

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
ICE
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**Segmentação e adaptação de vídeo para o
servidor de contexto
Utilizando uma abordagem simplificada do DASH**

Carlos Felipe Dias da mata

JUIZ DE FORA
FEVEREIRO, 2014

**Segmentação e adaptação de vídeo para o
servidor de contexto
Utilizando uma abordagem simplificada do DASH**

CARLOS FELIPE DIAS DA MATA

Universidade Federal de Juiz de Fora

ICE

DCC

Bacharelado em Ciência da Computação

Orientador: Eduardo Barrere

JUIZ DE FORA
FEVEREIRO, 2014

SEGMENTAÇÃO E ADAPTAÇÃO DE VÍDEO PARA O SERVIDOR DE CONTEXTO

Utilizando uma abordagem simplificada do DASH

Carlos Felipe Dias da mata

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO ICE DA UNIVERSIDADE
FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA
COMPUTAÇÃO.

Aprovada por:

Eduardo Barrere
Doutorado em Engenharia de Sistemas e Computação (Conceito CAPES 7).
Universidade Federal do Rio de Janeiro, UFRJ, Brasil.

Marcelo Ferreira Moreno
Doutorado em Informática. Pontifícia Universidade Católica do Rio de Janeiro,
PUC-Rio, Brasil.

Marcelo Caniato Renhe 2
Doutorado em andamento em Engenharia de Sistemas e Computação (Conceito CAPES
7). Universidade Federal do Rio de Janeiro, UFRJ, Brasil.

JUIZ DE FORA
DE FEVEREIRO, 2014

Primeiramente a Deus que me possibilitou estudar nesta universidade, graças a sua vontade e bondade cheguei

ate aqui

A minha querida mãe que não se encontra mais entre nós mas que me educou e me ajudou (e ainda ajuda de longe)

me chamando atenção, ficando feliz quando as coisas davam certo.

Ao meu pai que além de pai é meu melhor amigo, que me apoia e me ensina até hoje sempre me ajudando

certamente não estaria aqui sem ele.

Aos meus irmãos que torceram pelo meu sucesso e me motivaram a seguir em frente

Aos professores que me ensinaram com boa vontade as disciplinas do curso.

Aos meus amigos do curso, pelos bons momentos e pela motivação que me deram durante esta jornada,

compartilhamos momentos de alegria ao passar em disciplinas difíceis, também desespere e nas vésperas das

provas! Onde um ia explicando a matéria para o outro, foi bom estudar com vocês!

Resumo

Com o crescimento da internet tanto em número de usuários quanto em largura de banda, aliado a grande variedade de aparelhos as aplicações multimídia passaram a ser uma realidade, os usuários passaram a ter condições de assistir e transmitir vídeos através de aparelhos moveis e outros tipos de hardware, isso aumentou a interação entre pessoas e dispositivos, tanto é que uma boa parte do trafego na internet corresponde a conteúdo multimídia. Inserido neste âmbito esta o servidor de contexto a ser projetado e implementado pelo grupo de pesquisa da UFJF, chamado LApIC, que está em desenvolvimento por alunos de mestrado e TCC, na qual o presente trabalho esta incrustado. Existem problemas a ser resolvidos: hardwares diferentes têm características, objetivos e processamento diferentes. Logo cada aparelho vai precisar de um tipo diferente de codificação de vídeo (formato, codec, resolução, bitrate, etc) temos como objetivo implementar no servidor um modulo capaz de realizar a adaptação de vídeo no servidor afim de atender as necessidades do mesmo, que no caso do vídeo são: transcodifica-lo para diferentes hardwares e preparar a mídia para ser transmitida. Além disso, existem fatores como as diferenças na velocidade de conexão que influenciam a transmissão do vídeo, o grande tamanho dos arquivos de vídeo junto ao fato de usuários não gostarem de esperar para começar a assistir, que são problemas a ser resolvidos. A solução adotada foi utilizar o streaming onde a mídia é segmentada em pequenas partes que são transmitidas em sequência ate que todo o conteúdo seja transmitido, o tamanho destes segmentos influencia na qualidade da imagem e da transferência do vídeo.

Palavras-chave: adaptação, segmentos, contexto, multimidia, Segmentação de vídeo, Metadados, Transcodificação.

Abstract

With the growth of the internet both in number of users and bandwidth , combined with wide variety of multimedia devices applications have become a reality , users have come to be able to watch and stream videos through mobile devices and other types of hardware , this increased interaction between people and devices , so that is a great deal of traffic on the Internet corresponds to multimedia content . Within this context this server context to be designed and implemented by the research group UFJF called LAPIC , which is in development by MA and TCC students, on which this work is embedded. There are problems to be solved : hardware have different characteristics , objectives and different processing . Soon every appliance you will need a different video encoding (format , codec , resolution , bitrate , etc.) type we aim to implement a module on the server able to accomplish adaptation of video in the server in order to meet the needs of even that in the case of the video are : transcodes it to different hardwares and prepare media for transmission . In addition , there are factors such as differences in connection speeds that influence the transmission of video , the large size of video files together with the fact users do not like to wait to start watching , which are problems to be solved . The solution adopted was to use where media streaming is segmented into smaller parts that are transmitted in sequence until all content be transmitted, the size of these segments influences the quality of image and transference of the video.

Keywords: adaptation , segments , context, multimedia , video segmentation , Metadata , Transcoding .

Agradecimentos

A Deus que me deu todos os instrumentos e condições para concluir este projeto

A todos os meus parentes, pelo encorajamento e apoio.

Ao professor Dr. Eduardo Barrere pela orientação, amizade e principalmente, pela paciência, sem a qual este trabalho não se realizaria.

Aos professores do Departamento de Ciência da Computação pelos seus ensinamentos e aos funcionários do curso, que durante esses anos, contribuíram de algum modo para o nosso enriquecimento pessoal e profissional.

“Nunca, Nunca, Nunca deixe alguém te dizer que aquilo que você acredita é bobagem, que de repente o teu sonho não vai dar certo...”

Renato Russo

Sumário

Lista de Figuras	7
Lista de Tabelas	8
Lista de Abreviações	9
1 Introdução	10
2 FUNDAMENTAÇÃO TEÓRICA	13
2.1 Tecnologias de transmissão	13
2.2 Tecnologias de Suporte a vídeo	14
2.3 Arquitetura e protocolos de distribuição de vídeo na internet	18
2.4 Adaptação de vídeo	23
2.5 Segmentação de vídeo	26
2.6 Metadados	28
2.7 Qualidades e características do vídeo transmitido	30
2.8 Ciência de Contexto	33
3 SERVIDOR DE CONTEXTO	36
3.1 Arquitetura do servidor de contexto	36
4 Estudo de Caso: Implementação do Módulo de vídeo	39
4.1 Especificação do Módulo de Vídeo	39
4.2 Descrição do processamento no servidor de vídeo	39
4.3 Funcionamento do sistema	41
4.4 Processo de segmentação de vídeo	42
4.5 Decisões de Implementação	44
4.6 Resultados obtidos	46
4.7 Problemas encontrados	49
4.8 Tamanho das medias armazenadas	52
5 Conclusão	54
6 referência bibliografica	55

Lista de Figuras

1.1	fluxo de dados na internet[6]	11
2.1	Quadro chave puro	17
2.2	Quadro preditivo	17
2.3	Exemplo com quadro Bi-preditivo	18
2.4	Tipos de quadro[25]	18
2.5	arquitetura do sistema 3GP-DASH[18]	22
2.6	adaptação de vídeo difundida em ambientes de mídia para apoiar os terminais e redes heterogêneas[2]	24
2.7	tipos de adaptação de vídeo (a) Transcodificação de Formato -(b) Seleção / Redução -(c) substituição -(d) Síntese[2]	27
2.8	Largura de banda x Tamanho dos segmentos [6]	28
2.9	[18] semântica do MPD	29
2.10	[23] Comparação 320x240 e 640x480	31
2.11	[22] Resultados pesquisa sobre as qualidades dos vídeos com diferentes bit-rates	32
3.1	Arquitetura do Bus (orientada a eventos)	37
3.2	arquitetura do servidor de contexto	38
4.1	estados no modulo de vídeo	40
4.2	arquitetura do modulo de vídeo	41
4.3	Processo de criação dos segmentos	42
4.4	Processo de criação dos segmentos 2	43
4.5	MPD gerado	46
4.6	Parâmetros MPG to MP4	46
4.7	Parâmetros MPG/AVI to MPG	47
4.8	Parâmetros AVI to MP4	48
4.9	Sincronia ideal [20]	50
4.10	Problemas de sincronia[20]	51
4.11	Tamanho gasto por cada tipo de segmento agrupado pela qualidade	53

Lista de Tabelas

Lista de Abreviações

DCC	Departamento de Ciência da Computação
UFJF	Universidade Federal de Juiz de Fora
MPD	The Media Presentation Description
DASH	Dynamic Adaptive Streaming over HTTP
RTP	Real Time Transport Protocol
RSTP	Real Time Streaming Protocol
HTTP	Hyper Text Transfer Protocol

1 Introdução

Aplicações cientes de contexto têm como característica poder se adaptar ao ambiente que estão sendo executadas e às preferências do usuário. Para ampliar e potencializar uma aplicação dessa natureza é interessante um servidor de contextos bem estruturado com diferentes funcionalidades como a captura e persistência de contextos (usuário, dispositivo, temporal, espacial, etc.) e possibilidade de recuperação desses contextos, conforme a necessidade e características apresentadas pelas aplicações. Uma forma interessante é reunir informações de contexto sobre o dispositivo como por exemplo a Qualidade de Serviço (QoS) da rede, desta forma é possível potencializar a adaptação de vídeo [1]. A utilização de mídias adaptativas é um assunto que começou a ser tratada com maior ênfase no começo da década passada, essa área oferece um rico conjunto de técnicas para tentar solucionar questões desafiadoras em aplicações que envolvam a mídia vídeo [2], visando transformar a mídia de entrada em uma saída em forma de vídeo ou Multimídia modificada pela utilização de manipulações em vários níveis (sinal, estrutural) a fim de satisfazer as limitações de recursos diversificados e as preferências do usuário [3]. Desta forma, os usuários podem acessar e interagir com o conteúdo em diferentes tipos de redes e em diferentes dispositivos. Portanto, as novas aplicações têm a necessidade de lidar com a enorme variação das limitações dos recursos como largura de banda, capacidade de exibição, a velocidade de CPU, alimentação, entre outras.

Este projeto é parte integrante de um servidor de contexto e está focado no tratamento de vídeo . É fundamental ter um modulo dedicado a este tipo de mídia, pois a maior parte do fluxo de dados transmitido em aplicações multimídia, através da internet, corresponde à arquivos multimídia [6] a figura 1.1 o detalhadamente o trafego de dados na internet para dispositivos móveis.

Apesar da alta relevância existem vários fatores que prejudicam a experiência do usuário com arquivos multimídia [6], causando frustração e impaciência ao usuário, estes fatores são: formato de mídia improprio, protocolo inadequado, necessidade de instalar plug-ins, demora a iniciar a transmissão, baixa qualidade, etc. Esses problemas ocorrem

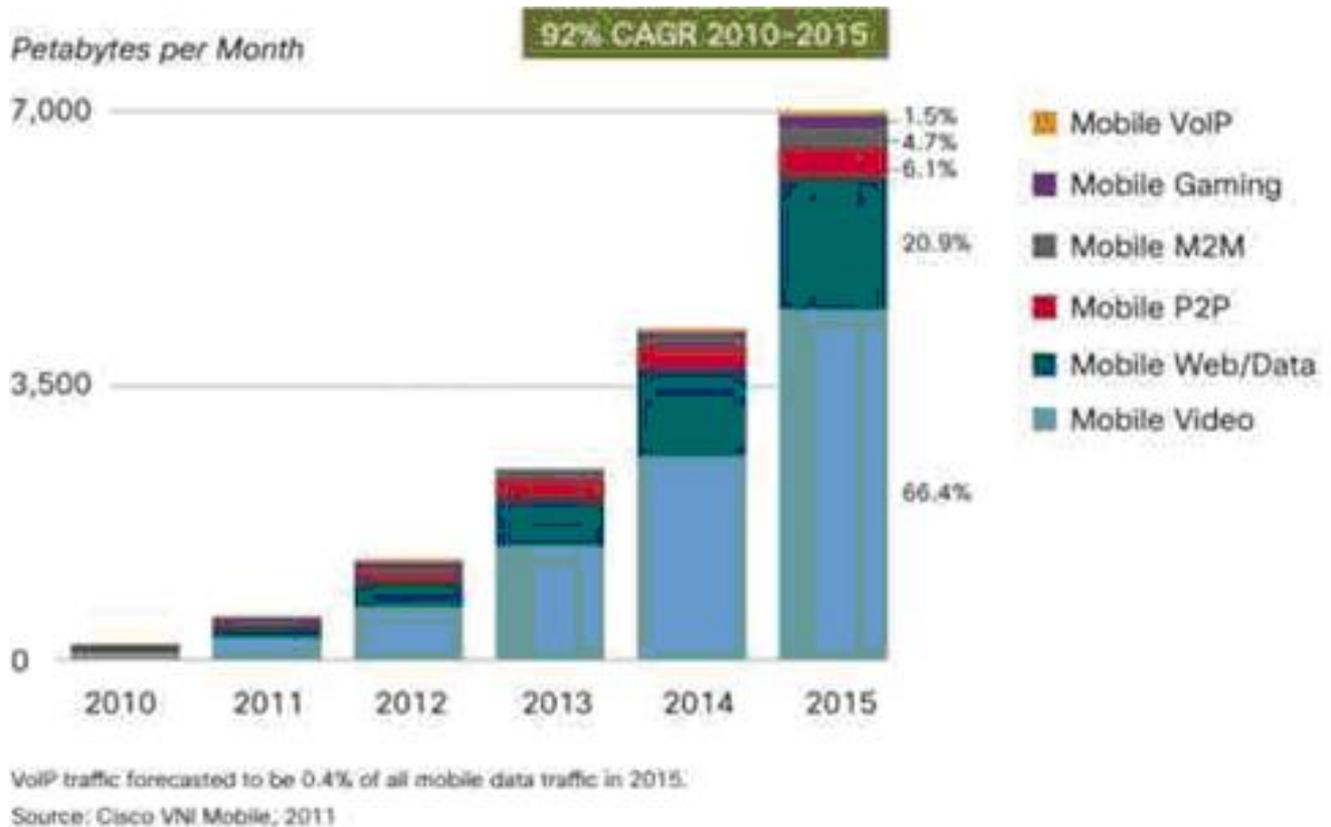


Figura 1.1: fluxo de dados na internet[6]

por causa das diferenças entre conexões de rede e diferentes tipos de hardware.

Quando temos diferença nas redes e no hardware devemos ter mecanismos de adaptação de vídeo [2]. Uma ferramenta de adaptação de vídeo deve tratar um ou vários arquivos multimídia atendendo suas necessidades, ou melhor, personalizações.

Um dos objetivos deste trabalho é adotar uma solução para a segmentação, dentro do projeto que envolve o desenvolvimento do servidor de contexto e aplicações que consumam os conteúdos (contextos e mídias relacionados ao vídeo) por ele providos.

