



Gerenciamento de buffer em Players de Vídeo IPTV

Marcos Paulo Mendes

JUIZ DE FORA
09 DE DEZEMBRO, 2014

Gerenciamento de buffer em Players de Vídeo IPTV

MARCOS PAULO MENDES

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Orientador: Marcelo Ferreira Moreno

JUIZ DE FORA
09 DE DEZEMBRO, 2014

GERENCIAMENTO DE BUFFER EM PLAYERS DE VÍDEO IPTV

Marcos Paulo Mendes

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS 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:

Marcelo Ferreira Moreno
Doutor em Informática

Eduardo Barrére
Doutor em Engenharia de Sistemas e Computação

Romualdo Monteiro de Resende Costa
Doutor em Informática

JUIZ DE FORA
DE 09 DE DEZEMBRO, 2014

Resumo

Com a evolução tecnológica, o uso sistemas IPTV tem crescido consideravelmente. Um dos serviços normalmente incluídos em IPTV é a TV Linear, que permite ao usuário a navegação de canais. Outro serviço é o Vídeo sobre Demanda, que permite ao usuário iniciar o consumo do conteúdo escolhido no momento em que lhe convier, oferecendo ainda operações como avanço, retrocesso e pausa. Neste contexto, desafios na distribuição do conteúdo multimídia destes serviços vem surgindo. Um desses desafios é justamente a percepção do usuário sobre a qualidade do serviço IPTV, mensurada em termos de qualidade de experiência (QoE). Sendo assim, este trabalho tem como objetivos desenvolver um player de áudio e vídeo específico para IPTV e analisar os problemas que podem ocorrer durante a recepção de conteúdo. Para os problemas identificados, propõe-se colocar em prática soluções, principalmente para a variação estatística do atraso de entrega do conteúdo multimídia, que pode ocorrer devido ao meio de transmissão ou nos pontos fim-a-fim, melhorando assim a qualidade de experiência do usuário.

Palavras-chave: IPTV, VoD, TV Linear, Buffer, Jitter.

Abstract

With technological developments, the use of IPTV systems has grown considerably. One of the services normally included in IPTV is the Linear TV, which allows the user to browse channels. Another service is the Video on Demand, which allows the user start the consumption of the chosen content at the time that suits, and providing operations as forward, rewind and pause. In this context, challenges in the distribution of multimedia content of these services is emerging. One such challenge is precisely the user's perception of the quality of the IPTV service, measured in terms of quality of experience (QoE). Thus, this study aims to develop an audio and video player specific for IPTV and analyze the problems that can occur when receiving content. To the problems identified , it is proposed to put in place solutions, mainly for statistical variation of the delivery delay of multimedia content that may occur due to the transmission media or in end-to-end points, thus improving the quality of experience user.

Keywords: ITPV, VoD, TV Linear, Buffer, Jitter.

Agradecimentos

A Deus pela força, proteção e por ser o principal responsável por mais esta conquista.

A meus pais e meu irmão pela força, apoio e confiança depositada em mim. Aos meus amigos, pelo encorajamento, pela força e pelo apoio.

Ao professor Marcelo Ferreira Moreno pela paciência, dedicação, incentivo e pelo seu conhecimento que muito me auxiliou no desenvolvimento deste trabalho.

Aos colegas do Laboratório de Aplicações e Inovação em Computação (LApIC), que fizeram parte da minha trajetória durante estes anos.

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.

*“Sabemos o que somos, mas não sabemos
o que poderemos ser”.*

William Shakespeare

Sumário

Lista de Figuras	7
Lista de Tabelas	8
Lista de Abreviações	9
1 Introdução	10
1.1 Justificativa	11
1.2 Objetivos	12
1.2.1 Objetivo Geral	12
1.2.2 Objetivos Específicos	12
1.3 Metodologia	13
1.4 Estrutura do trabalho	14
2 Revisão Bibliográfica	15
2.1 Conteúdo Audiovisual	15
2.1.1 Conteúdo ao vivo	15
2.1.2 Conteúdo armazenado	15
2.2 IPTV e a padronização ITU-T	16
2.2.1 Definição de IPTV	16
2.2.2 Padronização para terminais	16
2.3 Protocolo UDP	17
2.4 Protocolo RTP	18
2.5 Protocolo RTCP	20
2.6 Protocolo RTSP	20
2.7 Padrão MPEG-2	22
3 Solução Proposta	25
3.1 Introdução	25
3.2 Visão Geral	26
3.3 Gerenciamento do buffer pelo tempo	26
3.4 Controle do crescimento do ponto de reprodução	28
3.5 Elasticidade	29
3.6 Abordagem por algoritmo	29
3.7 Perda de pacotes	30
3.8 Solução e o ambiente IPTV	30
4 Implementação	31
4.1 TV Linear	32
4.2 Vídeo sobre Demanda	32
5 Resultados obtidos	33
5.1 TV Linear	33
5.2 VoD	34
5.2.1 Armazenamento físico	36

6	Considerações Finais	38
	Referências Bibliográficas	39

Lista de Figuras

1.1	Representação da atuação do buffer em uma reprodução	12
2.1	Representação do datagrama UDP	17
2.2	Representação do cabeçalho de um pacote RTP	19
2.3	Representação de uma comunicação cliente/servidor RTSP	22
2.4	Estrutura do fluxo MPEG-2 (Cavendish, 2005)	23
2.5	Composição do Program Stream (Inamori; Naganuma; Endo, 1999)	23
2.6	Composição do Transport Stream (Inamori; Naganuma; Endo, 1999)	24
2.7	Sincronização entre o codificador e o decodificador (Cavendish, 2005)	24
3.1	Variáveis e o que representam na gerência do buffer	27
5.1	Comparação entre a duração contida no buffer e o Ponto de reprodução T para a transmissão 1	33
5.2	Comparação entre a duração contida no buffer e o Ponto de reprodução T para a transmissão 2	34
5.3	Comparação entre a duração contida no buffer e o Ponto de reprodução T para o Vídeo 1	35
5.4	Comparação entre a duração contida no buffer e o Ponto de reprodução T para o Vídeo 2	36
5.5	Espaço utilizado para cada duração em buffer	36

