

Redes de Computadores

Transferência de Informação Multimédia

Departamento de Informática da
FCT/UNL

Objetivos do Capítulo

- As aplicações multimédia usam informação digital codificando áudio e vídeo
- Estas aplicações têm requisitos especiais
 - De quantidade de informação transferida
 - De qualidade de serviço da rede
- Mas nem sempre necessitam de absoluta fiabilidade
 - A semântica dos dados transferidos é parcialmente independente da resolução da Informação
 - O conceito de fiabilidade é **subjetivo neste contexto**
- Como podemos transmitir os dados destas aplicações?
- Como é que estas funcionam na Internet?

There are no facts, only interpretations.

- Autor: *Friedrich Nietzsche (1844-1900)*

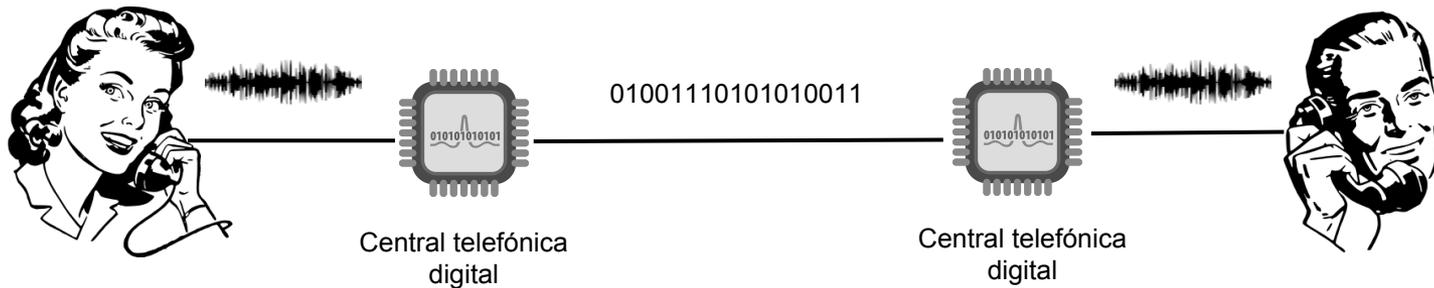
Os Desafios da Informação Multimédia

- Grande volume de dados
 - Quanto mais fidedigna for a informação multimédia, maior a quantidade de informação digital envolvida na transmissão
- O volume de informação a transmitir não é constante
 - Em particular devido à utilização de compressão
- Em alguns cenários não se toleram atrasos
 - Aplicações interativas (e.g., VoIP e jogos)
- Em alguns cenários não se tolera um *jitter* elevado
- Em contrapartida este tipo de informação não requer fiabilidade absoluta pois geralmente os destinatários são os órgãos humanos (vista, ouvido, ...)
 - Que têm grande capacidade de adaptação

Transmissão de Som de forma Analógica



Transmissão de Som Digitalizado



Informação Áudio Digital

- Amostragem do sinal analógico
 - São recolhidas amostras a intervalos de tempo fixos (*sampling*)
 - A frequência de amostragem é importante para a fidelidade (*sampling rate*)
 - Cada amostra corresponde a um valor real arbitrário
- Quantificação de cada amostra
 - Cada amostra é arredondada para um valor numa escala fixa de valores (# de valores distintos limitados ou resolução, representados através de um número fixo de bits
 - Por exemplo, com 8 bits apenas há 256 valores diferentes, com 10 bits há 1024, etc.

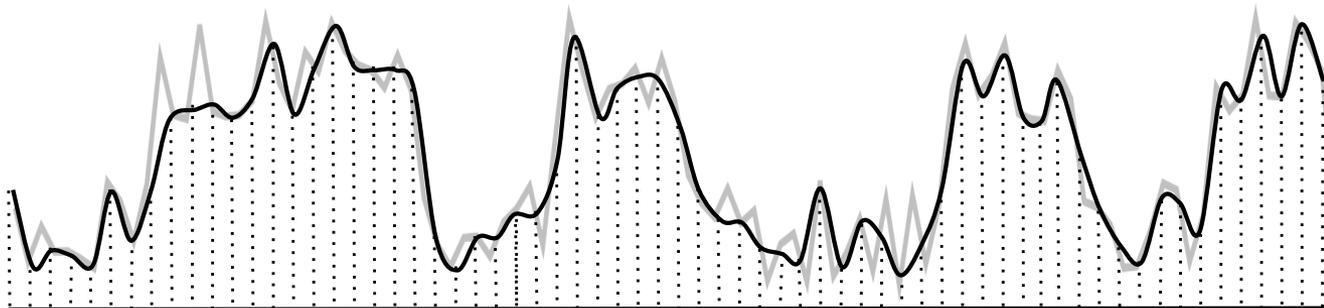
ADC e DAC

- Os dispositivos que realizam esta transformação chamam-se conversores analógico-digital (ADC - *Analog to Digital Converter*). A conversão inversa é feita por conversores digital-analógico (DAC - *Digital to Analog Converter*).
- Cada valor numérico diz-se uma amostra (*sample*), e a sequência de valores corresponde a uma sequência de amostras tomadas a intervalos regulares. A periodicidade, ou seja o intervalo de tempo que separa cada amostra, fica caracterizada pela frequência de amostragem (*sampling rate*).
- A frequência de amostragem mede-se em Hertz
- 1 Hertz corresponde a 1 ciclo por segundo

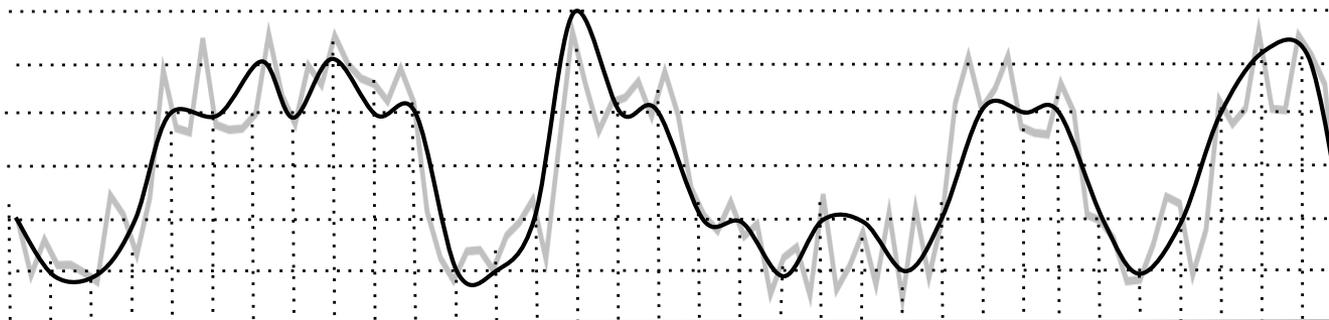
Impacto da Frequência e da Resolução



(a) sinal original



(b) sinal digitalizado com frequência de amostragem f



(b) sinal digitalizado com frequência de amostragem $f/2$ e com 3 bits de resolução