Durante o capítulo 2 é apresentada a fundamentação teórica, que serve como base para o presente projeto, onde abordaremos os elementos de vídeo, de rede, protocolos para transmissão de vídeo e meta dados, além de descrever o processo de adaptação e segmentação de vídeo. O capítulo 3 apresenta um breve resumo sobre a arquitetura do servidor de contexto proposto pelo departamento de ciência da computação da UFJF também será mostrado o relacionamento entre o servidor e o módulo de vídeo. No capítulo 4 é detalhado o módulo de vídeo aonde mostraremos sua especificação e quais decisões a

nível de arquitetura, projeto implementação foram tomadas além dos resultados obtidos e os problemas encontrados, finalmente no capítulo 5 será é apresentada a conclusão e os futuros projetos.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Tecnologias de transmissão

Para transmitirmos o conteúdo multimídia do servidor é importante compreender alguns princípios básicos em redes. A camada de transporte é responsável pela movimentação de dados de forma eficiente e confiável entre processos em execução nos equipamentos conectados a uma rede de computadores, deve poder regular o fluxo de dados e garantir confiabilidade, assegurando que os dados cheguem a seu destino sem erros e em sequência. Os serviços oferecidos pelo protocolo IP não oferecem confiabilidade. Problemas comuns como congestionamento, perda ou falta de ordenação de pacotes não são tratados. Entretanto as aplicações (HTTP, FTP e SMTP, por exemplo) necessitam prover um serviço de qualidade para o usuário. A camada de transporte pode oferecer um serviço confiável de entrega de dados das aplicações utilizando um serviço não confiável prestado pela camada de rede. Os principais serviços oferecidos pela camada de transporte são: Protocolos que oferecem serviços orientados e não orientados a conexão. Os serviços não orientados a conexão tem conceito semelhante ao funcionamento do protocolo IP. Já o serviço orientado a conexão, primeiramente estabelece uma comunicação entre usuários finais e só depois começa a transmissão.

- **TCP:** protocolo da camada de transporte presente na arquitetura Internet responsável por oferecer confiabilidade á transmissão, é orientado a conexão afim de ter o controle dos pacotes enviados e conseguir efetuar a fragmentação do arquivo a ser transmitido sem que hajam perdas, o TCP precisa que os usuários finais tenham o controle do que está sendo enviado, este protocolo especifica três fases durante uma conexão: estabelecimento da ligação, transferência e término de ligação. Para estabelecimento da conexão o TCP necessita que: O cliente inicie a conexão enviando um pacote TCP com a flag SYN ativa e espera-se que o servidor aceite a conexão enviando um pacote SYN+ACK. Se, durante um determinado espaço de

tempo, esse pacote não for recebido ocorre um timeout e o pacote SYN é reenviado. O estabelecimento da conexão é concluído por parte do cliente, confirmando a aceitação do servidor respondendo-lhe com um pacote ACK [26].

- **UDP** é um protocolo da camada de transporte não confiável e não orientado a conexão. Fornece apenas os serviços de endereçamento e fragmentação, não provendo confiabilidade (controle de fluxo, erro, congestionamento). Isso indica que o UDP não adiciona serviços ao protocolo IP. O que nos leva então a utilizar o UDP? O UDP por ser mais simples possui um cabeçalho menor gerando um menor overhead. Ideal para algumas aplicações onde a velocidade é mais importante que a confiabilidade como aplicações multimídia ao vivo. Afinal não faz sentido algum receber um trecho de um arquivo vídeo que já passou.

Para se realizar a transmissão de vídeo no servidor para um cliente utilizaremos algum destes protocolos.

2.2 Tecnologias de Suporte a vídeo

Para implementar o nosso módulo de vídeo devemos compreender e aplicar as técnicas descritas a seguir pois vídeo é nosso objeto base de estudo

- **Animação** Animação refere-se ao processo segundo o qual cada quadro de um filme é produzido individualmente, podendo ser gerado por computação gráfica ou através da fotografia de uma imagem desenhada repetidamente, situação na qual cada nova fotografia apresenta uma pequena alteração em relação á fotografia anterior . Quando os fotogramas são ligados entre si e o filme resultante é assistido a uma velocidade mínima de 16 quadros por segundo, ocorre uma ilusão de movimento continuo (ilusão de continuidade) devido a limitação da visão humana [9]).
- **Taxa de quadros** A taxa de quadros refere-se ao frame rate que a medida da frequência em que um dispositivo de processamento de imagens produz imagens consecutivas através de um dispositivo audiovisual qualquer, como uma câmara de cinema ou vídeo, uma webcam, um projetor cinematográfico ou de vídeo, etc.

Significando assim o número de quadros que tal dispositivo registra, processa ou exibe por unidade de tempo. A unidade de medida do Frame rate é o FPS (Frames Por Segundo). Exemplos: Cinema: 24 FPS, TV Analógica Preto-e-Branco: 30 FPS, TV Analógica Colorida: 29,97 FPS, TV Digital 1080i: 30 FPS, TV Digital 720p: 60 FPS, Monitor: variável, podendo superar os 60 FPS; O FPS é uma importante propriedade de vídeo, porém na implementação não estamos alterando-o em nem uma transcodificação.

- **Sistemas analógicos e digitais** segundo[10] em um sistema de vídeo analógico, tal como o sistema de televisão é muito difícil separar as imagens do vídeo, isto pois não há o armazenamento de imagens. Se nós desejarmos apresentar uma imagem de um vídeo durante, por exemplo, 5 minutos, esta imagem deve ser capturada, transmitida e apresentada repetidamente por 5 minutos. Ao contrário, em sistemas digitais há o armazenamento de imagens, se uma imagem deve ser apresentada durante certo tempo, temos que capturar e enviar a imagem uma única vez. Em sistemas analógicos, todos os componentes, câmera, transmissor, receptor e monitor, devem operar sincronamente. No caso de sistemas digitais estes componentes são independentes: a câmera registra uma imagem/cena, mas o transmissor pode transmitir seletivamente as imagens com base no seu conteúdo, poder de computação ou na largura de banda disponíveis; o monitor mostra aquilo que é disponível. A imagem mostrada não vai piscar se faltarem quadros subsequentes pois a tela será atualizada constantemente com o conteúdo disponível. A enorme capacidade de processamento de dados da visão humana se reflete nos requisitos para armazenamento e transmissão de imagem em movimento. Um dos problemas da utilização de sistemas de vídeo digitais está associado aos requisitos de largura de banda de transmissão. Assumindo a largura de banda de um vídeo analógico é B MHz, então a largura de banda mínima necessária para o canal de transmissão é B MHz. No caso de sistemas digitais, onde são transmitidos dados binários sobre um canal de B MHz, de acordo com o Teorema de Nyquist a taxa máxima de bits que o canal deve suportar é $2B$ Mbits/s. De qualquer maneira, quando um vídeo é digitalizado no formato vermelho, verde e azul, temos uma taxa de bits de $49B$ Mbits/s. Esta

comparação sugere que a transmissão não compactada de vídeo digital requer um canal de transmissão com um significativo aumento de largura de banda comparado aos sistemas analógicos. Mas se a compressão de vídeo digital for utilizada numa taxa de compressão 24:1 ou maior, nós precisaremos de um canal de transmissão cuja largura de banda é mais baixa que os dos vídeos analógicos.

- **Compressão de vídeo** Segundo[9] um vídeo pode consumir uma quantidade enorme de armazenamento no disco, 1 hora de vídeo sem compressão ocupa cerca de 72,4 Gb, isso tornaria inviável o uso dessa mídia. Existem diferentes tipos de compressão:

- **Compressão da cromaticidade:** Retira do vídeo tons de cores que o olho humano dificilmente será capaz de identificar;
- **Compressão intraquadros** semelhante ao JPEG (Joint Photographic Experts -Group). Aproveita e remove redundâncias e semelhanças dentro de um quadro
- **Compressão interquadros:** aproveita a semelhança entre quadros consecutivos do Vídeo devido à semelhança entre quadros;

Os algoritmos que aplicam as técnicas de compressão em vídeos são chamados de codecs (codificadores/decodificadores). Os principais codecs de vídeo são: - Sem Perdas: HuffYUV, MSU, MJPEG, H.264 e FFmpeg Video 1; -Com Perdas: Xvid, DivX, RMVB, WMV, Theora, Sorenson, MPEG

- **Tipos de quadro** Inicialmente temos apenas o quadro chave(I) que é um quadro que contém toda a informação referente a imagem do vídeo esses quadros ocupam um espaço maior, mas ao aplicarmos a compressão inter quadros, substituímos alguns quadros I por quadros preditivos(P) eles contêm apenas a diferença entre o quadro I anterior e o seguinte. Estes quadros ocupam um espaço muito menor que o quadro I, também temos o quadro bi-preditivo(B) que economiza ainda mais espaço pois utiliza as diferenças entre o quadro atual e ambos os quadros anteriores e posteriores sejam eles P ou I para especificar o seu conteúdo. As figuras 2.1, 2.2, 2.2, ilustram este processo:



Figura 2.1: Quadro chave puro

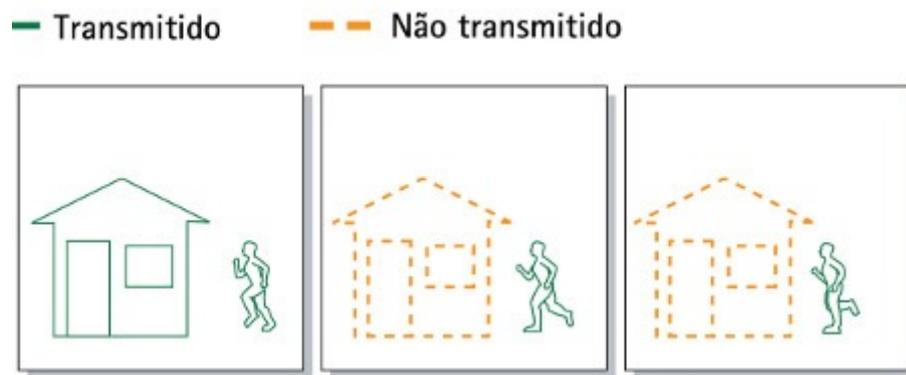


Figura 2.2: Quadro preditivo

Um exemplo da aplicação destas técnicas é o algoritmo de compressão H.264 que atualmente é o padrão de compressão de vídeo mais popular e eficiente. sua primeira versão foi lançada em 2003. O H.264, também conhecido como MPEG-4 foi originalmente concebido para transmitir dados de vídeo através da rede, tais como conferências, vídeos, filmes. Ele também encontrou aplicação em compressão de vídeo de alta definição, incluindo Blu-ray. O alto desempenho do H.264 é o resultado de previsão e estimativa de métodos para quadros selecionados do vídeo. Este codec implementa os 3 tipos de frames: I - Intra Coded, P - Predictive, B - Bi - predictive. O tamanho de cada quadro depende de muitos fatores, mas pode-se supor que o tamanho da frame P é cerca de 60 por cento do tamanho das frames I, e B pode ser reduzido até 10 por cento do tamanho da frame I.

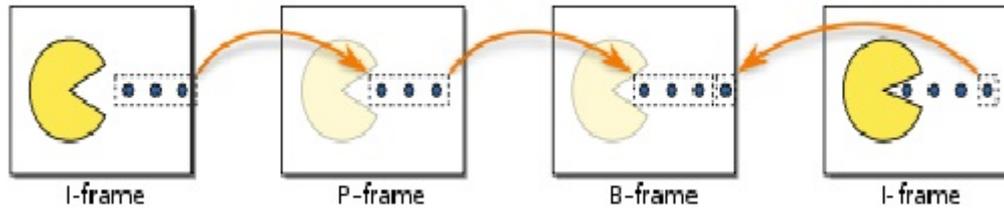


Figura 2.3: Exemplo com quadro Bi-preditivo

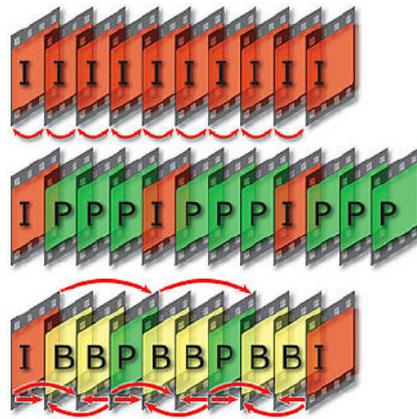


Figura 2.4: Tipos de quadro[25]

2.3 Arquitetura e protocolos de distribuição de vídeo na internet

Para transmitirmos arquivos pela internet necessitamos de estabelecer algum protocolo para a comunicação entre emissor e receptor ser bem sucedida, no caso de arquivos de vídeo não é diferente, nesta seção serão mostrados os principais protocolos utilizados para este tipo de mídia na atualidade. HTTP streaming e especialmente o streaming adaptativo por HTTP tem recebido muita atenção nos últimos anos. Sistemas proprietários como Microsofts Smooth Streaming [11], Adobes HTTP Dynamic Streaming [12] or Apples HTTP Live Streaming [13] implementam abordagens parecidas, aproveitando a rede de entrega de conteúdo existente (CDN) e infra-estruturas de cache de proxy cujos custos são significativamente menores do que outros produtos concorrentes.

Para distribuir vídeos na internet existem duas maneiras destinadas, são elas: Download progressivo ou Streaming. O aumento da disponibilidade de conexões de internet de alta velocidade em redes móveis como WLAN/3G/4G , junto ao enorme crescimento de

smartphones nos últimos anos aliado ao baixo custo do HTTP tem tornado o streaming de vídeo cada vez mais importante. mesmo com o grande crescimento no fluxo de vídeo na rede móvel, HTTP download/streaming básico não é adequado para ambientes móveis com consideráveis flutuações de banda larga.

- **Streaming** É uma tecnologia para envio de informações multi-mídia na Internet através de pacotes IP. Em geral com um protocolo de controle de sessão (ex.: RSTP) controlando a transmissão dos dados e de forma que após a reprodução os dados não sejam armazenados na máquina do cliente[19]. Apresenta inconvenientes, como: Dependência em relação à taxa de compressão e à largura de banda disponível na Internet. Pode ainda ocasionar diminuição de qualidade de reprodução ou mesmo momentos de interrupção. Os conteúdos estão armazenados em bibliotecas e são enviados, pela rede, quando estes são requisitados pelos softwares clientes (players) Os principais protocolos de transporte utilizados pelo etreaming são: RTP e HTTP.
- **RTP/RSTP** Há alguns anos atrás, a maioria dos sistemas de live streaming utilizava os protocolos RTSP (Real Time Streaming Protocol) e RTP (Real Time Protocol), afinal o RTP foi projetado para definir formatos de pacotes para conteúdo de áudio e vídeo. Os softwares utilizados para reproduzir os conteúdos usavam RTSP para fazer o pedido de streaming ao servidor e, após o pedido ser confirmado, entravam em loop para receber os conteúdos enquanto o servidor entrava em loop para enviar a mídia, aplicando o protocolo RTP sobre UDP. Neste modelo de comunicação RTSP/RTP, é o servidor quem decide qual o fragmento do conteúdo que deve ser enviado no próximo momento, baseando-se nos conhecimentos(acknowledgements) e informações sobre erros enviadas pelo cliente. O papel do cliente é apenas receber os dados enviados pelo servidor e reproduzi-los. Também este protocolo funciona bem em gestão de redes IP. Entretanto, na atualidade da internet as redes gerenciadas foram substituídas por redes de distribuição de conteúdo (CDN), muitas das quais não suportam streaming de RTP/UDP ou RTSP. Além disso, os pacotes RTP/UDP ou RTSP muitas vezes não são permitidos através de firewalls. Em paralelo a isso o grande crescimento da WEB fez com que o custo de transporte dos dados de áudio ou vídeo através de pacotes muito pequenos diminuísse. Conteúdos multimídia podem

agora ser entregues com mais eficiência mesmo através de segmentos maiores (em contraste com os pequenos pacotes) usando HTTP. Graças a estes fatores modelo HTTP live streaming começou a prevalecer.

- **Download Progressivo** É uma alternativa para transmissões convencionais de vídeo, aonde um cliente pode fazer o download normalmente através de HTTP de um arquivo de mídia que é encapsulado de forma contínua e podendo ser executada partir do armazenamento local.

