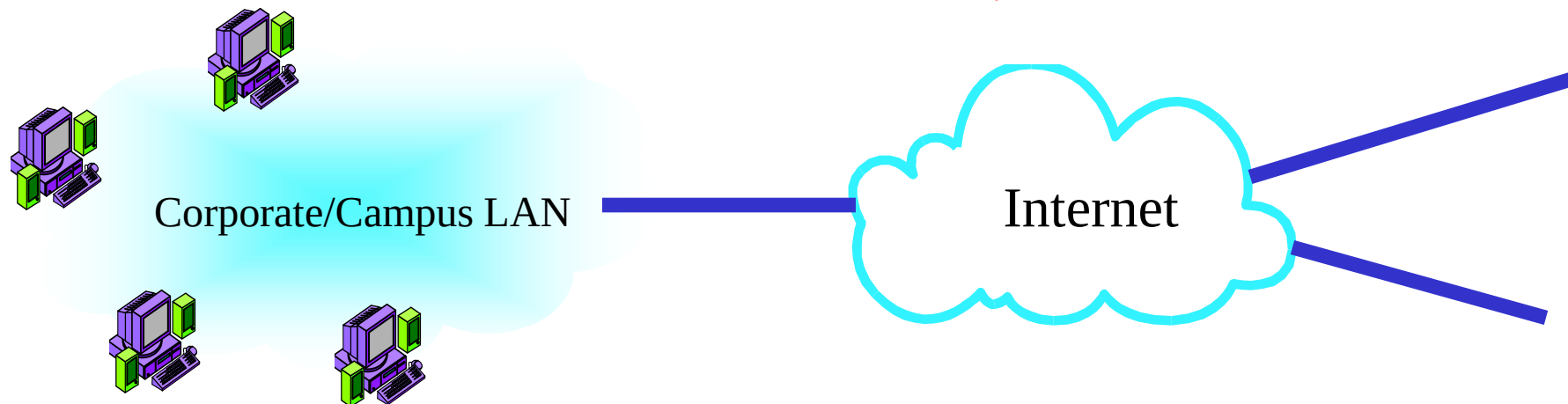
- Redução de custos
- Simplicidade
- Aplicações Avançadas
 - Call centers Web
 - Integração Voicemail e e-mail
- Aplicações colaborativas
 - Encaminhar, localizar, ...



Infra-estruturas Tradicionais

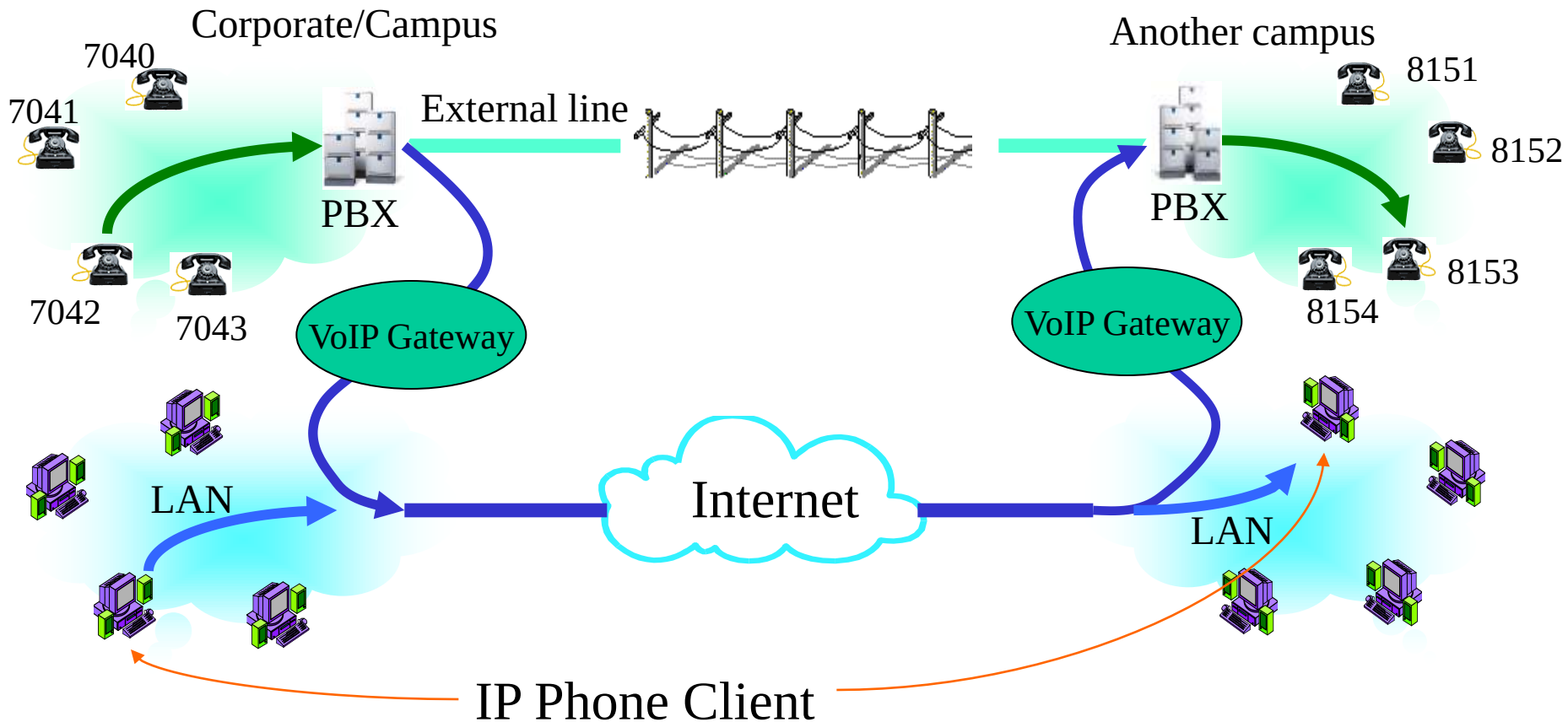


PSTN e Rede de Dados separadas



VoIP: Integração Dados/Voz

Integração das duas Redes



Protocolos do VoIP

O VoIP assenta essencialmente em 3 protocolos:

- ❑ **SIP** - Session Initiation Protocol para estabelecer a sessão correspondente a uma chamada
 - Protocolo de sinalização especificado pelo IETF
 - Ver adiante
- ❑ **RTP/RTCP** para a transferência do fluxo de voz
 - Ponto anterior
- ❑ **H.323** - família de protocolos completo de sinalização e streaming, especificado pelo ITU-T destinado a vídeo chamada e videoconferência
 - Favorece a integração com as redes telefónicas existentes



SIP - Session Initiation Protocol

- ❑ Definido pelo IETF RFC 3261 em 2002

Visão do SIP a longo termo

- ❑ Todas as chamadas telefónicas de voz e de vídeo, assim como a vídeo-conferência serão realizadas através da Internet
- ❑ Os intervenientes são identificados pelos endereços de e-mail, em vez de números de telefone
- ❑ O destinatário pode ser contactado onde quer que se encontre, qualquer que seja o dispositivo IP que esteja a utilizar



Serviços fornecidos pelo SIP

❑ Estabelecimento de Chamadas

- Alerta de um correspondente que está a ser chamado
- Acordo sobre o tipo de media e método de codificação
- Finalização de chamada

❑ Determinação do IP do destinatário

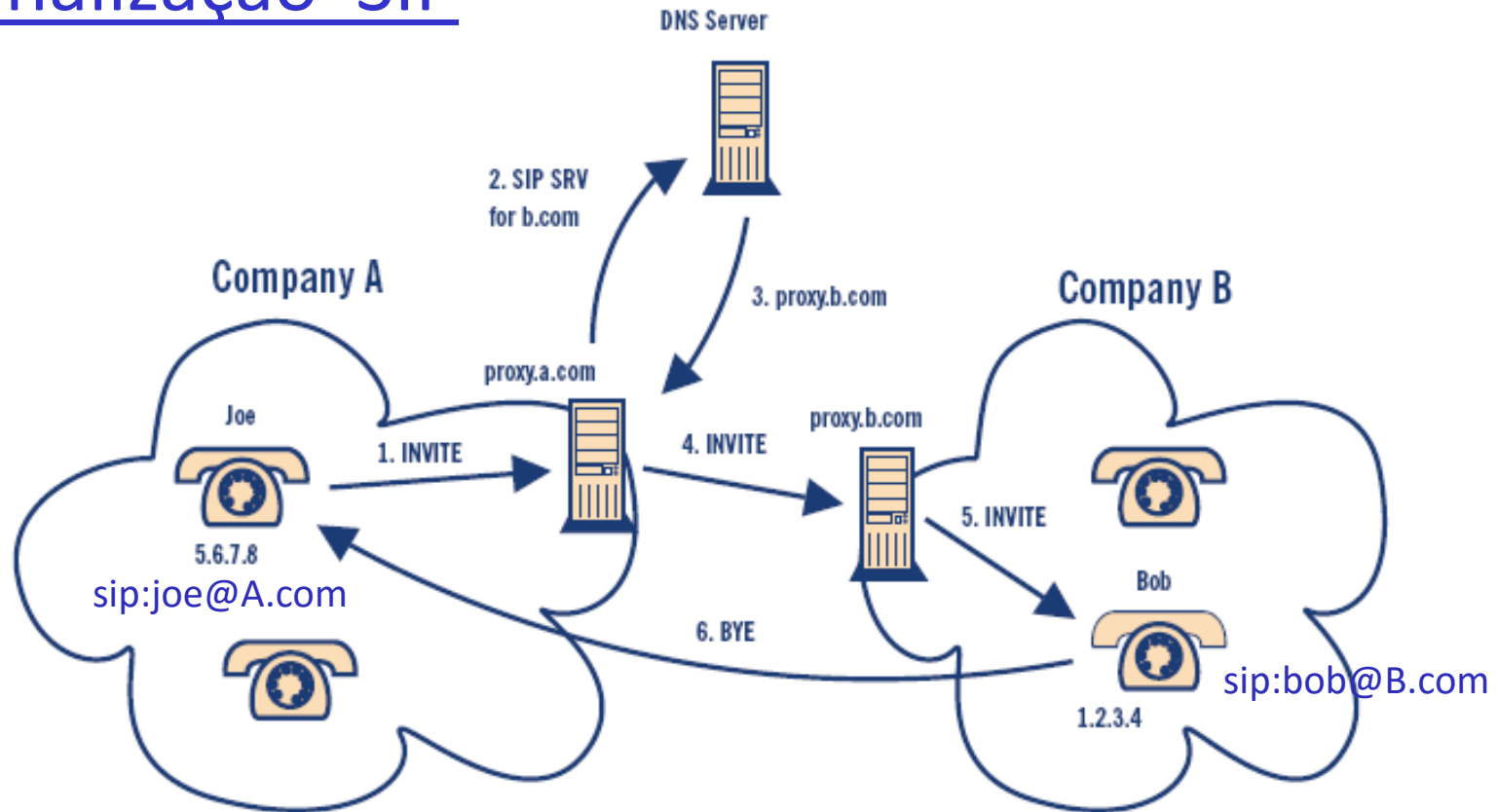
- Mapeamento de mnemónicas para endereços IP

❑ Gestão de Chamadas

- Introdução de novos fluxos de media numa chamada
- Mudança de método de codificação
- Convite de outros intervenientes (conferência)
- Transferência e chamada em espera