Digitalização do Som sem Compressão

- O sinal analógico é amostrado a um ritmo constante. Exemplos:
 - Telefone: 8,000 amostras / s
 - CD: 44.100 amostras / s
- Cada amostra é quantificada (aproximada por um valor)
 - Por exemplo $2^8=256$ valores possíveis
- Exemplo: 8.000 amostras / s, com 256 valores possíveis cada, implica uma velocidade de transmissão de $8.000 \times 8 = 64,000$ bps ou 64 Kbps
- O recetor volta a realizar a conversão para sinal analógico
 - Implica necessariamente alguma perda de informação

Exemplos Áudio

- Voz (codificação PCM - Pulse Coded Modulation)
 - Ritmo de amostragem: 8000 / s
 - Codificação: 8 bits por amostra
 - Resultado: 64 kbps
- Compact Disc (CD)
 - Ritmo de amostragem: 44.100 amostras /s
 - Codificação: 16 bits por amostra
 - Resultado: 705.6 kbps para mono,
1.411 Mbps para stereo
- Ambos os exemplos têm resolução constante e requerem débito constante (CBR - Constant Bit Rate)

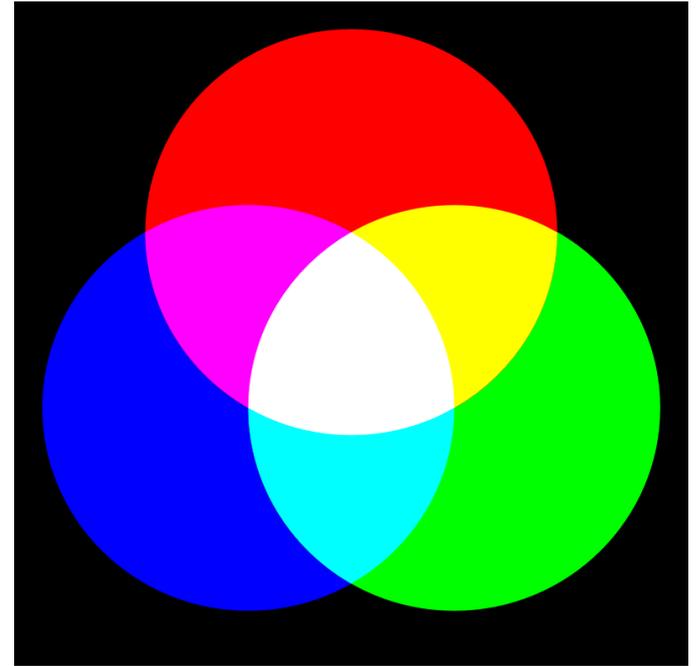
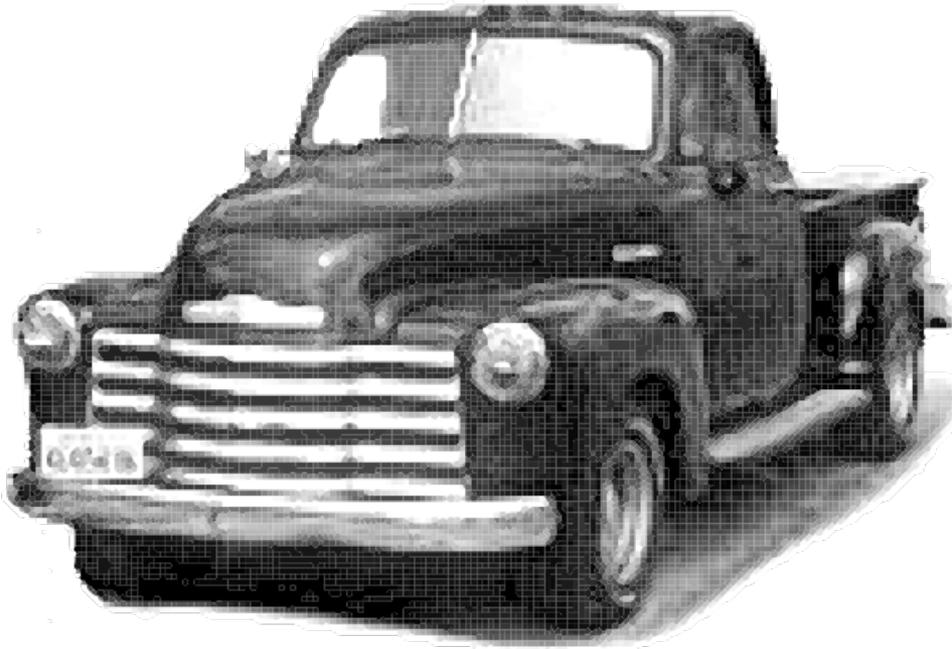
Compressão Áudio

- O débito necessário para transmitir áudio pode ser reduzido através de compressão
 - A qual consiste em reduzir o número de bits transmitidos por unidade de tempo, reduzindo a resolução
 - Ou, melhor ainda, suprimindo os detalhes que o ouvido não distingue ou dispensa
- Exemplos de formatos áudio comprimidos
 - Voz: GSM (13 K bps), G.729 (8 K bps), e G.723.3 (6.4 e 5.3 K bps)
 - Música: MPEG 1 layer 3 (MP3) de 96 a 320 K bps
- O débito passa a ser variável (VBR — Variable Bit Rate)

CODECS

- Um CODEC é um dispositivo hardware e/ou software, ou um circuito VLSI, que realiza a transformação do sinal analógico para uma codificação digital ou vice-versa (*code / decode*).
- Há CODECS simples como os CODECS PCM dos telefones digitais ou das centrais telefônicas. Estes dispositivos apenas transformam o som codificado de forma analógica em digital e vice versa através de uma amostragem de 8 bits com uma frequência de amostragem de 8 KHz.
- Há CODECS muito complexos como os CODECS MPEG-1, 2, 4 existentes nos DVDs, em placas para PC, em recetores TDT, em televisores ou em software. Estes CODECS codificam / separam e comprimem / descomprimem vários canais de vídeo e de voz
- Há CODECS públicos, isto é, normalizados, e CODECS proprietários, isto é cujas formas de codificação / descodificação são patenteados e de utilização sujeita a pagamento.