O download progressivo utiliza download normal do protocolo HTTP com o método GET ou pedidos parciais GET. A diferença entre de download regular e download progressivo é que o conteúdo pode ser criado progressivamente para download e o terminal reconhece que o conteúdo é adequado para ser transmitido utilizando o download progressivo. Um cliente que solicita um download de mídia contínua pode decidir começar playout(exibir, dar play) dos dados de mídia encapsulados antes que o download do arquivo de mídia seja concluído[18]

Este método não utiliza buffer e funciona melhor com aplicações multimídia pequenas (tamanho dos diretórios que podem reproduzir é limitado). É um método que descarrega todo o ficheiro e começa a reproduzi-lo assim que a parte já transferida permita a reprodução de forma ininterrupta (ao contrário do streaming que envia pequenas porções do conteúdo em cada momento). A reprodução é de alta qualidade, independentemente da velocidade da conexão com a Internet porem conexões de baixa velocidade esperam mais tempo antes de começar a reprodução.

Segundo [19] o download progressivo apresenta algumas desvantagens, são elas:

- **Segurança** O Download progressivo copia completamente a mídia preparando o hardware do usuário, exatamente como acontece com um download de um arquivo qualquer, fazendo com que esta solução seja relativamente insegura, dado que existem diversas formas extremamente simples de se obter, copiar e distribuir ilegalmente o conteúdo.
- **Ineficiência para transmissão ao vivo** apesar de existem algumas técnicas que permitem que isto aconteça elas são muito ineficientes pois e necessário

identificar no sinal de vídeo codificado que está sendo gerado a posição do último key-frame, para que a cada nova conexão seja atendida a partir deste ponto. Com isso, é necessário incluir um header semelhante ao header de um arquivo previamente codificado, além de definir um tamanho fictício de conteúdo na resposta do servidor web. Com isso, o cliente processa o conteúdo pensando que o mesmo é um arquivo comum, apesar de não ter um final definido

- **Inflexibilidade** Não se ajusta automaticamente à velocidade de conexão do usuário, além de não permitir ir até o final, ao visionar o vídeo, não pode avançar na sua reprodução até que o seu download esteja completo

- **Adaptive Streaming over HTTP** fornece serviços de streaming via HTTP. Isso permite a entrega de conteúdo de servidores HTTP padrão para um cliente HTTP-Streaming e habilita o conteúdo em cache através de caches padrão do HTTP. O descritor de apresentação da mídia (MPD) descreve a apresentação de mídia, ou seja, uma apresentação limitada ou não do conteúdo de mídia. Em particular, ela define formatos para anunciar identificadores de recursos para os segmentos além de prover o contexto para os recursos identificados dentro de uma apresentação multimídia. No âmbito desta parte da norma, os identificadores dos recursos são exclusivamente URLs. No entanto, esta especificação permite adicionalmente a restrição destes URLs por um atributo de intervalo de bytes.

O MPD proporciona informação suficiente a um cliente para fornecer um serviço de transmissão ao o utilizador através do acesso aos segmentos por meio do protocolo indicado no esquema dos recursos definidos, no contexto da presente especificação é exclusivamente sobre o HTTP/1.1

- **Dinamic Adaptive Streaming over HTTP** Mesmo o Adaptive Adaptive Streaming over HTTP ainda não é suficiente para suprir às necessidades atuais no que tange serviço de transporte para conteúdo multimídia, pois fatores como oscilação na qualidade da banda e diferença nos displays dos diversos dispositivos não foram contemplados até então. Temos a necessidade de adaptar dinamicamente o conteúdo de vídeo a ser transmitido em função do contexto computacional (qualidade da co-

nexão, tipo de aparelho, etc) em que nos encontramos no exato momento. Para[6] O fluxo de vídeo deve se adaptar às capacidades de largura de banda diferentes, a fim de entregar ao usuário um fluxo de vídeo contínuo, sem interrupções na melhor qualidade possível para o momento, que é conseguido do dynamic adaptive streaming over http(DASH) A Figura 2.5 mostra uma arquitetura em que os formatos definidos nesta especificação são tipicamente utilizados. Caixas com linhas contínuas indicam os dispositivos que são mencionados nesta especificação como hospedar ou processar os formatos definidos nesta especificação enquanto caixas tracejadas são conceituais ou transparente. A presente especificação diz respeito à definição de formatos que são acessíveis sobre a interface para o cliente 3GP-DASH, indicado pelas linhas sólidas. Quaisquer outros formatos ou interfaces não estão no escopo desta especificação. No cenário de implantação considerado, presume-se que o cliente 3GP-DASH tem acesso a um MPD. O MPD fornece informações suficientes para que o cliente 3GP-DASH para fornecer um serviço de streaming para o usuário, solicitando Segmentos de um servidor HTTP e demultiplexação, decodificar e processar os dados de mídia incluídos de forma adequada[18].

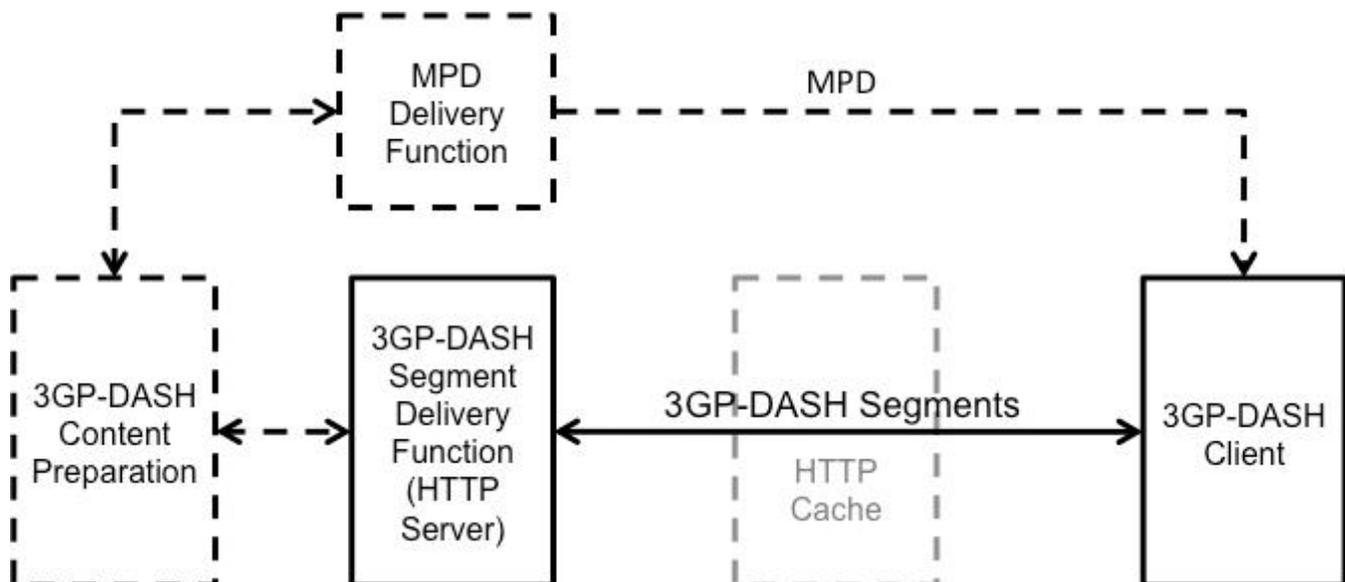


Figura 2.5: arquitetura do sistema 3GP-DASH[18]

2.4 Adaptação de vídeo

A adaptação de vídeo é uma das principais senão a principal funcionalidade do módulo de vídeo implementado neste projeto. Segundo [2] uma entidade se refere a uma unidade básica de vídeo (pixel, frame, componentes sintáticos e semânticos) que sofre alguma ação durante o processo de adaptação, operadores de diferentes adaptações podem ser definidos para os diversos tipos de entidades. Por exemplo, um quadro de vídeo pode ser reduzido mediante a resolução, a qualidade espacial, ou omitido, a fim de reduzir a largura de banda total. Um componente semântico (como uma história em um programa de notícias) pode ser resumido em uma forma visual ou textual. Um subconjunto de frames em uma sequência pode ser removido a fim de gerar uma versão compacta do vídeo,

Existem diversas configurações para um arquivo de vídeo, como formato, resolução, codec, quadros por segundo, etc. Essas configurações influenciam na qualidade, tamanho e processamento da mídia. Elas também são fundamentais para que o vídeo possa ser exibido em diferentes tipos de hardware, por exemplo, a resolução do Iphone é 320x480 já a da TV-HD é 1920x1080. Para fornecer o vídeo para os dois tipos de equipamentos citados, temos como possibilidade: A. Enviar o mesmo vídeo para os dois. Desta forma, ou o vídeo terá uma resolução muito grande para o dispositivo ou muito inferior à resolução suportada pelo dispositivo. B. Adaptar, em tempo real, a resolução, conforme a resolução do display. Isso gera diversos problemas, entre eles o grande processamento necessário na conversão e a ausência de escalabilidade no provimento do serviço. C. Gerar versões diferentes da mídia, conforme os possíveis displays previstos que consumirão o vídeo. Esta solução demanda pré-processamento do vídeo e grande espaço de armazenamento.

As soluções B e C envolvem a adaptação de mídia. É importante frisar que adaptação de vídeo e codificação de vídeo são duas coisas diferentes, a transcodificação é um tipo de adaptação de vídeo.

- **Adaptação:** Uma ferramenta ou sistema de adaptação de vídeo adapta um ou mais programas de vídeo afim de gerar uma nova apresentação com um formato de vídeo ou multimídia que atenda as necessidades dos usuários diferentes situação

situações, conforme a figura 2.6

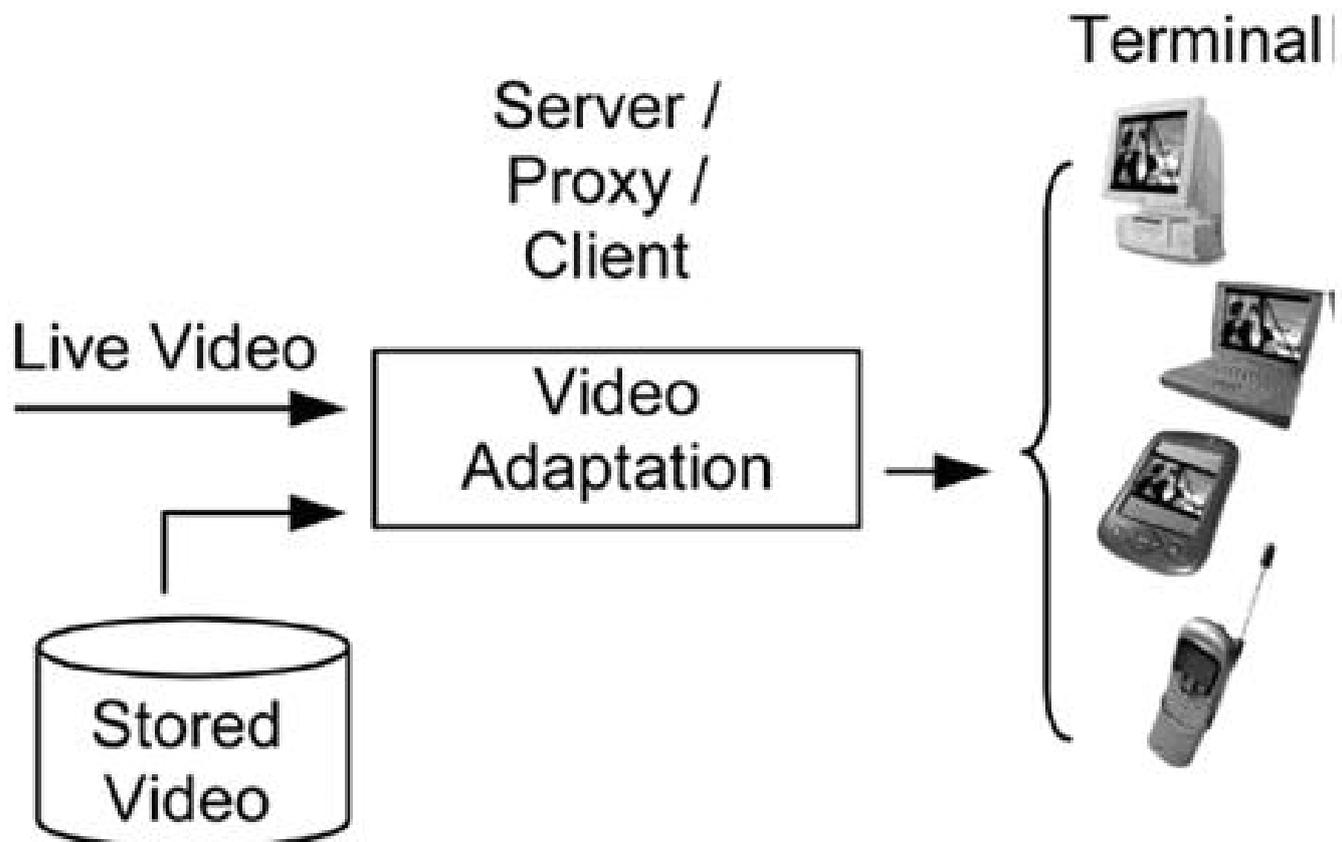


Figura 2.6: adaptação de vídeo difundida em ambientes de mídia para apoiar os terminais e redes heterogêneas[2]

- **Transcodificação:** : pode ser de dois tipos: homogeneia que modifica um conjunto de parâmetros do vídeo sem modificar o seu padrão de compressão, ou seja, realiza conversão da taxa de bits, da resolução espacial, da resolução temporal e mudanças da codificação VBR e heterogênea, onde se realiza a conversão de padrões de compressão ou a conversão entre os formatos entrelaçado e progressivo, Além desses dois tipos, um outro tipo de transcodificação consiste em incluir funções adicionais à sequência de vídeo codificada, tais como proporcionar resistência a erros [1].

Logo adaptação de vídeo se difere da codificação de vídeo no seu âmbito e os locais de aplicação pretendidos. Há uma grande variedade de abordagens de adaptação: o nível de sinal em relação nível estrutural contra nível semântico, transcodificação - seleção sumarização, ou largura de banda em função da potência versus tempo limitado.

Adaptação normalmente leva um vídeo codificado como entrada e produz um vídeo com codificação diferente ou uma apresentação multimídia aumentada. Outra diferença é que a adaptação geralmente é implantada nos locais intermediários, tais como proxy entre servidor e cliente, embora eles possam ser incluídos nos servidores ou clientes em algumas aplicações[2].

-Tipos de adaptação

- **Transcodificação de Formato:** Um processo de adaptação de base é para transcodificar o vídeo de um formato para outro, a fim de fazer o vídeo compatível com o novo ambiente de utilização. Isto não é surpreendente quando ainda existem muitos formatos diferentes que prevalecem nos setores de aplicação diferentes, como a radiodifusão, eletrônicos de consumo, e streaming na Internet. Uma implementação simples é concatenar o decodificador de um formato com o codificador do novo formato.
- **Seleção / Redução:** Seleção / Redução: É adequada em situações na qual os recursos são limitados, nesse tipo de adaptação alguns componentes da entidade são trocados com finalidade de promover a poupança de alguns recursos. Esses sistemas são geralmente implementados através de seleção e redução de alguns elementos em uma entidade de vídeo como fotos e quadros em um videoclipe, pixels em um quadro de imagem, planos de bits em pixels, componentes de frequência na representação transformada, etc. Além disso, alguns destes sistemas são tipicamente considerados como formas de transcodificação: mudar a taxa de bits, taxa de quadros, ou a resolução de um fluxo de vídeo codificado existente. Redução envolve uma etapa de seleção para determinar quais componentes específicos devem ser eliminados.
- **substituição:** Esta classe de adaptação substitui elementos selecionados em uma entidade de vídeo com os seus homólogos menos caros, com o objetivo de preservar a utilidade total percebida. Por exemplo, uma sequência de vídeo pode ser substituído por quadros ainda (por exemplo, quadros-chave ou visuais de representação) e narrativas associadas à produção de uma apresentação de slides. O requisito global de largura de banda pode, assim, ser drasticamente reduzido. Se a redução da largura

de banda não é uma grande preocupação, adaptações de tais métodos podem ser utilizadas para fornecer eficientes auxiliares de navegação visual que podem ser usados como resumos visuais, bem como os índices eficientes aos pontos importantes no vídeo original. Sem que o teor de substituição não tenha sido extraído do vídeo original.