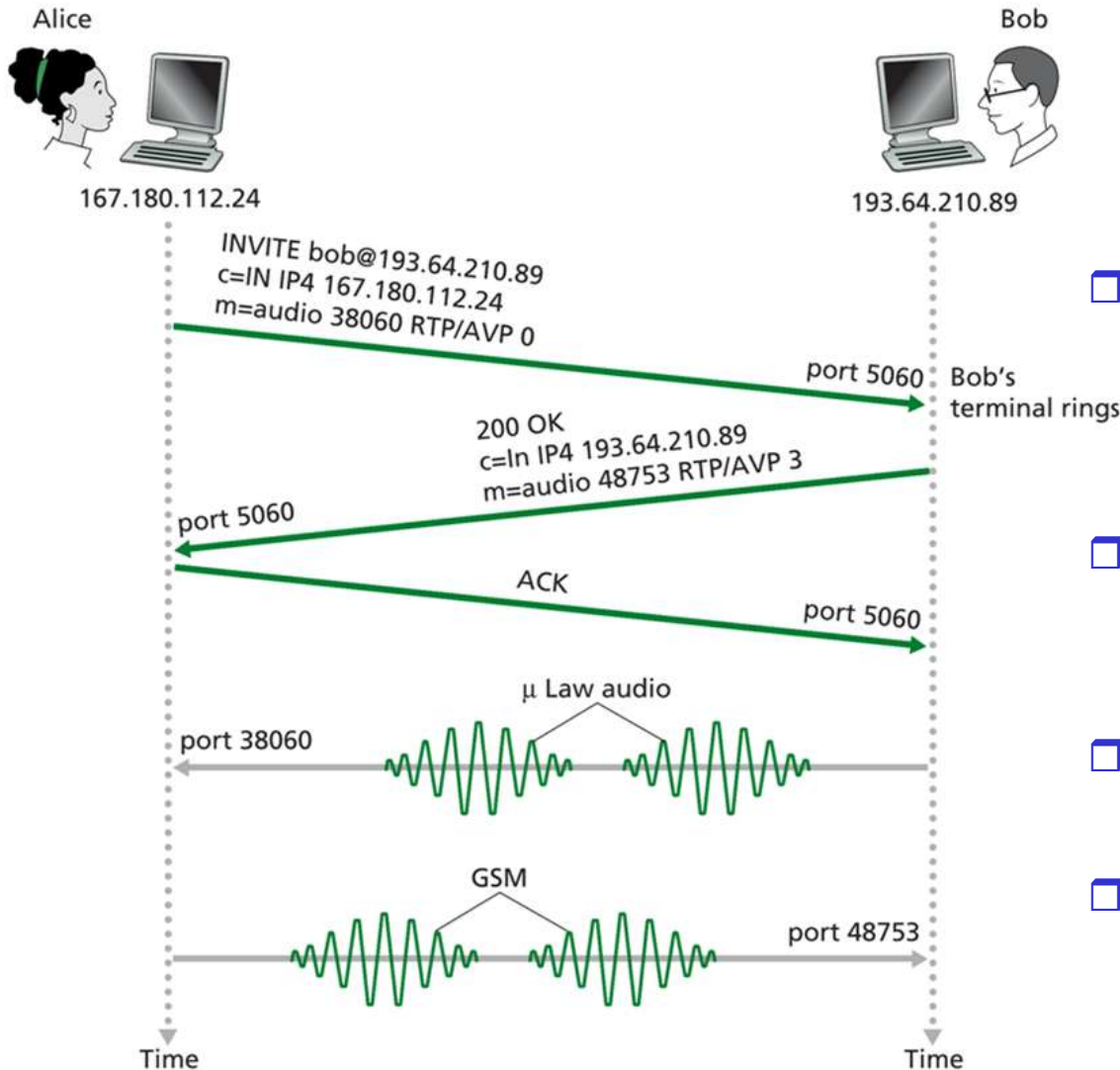


Sinalização SIP



- ❑ O SIP é um protocolo de Sinalização que serve apenas para o estabelecimento das chamadas
- ❑ Uma vez estabelecida, o *streaming* de voz é feito com base num outro protocolo, geralmente RTP

Ex: Chamada para um IP conhecido



- ❑ A mensagem SIP invite de Alice indica o seu endereço IP e número de porto e propõe codificação PCM μ law
- ❑ Bob responde com o seu número de porto indicando que prefere a codificação GSM
- ❑ As mensagens SIP podem ser transportadas em UDP ou TCP
- ❑ O porto por defeito do SIP é o 5060

Estabelecimento de Chamada

- ❑ Negociação do Codec:
 - Se Bob não tiver um codec para PCM μ law.
 - A resposta seria 606 Not Acceptable com a lista dos codecs de que dispunha
 - Alice iria enviar uma nova mensagem INVITE propondo novo codec entre os listados
- ❑ Rejeição da chamada
 - A chamada pode ser recusada com os motivos “busy,” “gone,” “payment required,” “forbidden”.
- ❑ O media pode ser enviado em RTP ou noutro protocolo acordado entre as partes



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@host.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 tipo HTTP
- ❑ URI do tipo sip:name@domain.tld
- ❑ sdp = Session Description Protocol
- ❑ Call-ID é único para cada chamada.

- ❑ Neste caso, o endereço IP de Bob não é conhecido, por isso é necessário utilizar servidores SIP intermediários (proxy) para resolver o nome
- ❑ Alice envia mensagens utilizando o porto SIP por defeito (5060)
- ❑ Alice especifica na directiva Via: que o seu cliente envia e recebe mensagens através de UDP



Exemplo de início de sessão VoIP

No. -	Time	Source	Destination	Protocol	Info
16	2.106345	10.0.0.2	10.0.0.4	SIP/SD	Request: INVITE sip:jrogado@10.0.0.4, with session descr
17	0.003807	10.0.0.4	10.0.0.2	SIP	Status: 100 trying -- your call is important to us
18	0.009649	10.0.0.4	10.0.0.2	SIP	Status: 180 Ringing
340	6.501599	10.0.0.4	10.0.0.2	SIP/SD	Status: 200 OK, with session description
342	0.005989	10.0.0.2	10.0.0.4	RTCP	Receiver Report[Malformed Packet]
343	0.014478	10.0.0.4	10.0.0.2	RTCP	Receiver Report
344	0.007823	10.0.0.4	10.0.0.2	RTP	Payload type=ITU-T G.711 PCMU, SSRC=1295595096, Seq=2568
345	0.013863	10.0.0.2	10.0.0.4	RTP	Payload type=ITU-T G.711 PCMU, SSRC=1352170915, Seq=7280
353	0.028829	10.0.0.2	10.0.0.4	RTP	Payload type=ITU-T G.711 PCMU, SSRC=1352170915, Seq=7281
354	0.030537	10.0.0.2	10.0.0.4	RTP	Payload type=ITU-T G.711 PCMU, SSRC=1352170915, Seq=7282

```

❏ Internet Protocol, Src: 10.0.0.4 (10.0.0.4), Dst: 10.0.0.2 (10.0.0.2)
❏ User Datagram Protocol, Src Port: 5060 (5060), Dst Port: 9782 (9782)
❏ Session Initiation Protocol
  ❏ Status-Line: SIP/2.0 200 OK
    Status-Code: 200
    [Resent Packet: False]
  ❏ Message Header
    Via: SIP/2.0/UDP 10.0.0.2:9782;rport=9782;branch=z9hG4bK-d87543-046ec739f357e374-1--d87543-
    Record-Route: <sip:127.0.0.1;r2=on;lr=on;ftag=b52ffe72>, <sip:10.0.0.4;r2=on;lr=on;ftag=b52ffe72>
    + To: "Jose Rogado" <sip:jrogado@10.0.0.4>;tag=qnhqt
    + From: "Jose Q. Rogado" <sip:jqr@10.0.0.4>;tag=b52ffe72
    Call-ID: Mjk1YWJmMjFhMTE0MDhinzhinju0Y2YxNzc0NTBIM2I.
    CSeq: 1 INVITE
    + Contact: <sip:jrogado@10.0.0.4:5062>
    Content-Type: application/sdp
    Allow: INVITE,ACK,BYE,CANCEL,OPTIONS,PRACK,REFER,NOTIFY,SUBSCRIBE,INFO
    Server: Twinkle/1.0
    Supported: replaces,norefersub
    Content-Length: 198
  ❏ Message body
    ❏ Session Description Protocol
      Session Description Protocol Version (v): 0
      ❏ Owner/Creator, Session Id (o): jrogado 964782761 2108236566 IN IP4 10.0.0.4
      Session Name (s): -
      ❏ Connection Information (c): IN IP4 10.0.0.4
        Connection Network Type: IN
        Connection Address Type: IP4
        Connection Address: 10.0.0.4
      ❏ Time Description, active time (t): 0 0
      ❏ Media Description, name and address (m): audio 8000 RTP/AVP 0 101
        Media Type: audio
        Media Port: 8000
        Media Proto: RTP/AVP
        Media Format: ITU-T G.711 PCMU
        Media Format: 101
      ❏ Media Attribute (a): rtpmap:0 PCMU/8000
      ❏ Media Attribute (a): rtpmap:101 telephone-event/8000
      ❏ Media Attribute (a): fmtp:101 0-15
      ❏ Media Attribute (a):ptime:20
  
```

Resolução de Nomes e Localização de Correspondentes

- ❑ O iniciador só tem o nome ou email do destinatário
 - Precisa de obter endereço IP do seu host actual
- ❑ O destinatário pode ter diferentes dispositivos IP (PC, PDA, automóvel)
- ❑ O DNS não sabe gerir este tipo de pedidos
- ❑ O resultado pode depender de uma série de factores:
 - Hora da chamada (casa, trabalho)
 - Estado actual do destinatário (ocupado, ausente, etc...)
 - Localização do próprio iniciador

O serviço é fornecido por diferentes servidores SIP :

- ❑ SIP registrar server
- ❑ SIP proxy server



SIP Registrar

- ❑ Quando o Bob inicia o seu cliente SIP, este envia uma mensagem ao Servidor de Registo (**Registrar**) semelhante à que é enviada a um servidor de Instant Messaging

Register Message:

```
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
```



SIP Proxy

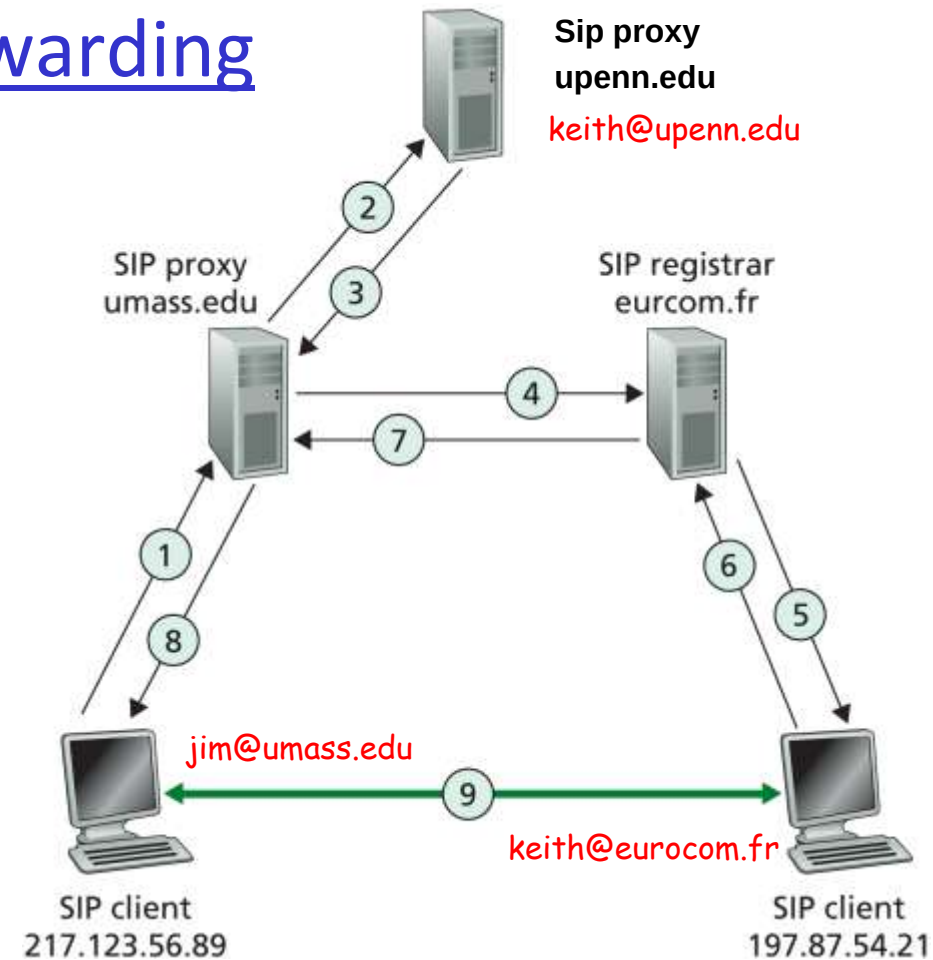
- ❑ Alice envia a mensagem ao seu servidor proxy
 - Com o endereço do destinatário sip:bob@domain.com
- ❑ O proxy é responsável por encaminhar as mensagens até ao destinatário
 - Possivelmente através de vários proxies
- ❑ O destinatário envia a sua resposta através do mesmo conjunto de proxies
- ❑ O proxy envia a mensagem de volta ao cliente de Alice
 - A resposta contém o endereço IP de Bob
 - Alice pode então contactar directamente Bob
- ❑ Nota: os proxies têm um papel análogo ao dos servidores DNS



Exemplo de Call Forwarding

Caller `jim@umass.edu`
places a call to
`keith@upenn.edu`

- (1) Jim sends INVITE message to umass SIP proxy.
- (2) Proxy forwards request to upenn registrar server.
- (3) upenn server returns redirect response, indicating that it should try `keith@eurocom.fr`

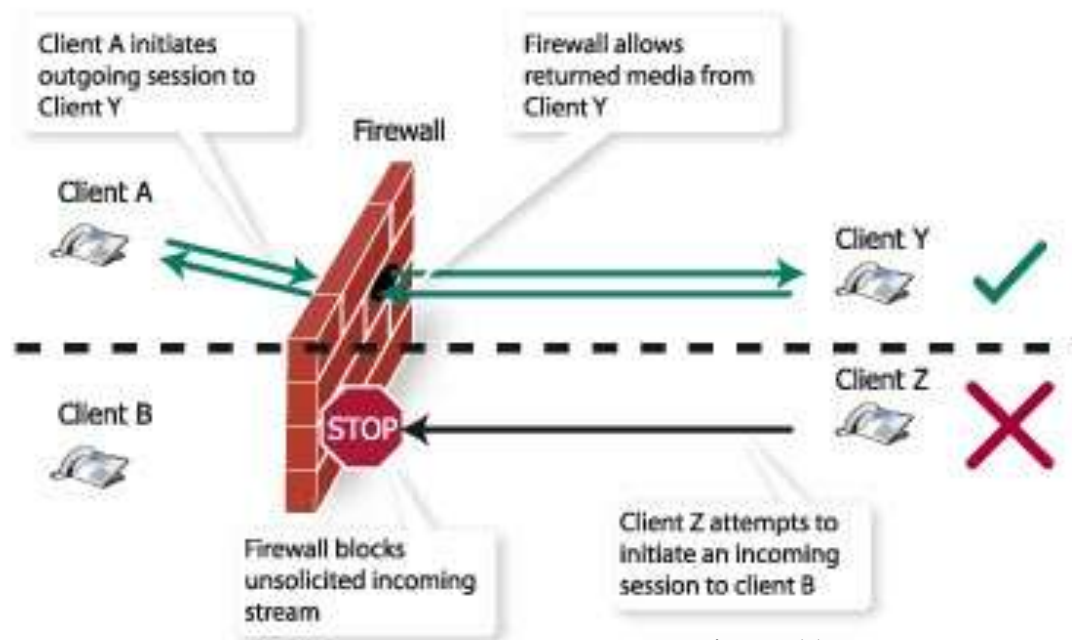


(4) umass proxy sends INVITE to eurecom registrar. (5) eurecom registrar forwards INVITE to 197.87.54.21, which is running keith's SIP client. (6-8) SIP response sent back (9) media sent directly between clients.