Lista de Tabelas

Lista de Abreviações

DCC	Departamento de Ciência da Computação
UFJF	Universidade Federal de Juiz de Fora
IPTV	Internet Protocol Television
RTP	Real-time Transport Protocol
RTCP	Real-time Transport Control Protocol
RTSP	Real Time Streaming Protocol
SBTVD	Sistema Brasileiro de Televisão Digital
NGN	Next Generation Networking
VoD	Video on Demand
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
MPLS	Multi Protocol Label Switching
IP	Internet Protocol
AAC	Advanced Audio Coding
ITU	International Telecommunication Union
SDL	Simple DirectMedia Layer
PTS	Presentation Time Stamp
DTS	Decode Time Stamp
PCR	Program Clock Reference
SCR	System Clock Reference
STC	System Time Clock
SDTV	Standard Definition Television
PES	Packetize Elementary Stream
FPS	Frames Per Second
TS	Transpor Stream

1 Introdução

É fato que a evolução tecnológica traz diversos benefícios ao cotidiano. Também é fato que esses benefícios tendem a ser cada vez mais exigentes em termos de uso de recursos computacionais. Sendo assim, com a evolução dos meios físicos de transmissão, permitindo um tráfego de informações cada vez maior, a possibilidade de serviços convergentes em uma única rede se torna cada vez mais real. Ou seja, um mesmo ponto de conexão poderá ser o responsável pelo acesso à internet, recepção do sinal de TV Digital, voz, videoconferência, dentre outros. Esse conceito é conhecido como Next Generation Networking (NGN), e é sobre ele que está o desenvolvimento da tecnologia de transmissão IPTV (Internet Protocol Television) e sua arquitetura (ITU-T Y.1910, 2008).

A arquitetura IPTV possibilita a transmissão de conteúdo em alta definição, com múltiplos canais de áudio e, junto com o conteúdo, é possível fazer a distribuição de dados, que podem ser aplicações multimídia. É possível também se ter um sistema de correção de erros que podem surgir da existência de ruídos na transmissão (Barbosa; Soares, 2008; Batista; Moreno; Soares, 2011). Sendo assim, diversos serviços podem ser oferecidos (Zapater, 2007) como Video on Demand (VoD), gravação seletiva de conteúdo e aplicações interativas (Barbosa; Soares, 2008) que são benéficas tanto para conteúdos de aprendizagem (ensino à distância) quanto para interesses como venda de produtos e prestação de serviço (Soares et al., 2010).

A distribuição de conteúdo IPTV é feita via protocolo IP (Internet Protocol) (IETF 4445, 2006). Uma vez que o áudio e vídeo são comprimidos e codificados, são entregues ao cliente utilizando o protocolo RTP (Real Time Transport Protocol) (IETF 3550, 2003) sobre UDP (User Datagram Protocol) (IETF 768, 1980; ITU-T Y.1910, 2008). Como a distribuição multimídia possui um alto nível de tolerância a perda de pacotes, se houver perda de alguns pacotes do conteúdo de áudio e vídeo não será crítico para a reprodução. Além disso, a distribuição multimídia possui um baixo nível de tolerância ao atraso na entrega. Sendo assim, a escolha do UDP é feita pelo fato da redução de sobrecargas para controle (*overhead*) que são geradas na transmissão por TCP (Transmission

Control Protocol) (IETF 793, 1981). Esse *overhead* inclui retransmissão de pacotes, que insere um atraso e o controle de congestionamento que, por sua vez, reduz a taxa de transmissão.

Essa distribuição do conteúdo é realizada em redes projetadas especificamente para tráfego multimídia, fazendo uso de recursos que garantem baixa latência e garantia de entrega. Por exemplo redes MPLS (Multi Protocol Label Switching) (Zapater, 2007) conseguem oferecer mecanismos que garantam a qualidade de serviço (QoS) (ITU-T Y.1901, 2009).

Apesar de haver a garantia de Qualidade de Serviço (QoS) na entrega de conteúdo IPTV (ITU-T Y.1901, 2009), o recebimento deste conteúdo não está livre de atrasos (IETF 4445, 2006), podendo ser desde uma variação na entrega por parte do servidor a um possível atraso no momento de recebimento e exibição do conteúdo por parte do cliente. Estes atrasos se tornam extremamente perceptíveis quando a sua existência não é considerada nas implementações de players multimídia (Tanenbaum; Wetherall, 2011).

1.1 Justificativa

Quando um conteúdo chega ao cliente, ele deve ser reproduzido no momento certo. Em geral, esse não será o momento em que o pacote RTP chegou ao cliente, pois os pacotes levarão tempos ligeiramente diferentes para serem transportados na rede. Mesmo que os pacotes sejam despachados exatamente com os intervalos certos entre eles no servidor, eles chegarão ao cliente com tempos relativamente diferentes (Tanenbaum; Wetherall, 2011). Segundo Begazo (2012), diversos fatores são responsáveis pela degradação da rede como a perda de pacotes, atraso fixo, variação de atrasos, corrupção de pacotes, duplicação de pacotes e reordenação de pacotes.