- **Síntese:** adaptação de Síntese vai além das classes acima mencionados por sintetizar novas apresentações de conteúdo com base no resultados da análise. A meta é para lhe proporcionamos uma experiência mais abrangente ou uma ferramenta mais eficiente para a navegação. Para obter por exemplo, mosaicos visuais (ou panoramas) podem ser produzidos por de análise de movimento e construção de uma cena. A exibição estendida fornece uma experiência aprimorada em compreender as relações espaço-temporais entre os objetos em uma cena. Além disso, a transmissão do fluxo sintetizado geralmente requer muito menos largura de banda do que a sequência de vídeo original uma vez que a informação redundante, no fundo não tem de ser transmitida. Outro exemplo de adaptação por síntese é o resumo hierárquica de vídeo, como se mostra na figura 2.7 (d). Os quadros-chave correspondentes para destacar segmentos em uma sequência de vídeo estão organizados em uma estrutura hierárquica para facilitar a navegação eficiente. As estruturas na hierarquia podem ser baseadas em decomposição ou temporal classificação semântica.

2.7.

2.5 Segmentação de vídeo

Também existe outra questão: como o vídeo será transmitido? Mesmo tendo uma boa largura de banda um arquivo de vídeo por ser geralmente grande pode demorar muito tempo para ter a transmissão concluída, se o usuário quiser ver o vídeo imediatamente ele vai ter que esperar, mas esta não é uma situação interessante. Para resolver este problema podemos adotar como solução as seguintes opções:

- **A:** Dividir o arquivo original em pequenos arquivos de tamanho fixo que serão enviados ao cliente, como utilizado no YouTube e no Sistema RIO [4].

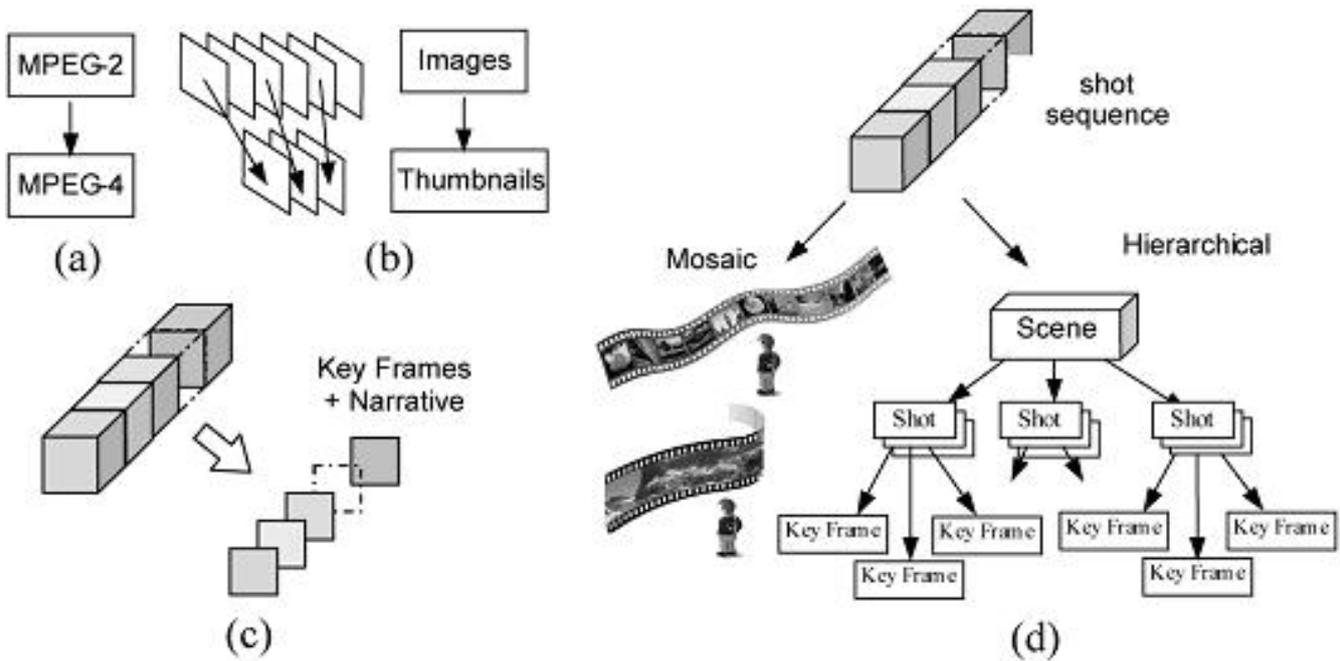


Figura 2.7: tipos de adaptação de vídeo (a) Transcodificação de Formato -(b) Seleção / Redução -(c) substituição -(d) Síntese[2]

- **B:** Utilizar alguma técnica de streaming (fluxo) de vídeo [5].
- **C:** Segmentar o vídeo em pedaços baseados em grandezas do próprio vídeo, como o tempo por exemplo. Segundo [6], o tamanho ideal dos segmentos é de 6 segundos para conexões não persistentes e 2 segundos para conexões persistentes.

Como o presente projeto tem como objetivo prover o vídeo para aplicações multimídia e estas tradicionalmente utilizam o tempo como elemento de sincronização do vídeo com as demais mídias, optou-se por utilizar a segmentação baseada em tempo para a implementação do projeto. Outras opções adotadas foram à adaptação baseada na resolução espacial (dividiu-se em grupos de dispositivos) e na taxa de bits, não adotando a adaptação na resolução temporal (retirar quadros do vídeo), pois a mesma pode gerar perdas significativas na qualidade do vídeo e no seu processo de compactação [1]. Optou-se também em utilizar segmentos de 4 segundos, pois eles atendem de forma satisfatória os dois tipos de conexão, conforme mostrado na Figura 2.8 vale a pena observar que a largura de banda utilizada é fora da realidade da maioria das cidades brasileiras, ainda mais se tratando de conexões em aparelhos moveis, então possivelmente esse tamanho pode ser alterado após a realização de testes. Segundo [6], seguimentos de tamanho menor

podem causar uma maior sobrecarga em termos de pedidos e, por outro lado seguimentos maiores podem dificultar a transmissão em clientes com conexão mais fraca, uma mídia segmentada em tamanhos menores acaba ocupando um espaço maior.

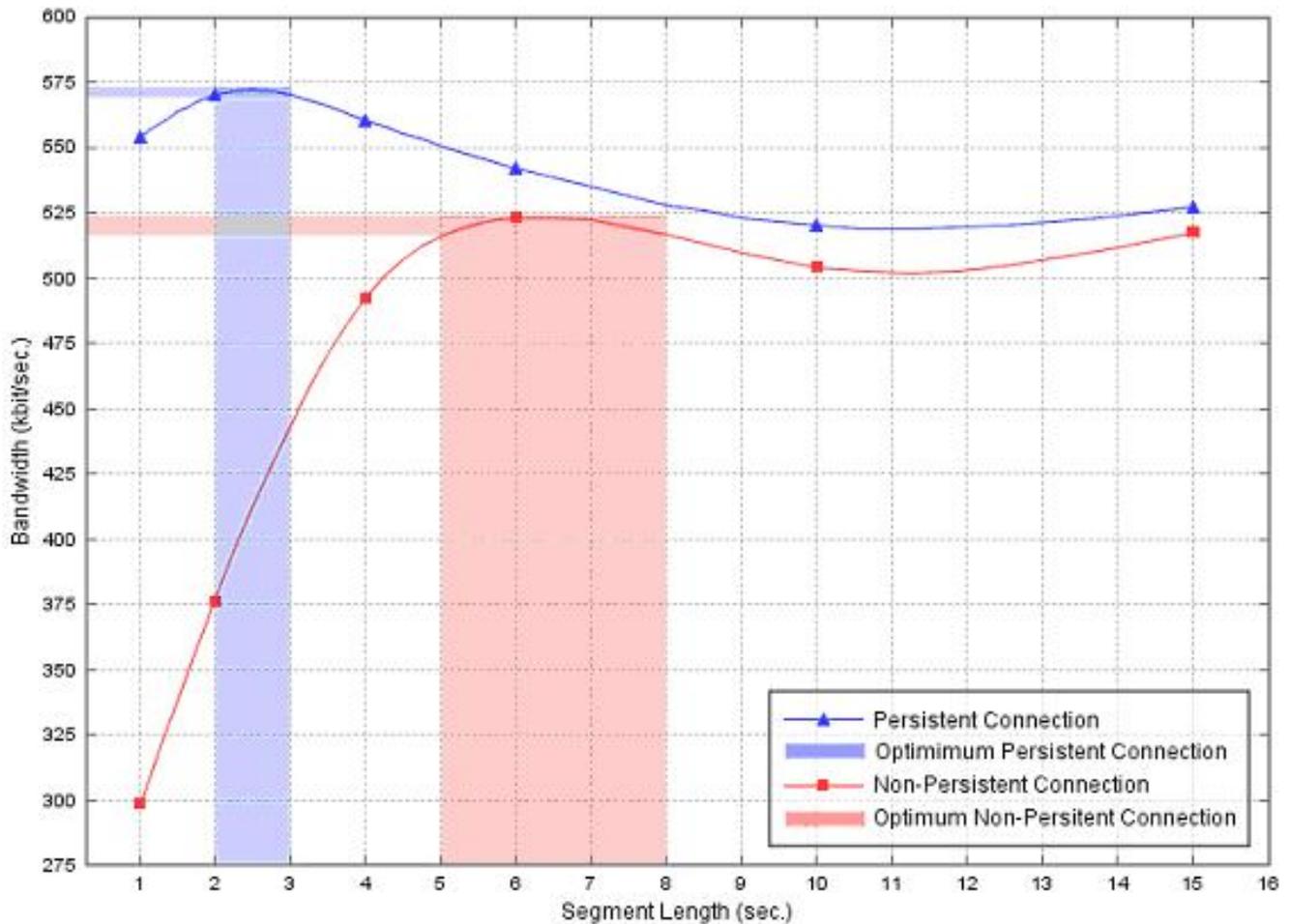


Figura 2.8: Largura de banda x Tamanho dos segmentos [6]

2.6 Metadados

Os metadados são a grosso modo dados que contém informações sobre outros dados . No presente projeto utilizamos os metadados para que a aplicação cliente tenha uma referência sobre o conteúdo multimídia armazenado no servidor. W3C define metadados como informação sobre objetos da Web compreensível por máquinas. Segundo[14] um metadado é por definição: dado que descreve atributos de um recurso, caracteriza suas relações, apoia sua descoberta e uso efetivo, e existe em um ambiente eletrônico. Usualmente consiste em um conjunto de elementos, cada qual descrevendo um atributo do

recurso, seu gerenciamento, ou uso.

Um documento é usualmente descrito e estruturado, no meio digital, por meio de metalinguagens de marcação como o XML. No âmbito deste trabalho os metadados serão utilizados para guardar informações necessárias as mídias para que elas sejam exploradas pelo servidor de contexto.

Por exemplo o MPD é o documento XML que representa a mídia segmentada, neste documento estão contidas todas as informações da media. Segundo [18] MPD contém informações que descrevem a apresentação de mídia e que é necessário para o cliente 3GP-DASH para a construção de informação adequada e para fornecer o serviço de streaming para o usuário, ainda de acordo com [18] neste xml existem vários atributos e elementos conforme a tabela 2.9:

Element or Attribute Name	Use	Description
MPD	1	The root element that carries the Media Presentation Description for a Media Presentation.
@profile	O	specifies a space delimited list specifying the Media Presentation profiles.
@type	OD default: OnDemand	'OnDemand' or 'Live'. specifies the type of the Media Presentation. Currently, on-demand and live types are defined.
@availabilityStartTime	CM Must be present for type='Live'	For @type='Live' this attribute shall be present. In this case it specifies the anchor for the computation of the segment availability start time for any Segment in the Media Presentation. For @type='OnDemand', if present, it specifies the segment availability start time for all Segments referred to in this MPD. If not present, all Segments described in the MPD shall be available.
@availabilityEndTime	O	specifies the latest segment availability end time for any segment in the Media Presentation. When not present, the value is unknown.
@mediaPresentationDuration	O	If present, it specifies the duration of the entire Media Presentation. In this case the attribute MPD@minimumUpdatePeriodMPD shall not be present.

Figura 2.9: [18] semântica do MPD

2.7 Qualidades e características do vídeo transmitido

- **Obtendo parâmetros para Dispositivos moveis** O mundo está sofrendo uma mudança histórica em termos de conectividade em função do avanço nas tecnologias de dispositivos móveis, na última década milhões de pessoas principalmente nos países em desenvolvimento adquiriram este tipo de dispositivo , o que garantiu a capacidade de fazer chamadas telefônicas e troca de mensagens de texto básicas (SMS) com o outras pessoas . Mas há uma segunda onda em curso, onde os telefones estão convergindo com a computação (smartphones e tablets) além disso a diminuição do custo da transmissão de dados possibilita um grande aumento no uso da internet através das redes móveis. Estima-se que em 2015 mais de 700 milhões de pessoas que terão acesso à internet exclusivamente através do seu telefone celular [21]. Mas a utilização de dados através da transmissão sem fio no mundo pode não ser tão rápida ou bem sucedida como a que vimos com a telefonia básica. Pois existem limitações técnicas e de custo inerentes à largura de banda, que pode prejudicar o uso de dados em redes sem fio. No nosso caso o foco é o consumo de vídeo. Supondo que os componentes técnicos e os preços foram fixados, será explorada a natureza do próprio conteúdo que no nosso caso é o vídeo. Degradando sua qualidade podemos reduzir o tamanho do arquivo e conseqüentemente seu custo de transmissão para os usuários. Segundo [21] o primeiro passo para alcançar esta estratégia é verificar ”o quão ruim é bom o suficiente?”: O quanto poderíamos degradar um vídeo antes de torna-lo indesejável? Os principais fatores que afetam a qualidade do vídeo são: resolução, taxa de bits de vídeo, taxa de bits de áudio, e a taxa de quadros. [21] Existe uma relação entre estes fatores na qual, a redução na taxa de quadros traz uma melhoria na qualidade de quadros, assim como aumentar a resolução do vídeo faz com que a nitidez do vídeo piore, Como a visualização de vídeo muda devido a contextos suscetíveis a esses trade-offs (por exemplo, pequenos dispositivos móveis, os limites de largura de banda para dados de streaming), os modelos de QoE foram responsáveis pelo efeito combinado da taxa de bits, taxa de quadros, tamanho da tela e do conteúdo. Segundo [23] um vídeo 320x240 tem 76.800 pixels em cada quadro, enquanto um arquivo de vídeo de 640x480 tem 307.200 pixels, ,ou seja, quatro

vezes mais, o que é evidente na Figura a 2.10. Isso significa que temos que aplicar quatro vezes a compressão para atingir a mesma taxa de dados, o que normalmente reduz visivelmente a qualidade. É por isso que taxa e resolução dados são integralmente ligada nas decisões de streaming de qualidade relacionados, ,por exemplo, uma taxa de 250 kbps de dados de vídeo deve produzir excelente qualidade a uma resolução de 320x240 , mas poderia ser desastrosa na resolução 640x480.