Note: there is also a SIP ack message, which is not shown.



Alguns Problemas do VoIP (i)

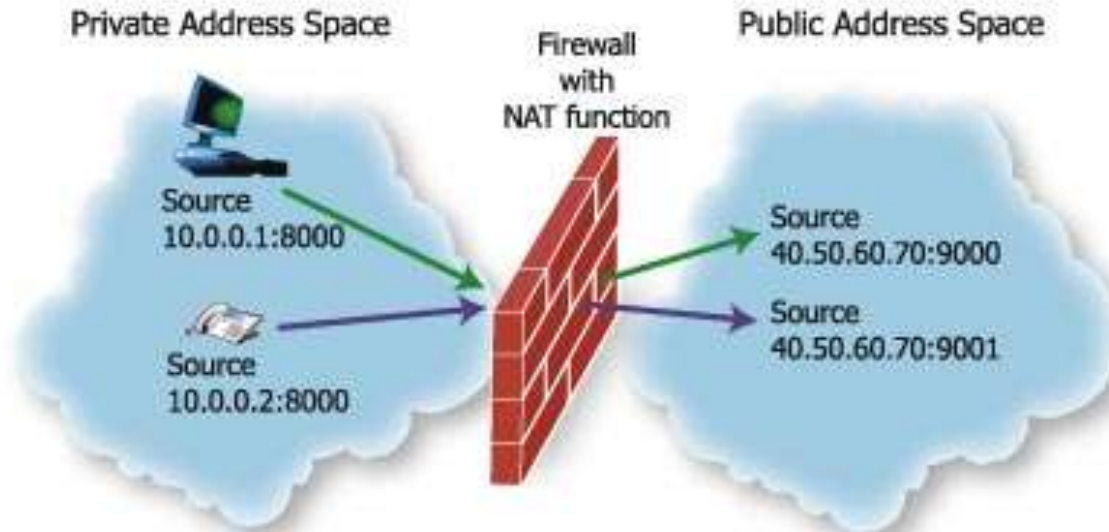


Source <http://www.newport-networks.com>

- ❑ Estabelecimento de sessões entre utilizadores posicionados atrás de routers e firewalls
 - O endereço IP do utilizador não é visível do exterior
 - As firewalls filtram o tráfego à entrada
- ❑ As sessões podem ser iniciadas do interior, mas não do exterior



Alguns Problemas do VoIP (ii)



Source <http://www.newport-networks.com>

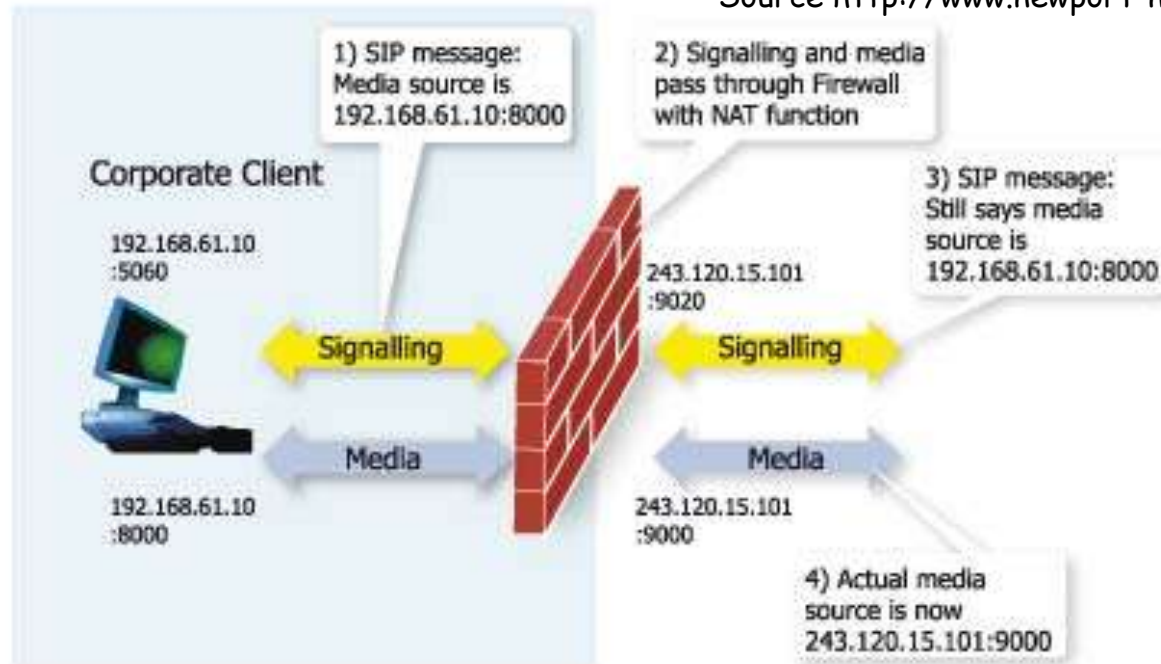
A maioria dos routers e firewalls implementa NAT

- ❑ O endereço IP e porto internos são transformados num IP e porto externos
- ❑ O porto de origem pode ser modificado, de várias formas
 - NAT cónico: o porto é mantido para todos os destinos e qualquer entidade externa pode enviar tráfego desde que conheça esse porto
 - NAT simétrico: o porto muda com o destino, e só o destinatário pode enviar tráfego de volta para esse porto



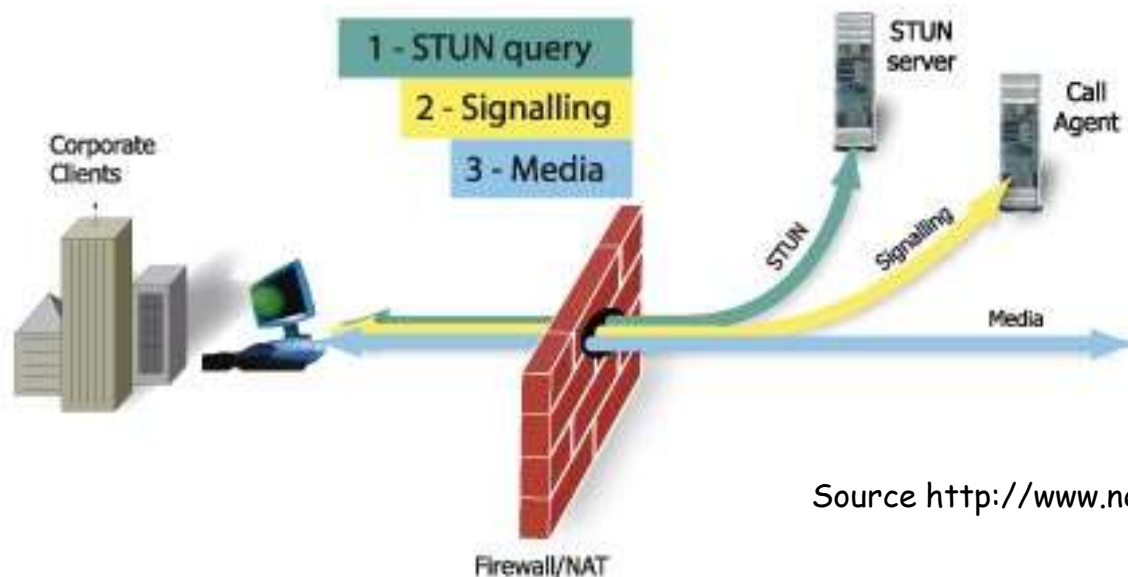
NAT e SIP

Source <http://www.newport-networks.com>



- ❑ O NAT introduz incongruência nas mensagens SIP
 - A mensagem SIP refere os verdadeiros endereços IP e portos para realizar o fluxo de voz
 - O acesso a esses IPs e portos não é possível do exterior
 - Os dados que chegam provêm de um endereço diferente do que é anunciado
- ❑ São necessários mecanismos para resolver estes problemas

STUN: Simple Traversal of UDP Through NAT

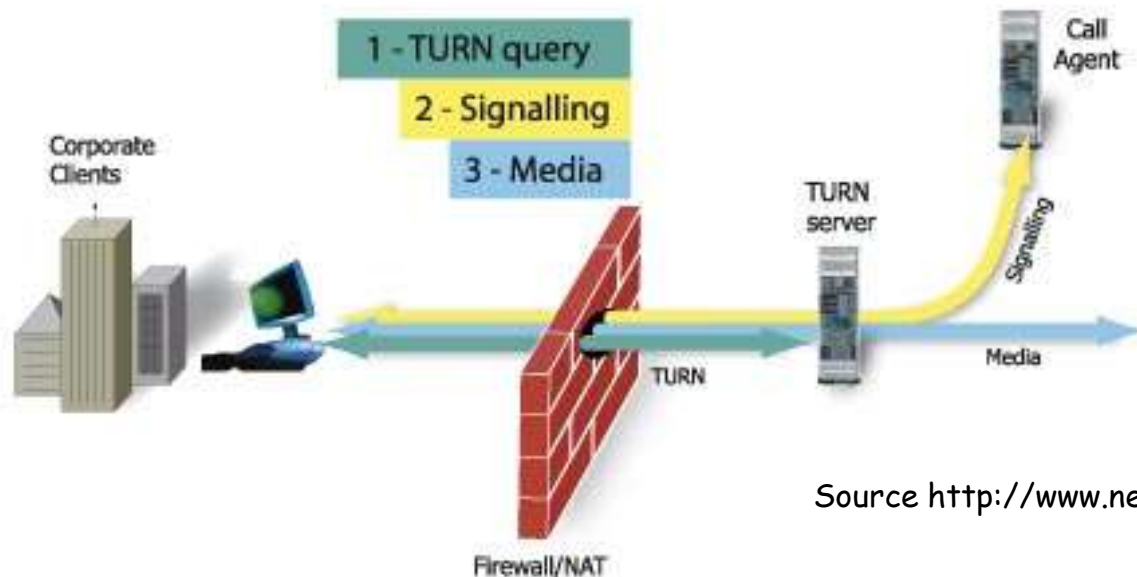


Source <http://www.newport-networks.com>

- ❑ O STUN introduz um servidor adicional fora da Firewall/Router
 - No início da ligação o cliente SIP utiliza-o para descobrir quais o endereço IP e porto públicos
 - As mensagens de sinalização são então modificadas com base nessa informação
- ❑ Só funciona com NAT cónico: o IP e o porto públicos são os mesmos para todos os pacotes provenientes de um dado IP e porto interno
- ❑ Não funciona com NAT simétrico: para cada destino externo distinto é alocado um porto diferente



TURN - Traversal Using Relay NAT

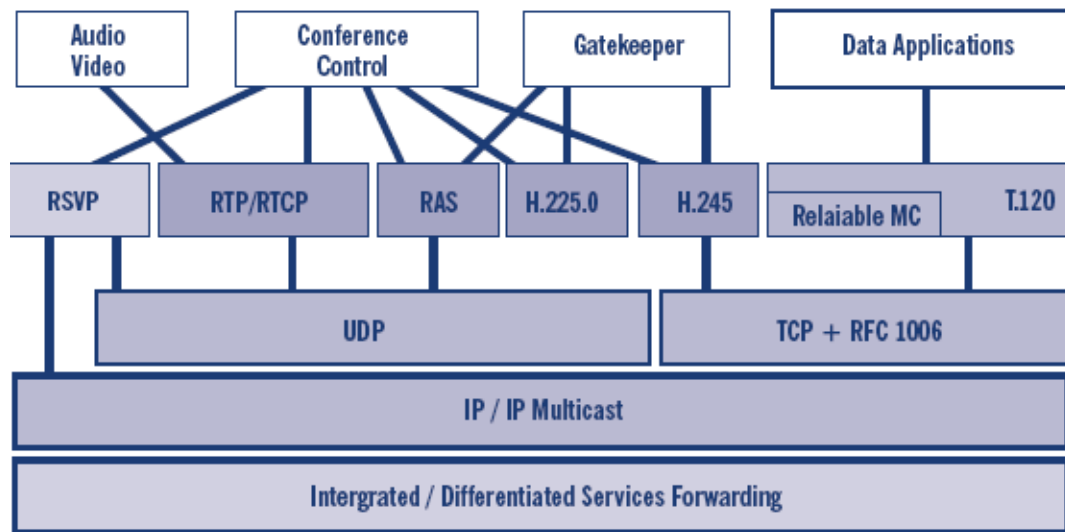


Source <http://www.newport-networks.com>

- ❑ O servidor TURN mantém-se no fluxo de dados realiza forwarding dos pacotes de ida e volta
- ❑ Não há mudança de destino nas mensagens pelo que pode funcionar com NAT simétrico
- ❑ Tanto o STUN como o TURN necessitam de funcionalidade adicional no cliente VoIP
 - Além do deployment do próprio servidor
- ❑ ICE - Interactive Connectivity Establishment integra estas tecnologias de modo a poderem ser usadas em função da arquitectura de deployment



Breve comparação com o H.323



Arquitectura do protocolo H.323

- ❑ O H.323 é um protocolo da ITU - T para aplicações interactivas em tempo real
- ❑ O H.323 é uma suite completa e integrada de protocolos para conferências multimédia: inclui sinalização (H.225), registo, controle de admissão (H.245), transporte (H.248.57) e codecs (H.263, H264).