Segundo Bhadada; Sharma (2010), a solução é a utilização de buffers na recepção, justamente para evitar interrupções na reprodução que podem ocorrer devido a variação do atraso. Essa ideia consiste em armazenar um determinado número de pacotes antes destes serem reproduzidos. Com isto, a variação estatística no atraso, conhecido como jitter, pode ser controlado de tal forma a se obter uma reprodução suave. Na figura 1.1 pode se ver como seria o funcionamento desta solução. Os quadrados cinzas numerados

são pacotes enviados pelo Servidor ao Cliente. Nota-se que a reprodução só será iniciada após a recepção de 4 pacotes, amenizando ao usuário a percepção do atraso que haveria entre os pacotes 2 e 3 e entre os pacotes 5 e 6.

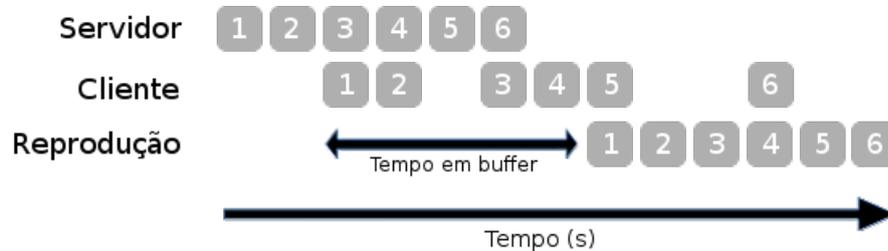


Figura 1.1: Representação da atuação do buffer em uma reprodução

Além disso, nesta solução se tem uma incógnita, que é justamente a determinação do tempo em que o cliente irá preencher o buffer. Se este valor for constante, o buffer poderá ser efetivo em alguns casos e totalmente ineficiente para outros casos.

Uma vez que o buffer consegue gerenciar a percepção do usuário com relação ao atraso, tem-se um aumento significativo na qualidade de experiência (QoE), que é um aspecto que tem se tornado cada vez mais importante para o sucesso de serviços e produtos voltados para IPTV (Duque, 2008).

1.2 Objetivos

1.2.1 Objetivo Geral

O principal objetivo deste trabalho é desenvolver uma solução para o gerenciamento avançado de buffer de conteúdo audiovisual especificamente para serviços IPTV. Além disso, será desenvolvido o player que inclua esta solução. Esta gerência deve ser dinâmica para evitar a ineficiência de se ter um tamanho constante de buffer.

1.2.2 Objetivos Específicos

Primeiramente, tem-se como objetivo implementar o player de áudio/vídeo para IPTV, capaz de suportar os serviços de TV Linear e de Vídeo sob Demanda. Para isto, será usada na recepção e decodificação do stream a biblioteca FFMpeg (Bellard; Niedermayer, 2014),

além da biblioteca SDL (Lantinga, 2014), para a manipulação gráfica como exibição dos frames de vídeo e reprodução do áudio.

Tem-se também como objetivo elaborar uma gerência de buffer para o player de vídeo de TV Linear e VoD, de forma que o mesmo se adapte à variação do atraso na transmissão. Para isto, será utilizado como métrica a quantidade em segundos de conteúdo (áudio e vídeo) recebido em um intervalo de tempo. Com isto, fatores específicos da rede e da demultiplexação que podem estar inserindo atraso serão analisados de forma indireta pois, se houver um destes fatores ativos, então a quantidade de conteúdo recebido pelo cliente neste intervalo de tempo será afetado diretamente.

1.3 Metodologia

A elaboração deste trabalho se baseou em trabalhos recentes na área de multimídia e análise de buffer para streaming de conteúdo multimídia. Este trabalho pode ser dividido nas seguintes etapas: estudo teórico, desenvolvimento da solução, análise da solução, implementação e testes, análise dos resultados, redação da monografia e entrega.

O estudo teórico foi realizado fazendo uso de artigos de congressos e seminários voltado à área de multimídia, como por exemplo o XVII Simpósio Brasileiro de Sistemas Multimídia e Web. Além disso, foi feito o uso de diversas dissertações que poderiam contribuir para o desenvolvimento da solução.

Com a base teórica anteriormente referida, se iniciou então o desenvolvimento da solução. Foram analisados diversos cenários na distribuição dos conteúdos IPTV, com o propósito de obter uma solução que atendesse a estes cenários.

Com a solução proposta, se iniciou o desenvolvimento desta solução no player de TV Linear e VoD. Estes players foram implementados conforme o padrão proposto pelo ITU-T (ITU-T H.721, 2009) para ambientes IPTV.

Com a implementação finalizada, foram realizados os testes a fim de se obter os resultados para a solução proposta. Os resultados foram analisados com o princípio de ambiente IPTV.

No processo de redação da monografia foi finalizado a escrita da monografia, como todo, que se iniciou no processo anterior.

Na entrega será feita tanto a entrega da monografia quanto do código implementado.

1.4 Estrutura do trabalho

O capítulo 2 contém a revisão bibliográfica. Nele são apresentados os princípios que foram utilizados para a análise e elaboração da solução.

O capítulo 3 mostra a solução proposta, tanto de uma perspectiva matemática quanto de uma perspectiva de algoritmo.

O capítulo 4 apresenta as etapas de implementação deste trabalho. Esse capítulo descreve como o processo de desenvolvimento foi realizado.

O capítulo 5 contém os resultados obtidos com a solução proposta implementada.

O capítulo 6 apresenta as considerações finais, onde constam possíveis trabalhos futuros.