Figura 2.10: [23]Comparação 320x240 e 640x480

Trabalhos recentes revelaram limites específicos em que a qualidade de vídeo móvel se torna inaceitável [22], onde o valor da taxa de bits sugerido para a tela do smartphone deve estar entre 120kbps e 500kbps, dependendo do tipo de conteúdo.

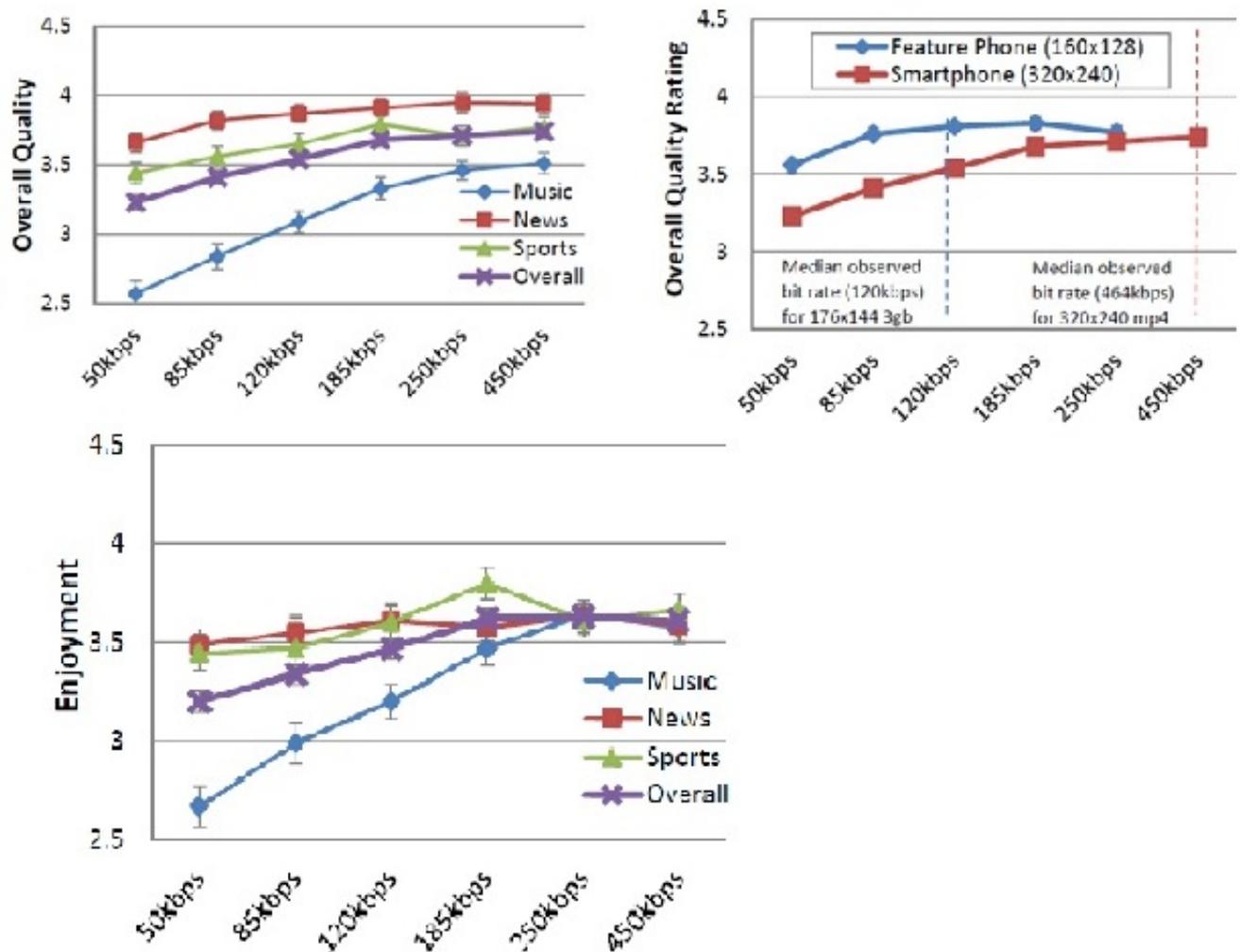


Figura 2.11: [22] Resultados pesquisa sobre as qualidades dos vídeos com diferentes bit-rates

- **TV e PC** Estes dispositivos geralmente (na teoria) não estão sujeitos a grandes oscilações na qualidade da conexão, pois utilizam Banda larga, atualmente no Brasil temos uma taxa que varia de 2 a 10 mbps em cidades como Juiz de Fora. Estou assumindo que a velocidade de transmissão estará sempre no intervalo entre 400 kbps e 5 mbps pois normalmente as empresas de telecomunicações entregam uma velocidade bem abaixo da estabelecida no contrato, e não tem como mensurar isso. Segundo [24] as melhores configurações de resolução de tela para PCs e notebooks são:

Tamanho do monitor	Resolução recomendada (em pixels)
Monitor LCD de taxa padrão de 19 polegadas	1280 × 1024
Monitor LCD de taxa padrão de 20 polegadas	1600 × 1200
Monitores LCD widescreen de 20 e 22 polegadas	1680 × 1050
Monitor LCD de taxa padrão de 24 polegadas	1920 × 1200

Tamanho da tela do laptop	Resolução recomendada (em pixels)
Tela de laptop taxa padrão de 13 a 15 polegadas	1400 × 1050
Tela de laptop widescreen de 13 a 15 polegadas	1280 × 800
Tela de laptop widescreen de 17 polegadas	1680 × 1050

As Resoluções das TV são HD - As TVs com resolução HD exibem imagens com resolução de 1.280 x 720 linhas. TVs com esta resolução representam 20 por cento das vendas. Full HD - A resolução tem 1.920 x 1.080 linhas e exibe cerca de 2 milhões de pixels. Ela está presente em mais de 70 por cento das TVs vendidas no Brasil.

2.8 Ciência de Contexto

Antes de mostrar a implementação e as características deste projeto é necessário abordar um estudo sobre Ciência do contexto, visto que o projeto está inserido em um servidor de contexto idealizado pelo LApIC.

- **Computação pervasiva** O conceito de computação pervasiva implica que o computador está embarcado no ambiente de forma invisível para o usuário. Nesta con-

cepção, o computador tem a capacidade de obter informação do ambiente no qual ele está embarcado e utilizá-la para dinamicamente construir modelos computacionais, ou seja, controlar, configurar e ajustar a aplicação para melhor atender as necessidades do dispositivo ou usuário. O ambiente também pode e deve ser capaz de detectar outros dispositivos que venham a fazer parte dele. Desta interação surge a capacidade de computadores agirem de forma inteligente no ambiente no qual nos movemos, um ambiente povoado por sensores e serviços computacionais[15]. Para que se consiga obter uma computação pervasiva, é preciso aplicações cientes de contexto, que adaptem seu comportamento baseando-se nas informações físico e computacional adquiridas do ambiente.

- **Definição de contexto** Segundo [16] Contexto é qualquer informação que pode ser utilizada para caracterizar a situação de uma entidade. Uma entidade é uma pessoa, lugar, ou objeto que é considerado relevante para a interação entre um usuário e uma aplicação, incluindo o usuário e a aplicação em si.
- **categorias de contexto**
 - **Contexto computacional** Conectividade de rede, largura de banda, custo computacional e recursos próximos como impressoras e terminais.
 - **Contexto do usuário** Preferências do usuário, localização, mobilidade do usuário, pessoas próximas.
 - **Contexto físico** Temperatura, luminosidade, pressão e umidade.
 - **Contexto de tempo** Dia, mês ou hora.
- **Computação Ciente de Contexto** Um sistema é ciente de contexto, se ele usa contexto para enviar informações ou serviços ao usuário. Aplicações cientes de contexto são capazes de modificar seu comportamento baseado nas informações de contexto ou são aplicações que mostram ao usuário informações de contexto[17]. Existem diversas classificações de aplicações ciente de contexto:
 - **Ciente de contexto ativo** uma aplicação adapta-se automaticamente ao contexto descoberto, mudando o comportamento da aplicação.

- **Ciente de contexto passivo** uma aplicação apenas apresenta informações de contexto ao usuário ou as armazena para que o usuário possa consultar posteriormente.
- **Sensoriamento do contexto (contextual sensing)** é a habilidade de detectar informações sobre o contexto e apresentá-las ao usuário.
- **Adaptação ao contexto (contextual adaptation)** é a habilidade de executar ou modificar um serviço automaticamente, baseando-se no contexto atual. São baseados numa simples regra de if-then-else. Um comando é executado quando existe uma determinada combinação de contexto.
- **Aumento do contexto (contextual augmentation)** é a habilidade de associar informações ao contexto para recuperar posteriormente. São aplicações que permitem associar dados digitais ao contexto do usuário

3 SERVIDOR DE CONTEXTO

Aplicações cientes de contexto têm como característica poder se adaptar com base no ambiente que estão sendo executadas e nas preferências do usuário. Para ampliar e potencializar uma aplicação dessa natureza é interessante um servidor de contextos bem estruturado com diferentes funcionalidades como a captura e persistência de contextos (usuário, dispositivo, temporal, espacial, etc.) e possibilidade de recuperação desses contextos, conforme a necessidade e características apresentadas pelas aplicações.

3.1 Arquitetura do servidor de contexto

- **Aplicações sensíveis ao contexto:** A fim de prover uma melhor experiência ao usuário com recursos multimídia, como fotos, vídeos e áudios que podem ser gerados pelos usuários na execução de suas atividades diárias, Aplicações de busca necessitam considerar outras informações implícitas relacionadas ao recurso. Logo surgiu a necessidade de uma nova classe de aplicações, as sensíveis ao contexto, que utilizam informações retiradas da situação do usuário ou dispositivo no momento em que ele desempenha sua atividade. Estes dispositivos devem estar conectados em uma rede sem fio e devem ter um poder de processamento no mínimo razoável, além de sensores para captar mudança no contexto, exemplo: GPS. Também essas aplicações necessitam de um alto grau de controle sobre o processamento, o acesso à rede sem fio e sensores, além de organizar as informações extraídas do contexto e anotá-las ao recurso.
- **ContextF:** segundo[27] é um framework que visa contribuir com a implementação de aplicações sensíveis ao contexto. Sua API simplificada permite que aplicação passe o recurso a ser anotado com informações sem se preocupar com todo processo de extrair-las do contexto
- **BUS** Esta arquitetura parte do princípio que softwares que podem receber e enviar

mensagens utilizando um ou mais canais de comunicação. Podendo assim interagir com outros softwares sem precisar conhecer os detalhes específicos sobre cada um. Message BUS é um estilo de estruturar aplicações onde a interação entre as aplicações é realizada com passagem de mensagem (normalmente assíncronas) sobre um barramento comum. O BUS viabiliza arquitetura orientada a eventos que registra os componentes para reagirem aos eventos disparados por ele. 3.1.

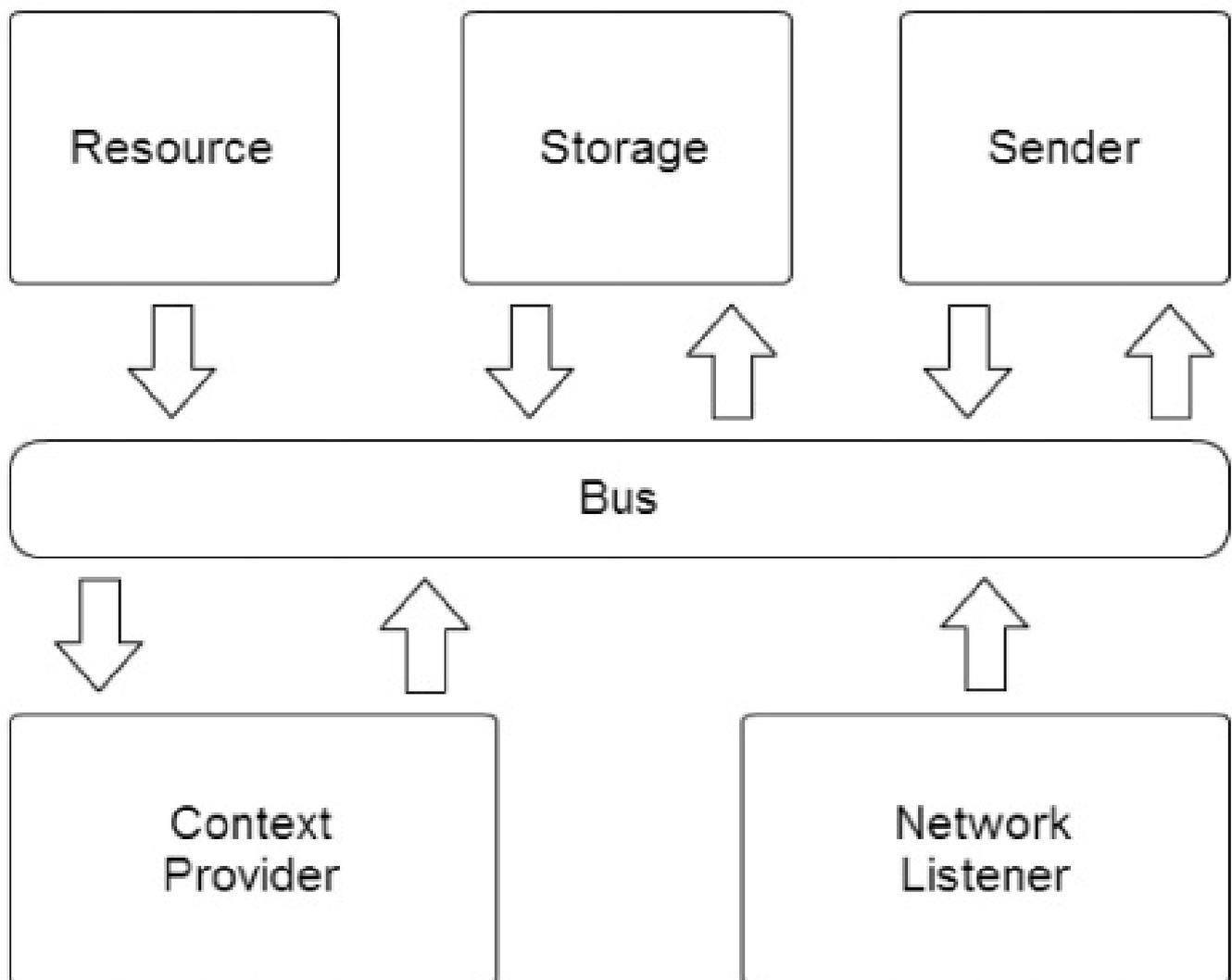


Figura 3.1: Arquitetura do Bus (orientada a eventos)

O servidor de contexto permite que vários clientes acessem de maneira concorrente as informações dos dispositivos. É similar a arquitetura cliente-servidor, onde o servidor terá ao seu acesso aos dispositivos e as informações geradas e o cliente irá requisitar essas informações. Problemas como protocolo de comunicação, desempenho de rede e qualidade

de serviço devem ser levados em conta ao se usar um servidor de contexto. Cada etapa do processo de contextualização tem um comportamento bem definido implementado por um componente baseado em técnicas e linguagens orientadas a objetos[27]. Alguns dos componentes são: componente de extração, sincronização, armazenamento, etc.