- ❑ O SIP é um simples componente, funciona com o RTP mas não o impõe
- ❑ Pode ser combinado com outros protocolos e serviços
- ❑ O H.323 provem da ITU e do mundo da telefonia.
- ❑ O SIP é originário do IETF: muitos dos seus conceitos são inspirados do HTTP.
- ❑ Embora provenientes de mundos diferente, os protocolos interoperam através de uma gateway (Asterisk)
- ❑ O SIP usa o princípio KISS: Keep It Simple & Stupid !



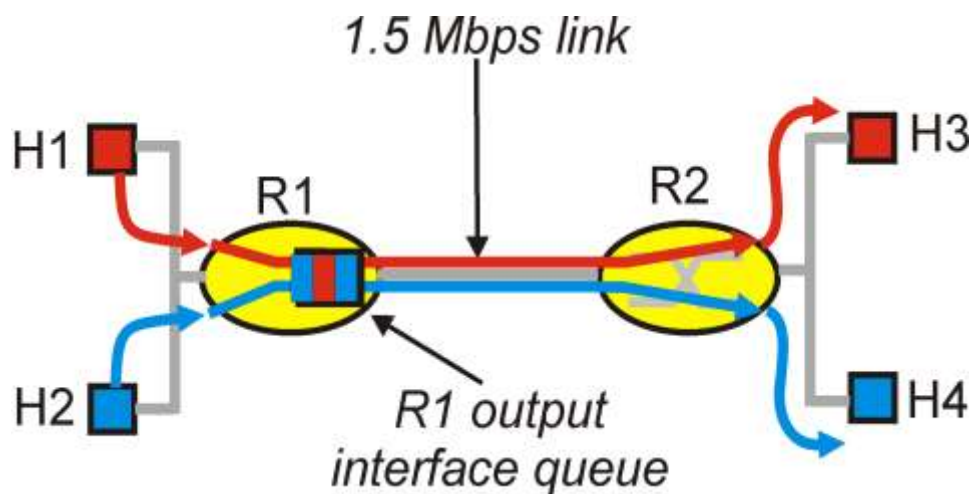
Capítulo 3: Redes Multimédia

- ❑ Características das Aplicações Multimédia
 - Características
 - Necessidades
 - Compressão de Dados
- ❑ Fluxo de Áudio e Vídeo
 - Acesso Web
 - Servidores de Streaming
 - RSTP
- ❑ Multimédia em Tempo Real
 - Limitações do “Best Effort”
 - Características e necessidades do VoIP
- ❑ Protocolos para Aplicações Interactivas em Tempo Real
 - RTP / RTCP
 - SIP / H.323
- ❑ Qualidade de Serviço
 - Necessidades
 - Exemplos
- ❑ Implementação de QoS
 - Mecanismos de Escalonamento
 - Disciplinas de Serviço
- ❑ Arquitecturas de QoS
 - Serviços Integrados (Intserv) e Diferenciados (Diffserv)
- ❑ Protocolo de Sinalização: RSVP



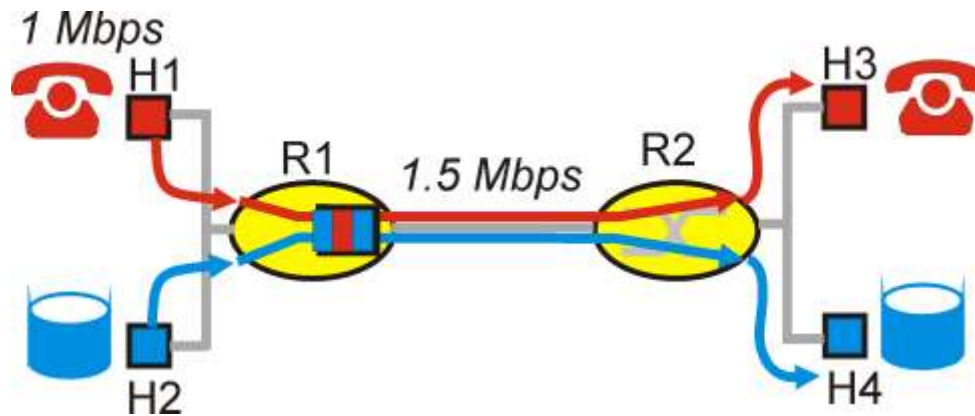
Introdução de QoS em redes IP

- ❑ Nas secções anteriores vimos a forma de tirar o melhor partido do conceito de “melhor esforço”
- ❑ Vamos agora abordar as modificações que podem ser introduzidas na Internet para garantir realmente QoS :
 - **Integrated Services**: integração na arquitectura
 - **RSVP**: sinalização para reserva de recursos
 - **Differentiated Services**: tráfego diferenciado
- ❑ Vamos utilizar um modelo simples para analisar os problemas de congestão



Garantia de QoS - Marcação e Prioritização

- ❑ Exemplo: tráfego VoIP a 1Mbps e FTP a partilhar uma ligação de 1.5 Mbps.
 - Em modo *best-effort* (FIFO), as sequência de pacotes FTP podem saturar o router, e causar perda de dados áudio
- ❑ Possibilidade de dar prioridade do áudio relativamente ao FTP
 - Introdução de prioridade no encaminhamento de pacotes baseada no tipo de tráfego ou no tipo de subscrição

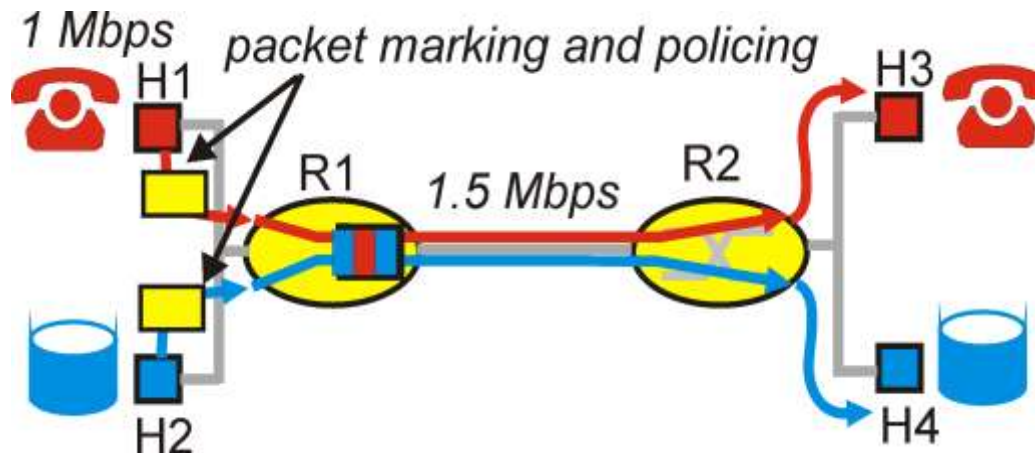


Princípio 1

É necessário marcar os pacotes para que os routers possam distinguir diferentes classes de tráfego, e alterar os algoritmos de encaminhamento do routers de forma a tratarem os pacotes de forma diferenciada

Garantia de QoS - Isolamento e Regulação

- ❑ Mas as aplicações podem ter um comportamento incorrecto
 - VoIP enviar áudio a uma taxa mais alta do que a anunciada
 - Pode levar a que o tráfego seja todo atribuído à classe mais prioritária levando ao bloqueamento das outras
- ❑ Necessidade de isolamento e regulação do tráfego:
 - Necessidade de forçar os vários tipos de tráfego a respeitar as características acordadas

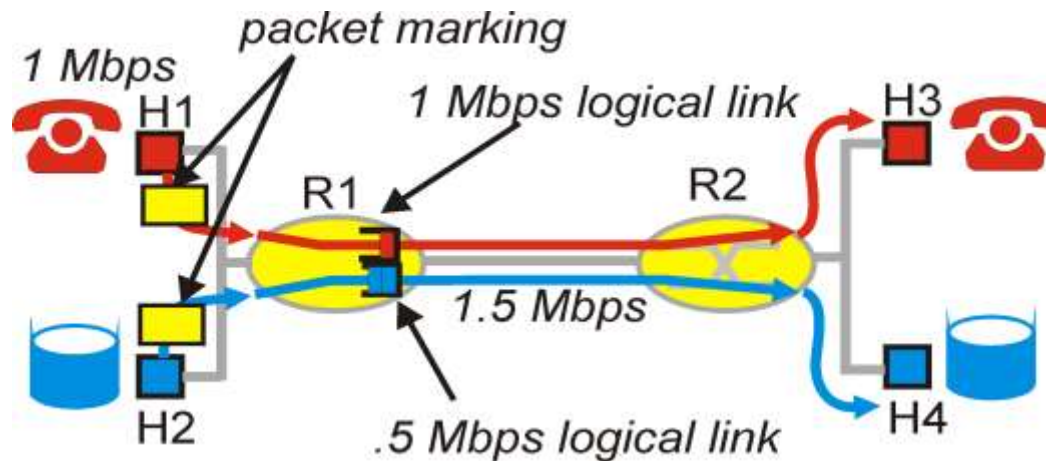


Princípio 2

É necessário isolar as diferentes classes de tráfego e garantir o respeito das regras estabelecidas

Garantia de QoS - Eficácia

- ❑ A alocação de largura de banda fixa a classes de tráfego distintas pode levar a utilização ineficiente dos recursos em caso de inexistência de tráfego
 - Problema idêntico ao da comutação de circuitos
- ❑ A separação e isolamento do tráfego deve feita de forma dinâmica

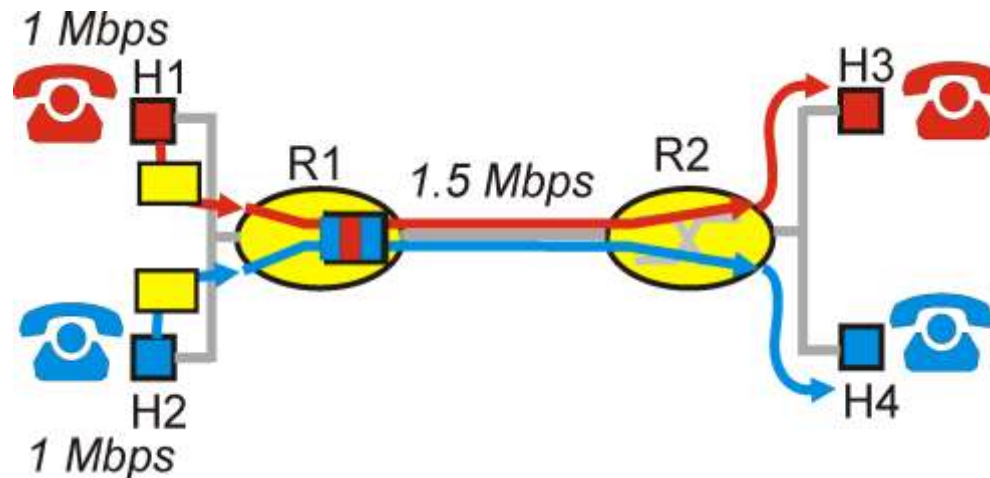


Princípio 3

Embora seja necessário isolar e proteger classes distintas de tráfego, a alocação de recursos deve ser feita de forma dinâmica de modo a tirar partido das suas possíveis variações