Imagem Digital



- Uma imagem é convertida num conjunto de pixels por um sensor de imagem
- Um pixel (PIcture Element) ocupa uma superfície e tem uma cor e uma intensidade
- A resolução depende do número de pixels por unidade de superfície
- Codificação RGB (Red, Blue, Green)
 - E.g., 24 bits, com 8 bits por cor

Compressão de Imagem

- **Compressão de uma imagem**
 - Explora-se a redundância espacial (e.g., regiões com a mesma cor)
 - Suprimem-se detalhes que os humanos não distinguem
 - Diminui-se o número de pixels por unidade de superfície
- **Exemplos de formatos / CODECS**
 - Joint Pictures Expert Group (JPEG)
 - Graphical Interchange Format (GIF)

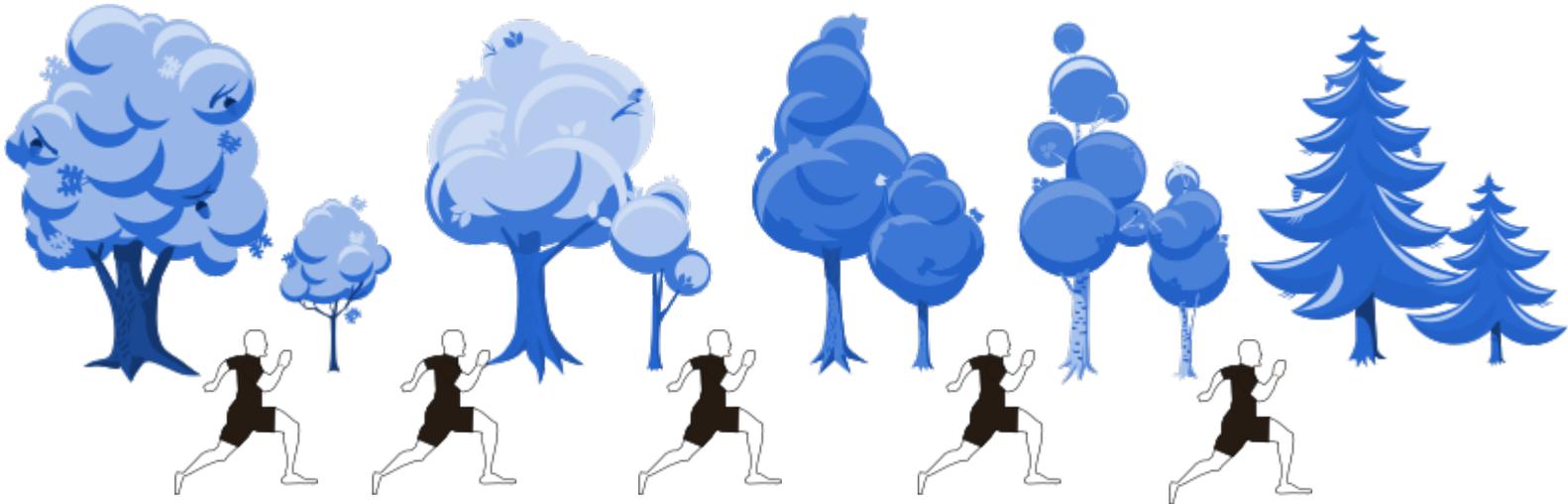


Filmes

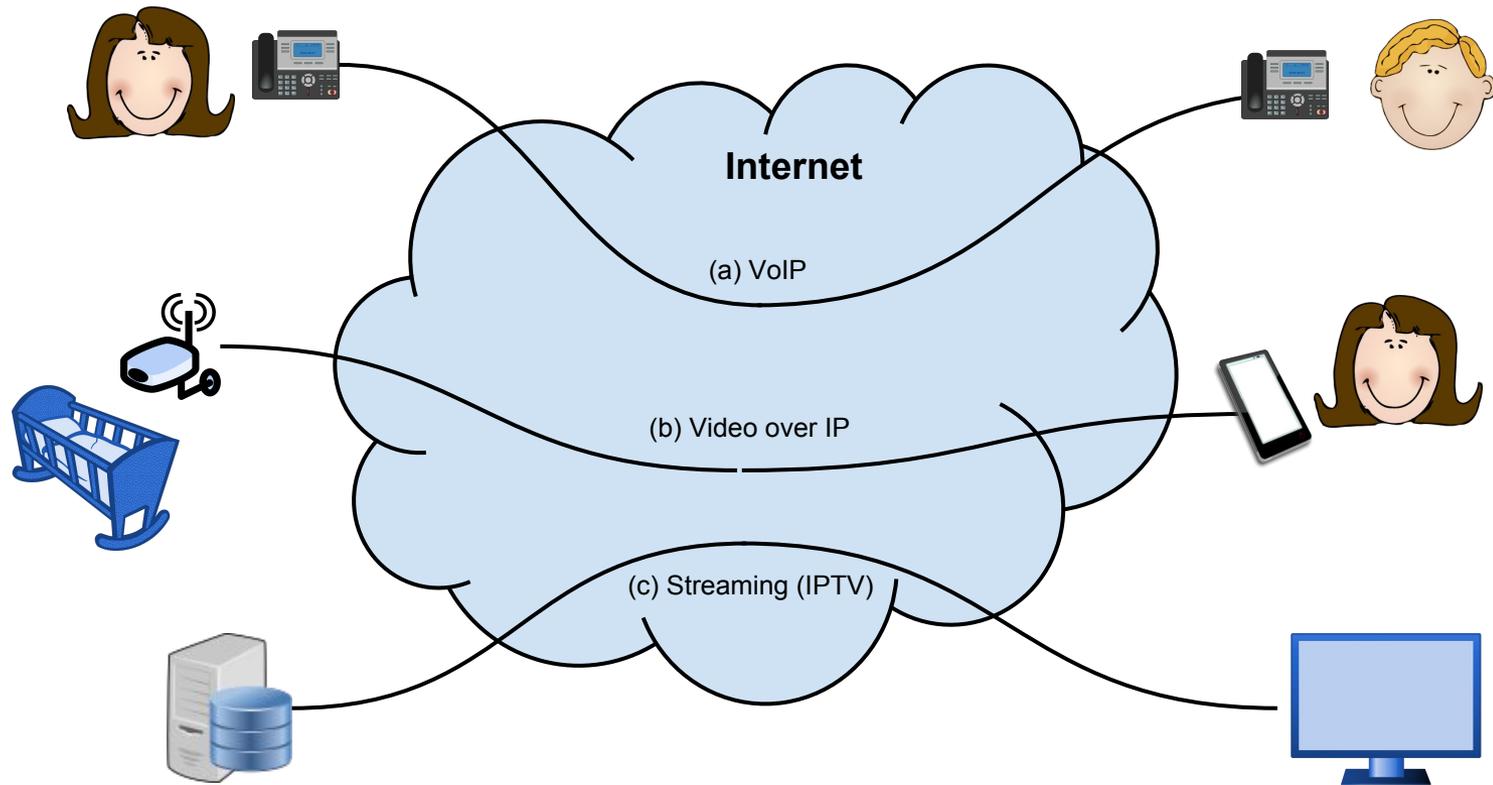
- Uma imagem animada corresponde a uma sucessão de imagens fixas (ditas *frames*)
 - O ritmo da sucessão diz-se o *frame rate*
 - A partir de cerca de cerca de 30 *frames* por segundo a vista humana já não distingue umas imagens das outras
- Um filme poderia ser uma simples sucessão de imagens comprimidas
 - Mas como muitas dos *frames* em sucessão são parecidos, aumenta a oportunidade para maior compressão
 - A maioria dos formatos comprimidos baseiam-se em indicar apenas as diferenças face ao *frame* anterior
 - Mas periodicamente são enviados *frames* completos (e.g. MPEG *I-frames*)
 - Por estas razões o vídeo comprimido é sempre VBR (Variable Bit Rate)