A figura 3.2 mostra a arquitetura do servidor de contexto no qual este projeto esta inserido:

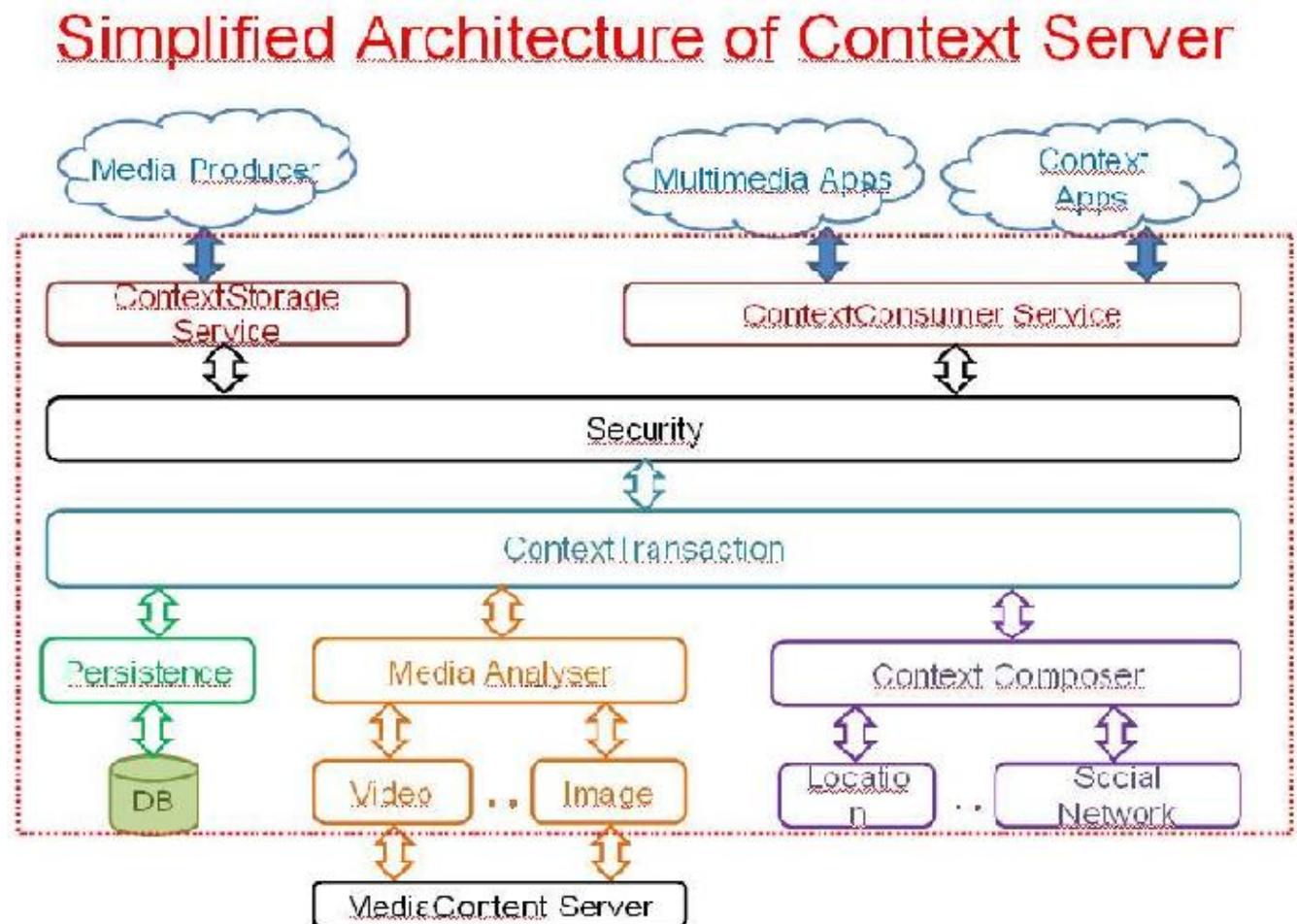


Figura 3.2: arquitetura do servidor de contexto

4 Estudo de Caso: Implementação do Módulo de vídeo

4.1 Especificação do Módulo de Vídeo

O módulo de vídeo deve apresentar uma API para receber requisições do servidor de contexto, essas requisições correspondem às mídias de vídeo submetidas pelos usuários ao servidor de contexto. Após receber uma solicitação, este módulo deve fazer um download do arquivo de vídeo caso ele não tenha sido enviado ao servidor, em seguida o vídeo deve ser adaptado para as mídias Celular, computador e TV, cada mídia terá diferentes bitrates em função de diferença nas conexões de rede. O ultimo passo é a segmentação dos vídeos adaptados e a geração de metadados correspondentes aos mesmos.

4.2 Descrição do processamento no servidor de vídeo

A maneira como a mídia de vídeo é tratada pelo servidor de contexto pode ser descrita de acordo com a maquina de estados descrita na figura 4.1.

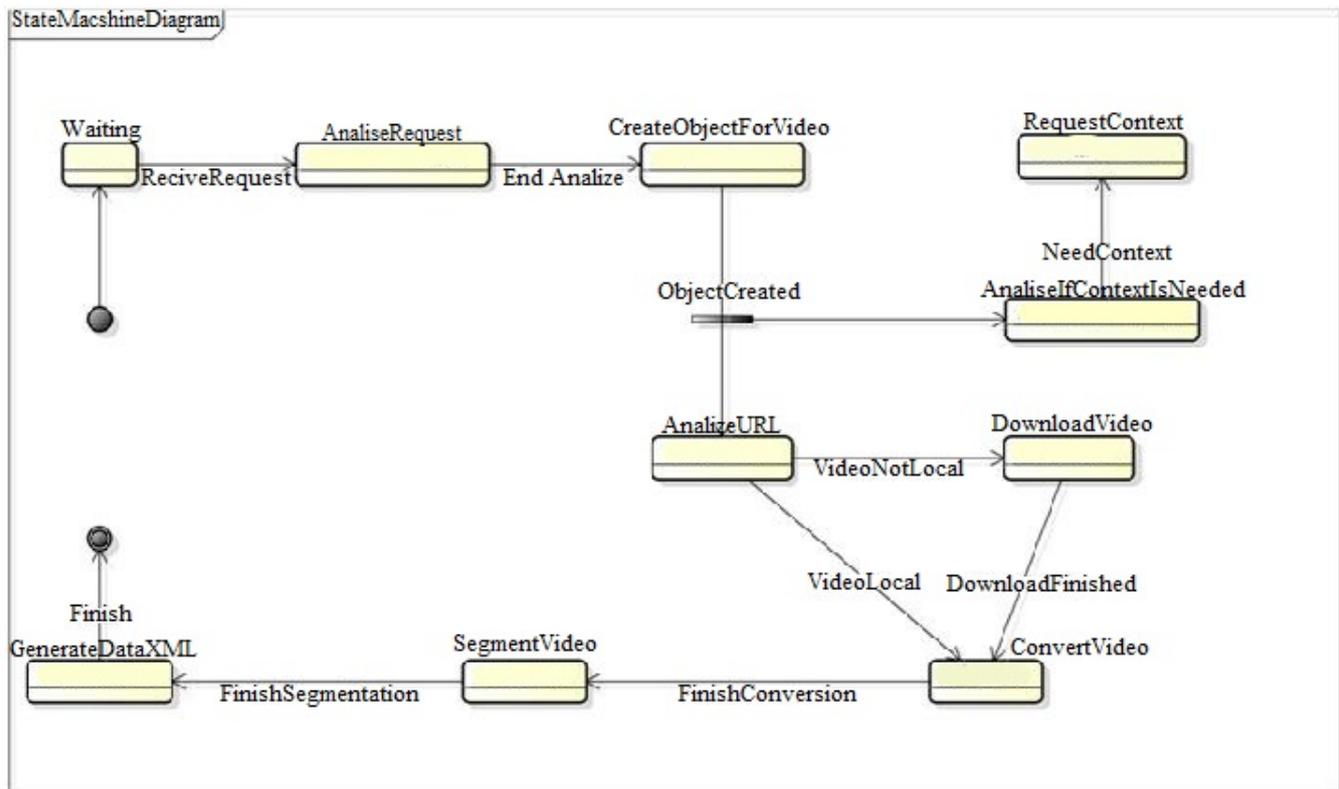


Figura 4.1: estados no modulo de vídeo

Inicialmente o vídeo module espera que o media analyser faça uma requisição para que uma nova mídia de vídeo seja processada no servidor. Nesta requisição são passados como parâmetros o XML e o ID correspondentes à mídia, um dos elementos contidos no XML é a URL do vídeo, para ler documentos XML devemos analisar o mesmo através de um parser(AnalizeRequestVideoXML) e popular objetos(CreateObjectForVideo) cujos atributos correspondam aos elementos e atribuídos do XML. O arquivo de vídeo pode estar armazenado localmente, em outra maquina da rede ou na internet, isso é verificado no estado(AnalizeVideoURL) caso não esteja armazenado localmente o vídeo é baixado para o servidor(DownloadVideo) em paralelo é verificado a necessidade de obter informações referentes ao contexto do video(AnalizeIfContextIsNeeded) se o contexto for relevante o sistema vai disparar uma requisição para o Media Analyser, em seguida é iniciada a conversão do video(ConvertVideo) para diferentes dispositivos como Mobile, PC, TV, tablet depois de finalizado esse processo é iniciada a fase da segmentação, onde cada formato de vídeo é segmentado em vários pedacinhos com duração entre 1 e 10 segundos, conforme 2.5 utilizaremos o tamanho de 4 segundos. O passo final é a geração de outro

XML contendo um cabeçalho com informações de contexto e algumas informações técnicas e uma estrutura contendo meta-dados sobre os segmentos de vídeo.

4.3 Funcionamento do sistema

Todo o processamento presente no módulo de vídeo é coordenado por um BUS que gerencia os módulos correspondentes às requisições e eventos que acontecem figura 4.2 Como

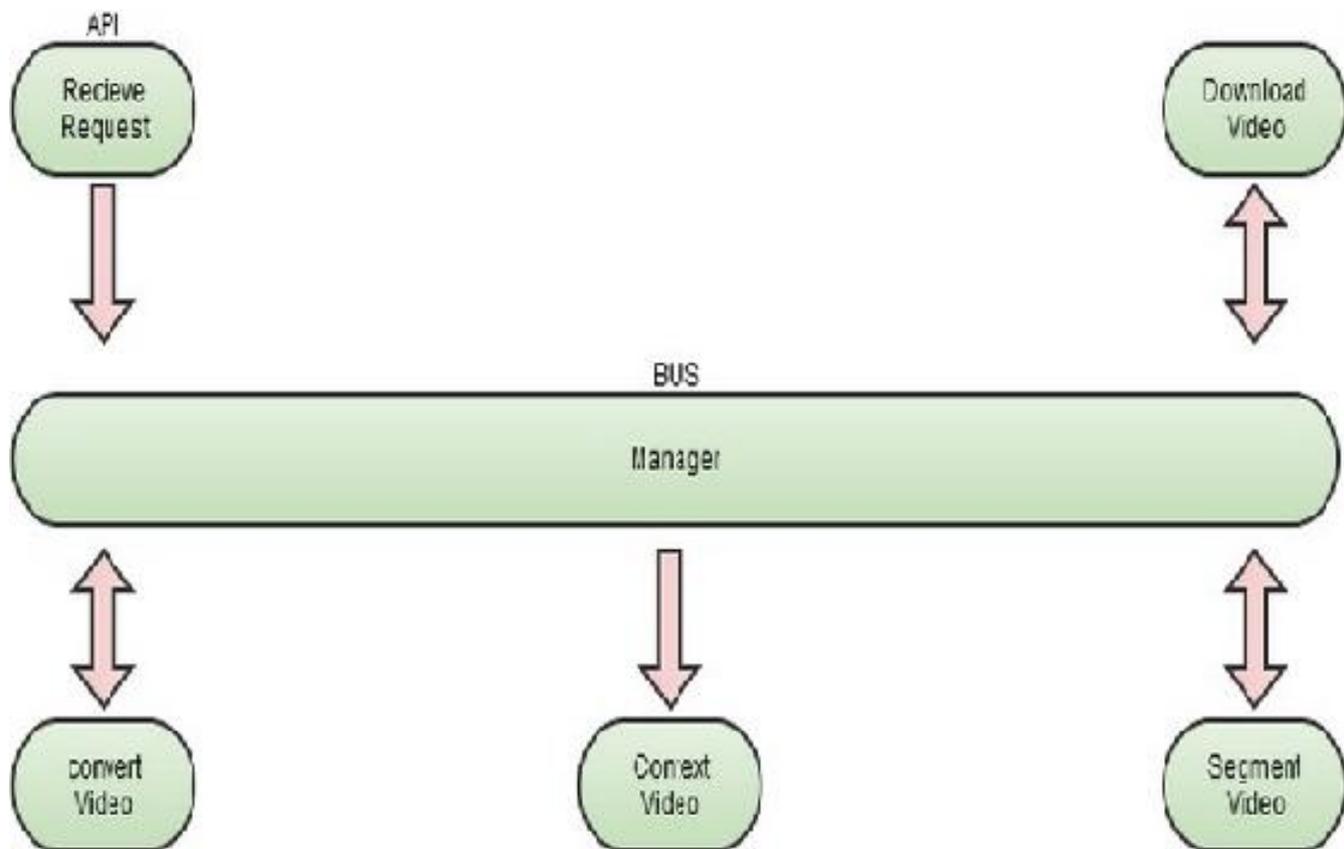


Figura 4.2: arquitetura do modulo de vídeo

mostrado no esquema acima, cada módulo se comunica apenas com o BUS no caso é o manager. o Vídeo circula dentro do BUS, os módulos que executam alguma ação sobre o vídeo requisitam ao BUS para que lhes entregue a informação de vídeo solicitada.

4.4 Processo de segmentação de vídeo

cada vídeo original será convertido para até três resoluções diferentes correspondentes aos diferentes tipos de hardware, e para cada resolução serão gerados três grupos de segmentos, cada um com uma taxa de bits diferentes. Assim é possível disponibilizar o vídeo mais adequado para o contexto de display do equipamento e ainda disponibilizar os segmentos conforme o QoS da rede (taxas de bits distintas). Depois todas as mídias adaptadas são segmentadas em tamanhos de 4 segundos a figuras a 4.3 e 4.4 mostram duas maneiras de implantar este processo.