Garantia de QoS - Admissão/Rejeição

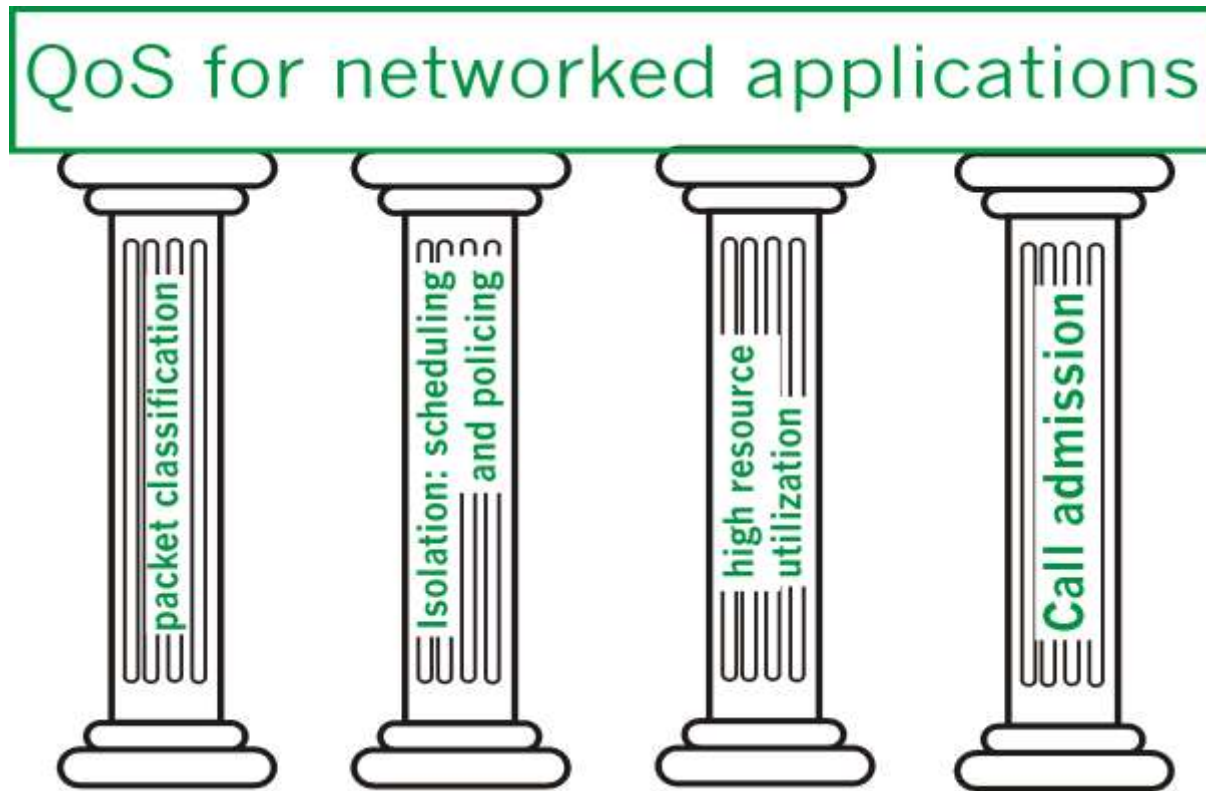
- ❑ Não é possível aceitar pedidos de tráfego para além da capacidade de transmissão da ligação
 - Mesmo respeitando os princípios anteriores não podem ser estabelecidas duas ligações VoIP de 1 Mbps
 - A alocação deve ser nesse caso rejeitada, da mesma forma que nas redes telefónicas saturadas



Princípio 4

É necessário existir um processo de admissão de tráfego capaz de aceitar pedidos com o tipo de QoS requerido, e de rejeitar os que excedem as capacidades físicas da rede

Recapitulação dos Princípios de QoS



Vamos agora examinar as formas de implementar QoS



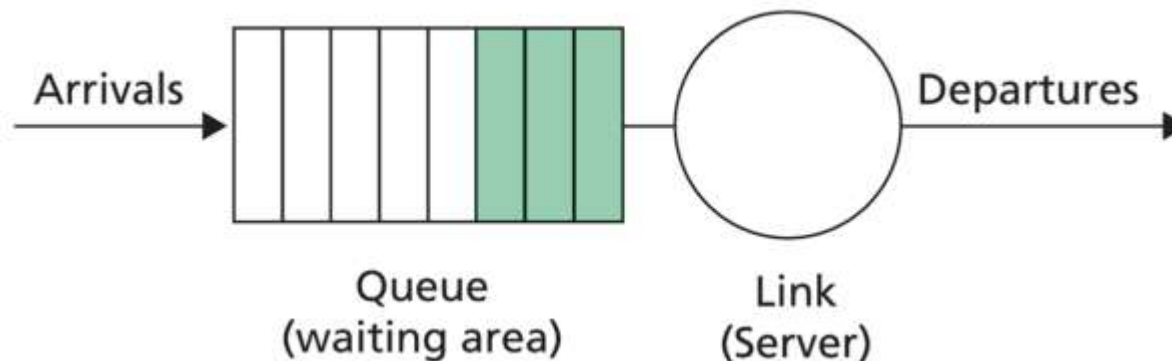
Capítulo 3: Redes Multimédia

- ❑ Características das Aplicações Multimédia
 - Características
 - Necessidades
 - Compressão de Dados
- ❑ Fluxo de Áudio e Vídeo
 - Acesso Web
 - Servidores de Streaming
 - RSTP
- ❑ Multimédia em Tempo Real
 - Limitações do “Best Effort”
 - Características e necessidades do VoIP
- ❑ Protocolos para Aplicações Interactivas em Tempo Real
 - RTP / RTCP
 - SIP / H.323
- ❑ Qualidade de Serviço
 - Necessidades
 - Princípios
- ❑ Implementação de QoS
 - Mecanismos de Escalonamento
 - Regulação de Serviços
- ❑ Arquitecturas de QoS
 - Serviços Integrados (Intserv) e Diferenciados (Diffserv)
- ❑ Protocolo de Sinalização: RSVP



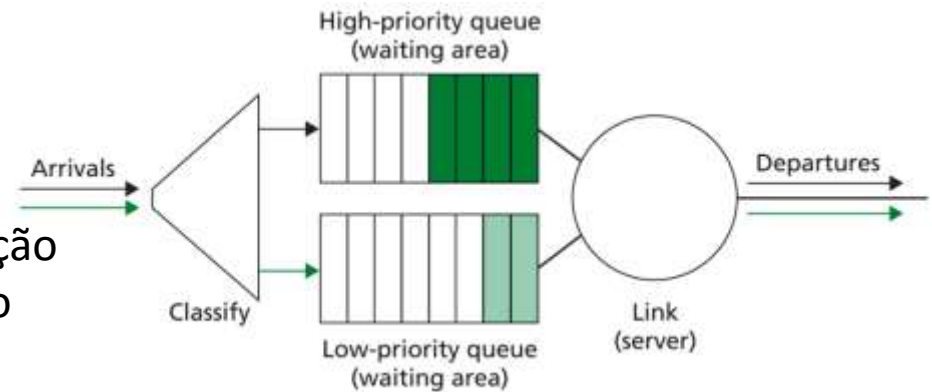
Mecanismos de Escalonamento

- ❑ **Escalonamento:** escolha do próximo pacote a enviar na ligação
 - Analogia com o escalonamento de processos
- ❑ **Escalonamento FIFO (first in first out):** enviar na mesma ordem de chegada
 - típico do *best effort*
 - Exemplo - fila de espera de um balcão atendimento
 - **Política de rejeição:** se um pacote chegar e a fila estiver cheia: que pacote se descarta?
 - Tail drop: descarta o pacote que chega
 - Priority drop: descartar/remover com base na prioridade
 - Random drop: descartar/remover aleatoriamente

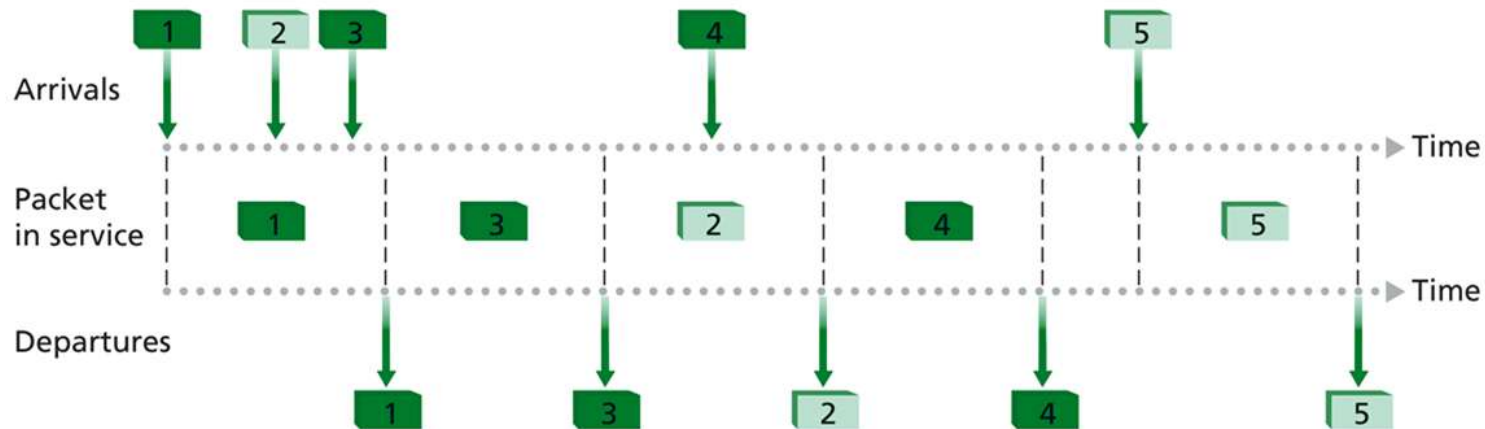


Escalonamento Prioritário

- ❑ É **sempre** transmitido o pacote com maior prioridade
- ❑ *Classes* múltiplas, com diferentes prioridades
 - A classe pode depender de marcação ou de outra informação contida no header:
 - p.ex. ToS, endereço IP de origem ou destino, número de porto, etc.



Implementação

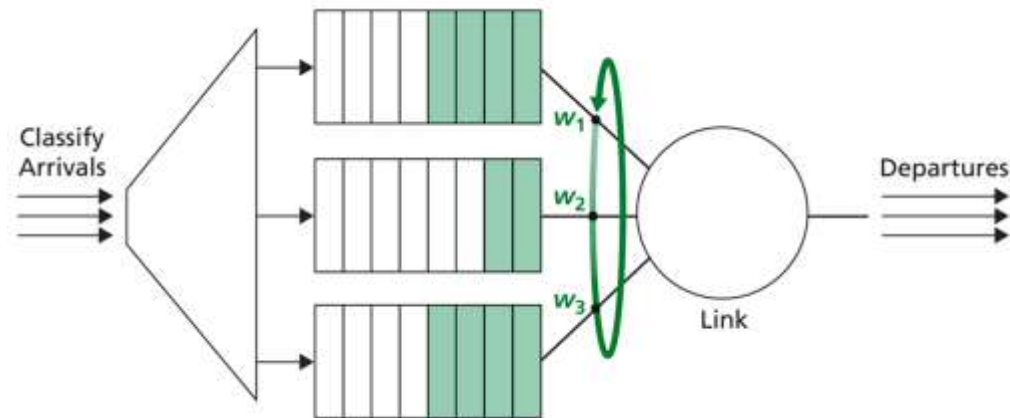


Funcionamento

Escalonamento Round-Robin e Weighted Fair Queueing

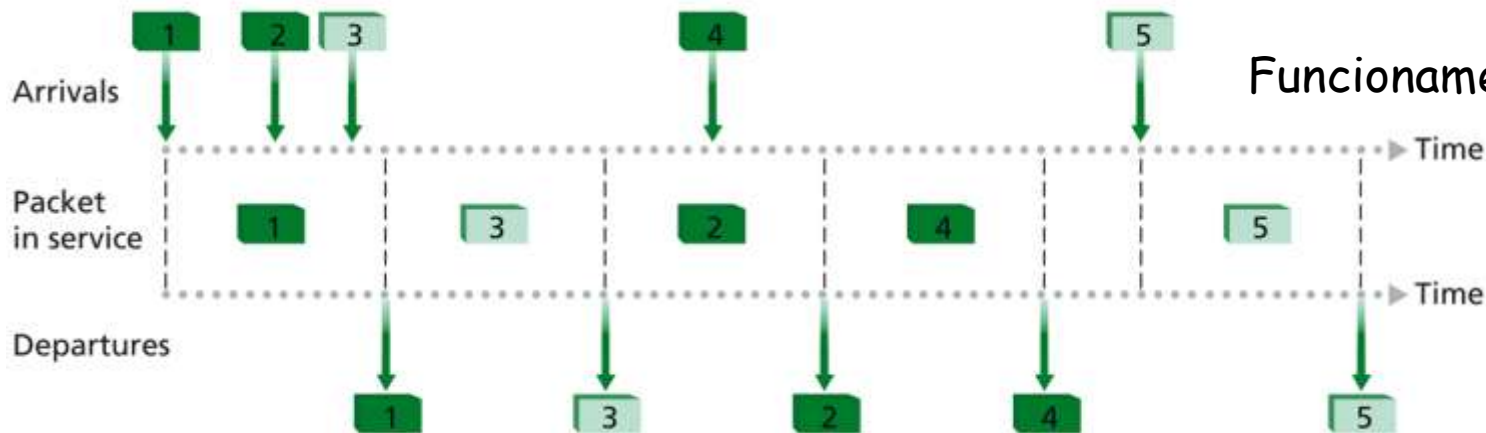
- São estabelecidas múltiplas classes
- As filas de espera das várias classes são percorridas ciclicamente, sendo um pacote retirado de cada uma (caso exista)
- Pode ser atribuído um peso diferente a cada fila:
 - Retira i pacotes da fila de peso i
 - Weighted Fair Queueing (WFQ)

Implementação



$$r_i = R \cdot \frac{w_i}{\sum w_j}$$

Funcionamento RR



Mecanismos de Regulação

Objectivo: limitação do tráfego para não exceder os parâmetros contratados

Três critérios utilizados mais frequentemente:

- ❑ ***Taxa Média:*** quantos pacotes podem ser transmitidos por unidade de tempo a longo termo
 - Questão crucial: qual o intervalo de medida 100 pacotes por segundo (mais restritivo) ou 6000 pacotes por minuto?
- ❑ ***Taxa Máxima:*** e.g., 6000 pacotes por min. (ppm) de média e 1500 ppm de taxa máxima (*peak rate*)
- ❑ ***Sequência Máxima:*** máximo numero de pacotes enviados consecutivamente (sem período de inactividade)