Compressão de vídeo com várias imagens

- Compressão entre imagens
 - Explora a redundância entre imagens
- Exemplos de formatos / CODECS
 - MPEG 1 - qualidade de um CD-ROM (1 a 1,5 M bps)
 - MPEG 4 - qualidade elevada, típica de um DVD (3 a 6 M bps)



Aplicações Multimédia



Requisitos e Alternativas

- Informação diferida ou *live*
 - Transferir como se transfere um ficheiro, isto é transferindo primeiro tudo e só depois visualizar
 - Ou vai-se mostrando conforme se recebe transferindo de avanço uma parte que acomoda variações de qualidade
 - E se a transferência é em *tempo real*? (isto é, o *stream* está agora a ser produzido)
- Com ou sem interatividade
 - De forma a suportar uma conversa ou um jogo, que são mais exigentes
- Um ou mais recetores
 - Difusão ou *broadcasting*

Problemas

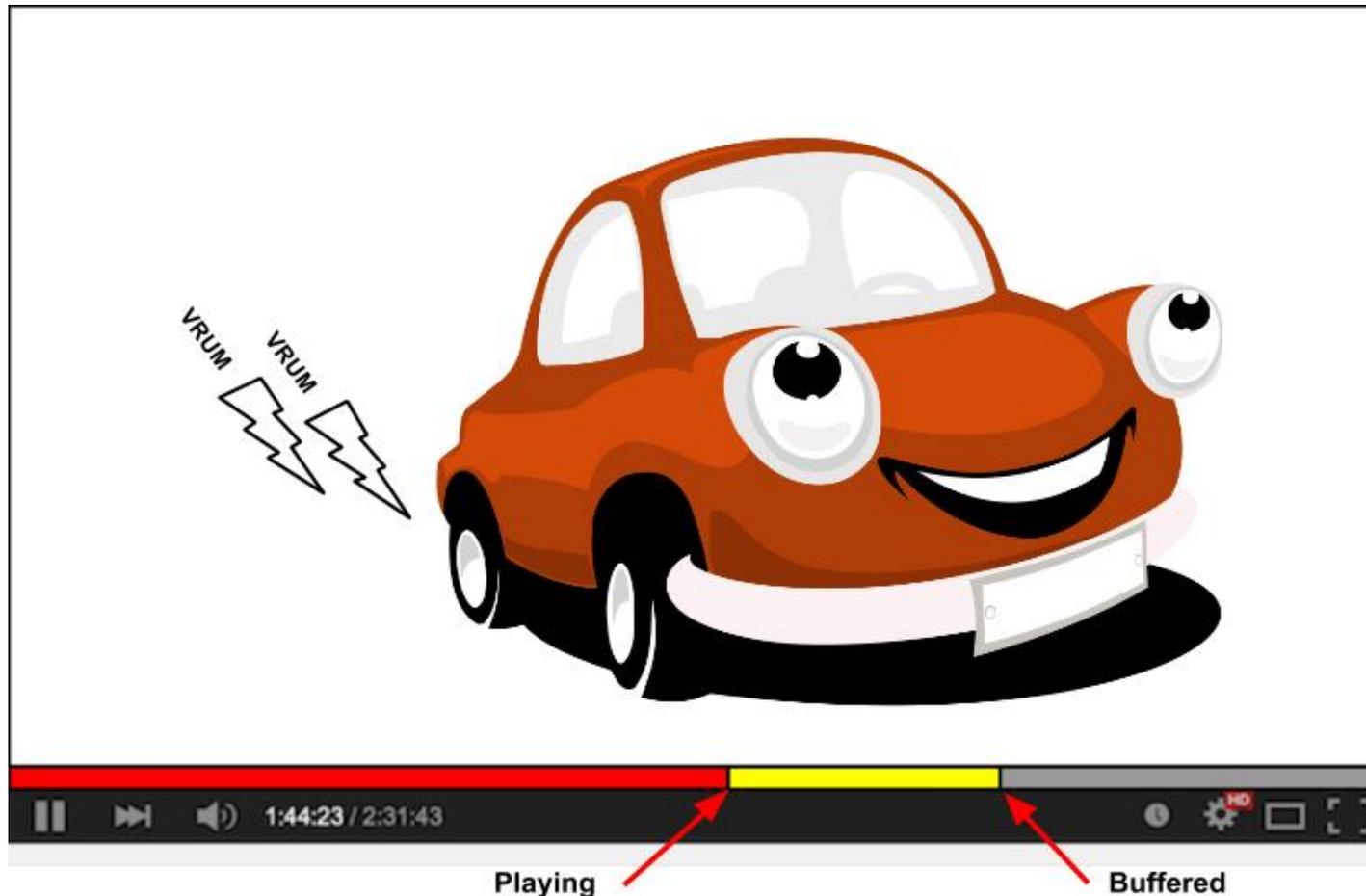
- Qualidade de serviço da rede
 - Débito médio extremo a extremo
 - Taxa de perda de pacotes extremo a extremo
 - Tempo de trânsito extremo a extremo
 - Variância do tempo de trânsito - *jitter*
- Protocolo de transporte
 - TCP é fácil de usar mas não suporta diretamente vários recetores
 - Com TCP, as percas de pacotes traduzem-se em maiores atrasos e maior *jitter*
 - UDP é não fiável mas suporta vários recetores e as perdas de pacotes não se traduzem em maior *jitter*