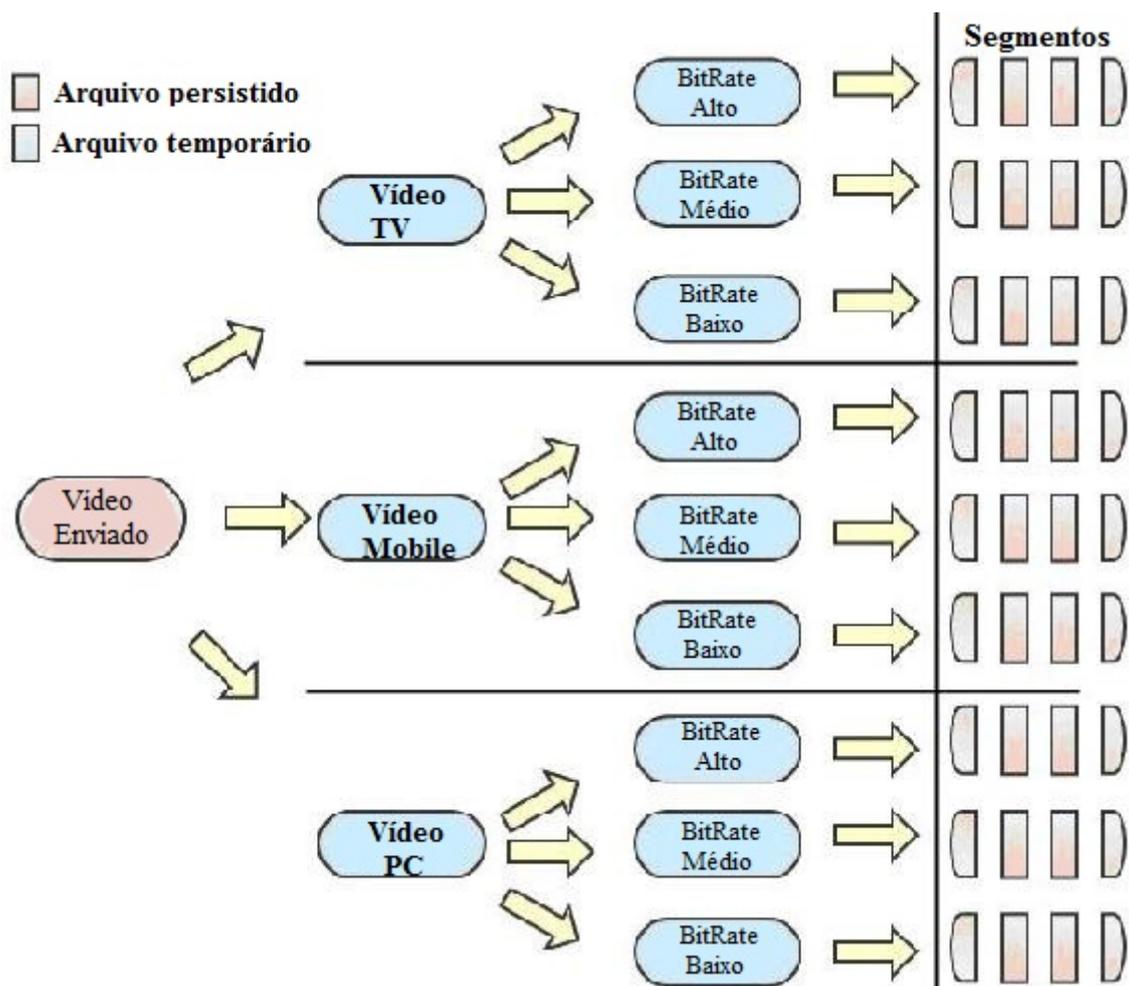


Figura 4.3: Processo de criação dos segmentos

na figura 4.3 executamos primeiro a conversão em 2 etapas, para então realizarmos a segmentação, este foi a maneira inicial de realizar este processo

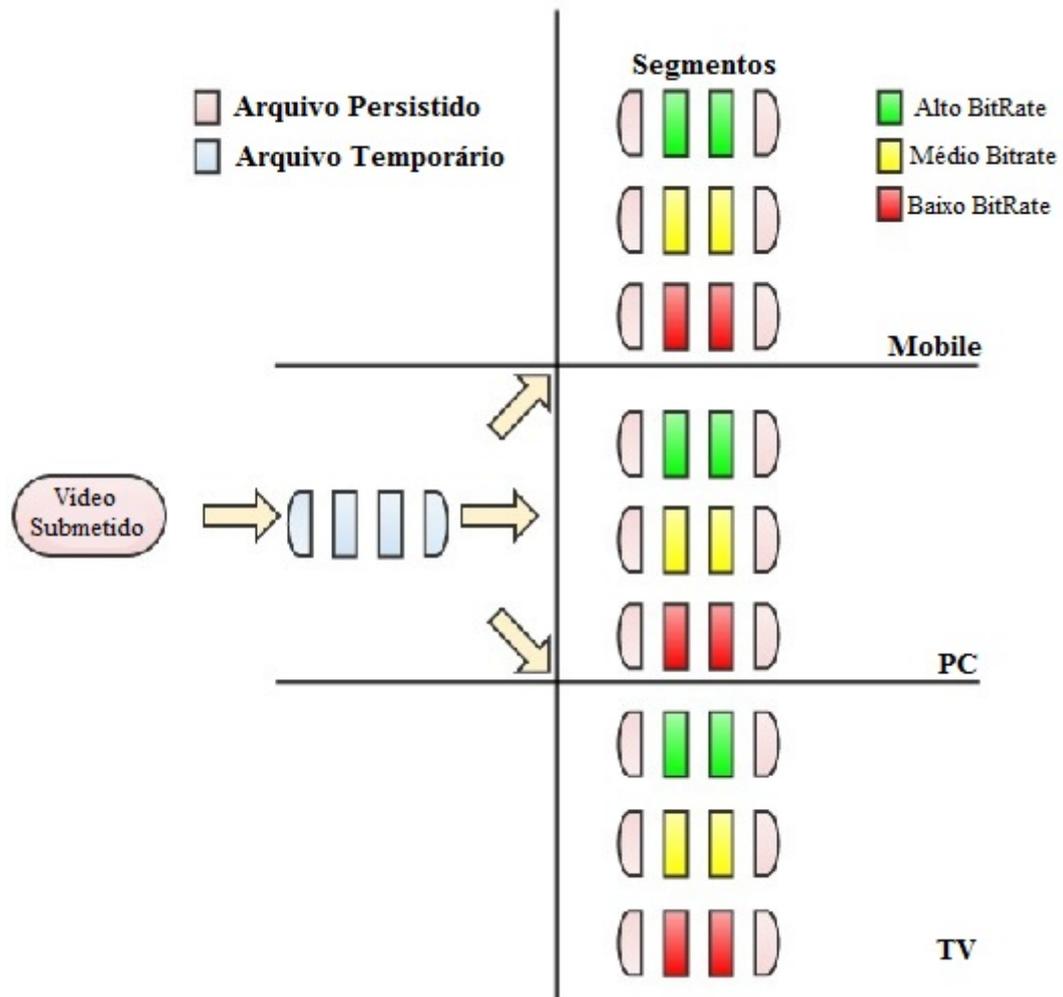


Figura 4.4: Processo de criação dos segmentos 2

já a figura 4.4 mostra primeiramente a segmentação, e depois a conversão em 1 passo, este modelo foi adotado em função de problemas com o modelo 4.3, esses problemas serão explicados na seção 4.7

4.5 Decisões de Implementação

É interessante que o sistema operacional do servidor possa ser tanto um sistema Linux quanto Windows por uma questão de flexibilidade e portabilidade, essa é uma das razões pela qual escolhemos a linguagem Java para implementarmos esse módulo de vídeo, outros fatores fundamentais para escolher essa linguagem foram: Quantidade de bibliotecas existentes, e o tamanho da comunidade, que torna mais fácil encontrar respostas para dificuldades técnicas que costumam aparecer. Para realizar a adaptação e a segmentação de vídeo foram testadas 2 tecnologias: o Xuggler e o FFMPEG.

- O Xuggler é uma biblioteca integrada ao Java, que possui o FFMPEG embutido, a principio é mais fácil trabalhar com ele, mas aconteciam alguns problemas para converter alguns formatos de vídeo, como de flv para mp4, costumavam ocorrer alguns erros de execução, portanto não estamos utilizando o Xuggler para nem uma ação onde ocorra processamento de vídeo, mas estamos utilizando-o para obter algumas informações de vídeo, como a duração que é uma informação importante.
- O FFMPEG é uma solução clássica que além de atender o Java, também atende linguagem como Php, c, c++, etc. o FFMPEG é executado por linha de comando, e devemos passar os parâmetros necessários para que ele funcione alguns parâmetros são: diretório origem, destino, formato saída, tamanho dos segmentos, etc.
- Quando o Media Analyzer envia uma requisição ao módulo de vídeo, além de submeter o arquivo de vídeo, ele entrega um arquivo de metadados do vídeo para a API, o arquivo de metadados é um XML, poderíamos utilizar outras tecnologias como o JASON, mas o XML esta mais difundido e é um padrão do W3C. Para escrevermos arquivos de metadados. No Java estamos usando uma biblioteca chamada XStream. nesse XML estão contidas informações vitais para a execução do modulo de vídeo, estas informações são: ID, URL

- O Google guava é uma biblioteca do Google que auxilia desenvolvedores Java, neste projeto foi utilizada o guava EventBus para implementar o manager, quando a API recebe uma requisição o manager é notificado automaticamente e começa a tratá-lo
- ThreadPool Um pool de threads é uma coleção de threads disponíveis para realizar tarefas. Pools geralmente proveem melhor performance quando se executam um grande número de tarefas devido a um overhead reduzido de chamadas por tarefa, uma forma de limitar recursos consumidos (incluindo threads) ao executar uma coleção de tarefas. Um programa com milhares de threads pode sofrer em desempenho, adicionalmente, com um pool de threads não é necessário preocupar-se com o ciclo de vida dos threads (criação, destruição), pode se concentrar em criar a lógica em vez de gerenciar threads. As implementações permitem que você estabeleça: O número básico e máximo do pool (número de threads), tipo de estrutura de dados para armazenar as tarefas, como tratar tarefas rejeitadas, como criar e terminar threads.

Neste projeto para cada requisição o manager inicia uma thread para adaptar e segmentar o vídeo correspondente a essa requisição, se recebermos varias requisições em um curto espaço de tempo o processador do servidor pode ser sobrecarregado, então nos gerenciamos essas threads, estabelecendo um numero máximo de threads que possam ser executadas ao mesmo tempo, isso é feito através do thread pool.

- Para gerarmos o MPD também Utilizamos a biblioteca XStream, visto que o MPD é um XML. apesar de ser utilizado [18] como uma referência, neste projeto foi implementada uma simplificação, utilizando poucos atributos e elementos conforme a figura 19, caso seja necessário futuramente podemos adicionar mais informação em nosso MPD

```

<mpd>
  <mediaName>troteCC</mediaName>
  <id>1</id>
  <tipo>onDemand</tipo>
  <duracao>120</duracao>
  <tamanhoSeg>4</tamanhoSeg>
  <tempoInicial>0</tempoInicial>
  <representacao>
    <mobileHighRate>
      <segmentos>
        ...
      </segmentos>
    <segmentos>
      <id>_ID_1-PT-3</id>
      <URL>C:\servidor_contexto\videos\1\mobile\HighRate\troteCC_ID_1-PT-3-600.ap4</URL>
      <inicio>112</inicio>
      <resolucao>360x240</resolucao>
      <banda>600k</banda>
      <bitRate>600</bitRate>
    </segmentos>
  </mobileHighRate>
  <mobileMediumRate>
  <mobileLowRate>
  <tvHighRate>
  <tvMediumRate>
  <tvLowRate>
  <pcHighRate>
  <pcMediumRate>
  <pcLowRate>
</representacao>
</mpd>

```

Figura 4.5: MPD gerado

4.6 Resultados obtidos

- **Qualidade do video e comandos FFmpeg utilizados** Foram testados como arquivos de entrada videos no formato AVI, e MPEG, para saidas MP4(h264 AVC) e mpeg. os resultados foram

– mpeg para mp4:

```

...MPG vcodec libx264 vprofile baseline -preset slow -crf 22 -threads 0 -acodec libfaac
      1           2           3           4           5           6
-ab 256k -ar 44100...MP4
      7           8

```

Figura 4.6: Parâmetros MPG to MP4

- * **1-** Aplicar o codec H.264.
- * **2-** Aplicar perfil para o H.264(Baseline é de qualidade menor, High tem mais qualidade).
- * **3-** Velocidade de compactação do FFMPEG, quanto mais lento melhor a qualidade.
- * **4-** Isto proporciona a máxima eficiência de compressão, com uma única passagem, um valor. muito baixo a perda é menor, um valor muito auto pode comprometer a qualidade..
- * **5-** Número de threads .
- * **6-** Codec de audio, se o libfac não funcionar, podemos utilizar o aacenc.
- * **7-** Bitrate de audio.
- * **8-** Frequencia do audio.

O tamanho do arquivo ficou um pouco maior e a qualidade se manteve no mesmo nivel

– **MPG/AVI para MPG:**

```
...AVI -preset slow -b:v 600k -maxrate 600k -bufsize 1200k -s 360x240 -threads 0 -ab 128k ...MPG  
...MPG      1           2           3           4           5           6           7
```

Figura 4.7: Parâmetros MPG/AVI to MPG

- * **1-** Velocidade de compactação do FFMPEG, quanto mais lento melhor a qualidade
- * **2-** determinar o bit rate.
- * **3-** Determino um limite para o bit rate(mas obriga a usar o parâmetro bufsize).
- * **4-** Tamanho do buffer do FFmpeg usado durante a compressao.
- * **5-** Determinar a resolução do display.
- * **6-** Número de threads criadas pelo FFMPEG
- * **7-** Bitrate de audio.

De AVI para MPG, o tamanho diminuiu em cerca de 50 por cento, mas a qualidade piorou bastante. De MPG para MPG pouca coisa mudou, essa conversão só faz sentido se for para mudar parâmetros como bitrate, resolução, etc.

– **AVI para MP4:**

```
...AVI -vcodec libx264 -vprofile baseline -preset slow -b:v 600k -maxrate 600k {continua}
      1           2           3           4           5
{...}-bufsize 1200k -s 360x240 -threads 0 -acodec libvo_aacenc -ab 128k ...MP4
      6           7           8           9           10
```

Figura 4.8: Parâmetros AVI to MP4

- * **1-** Aplicar o codec H.264.
- * **2-** Aplicar perfil para o H.264(Baseline é de qualidade menor, High tem mais qualidade).
- * **3-** Velocidade de compactação do FFMPEG, quanto mais lento melhor a qualidade.
- * **4-** determinar o bit rate.
- * **5-** Determinar um limite para o bit rate(mas obriga a usar parâmetro o bufsize).
- * **6-** Tamanho do buffer do FFmpeg usado durante a compressão.
- * **7-** resolução de saída.
- * **8-** Número de threads criadas pelo FFMPEG.
- * **9-** Selecionar codec de áudio.
- * **10-** bitrate de áudio.

O tamanho foi reduzido em cerca de 50 por cento, a qualidade do vídeo se manteve boa

Uma observação importante é que eu não coloquei todos os parâmetros utilizados na codificação, pois seria redundante, para diferentes dispositivos passo diferentes resoluções (-s), bitrates(-br), vprofiles para saída .mp4, etc.

4.7 Problemas encontrados

Alguns problemas afetam a qualidade da mídia, e do processo de adaptação.

- **Congelamento do vídeo** Após realizar uma transcodificação no ffmpeg para o formato mp4 h.264 avc, em alguns vídeos ocorria o congelamento a imagem, apenas o som fluía normalmente, alguns casos foram bem intrigantes, pois ao executar arquivos de mídia diferentes mas com mesmo formato e codec uma das mídias resultantes funcionava normalmente e a outra congelava a imagem. Em algumas pesquisas em sites e fóruns informais como Stack Overflow, GUJ encontrei a explicação de que se o arquivo de origem possuir um quadro chave corrompido, ao executar o ffmpeg

ele descarta o quadro, mas pode ficar esperando para exibir o próximo quadro até o fim da transmissão, esse problema foi resolvido modificando alguns parâmetros passados para o ffmpeg, mas isso interferiu na qualidade.