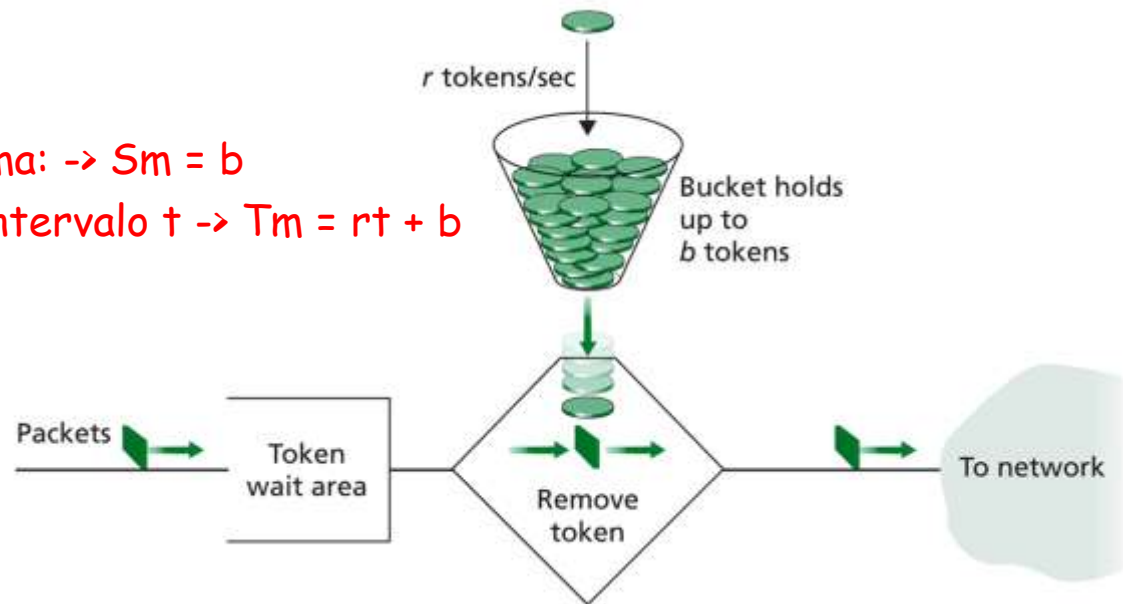


Mecanismo de Regulação: “Balde Furado”

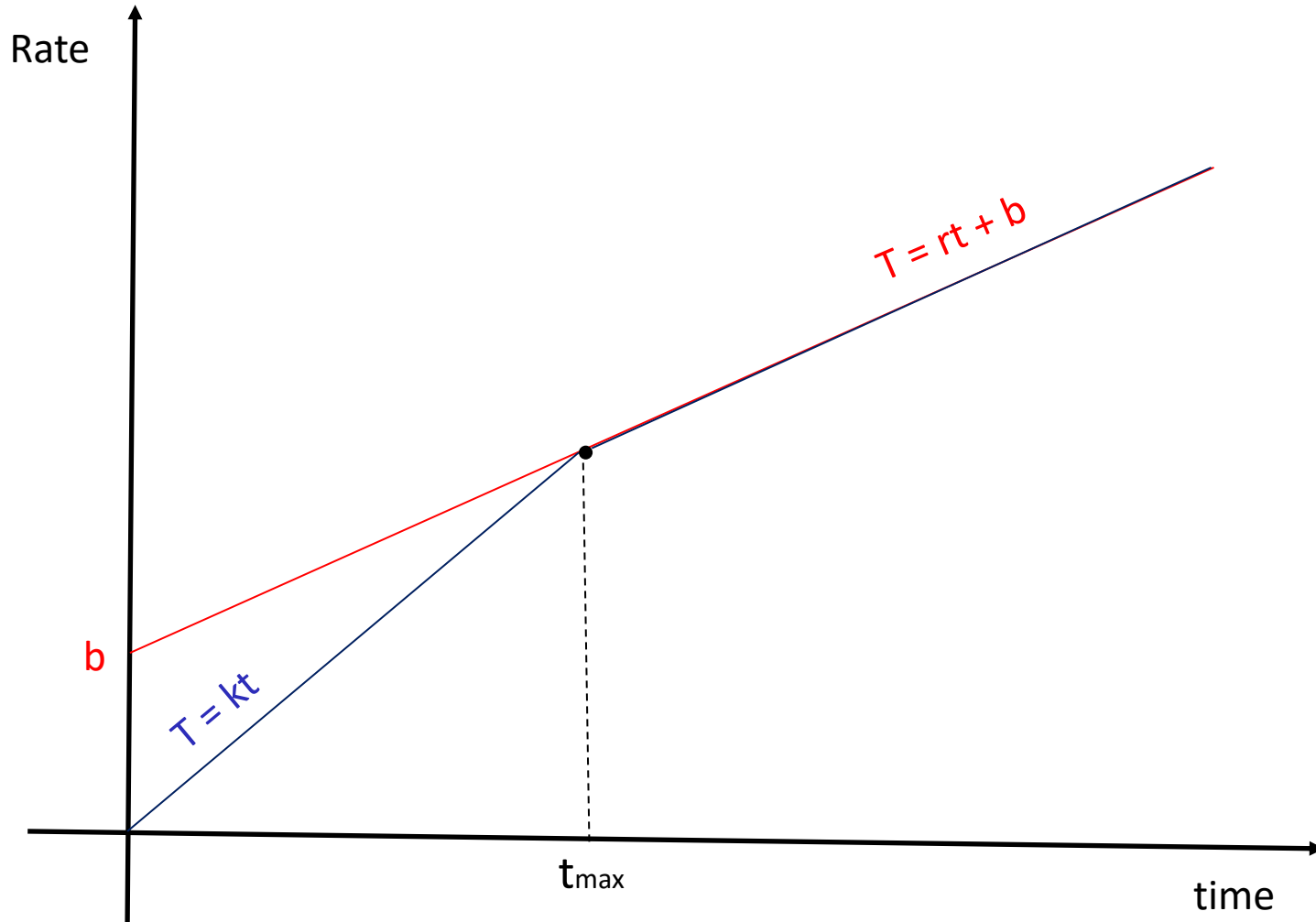
- ❑ Leaky Bucket: mecanismo que permite regular os parâmetros Sequência Máxima e Taxa Média.
 - O balde contém **b** fichas (tokens)
 - Quando o nível baixa, são geradas novas fichas a uma taxa de **r** fichas/segundo
- ❑ Funcionamento: de cada vez que um pacote é transmitido, tem de ser retirada uma ficha do balde
 - Se o balde estiver vazio, o pacote espera

Sequência Máxima: $\rightarrow S_m = b$

Taxa Média no intervalo t $\rightarrow T_m = rt + b$



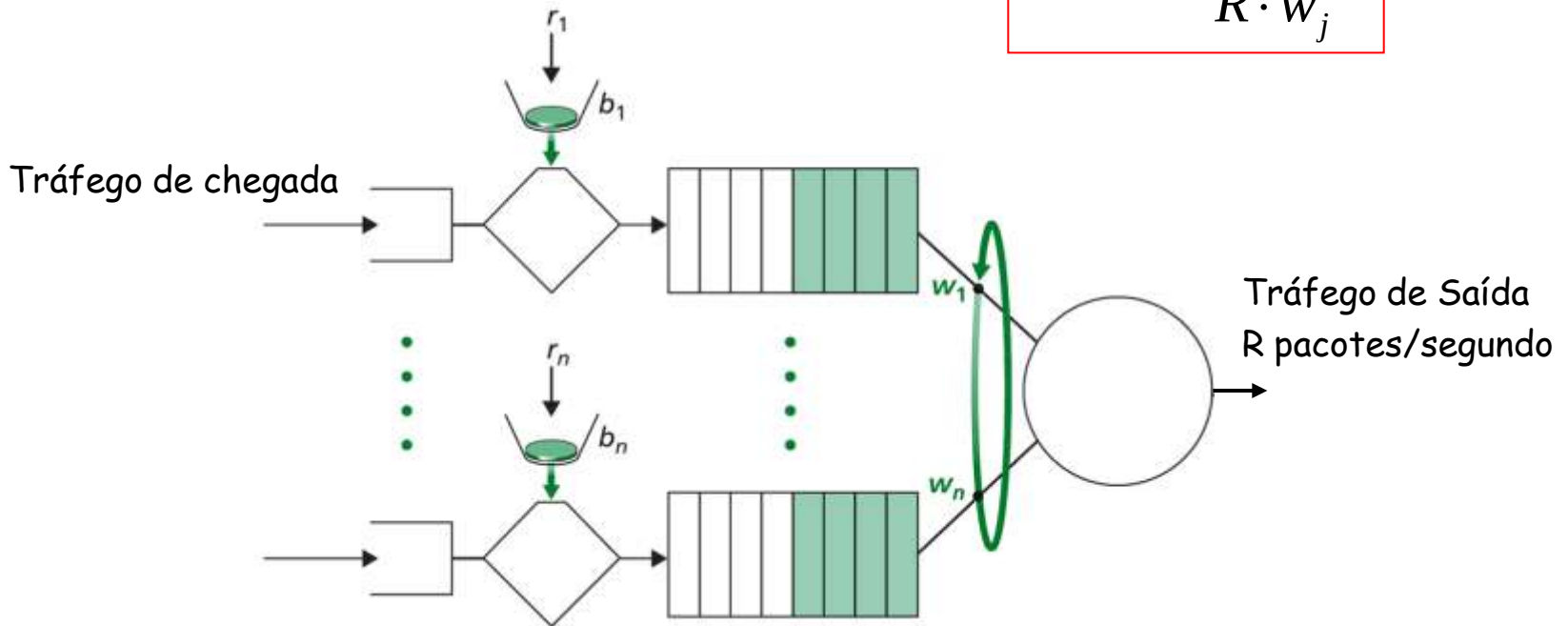
Cálculo da Duração de Taxa Máxima



Combinação de Mecanismos

- ❑ A combinação do “*leaky bucket*” com Weight Fair Queue permite garantir um limite superior no tempo de espera
 - A taxa de chegada em cada ligação é controlada por *leaky bucket*
 - A saída de pacotes é escalonada por WFQ
- ❑ Garantia de QoS
 - Atraso máximo em cada fila é dado por:

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



Capítulo 3: Redes Multimédia

- ❑ Características das Aplicações Multimédia
 - Características
 - Necessidades
 - Compressão de Dados
- ❑ Fluxo de Áudio e Vídeo
 - Acesso Web
 - Servidores de Streaming
 - RSTP
- ❑ Multimédia em Tempo Real
 - Limitações do “Best Effort”
 - Características e necessidades do VoIP
- ❑ Protocolos para Aplicações Interactivas em Tempo Real
 - RTP / RTCP
 - SIP / H.323
- ❑ Qualidade de Serviço
 - Necessidades
 - Exemplos
- ❑ Implementação de QoS
 - Mecanismos de Escalonamento
 - Disciplinas de Serviço
- ❑ **Arquitecturas de QoS**
 - **Serviços Integrados (Intserv) e Diferenciados (Diffserv)**
- ❑ Protocolo de Sinalização: RSVP



Integrated Services IETF (RFC 1633)

- ❑ Uma arquitetura para garantia de QoS em redes IP em sessões aplicacionais
- ❑ **Reserva de Recursos:** os routers mantêm informação de estado sobre os recursos reservados pelos requisitos de QoS através de um protocolo específico (RSVP)
 - cf. Circuitos Virtuais
- ❑ **Estabelecimento de Sessão:** os recursos necessários para uma sessão devem ser negociados previamente com todos os routers do caminho, havendo a possibilidade de admitir ou recusar novos pedidos

Decisão: o novo fluxo pode ser admitido na rede com os requisitos indicados sem degradar as garantias atribuídas aos fluxos existentes?