2 Revisão Bibliográfica

2.1 Conteúdo Audiovisual

O conteúdo audiovisual é um conteúdo (dados) de áudio, vídeo, ou ambos multiplexados, com o propósito de ser consumido pelos usuários de um serviço, o qual pode consistir em uma transmissão ao vivo ou de conteúdo armazenado.

2.1.1 Conteúdo ao vivo

Nesta abordagem, o cliente faz solicitação ao servidor de um conteúdo que está sendo gerado, como por exemplo um jogo de futebol ou uma programação de um canal de televisão. Sendo assim, as operações de avanço, retrocesso e pausa não estarão disponíveis no servidor. Esta abordagem é conhecida como TV Linear no âmbito de serviços IPTV.

Como esta abordagem consiste na apresentação de um conteúdo que está sendo gerado em tempo real, um tamanho de tempo em buffer com valor elevado pode afetar diretamente a QoE. Segundo Zapater (2007), um aspecto importante relacionado ao atraso é a perspectiva do usuário quando compara o atraso do sinal IPTV com o sinal dos canais transmitidos via serviços tradicionais de TV aberta. Conteúdos ao vivo são, portanto, mais suscetíveis a atrasos iniciais inseridos por técnicas de bufferização.

2.1.2 Conteúdo armazenado

Nesta abordagem, o cliente faz uma solicitação de um conteúdo previamente armazenado no servidor. Como este conteúdo já existe por completo no servidor, operações de avançar, retroceder, pausar estarão disponíveis para esta reprodução. Esta abordagem é conhecida como VoD (Vídeo sobre demanda) no âmbito de serviços IPTV.

Como esta abordagem representa a apresentação de um conteúdo que está armazenado por completo no servidor, um tamanho de tempo em buffer não é tão crítico quanto na TV Linear, visto que o conteúdo não está sendo gerado em tempo real.

2.2 IPTV e a padronização ITU-T

2.2.1 Definição de IPTV

A União Internacional de Telecomunicações define IPTV como serviço multimídia incluindo áudio, vídeo, imagens, textos, entregue por meio de protocolo IP em redes gerenciadas, para garantir os níveis necessários de qualidade de serviço (QoS), qualidade de experiência (QoE), segurança e interatividade (ITU-T Y.1901, 2009).

A plataforma IPTV oferece uma melhora significativa na qualidade de vídeo do conteúdo transmitido, suportando vídeos estéreo para a exibição em 3D ou qualidades superiores a Full HD para expansão futura (Mokarzel, 2010). Por fazer uso da rede IP, a interatividade se torna possível e natural, haja visto que a mesma rede que traz o conteúdo pode ser a rede de retorno.

Um dos serviços que podem ser prestados fazendo uso de IPTV é justamente o de TV Linear. Nele o usuário tem um conjunto de canais com suas programações que pode assistir alternando-os através de um controle remoto. A grande vantagem é que não há limites para o número de canais que podem ser oferecidos, além de outras vantagens, como as já citadas.

Outro serviço é o Vídeo sob Demanda onde o usuário poderá fazer a escolha, por exemplo, de um filme que queira assistir e em qual momento. Além disso, ele poderá ter comandos de controle de reprodução similares ao do aparelho de Blu-Ray.

2.2.2 Padronização para terminais

Em 2009, o ITU publicou um padrão para o terminal IPTV (ITU-T H.721, 2009). Este padrão é a base para a especificação dos formatos do áudio, vídeo, transmissão e multiplexação utilizados neste trabalho. Os padrões são:

- **VoD:** O formato para a codificação do vídeo é H.264 e o áudio é AAC. Para o transporte, o padrão define RTP, com o RTSP (Real Time Streaming Protocol) (IETF 2326, 1998) sendo o protocolo de controle de transmissão. Para a multiplexação áudio/vídeo/dado, o padrão define o uso do MPEG-2 TS (ISO/IEC 13818-1, 2000).

- **TV Linear:** O formato para a codificação do vídeo é H.264 e o áudio é AAC. Para o transporte, o padrão define RTP. Para a multiplexação áudio/vídeo/dado, o padrão define o uso do MPEG-2 TS (ISO/IEC 13818-1, 2000).

2.3 Protocolo UDP

O protocolo UDP (User Datagram Protocol) é um protocolo da camada de transporte voltado para a transmissão de datagramas (Tanenbaum; Wetherall, 2011). É bastante utilizado em aplicações multimídia (Calado, 2010). As características do UDP são (Duque, 2008):

- Não estabelece conexão;
- Os dados são enviados em blocos (datagramas);
- Cada datagrama contém o endereço e porta de destino;
- Permite tanto transmissão unicast quanto multicast;
- Não garante a entrega dos dados;
- Não garante ordem na entrega dos datagramas;
- Não faz uso de confirmação de recebimento de pacote;
- Não implementa mecanismo para controle de fluxo;
- Não implementa mecanismo para controle de congestionamento;

Porta de origem	Porta de destino
Comprimento do UDP	Checksum do UDP
Dados	

Figura 2.1: Representação do datagrama UDP

Na figura 2.1 tem-se a representação de um datagrama no protocolo UDP. Nota-se nesta figura que o cabeçalho não contém nenhum campo que permita ter número sequencial de datagrama. Sendo assim, caso haja necessidade de ordem dos pacotes, cabe ao protocolo que o utilizar fazer tal controle.

2.4 Protocolo RTP

O protocolo RTP (Real-time Transport Protocol) é um protocolo de transporte em tempo real que normalmente trabalha sobre o UDP. Uma aplicação multimídia consiste em vários fluxos de áudio, vídeo, texto e possivelmente outros fluxos. A função básica do RTP é multiplexar esses diversos fluxos de dados em tempo real sobre um único fluxo de pacotes UDP (Tanenbaum; Wetherall, 2011).