- **Sincronia entre áudio e vídeo** O programa ffmpeg tem vários interruptores que ajudam a ajustar e converter arquivos de áudio e vídeo, O interruptor `itsoffset` é usado para "empurrar" (para frente ou para trás), o horário de início de cada "stream" de áudio ou vídeo. Uma típica câmera de vídeo irá gravar um stream de vídeo e um fluxo de áudio que são mesclados em um único arquivo. A figura 4.9 mostra a situação desejada para nossos arquivos de vídeo



Figura 4.9: Sincronia ideal [20]

a figura 4.6 mostra problemas que devem ser corrigidos

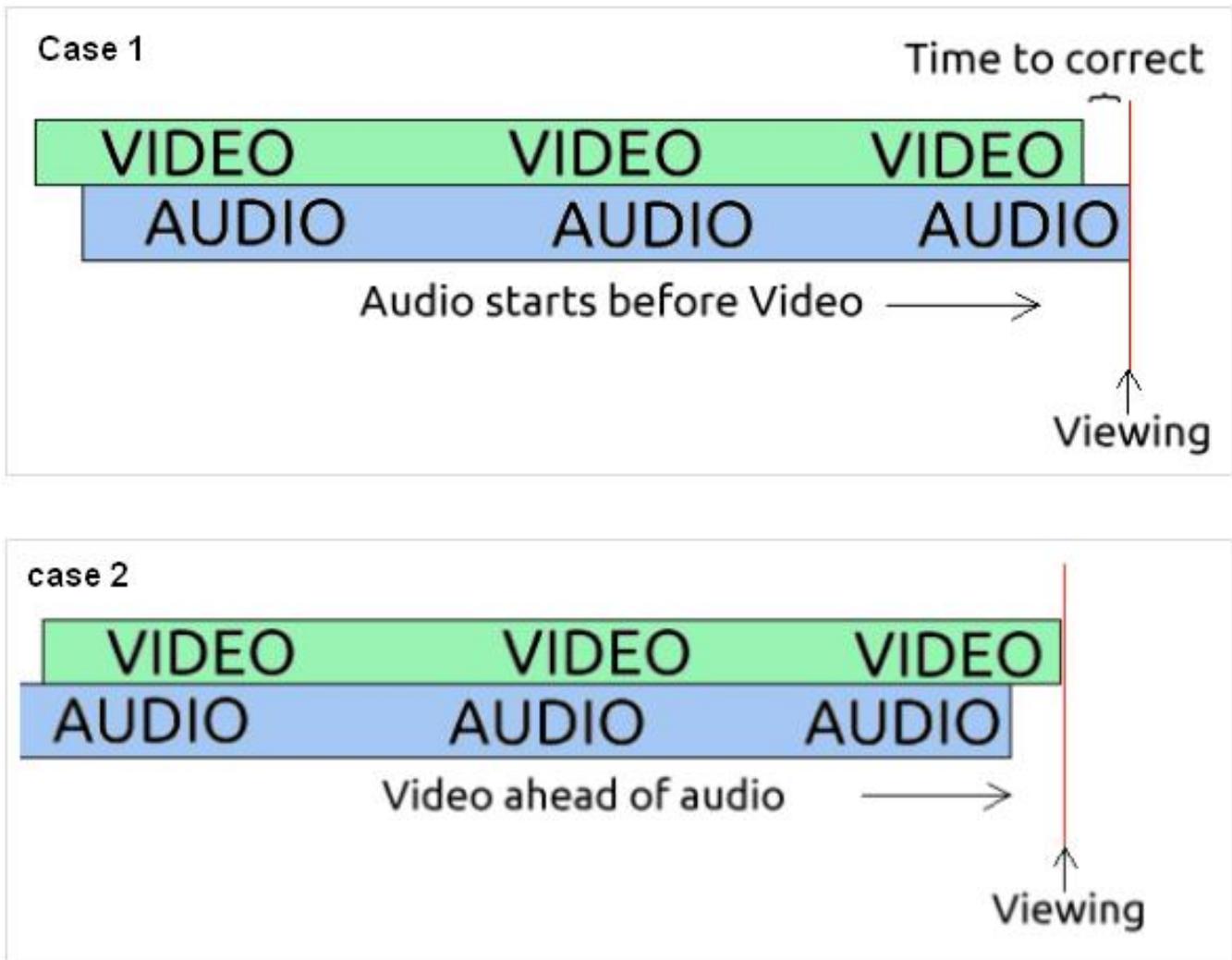


Figura 4.10: Problemas de sincronia[20]

para corrigir o caso 1 aonde o áudio começa antes do vídeo utilizamos o seguinte comando: `ffmpeg -i entrada.mp4 -itsoffset 0.150 -i entrada.mp4 -vcodec copy -acodec copy -map 0:0 -map 1:1 saida.mp4` para corrigir o caso 2 aonde o áudio começa depois do vídeo utilizamos o seguinte comando: `ffmpeg -i entrada.mp4 -itsoffset 0.150 -i entrada.mp4 -vcodec copy -acodec copy -map 0:1 -map 1:0 saida.mp4`**Ausência de som** em algumas situações o som desaparecia, esse problemas foi relativamente simples, batava passar corretamente os parâmetros de áudio e bitrate de áudio corretos. **vídeo muito lento** o stream de vídeo costuma ser exibido de maneira irregular, esse problema foi resolvido sem muita dificuldade.

- **Ausência de vídeo em alguns segmentos** foi um problema complicado, que levou a uma mudança na ordem do fluxo de processamento, antes o sistema convertia o vídeos para depois segmenta-los, tive que inverter a ordem, pois em função da compressão do h.264 tive fortes indícios de que alguns segmentos não possuíam o I-frame, tendo como consequência uma falha. Devido a isso, foi implementada o modelo 4.4 na seção 4.4.

4.8 Tamanho das medias armazenadas

O sistema adaptou um mesmo arquivo de vídeo para diferentes tamanhos: 2s, 4s, 8s e 16s, os arquivos de vídeo correspondentes a estes segmentos foram armazenados no servidor, afim de verificar o espaço gasto para cada tamanho foi implementada uma pequena aplicação para varrer os diretórios de cada tipo de segmento pegando o tamanho e agrupando pela qualidade do vídeo afim de plotar um gráfico que será mostrado na figura 4.11

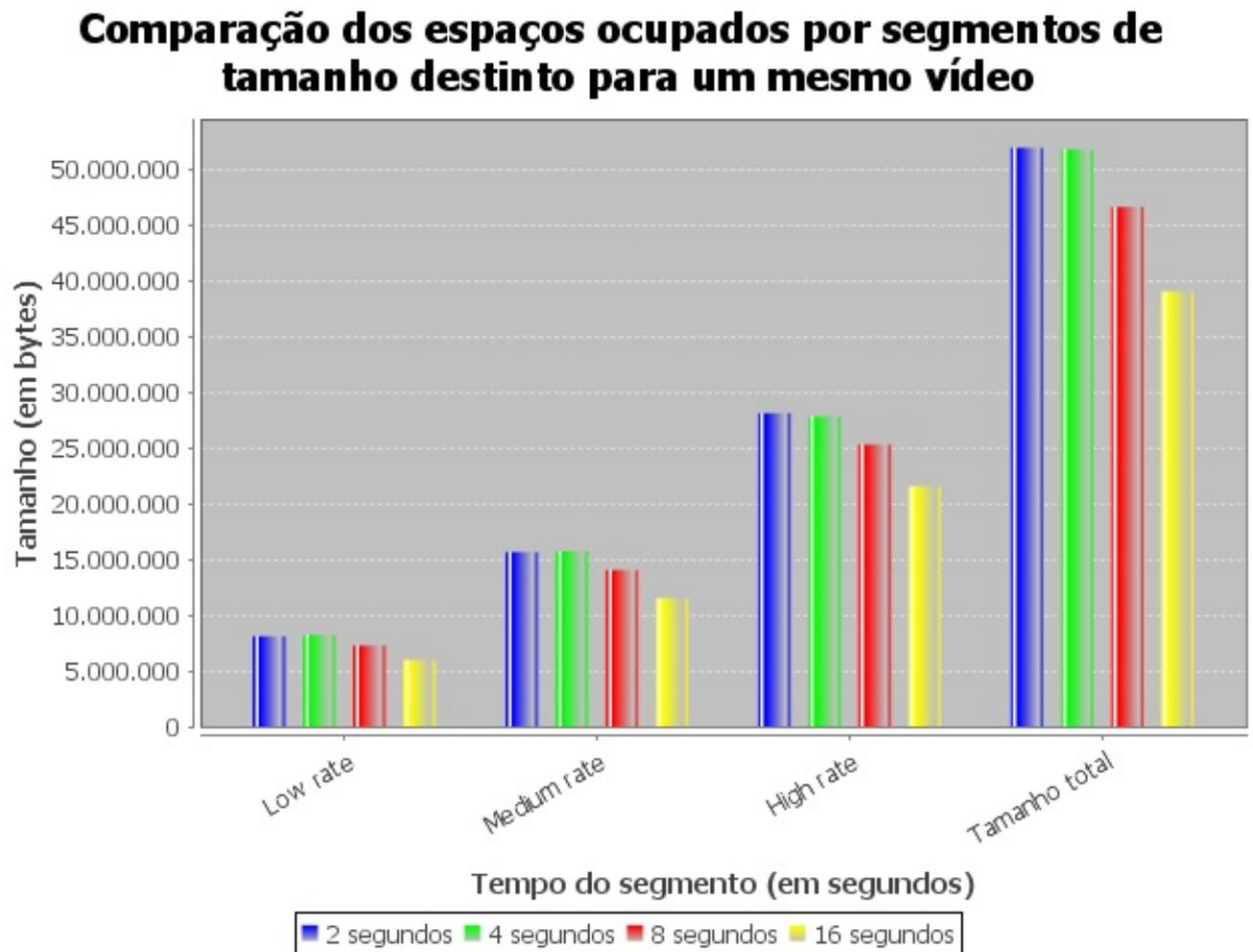


Figura 4.11: Tamanho gasto por cada tipo de segmento agrupado pela qualidade

5 Conclusão

A implementação do módulo foi realizada de maneira satisfatória, apesar de ainda haverem melhorias e otimizações que podem ser feitas. Durante a implementação pôde-se notar que alguns pontos da teoria foram confirmados, como o fato de seguimentos com menor tamanho ocuparem um pouco mais de espaço conforme 2.5 , também foram encontrados problemas para segmentar o vídeo em função das técnicas de compressão, o que resultou em algumas mudanças no fluxo da aplicação.

Ainda existem complicações referentes a baixa qualidade das redes móveis no Brasil, e por isso possivelmente as propriedades dos arquivos de vídeo deverão ser modificadas. Futuramente este módulo devera ter um servidor dedicado, que apenas receberá as requisições do servidor de contexto e realizará a adaptação e segmentação do vídeo.

Existem várias possibilidades de projetos futuros relacionados ao servidor de contexto e em especial a este módulo, por exemplo uma aplicação cliente para consumir os segmentos gerados. Um importante trabalho seria a realização de benchmarks afim de medir o tempo de processamento para as transcódificações, verificando por exemplo o impacto que cada parâmetro adicionado ou removido causa no processamento, isso é importante pois além de nos ajudar a definir uma infra-estrutura adequada, pode ajudar a otimizar os comando passados para o FFMPEG.

6 referência bibliografica

[1] Regis, C. D. M. et al. 2008. Transcodificação de vídeo digital para receptores portáteis. IV SEGeT Simpósio de Excelência em Gestão e Tecnologia.

[2] Chang, S. and Vetro, A. 2005. Video Adaptation: Concepts, Technologies and Open Issues. Mitsubishi Electric Research Laboratories. In: <http://www.merl.com/>.

[3] Stockhammer, T. 2011. Dynamic adaptive streaming over HTTP: standards and design principles. ACM-MMSys '11. Proceedings of the second annual ACM conference on Multimedia systems. Pages 133-144.

[4] Silva, E. S. et al. 2006. Multimedia Supporting Tools for the CEDERJ Distance Learning Initiative applied to the Computer Systems Course. 22th ICDE World Conference on Distance Education. pages 1-11.

[6] Lederer, S. Müller, C. and Timmerer, C.. 2012. Dynamic Adaptive Streaming over HTTP Dataset. ACM-MMSys '12 - 3rd Multimedia Systems Conference. Pages 89-94.

[8] Carlos Danilo M. Regis, Gilney Christierny B. dos Anjos, Jean Felipe F. de Oliveira, Jerônimo S. Rocha, Mylène Christine Q. Farias, Marcelo S. de Alencar. Transcodificação de vídeo digital para receptores portáteis, IV SEGeT Simpósio de Excelência em Gestão e Tecnologia

[9] Prof. Roberto Tenorio Figueiredo Apostila de Sistemas Multimídia; FACULDADE DE CIÊNCIAS APLICADAS E SOCIAIS DE PETROLINA, 2010

[10] Prof. Roberto Willrich, Sistemas Multimídia distribuídos, UFSC, 2000

[11] Microsoft Smooth Streaming, <http://www.iis.net/download/smoothstreaming> ; 2011.

[12] Adobe HTTP Dynamic Streaming, http://help.adobe.com/en_US/HTTPStreaming/1.0/Using/index.html; 2011.

[13] Pantos, R., May, W. 2010. HTTP Live Streaming, IETF draft ;2010 <http://tools.ietf.org/html/draft-pantos-http-live-streaming-04> ; 2010.

[14] Luiz Fernando de Barros Campos. METADADOS DIGITAIS: revisão bibli-

ográfica da evolução e tendências por meio de categorias funcionais, ECI-UFMG; 2007

[15] Regina Borges de Araújo. *Computação Ubíqua: Princípios, Tecnologias e Desafios*, UFSCar, XXI Simpósio Brasileiro de Redes de Computadores; 2003

[16] Dey, Anind K.; Abowd, Gregory D.. *Towards a Better Understanding of Context and Context-Awareness*. 1st International Symposium on Handheld and Ubiquitous Computing, 1999.

[17] Carlos Ferraz, Paulo Martinelli Hemmlepp. *Aplicações de TV Digital sensíveis a contexto para dispositivos móveis*, Universidade Federal de Pernambuco Graduação em Ciência da Computação; 2006

[18] ETSI TS 126 247 V10.0.0, 2011. *Universal Mobile Telecommunications System (UMTS); LTE; Transparent end-to-end Packet-switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH) (3GPP TS 26.247 version 10.0.0 Release 10)* http://www.etsi.org/deliver/etsi_ts/126200_126299/126247/10.00.

[19] Marcello de Lima Azambuja, Rafael Silva Pereira, *globo.com*, 2008; *Arquituras de Distribuição de Vídeos na Internet*, REVISTA DE RADIODIFUSÃO - VOLUME 03, N° 03

[20] W. Jason Woodrow, *Computer Geekery*, *Correcting for audio/video sync issues with the ffmpeg programs ITSOFFSET switch*, 2013 <http://wjwoodrow.wordpress.com/2013/02/04/correcting-for-audiovideo-sync-issues-with-the-ffmpeg-programs-itsoffset-switch/>

[21] *How Bad is Good Enough? Exploring Mobile Video Quality Trade-offs for Bandwidth-Constrained Consumers* Anne Oeldorf-Hirsch; Pennsylvania State University Jonathan Donner; Microsoft Research India Edward Cutrell; Microsoft Research India

[22] Song, W., Tjondronegoro, D., and Docherty, M. *Saving bitrate vs. pleasing users*. *Proceedings of the 19th ACM international conference on Multimedia - MM 11*, ACM Press (2011), 403.

[23] *Streaming Learning Center; Streaming 101: The Basics - Codecs, Bandwidth, Data Rate and Resolution -*

See more at: <http://www.streaminglearningcenter.com/articles/>

[streaming-101-the-basics—codecs-bandwidth-data-rate-and-resolution.html](http://www.streaminglearningcenter.com/articles/streaming-101-the-basics—codecs-bandwidth-data-rate-and-resolution.html) [hash.K1t6y7kH.dpuf](https://www.dpu.com/101-the-basics—codecs-bandwidth-data-rate-and-resolution.html)

[24] <http://windows.microsoft.com/pt-br/windows/getting-best-display-monitorgetting-best-display-monitor=windows-7>

[25] An HD Primer Douglas Spotted Eagle/VASST Instructor ©2007 Sundance Media Group <http://www.sundancemediagroup.com/articles/camcorders/HDprimer.htm>

[26] Vinton G. Cerf, Robert E. Kahn, A Protocol for Packet Network Intercommunication, IEEE Transactions on Communications, Vol. 22, No. 5, May 1974 pp. 637-648

[27] ContextF: Framework de apoio a construção de aplicações móveis sensíveis ao contexto Marcus Henrique Barbosa Universidade Federal de Juiz de Fora, Eduardo Barrére Universidade Federal de Juiz de Fora; 2013