Arquitetura Diffserv (RFC 2475)

Distinção entre funções periféricas e internas:

Router periférico:

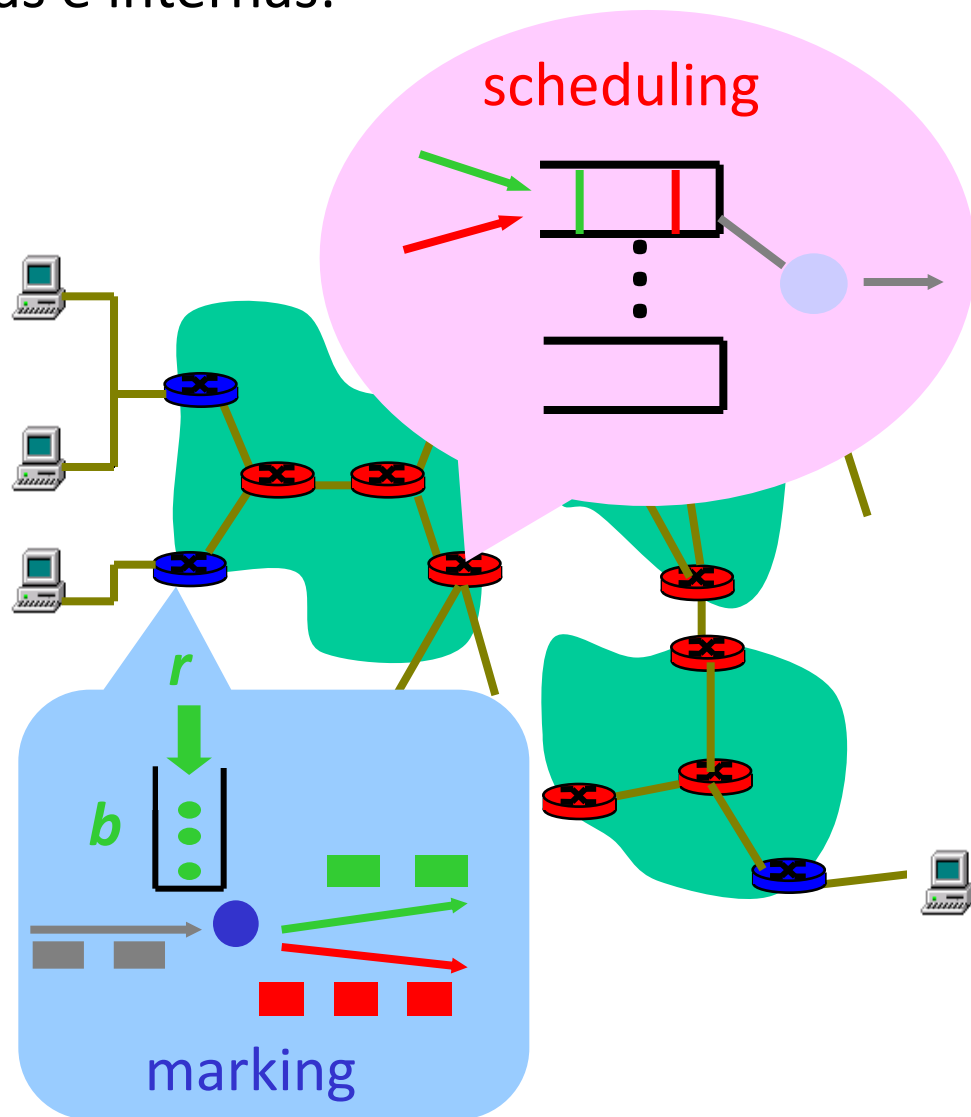


- ❑ Classificação dos pacotes e condicionamento do tráfego
- ❑ Marcação dos pacotes de acordo com a classe da aplicação

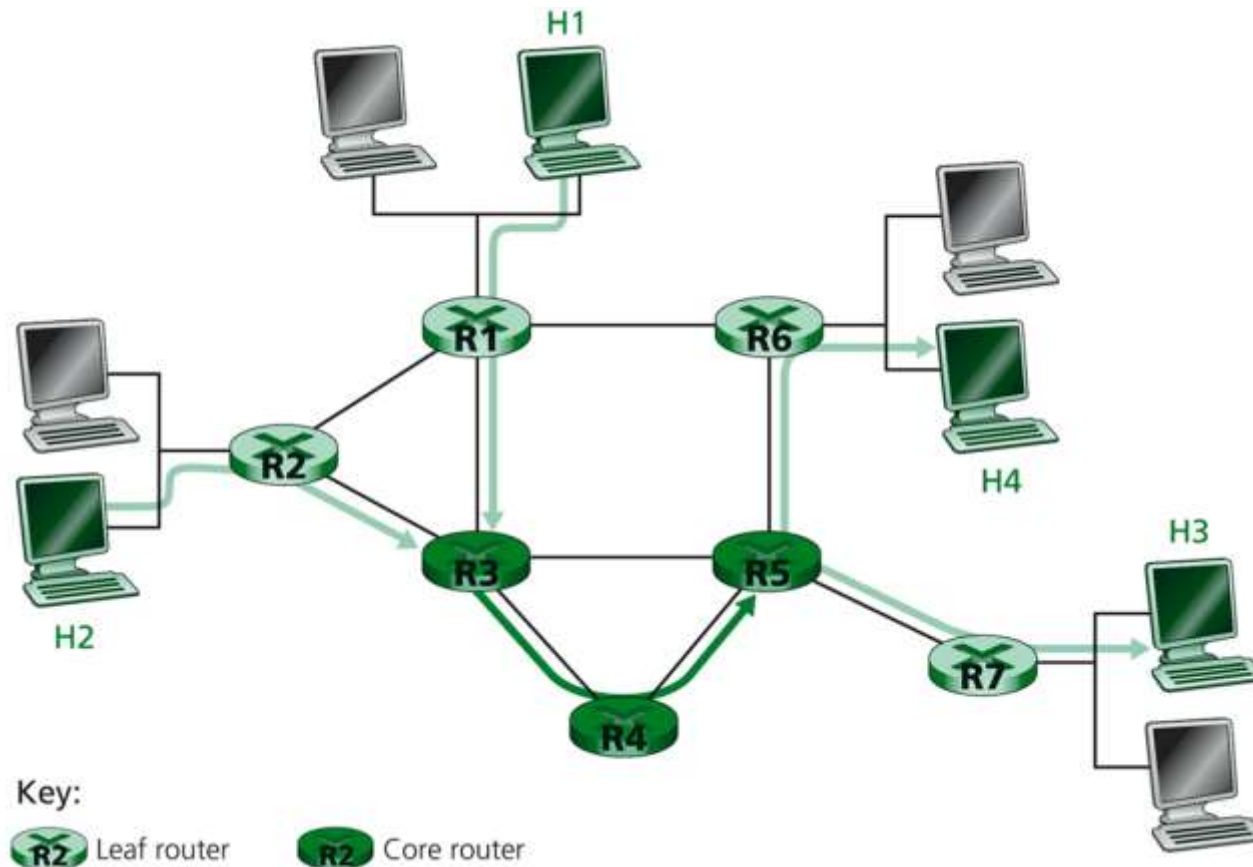
Router interno:



- ❑ Gestão dos pacotes de acordo com a marcação
- ❑ O encaminhamento é somente realizado de acordo com o **PHB - Per-Hop Behaviour** (ver slide 110) associado à classe de marcação



Diferenciação do Tráfego

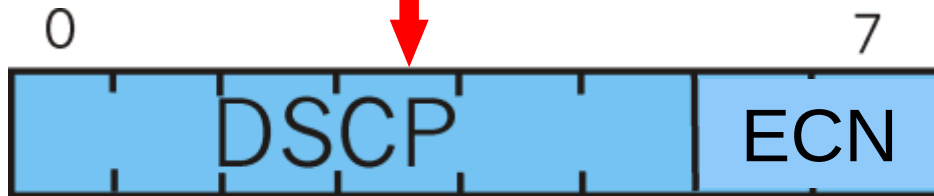


- ❑ O encaminhamento do tráfego em R4 não precisa de saber se provém de R1 ou de R2
- ❑ Baseia-se unicamente no PHB associado à marcação

Marcação de Pacotes

- ❑ Os pacotes são marcado utilizando o campo Type of Service (TOS) em IPv4, e Traffic Class em IPv6
- ❑ 6 bits são utilizados para o **Differentiated Service Code Point (DSCP)** e determinam o PHB que será aplicado ao pacote

+	Bits 0–3	4–7	8–15	16–18	19–31
0	Version	Header length	Type of Service (now DiffServ and ECN)	Total Length	
32	Identification			Flags	Fragment Offset
64	Time to Live		Protocol	Header Checksum	
96	Source Address				
128	Destination Address				

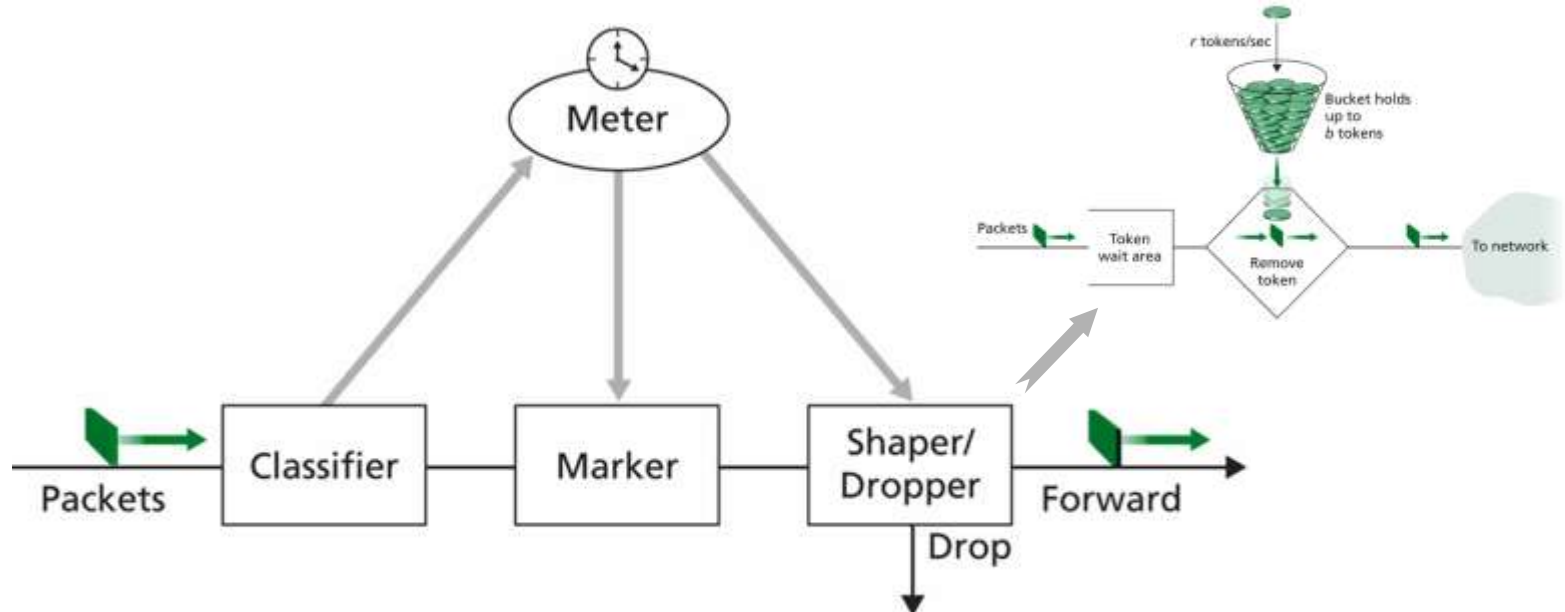


ECN: Explicit Congestion Notification

Classificação e Regulação

À entrada de um router periférico Diffserv os pacotes são:

- ❑ Classificados de acordo com um critério de selecção (endereços IP, portas, protocolos, ...)
- ❑ Marcados com um identificador associado ao **perfil** do QoS contratado
- ❑ Validados relativamente às características de tráfego que devem obedecer aos parâmetros do perfil
- ❑ Se o fluxo não obedece às características do perfil, os pacotes podem ser descartados, passados para outro perfil ou postos em espera para reintegrarem (*reshape*) as características do perfil



Encaminhamento: Per-Hop Behaviour

- ❑ O PHB implica um encaminhamento com características de desempenho observáveis e mensuráveis
- ❑ O PHB não especifica que mecanismos de scheduling devem ser usados nos routers internos para conseguir essas características
- ❑ Exemplos:
 - A Classe de tráfego A recebe x% da largura de banda em termos estatísticos
 - Os pacotes da Classe A são encaminhados primeiro que os pacotes de Classe B
 - Etc.



Tipos de PHB

Existem vários tipos de perfis de PHB:

- ❑ **Default:** no caso de não ser especificado nenhum PHB este corresponde ao tráfego “Best Effort”
- ❑ **Expedited Forwarding:** (RFC 3246) a taxa de entrega de pacotes é garantidamente igual ou superior a uma taxa contratada
 - Utilização de escalonamento prioritário estrito
 - Para evitar saturação, os operadores limitam o tráfego em EF a cerca de 30% das capacidades da rede
- ❑ **Assured Forwarding:** (RFC 2597) é garantido o encaminhamento de todo o tráfego que não excede uma taxa contratada. O tráfego em excesso tem grandes probabilidades de ser descartado em caso de congestão.
 - Fornece 4 classes de tráfego com níveis de priorização relativos

	Class 1	Class 2	Class 3	Class 4
Low Drop	AF11	AF21	AF31	AF41
Med Drop	AF12	AF22	AF32	AF42
High Drop	AF13	AF23	AF33	AF43



Críticas ao Diffserv

- ❑ Embora seja um modelo coerente de fácil implementação, a sua utilização implica uma conformidade de PHBs entre operadores que está longe de ser conseguida
 - Do ponto de vista comercial, é difícil para um operador vender um serviço diferenciado se os outros operadores que participam no encaminhamento do tráfego não têm as mesmas referências
- ❑ O serviço só funciona se os routers e hosts periféricos respeitarem as convenções de marcação dos pacotes
 - Algumas implementações classificam sistematicamente o tráfego com classe de prioridade elevada
- ❑ Existe a tendência para o incremento exponencial da largura de banda oferecida a nível da camada física, que acaba por ser a maneira mais simples de garantir QoS...
 - O Diffserv poderá servir apenas para limitar o tráfego dos clientes de classe C...