Como UDP não garante ordem na entrega dos datagramas, o RTP possui em seu cabeçalho um campo que recebe um número de uma unidade maior do que o seu predecessor (IETF 768, 1980). Através disso, é possível o cálculo do número de pacotes que estão faltando. A retransmissão de pacotes faltando neste caso não é viável, pois os pacotes retransmitidos poderiam chegar muito tarde (Tanenbaum; Wetherall, 2011).

As características do RTP são (Duque, 2008):

- Numeração sequencial para os pacotes;
- Selo de temporização, possibilitando a correta temporização dos pacotes contendo áudio ou vídeo;
- Envio de pacotes sem retransmissão;
- Identificação da fonte do dado;
- Identificação do conteúdo;
- Possibilita sincronismo entre diferentes fluxos;

Na figura 2.2 tem-se uma representação de cabeçalho RTP. Os campos nela representados têm o seguinte significado (Muezerie, 2005):

Ver.	P	X	CC	M	Tipo do dado	Número de Sequência
Período de tempo (timestamp)						
Identificador da origem						

Figura 2.2: Representação do cabeçalho de um pacote RTP

- **Ver.:** Contém a versão sendo usada;
- **P:** Indica se o pacote foi completado até chegar a um múltiplo de 4 bytes;
- **X:** Indica que um cabeçalho de extensão está presente;
- **CC:** Indica quantas fontes estão presentes;
- **M:** Um marcador de uso pela aplicação;
- **Tipo de dado:** Qual a codificação está sendo usada;
- **Número de sequência:** Contador que é incrementado a cada envio de pacote RTP;
- **Período de tempo (timestamp):** Indica quando a primeira amostra do pacote foi criada na origem.
- **Identificador da Origem:** usado na demultiplexação, indica a qual fluxo o pacote pertence;

O RTP adiciona seus próprios timestamps a cada stream de áudio e vídeo independentemente, permitindo a transmissão independente sobre a rede. O cliente usa então os timestamps do RTP para estabelecer o sincronismo entre o áudio e o vídeo recebidos sobre RTP/UDP/IP. (Duque, 2008). Isso somente é possível se o mesmo relógio de sincronização for usado para ambas as fontes, áudio e vídeo. No caso em que relógios diferentes são usados, clientes RTP podem contar com o auxílio do protocolo descrito na próxima Seção.

2.5 Protocolo RTCP

O protocolo RTCP (IETF 3550, 2003) (Real-time Transport Control Protocol) é usado para obter informações, tanto pelo servidor quanto pelo cliente, não realizando o transporte de nenhuma amostra de mídia. Pode ser usado para fornecer ao servidor informações sobre o atraso, a variação do atraso (jitter), largura de banda, congestionamento e outras propriedades de rede. Ele também é usado para se conseguir o sincronismo entre fluxos distintos. Além disso, ele pode também fornecer um modo para nomear as diversas origens, por exemplo em texto. Essas informações podem ser exibidas na tela do cliente a fim de indicar quem está comunicando no momento (Tanenbaum; Wetherall, 2011), o que é útil em comunicação muitos-para-muitos, por exemplo (videoconferência).

2.6 Protocolo RTSP

A principal função do protocolo RTSP (Real Time Streaming Protocol) é o controle de sessão, permitindo à transmissão ter operações similares as que possuem um aparelho de Blu-Ray como Play, Pause e Rewind (Zapater, 2007). Por operar predominante sobre comunicações unicast, essas operações não são realizadas localmente e sim no servidor de streaming, ou seja, quando um Pause é solicitado, o servidor para de enviar o streaming ao cliente. Trata-se de uma abordagem acertada, visto que economiza recursos da rede e do servidor.

As operações possíveis no servidor RTSP são (IETF 2326, 1998):

- **DESCRIBE.:** Obtém uma descrição da apresentação ou objeto de mídia identificados pela URL informada.
- **OPTIONS.:** Retorna os comandos possíveis a serem executados pelo cliente. Esse comando pode ser executado a qualquer momento, pois ele não influencia no estado do servidor.
- **ANNOUNCE.:** Tem 2 propósitos: quando enviado do cliente para o servidor, envia a descrição do objeto de mídia identificado pela URL requisitada ao servidor. Se for do servidor para o cliente, então ele está notificando ao cliente uma atualização

na descrição da sessão em tempo real;

- **SETUP:** para uma URI informada, este comando serve para especificar o mecanismo de transporte a ser usado para transmitir o conteúdo. Este comando pode ser executado quando já tem conteúdo em reprodução, com o propósito de alterar os parâmetros de transporte que um servidor permite mudar.
- **PLAY:** Informa ao servidor para iniciar o envio de dados utilizando o mecanismo especificado com SETUP. Como parâmetro deste comando, é possível informar um intervalo de reprodução. Isto permite que seja possível a operação Seek por parte do cliente.
- **PAUSE:** Informa ao servidor para interromper o envio do conteúdo temporariamente.
- **TEARDOWN:** Informa ao servidor que deve parar o envio dos conteúdos referente a URI informada. Basicamente, encerra a transmissão.
- **GET_PARAMETER:** Obtém o valor de um parâmetro de uma apresentação ou stream especificado pela URI informada.
- **SET_PARAMETER:** Altera o valor de um parâmetro de uma apresentação ou stream especificado pela URI informada.
- **REDIRECT:** Informa ao cliente que ele deve se conectar a outro servidor.
- **RECORD:** Inicia a gravação de um intervalo de dados do conteúdo de acordo com a descrição da apresentação. O servidor é quem decide onde armazenar os dados gravados, na URI solicitada ou em outra URI.