Débito e *Jitter*

- Débito extremo a extremo
 - Quando há necessidade de alguma simultaneidade (*tempo real*) entre a transmissão e a utilização pelo receptor a codificação e ritmo da emissão têm de ser compatíveis com o débito médio da rede
- No mesmo cenário é também necessário compensar as perdas e o *jitter*
 - Senão a informação multimédia é processada e apresentada ao utilizador ao ritmo de chegada e com eventuais falhas ou variações

Jitter: Solução Típica

- As aplicações com som ou vídeo usam *buffers* com dados de avanço para compensar o *jitter*



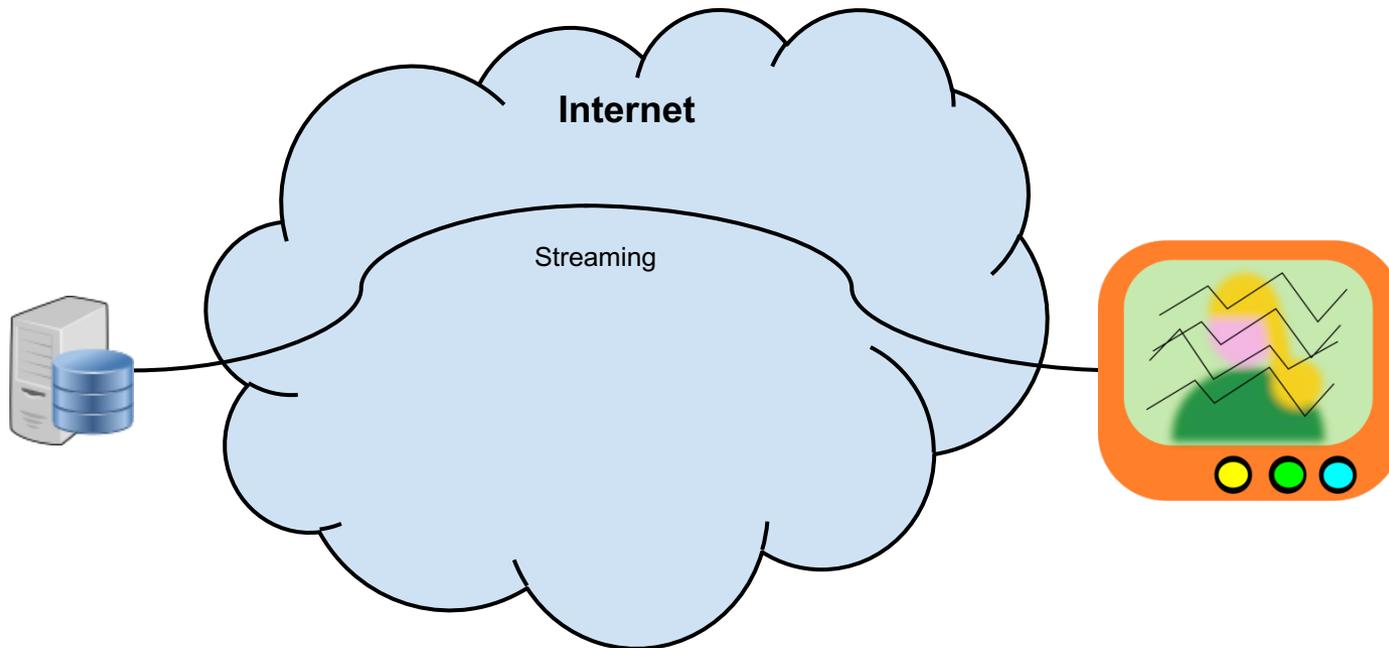
Playout Delay

- Quanto maior melhor ?
 - Nas aplicações uni-direcionais (*progressive streaming*) um grande *playout delay* apenas provoca uma grande espera no início ou quando se muda o ponto de visualização (e.g. "rebobinar")
 - Se for pequeno provoca *rebuffering* frequente
 - Se for grande o utilizador "desiste"
 - No limite poder-se-ia fazer o *download* integral do vídeo e só depois o visualizar

Débito

- Para que o *streaming* ou outras aplicações *tempo real* funcionem é necessário que o débito seja adequado
- Tudo depende da capacidade requerida pelo *stream* e da capacidade disponível na rede
- Se a rede não suporta a capacidade requerida
 - É necessário alterar a resolução e/ou o algoritmo de compressão (CODEC)
 - Senão a qualidade de serviço será muito deficiente

Streaming a Pedido Sobre TCP



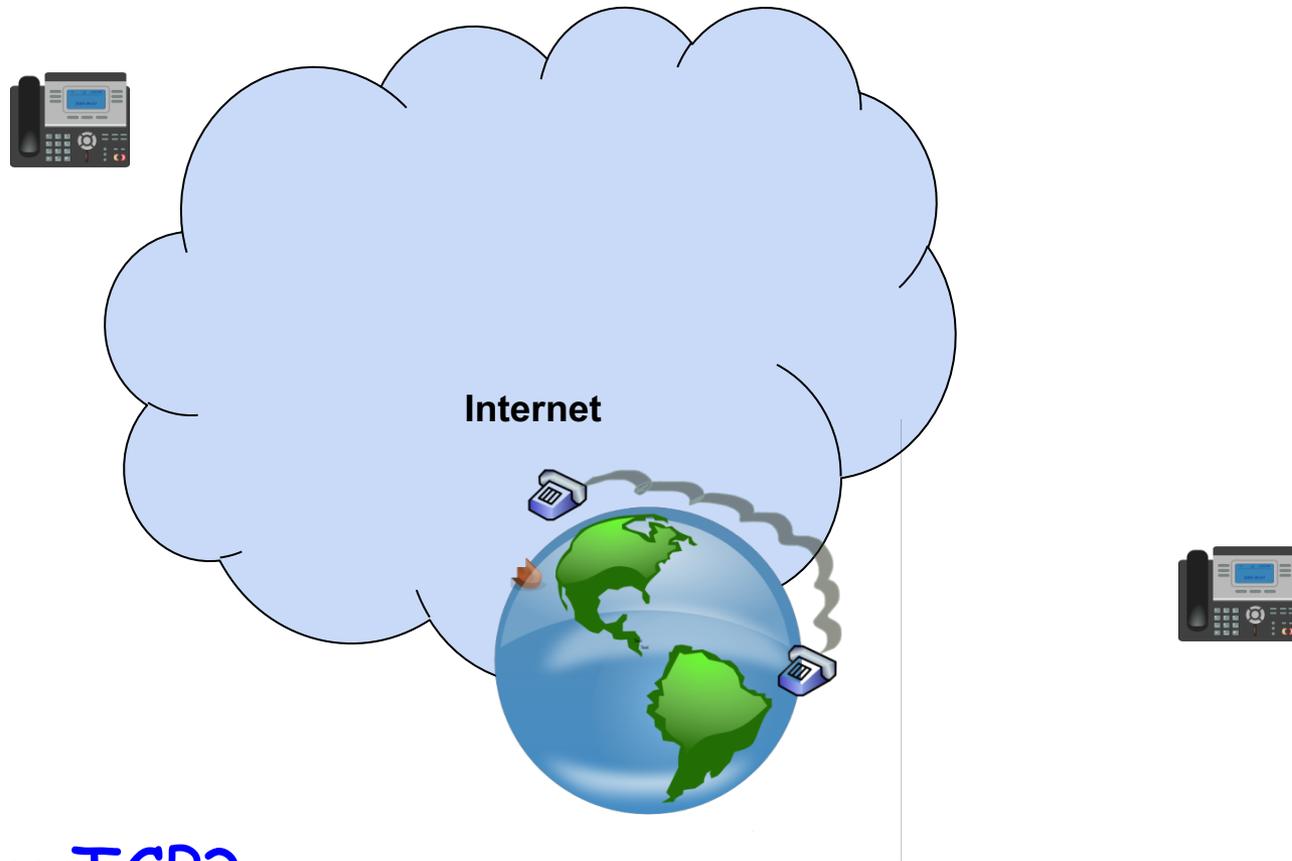
Adaptar a Resolução ao Débito Disponível