Capítulo 3: Redes Multimédia

- ❑ Características das Aplicações Multimédia
 - Características
 - Necessidades
 - Compressão de Dados
- ❑ Fluxo de Áudio e Vídeo
 - Acesso Web
 - Servidores de Streaming
 - RSTP
- ❑ Multimédia em Tempo Real
 - Limitações do “Best Effort”
 - Características e necessidades do VoIP
- ❑ Protocolos para Aplicações Interactivas em Tempo Real
 - RTP / RTCP
 - SIP / H.323
- ❑ Qualidade de Serviço
 - Necessidades
 - Exemplos
- ❑ Implementação de QoS
 - Mecanismos de Escalonamento
 - Disciplinas de Serviço
- ❑ Arquitecturas de QoS
 - Serviços Integrados (Intserv) e Diferenciados (Diffserv)
- ❑ **Protocolo de Sinalização: RSVP**



Protocolo de Sinalização: RSVP

Ausência de estado
Encaminhamento
pacotes IP

+

Serviço de
Melhor Esforço =

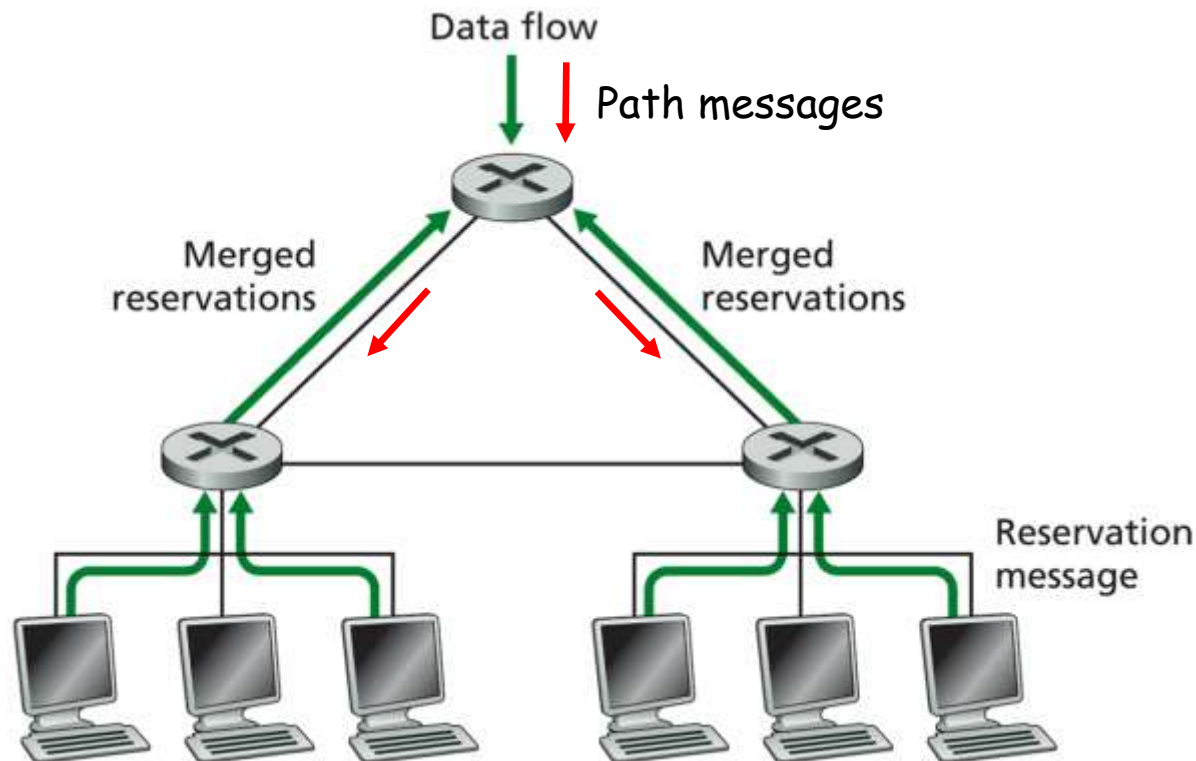
Não existem protocolos de
sinalização no desenho
original do IP

- ❑ **Novo requisitos:** é necessário reservar recursos ao longo do caminho ponto a ponto para aplicações multimédia com requisitos de QoS
- ❑ **RSVP:** resource ReSerVation Protocol [RFC 2205]
 - “ ... allow users to communicate requirements to network in robust and efficient way.” i.e., signaling !
- ❑ Existiu uma versão experimental anterior
 - Internet Stream protocol: ST-II [RFC 1819]



RSVP: Esquema de Funcionamento

- ❑ A reserva de largura de banda é feita por estruturas de multicast (árvores) que integram os receptores de um fluxo
 - Os emissores anunciam o envio de fluxos com QoS
 - Os receptores de fluxos de dados iniciam as reservas de recursos ao longo do caminho até aos emissores

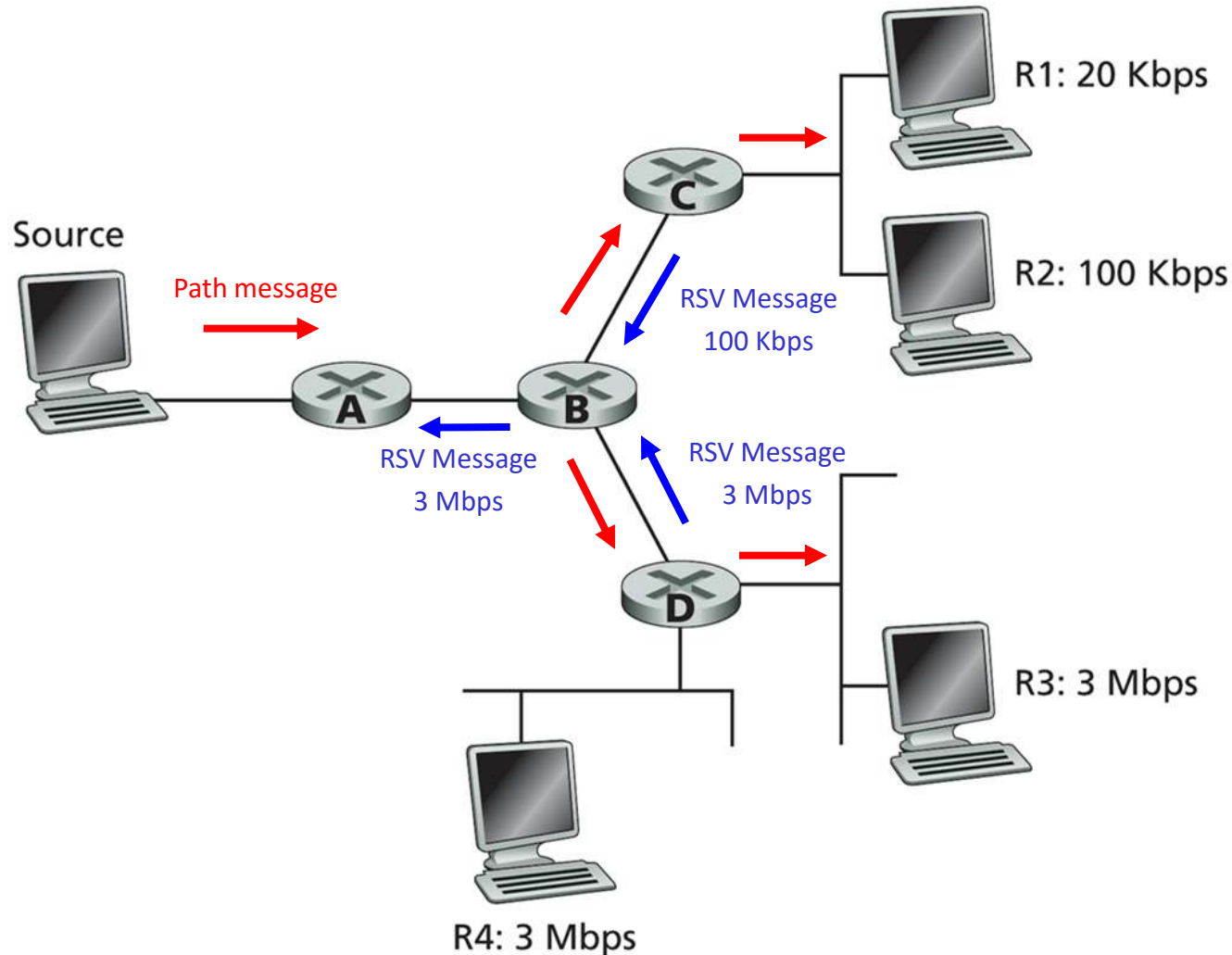


RSVP: Tipos de Sinalização

- ❑ Sinalização **sender-to-network**
 - *path message*: permite dar a conhecer um emissor aos routers
 - path teardown: apaga a rota para o emissor
- ❑ Sinalização **receiver-to-network**
 - *reservation message*: realiza a reserva recursos de um emissor para o receptor
 - reservation teardown: apaga as reservas do receptor
- ❑ Sinalização **network-to-end-system**
 - Erros de rota
 - Erros de reserva

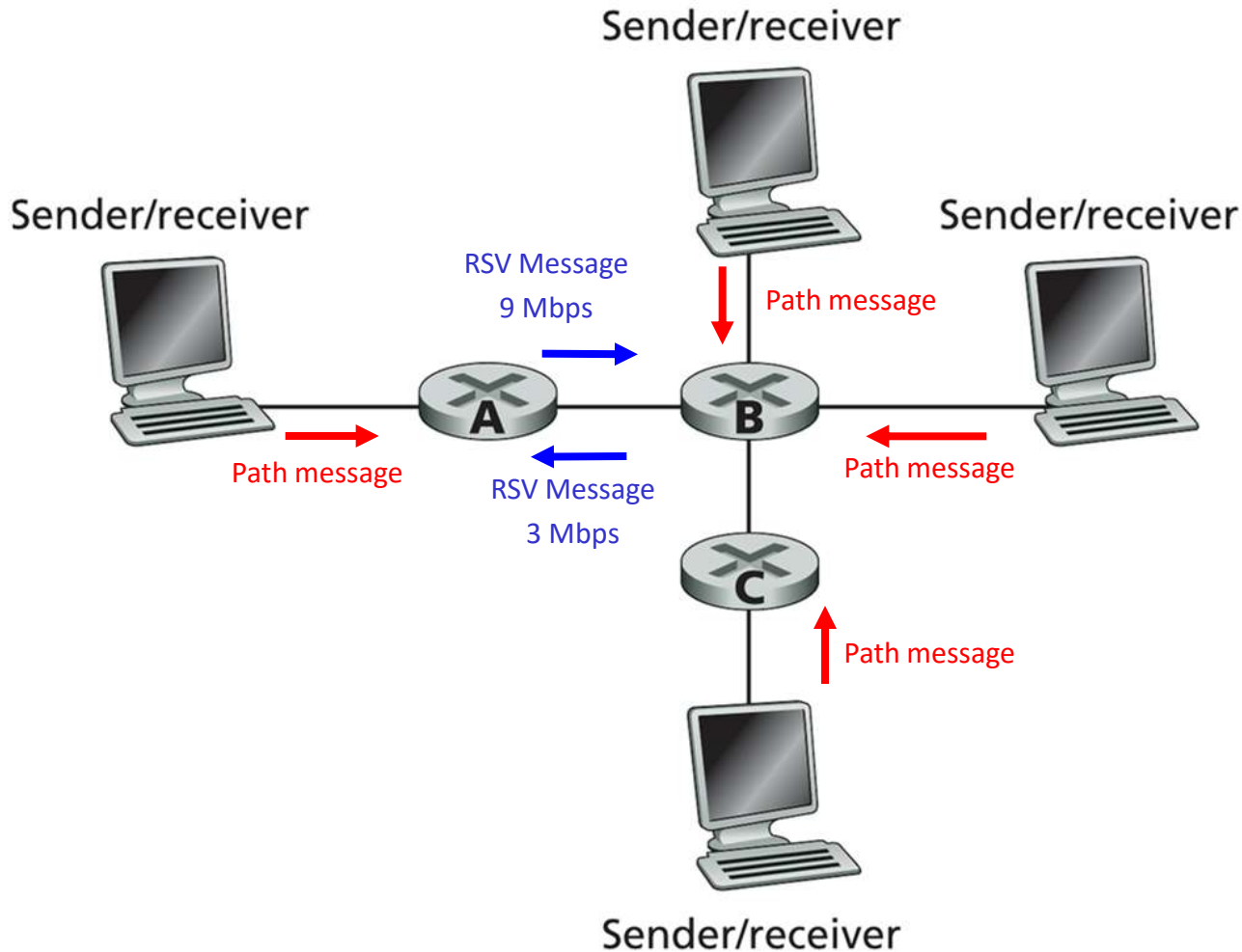


Exemplo: Broadcast de Emissão de Vídeo



- As reservas de banda de cada router correspondem ao máximo dos pedidos dos receptores a que dá acesso visto o conteúdo ser o mesmo para todos

Exemplo: Videoconferência



- Cada participante recebe simultaneamente tráfego dos 3 outros:
 - A reserva deve ser de 3 vezes o tráfego nominal (9 Mbps) em recepção mais uma vez (3 Mbps) em emissão

Redes Multimédia: Sumário

- ❑ Uma área em constante expansão com inúmeros desenvolvimentos em curso
 - Protocolos, aplicações, optimizações, ...
- ❑ Analisámos os vários tipos de aplicações multimédia e os seus requisitos
- ❑ Vimos como tirar partido do “melhor esforço” e os seus limites
- ❑ Estudámos mecanismos de escalonamento e regulação
- ❑ Finalmente abordámos as arquitecturas de QoA para a Internet: Intserv, RSVP, Diffserv
- ❑ Não acaba aqui...
 - Cadeira de 2º Ciclo sobre específica sobre Sistemas Multimédia