Na figura 2.3 tem-se uma representação de um exemplo de comunicação entre um cliente e servidor RTSP. Este protocolo não faz o transporte do conteúdo em si, ele apenas fornece um meio de controle a transmissão através de um identificador de sessão. A seta indicando o Fluxo de conteúdo, pode estar sendo transportado, por exemplo, através do protocolo RTP.

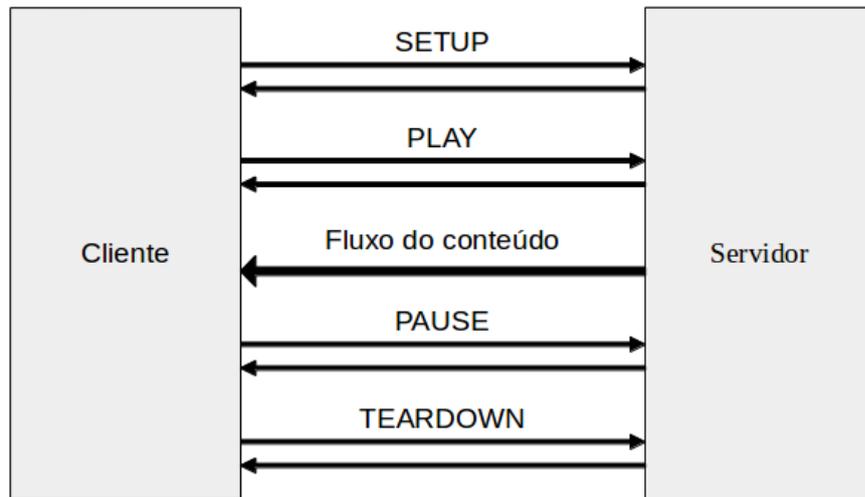


Figura 2.3: Representação de uma comunicação cliente/servidor RTSP

2.7 Padrão MPEG-2

O MPEG-2 (ISO/IEC 13818-1, 2000), foi iniciado em 1990 como uma evolução do MPEG-1 e publicado em 1995 e uma segunda edição publicada em 2000. O objetivo deste padrão é prover uma taxa de vídeo de 1,5 Mbps a 15 Mbps, adequados para sinais de televisão padrão (SDTV) e taxas de 15 Mbps a 30 Mbps para HDTV. Nele é estabelecido a forma de compressão para o fluxo de sistemas, para vídeo e áudio. Nele foi definida uma camada de sistema, MPEG-2 Systems, responsável pela divisão e encapsulamento de cada fluxo comprimido em pacotes, pela inserção de informações de sincronização entre fluxos de mídias distintas, pela multiplexação do fluxos encapsulados e pelo transporte da informação de referência do relógio utilizado no codificador (Cavendish, 2005).

No MPEG-2 Systems, um esquema de empacotamento hierárquico é usado. Primeiramente, o fluxo elementar do codificador (e.g. fluxo de áudio) é empacotado na subcamada PES (Packetized Elementary Stream) (Medeiros, 2010). Essa subcamada tem a função de identificar cada fluxo através de ID de fluxo, e a sincronização intra e intermídia (Cavendish, 2005). Uma vez que esta camada é processada, o resultado é empacotado em um Transport Stream ou Program Stream, dependendo da aplicação. Alguns dos valores do cabeçalho PES são determinados através da análise do fluxo elementar. A marca de tempo de apresentação PTS (Presentation Time Stamp) e a marca de tempo de decodificação DTS (Decoding Time Stamp), usadas para o sincronismo entre áudio

e vídeo, são exemplos desses valores (Inamori; Naganuma; Endo, 1999). Na figura 2.4 pode-se observar como é a representação da estrutura do fluxo MPEG-2. Nas figuras 2.5 e 2.6 pode-se observar como é gerado, a composição e como é demultiplexado bem como a localização do PTS e DTS no fluxo.

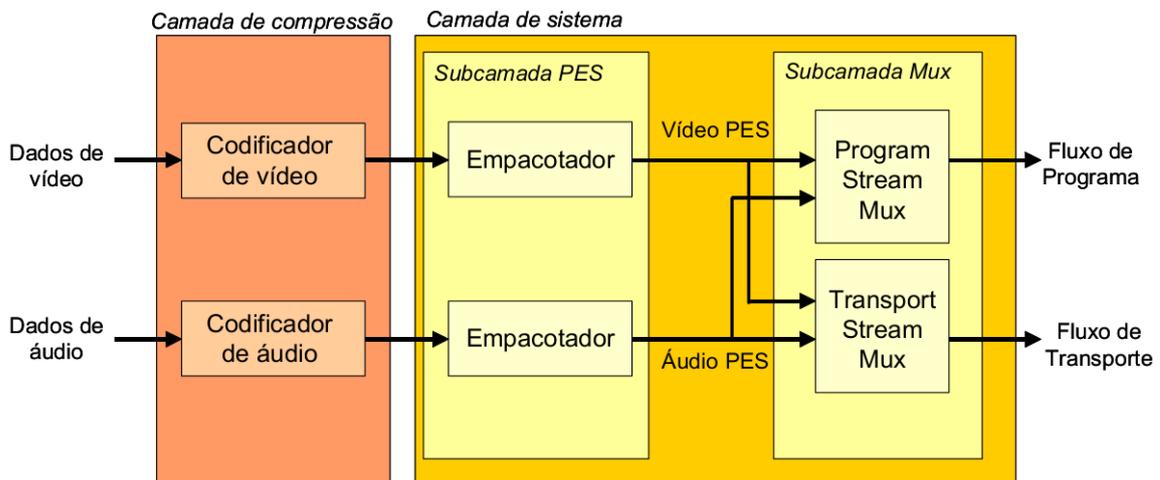


Figura 2.4: Estrutura do fluxo MPEG-2 (Cavendish, 2005)