- Os diferentes utilizadores têm diferentes capacidades de rede
 - É aborrecido ter de estar a escolher a resolução
 - A capacidade da rede pode variar no tempo
- Porque não tentar variar e adaptar a resolução dinamica e automaticamente?
- DASH (Dynamic Adaptive Streaming over HTTP)
 - Vídeo (*live* ou diferido) é partido em pedaços contendo uma sequência de informação multimédia (e.g. 2 a 10 segundos)
 - Para cada período temporal existem diferentes pedaços para cada resolução
 - Em função do tempo de *download* de cada pedaço, o *player* decide a resolução a usar para o pedaço seguinte

Interatividade

- Numa conversa frente a frente estamos habituados a ver e ouvir o interlocutor - o telefone simula esta situação
- Mas o tempo de propagação e o *jitter* não devem ser perceptíveis
 - Se o tempo de transito for inferior a 150 ms o mesmo não é perceptível
 - Se for superior a 400 ms os interlocutores têm grande dificuldade
- UDP ou TCP?

IP Phone, Vídeo Conference, ...



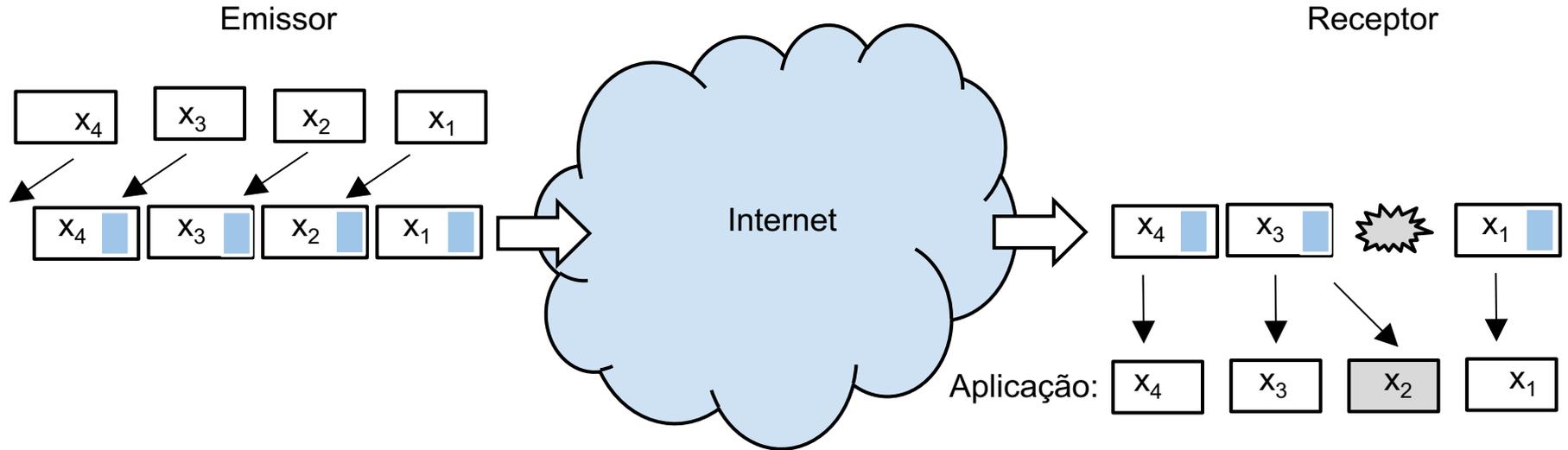
- UDP ou TCP?

- UDP é não fiável mas não aumenta o tempo de trânsito nem o *jitter* quando há perda de pacotes