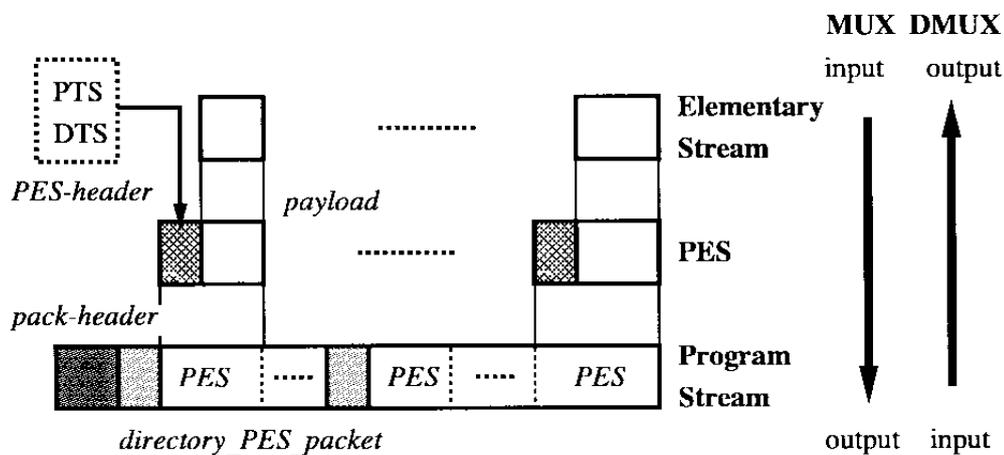


Figura 2.5: Composição do Program Stream (Inamori; Naganuma; Endo, 1999)

O PTS é entregue em unidades relacionadas com o clock de referência global, como o PCR (Program Clock Reference) e o SCR (System Clock Reference), o qual também é transmitido no Transport Stream ou Program Stream. Os valores do PCR e do SCR são definidos em termos de um relógio de sistema comum denominado STC (System Time Clock) e significam o instante de tempo em que o último bit desses campos entra no decodificador (Cavendish, 2005). Na figura 2.7 pode-se notar como ocorre a idéia do

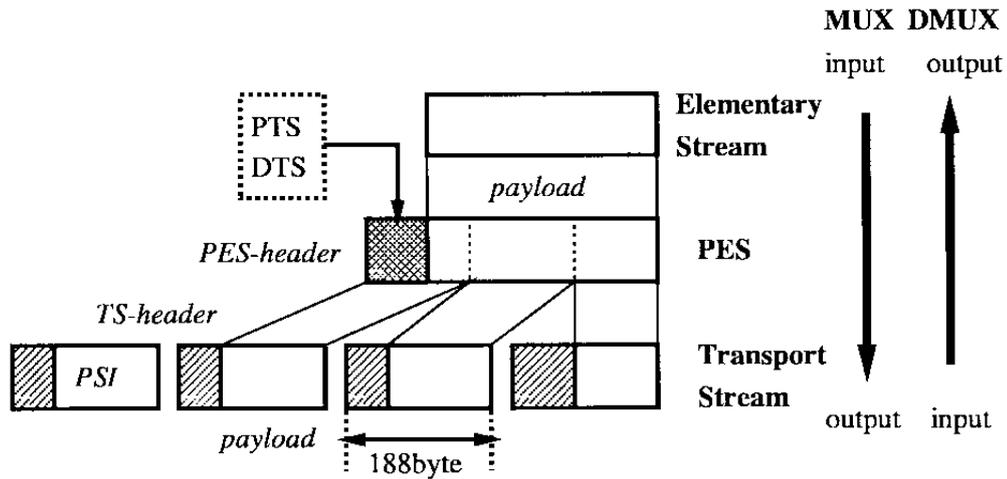


Figura 2.6: Composição do Transport Stream (Inamori; Naganuma; Endo, 1999)

sincronismo entre o codificador e o decodificador.

O DTS, presente apenas no fluxo de vídeo, indica o instante de tempo em que a unidade de apresentação deve ser entregue ao respectivo decodificador e é utilizado quando é necessária a reordenação de quadros, no decodificador (Cavendish, 2005).

Uma vez que essa marca de tempo é informada, o sincronismo entre diferentes fluxos elementares pode ser realizado ajustando a decodificação dos fluxos elementares a uma base de tempo central. Essa base de tempo pode ser o clock de um dos decodificadores, pode ser o clock da fonte do dado, ou pode ser algum clock externo.