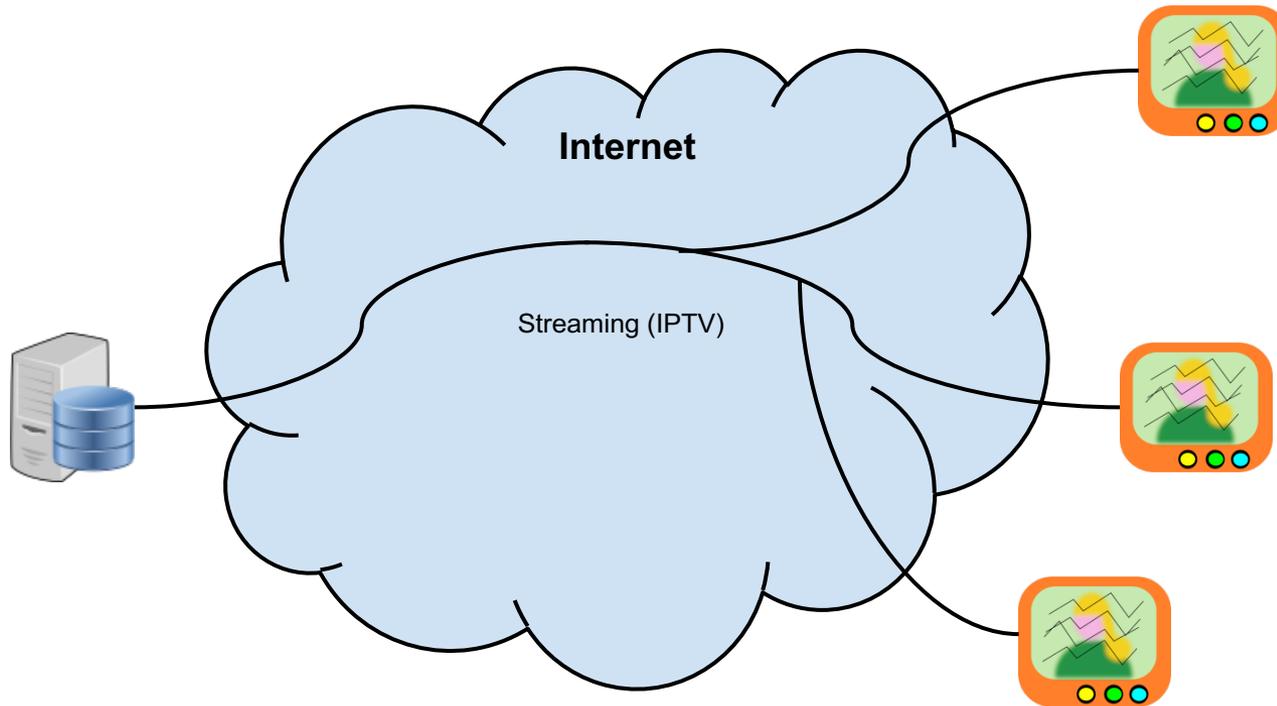
Soluções Baseadas em UDP

- Não sofrem de tantos atrasos devido a perdas de pacotes como no TCP
- Mas também compensam o *jitter com buffers e playout delays*
- Compensam as perdas de pacotes ao nível aplicação através de técnicas especiais
 - Interpolação simples pelo recetor
 - *Forward error correction (FEC)*
 - *Interleaving* (enviar em cada pacote partes diferidas do *stream*)
 - *Interleaving and multi-resolution* (idem com diferentes resoluções)

Forward Error Correction (FEC)



Streaming Multicast

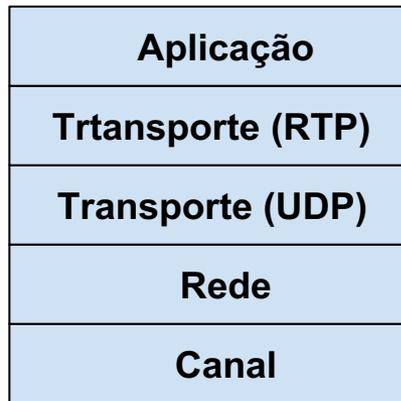


Televisão IP *Live* (*Live* IP TV)

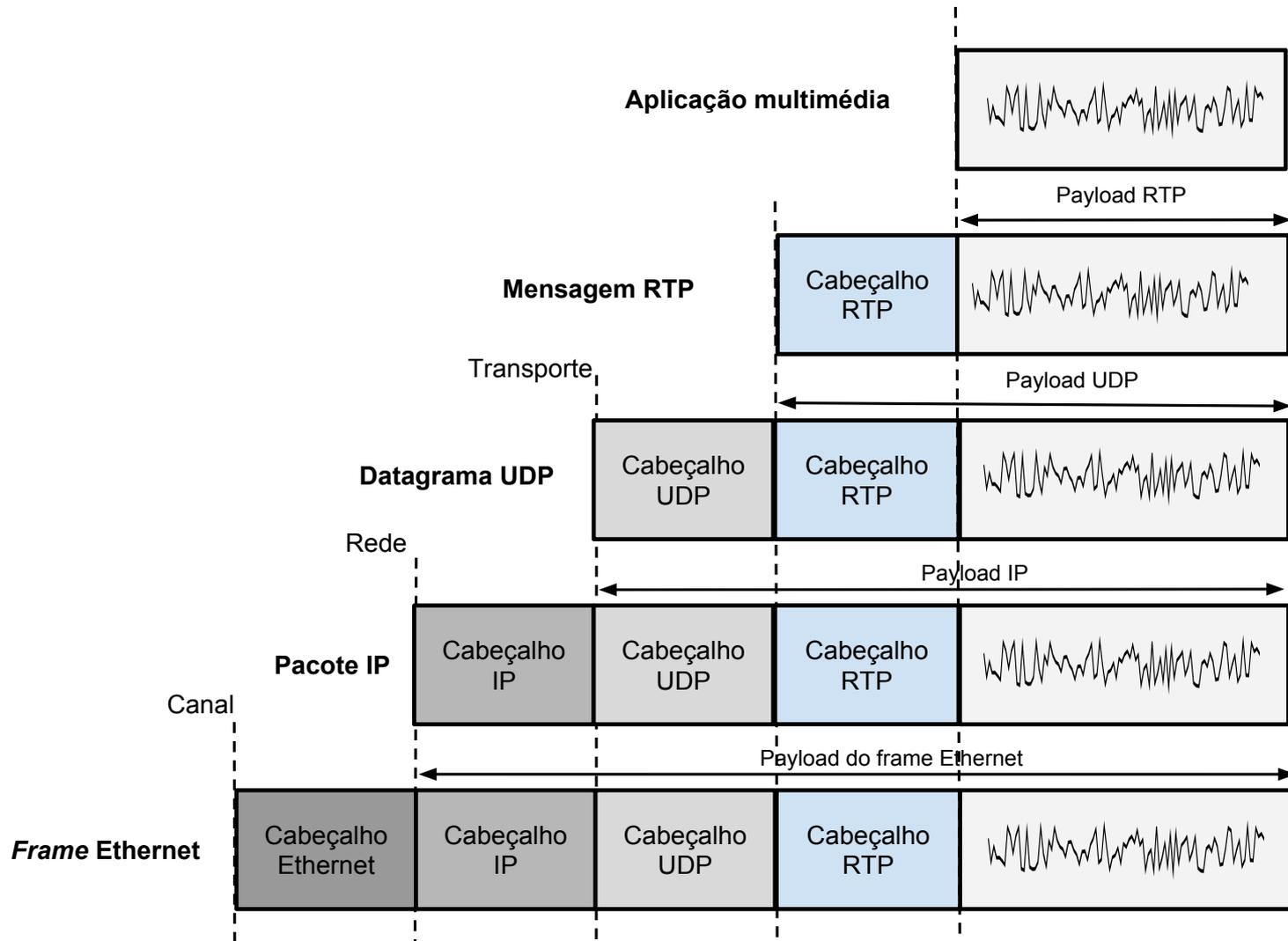
- Cada canal de televisão é difundido através de um *stream* UDP com resolução muito elevada (vários Mbps)
- Esta capacidade só está disponível para os canais difundidos pelo operador para os seus clientes diretos pois requer reserva e afectação de capacidade na rede
- Cada *stream* é difundido para um grupo IP *Multicast* distinto
- Utiliza as técnicas de FEC que vimos atrás
- De alguma forma a TDT utiliza técnicas semelhantes mas o canal de difusão é *broadcasting* sem fios numa gama de frequências reservadas

RTP - Real Time Transport Protocol

- Pacotes contendo informação multimédia transportados através de datagramas UDP
- Formato normalizado para transmissão de informação multimédia em pacotes
- Esses pacotes são geralmente encapsulados em UDP



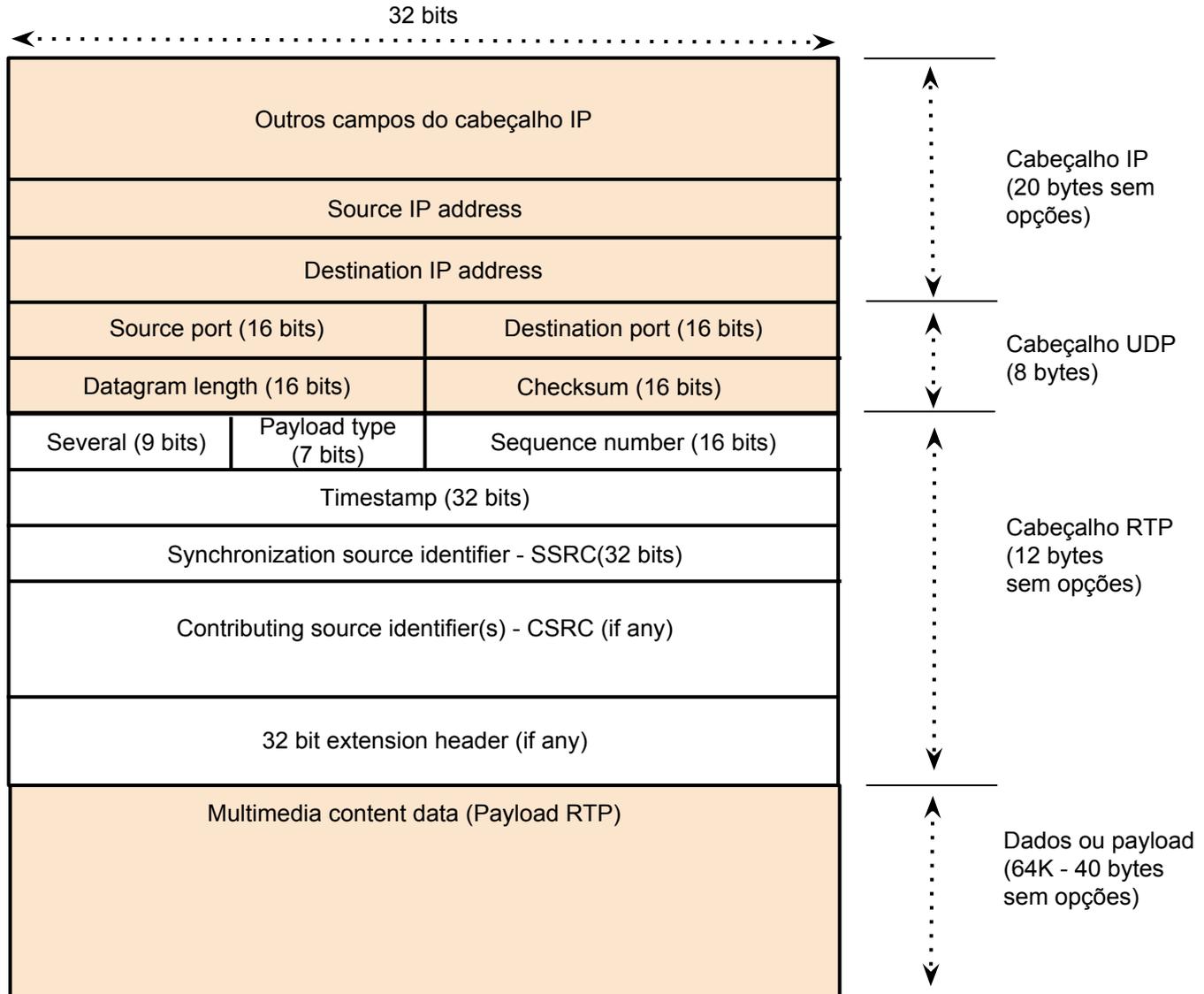
RTP - Real Time Transport Protocol



Real-Time Protocol (RTP)

- O RTP especifica a estrutura dos pacotes que contém áudio e vídeo
- RFC 1889.
- Providencia
 - Identificação do tipo do conteúdo
 - Números de sequência
 - Etiquetas temporais (*timestamps*)
 - Identificação dos *streams*
- O RTP é um protocolo transporte / aplicativo só conhecido dos sistemas finais.
- Os pacotes ou datagramas RTP são transportados em segmentos UDP
- Interoperação: se duas aplicações de IP Phone distintas usam RTP, devem poder funcionar em conjunto

Cabeçalho RTP



Cabeçalho RTP

Payload Type (7 bits): Indica o tipo de codificação (CODEC) usado. Se o emissor resolver alterá-lo, indica-o ao recetor mudando este campo.

Exemplos:

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): incrementado em cada pacote enviado o que permite detetar a perda ou troca dos pacotes

Timestamp: posição temporal da informação no *stream*

Continuação

- **Timestamp field (32 bits)**. Reflete o momento em que os dados contidos no pacote foram gerados em termos do relógio usado para realizar a amostragem para a digitalização:
 - Com som, o *timestamp clock* é tipicamente incrementado de 1 por cada período de amostragem (em cada 125 micro segundos quando se faz amostragem a 8 KHz ou 8.000 vezes por segundo)
 - Se cada pacote contiver 160 amostras (codificadas em 160 bytes), o valor deste campo é incrementado de 160 em cada pacote.
 - O valor deste relógio continua a ser incrementado mesmo que a fonte esteja inativa e não emita pacotes.
- **SSRC field (32 bits)**. Identifica a fonte do *stream*. Uma sessão RTP pode ter vários *streams* e cada um deve ter um valor de SSRC diferente.

Conclusões

- A informação multimédia é volumosa e requer capacidade variável de transferência e também características do tipo *tempo real*
- A Internet tem qualidade de serviço que nem sempre é adequada às necessidades das aplicações multimédia
- No entanto, diversas técnicas podem ser usadas para compensar esses problemas
 - Técnicas de compressão
 - Utilização de playout delays
 - Adaptação dinâmica da resolução à capacidade disponível
 - Utilização de UDP e compensação da qualidade de serviço ao nível aplicacional