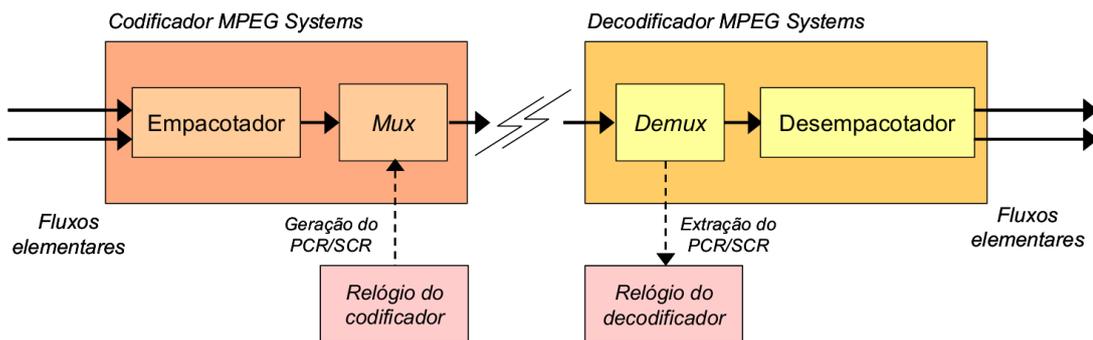


Figura 2.7: Sincronização entre o codificador e o decodificador (Cavendish, 2005)

3 Solução Proposta

3.1 Introdução

A análise para se determinar o tempo em buffer pode ser realizada levando em consideração diferentes parâmetros. Como pode ser visto em Bhadada; Sharma (2010), parâmetros como a velocidade do meio de transmissão, a taxa de frames por segundo e o tamanho total do conteúdo são utilizados para calcular o tempo de buffer inicial. Em Duque (2008), é proposto um cálculo de dimensionamento de buffer, porém o valor calculado é estático e não leva em conta as condições do servidor para entregar o conteúdo em um determinado momento e nem do cliente, como por exemplo a demultiplexação deste conteúdo. Sendo assim, fatores como o tempo de demultiplexação em um instante no cliente ou uma pequena sobrecarga no servidor não são considerados.

Uma vez que o fator de análise é o tempo de duração do conteúdo disponível para reprodução, outros parâmetros como a velocidade do meio, taxa de frames por segundo e outros, são descartados dessa análise, não por ser menos importantes, mas pelo fato de estarem ligados diretamente à duração do conteúdo disponível. Se houver uma diminuição na velocidade do meio de transmissão, por exemplo, a duração de conteúdo recebida irá reduzir, o que implica diretamente a existência de um possível atraso detectado pela duração. Além disso, o tempo de duração recebido também é afetado diretamente por um atraso por parte do servidor na entrega de um conteúdo em um dado momento, fator este que não é levado em consideração analisando-se apenas os fatores citados anteriormente, como a velocidade do meio. Como o tempo de duração recebido será analisado mediante a demultiplexação de um conteúdo, outro fator que será analisado de forma direta é justamente o atraso por parte do cliente em demultiplexar um conteúdo. Esse fator também não é considerado levando em conta apenas os parâmetros citados anteriormente.

Como se pode observar, a análise pelo tempo de duração do conteúdo pode determinar atrasos até mesmo não previstos que podem afetar diretamente o recebimento de conteúdo do servidor pelo cliente. Como visto, um fator de atraso que pode ocorrer no

servidor ou no cliente não pode ser determinado apenas levando em consideração o meio de transmissão, mas também deve levar em consideração o estado comportamental dos dois pontos da comunicação.

3.2 Visão Geral

Uma vez que é possível obter o PTS e a base de tempo de um frame ou amostra de áudio, é possível convertê-lo para uma unidade de tempo como segundos, para fins de análise e controle. Sendo assim, é possível o cálculo de duração de um frame ou amostra de áudio, bastando fazer a diferença entre o PTS do sucessor com seu respectivo PTS. Esse resultado é obtido na unidade do fluxo. Sendo assim, multiplicando essa diferença por uma base obtida em como o vídeo foi codificado, obtêm-se este valor em segundos. O cálculo é feito da seguinte forma:

$$d(i) = (P_{i+1} - P_i)B \quad (3.1)$$

Onde:

- $d(i)$: É a duração do i ésimo frame ou amostra de áudio, em segundos;
- P : É o valor do PTS;
- B : Base.

Com este cálculo em mãos, a análise passa a ser pela duração em segundos que se tem na recepção dos dados. Sendo assim, é esperado que dentro de um intervalo de tempo t , o valor da soma das durações dos dados recebidos não seja menor que o valor de t , a menos que esse valor possa ser suprido pelo tamanho do início da reprodução T .

3.3 Gerenciamento do buffer pelo tempo

Para ser feita a análise, um intervalo de tempo I será definido. Para uma melhor precisão na obtenção do cálculo, este intervalo deve ser mensurado em tempo real (wall clock) e

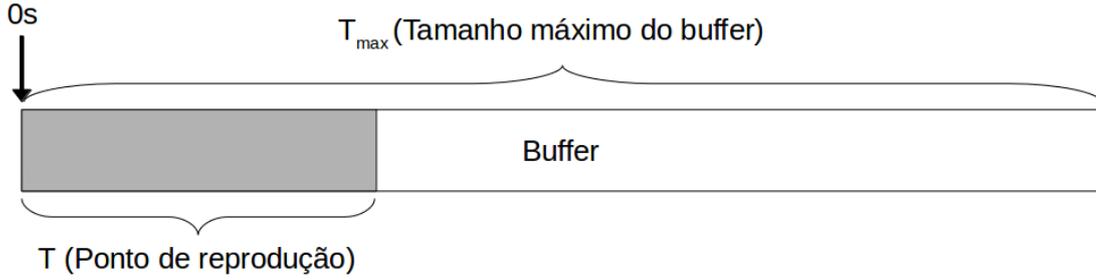


Figura 3.1: Variáveis e o que representam na gerência do buffer

deve-se evitar mensurá-lo com contadores de clock do processador, pois o Sistema Operacional hospedeiro da aplicação pode ser multitarefa, o que implica que o atraso por outro processo estar utilizando o processador também é um atraso a ser considerado.

Considerando que com a função $d(i)$ se obtém a duração de um frame ou amostra de áudio, a duração no intervalo de tempo I pode ser calculada somando a duração de cada frame recebido no intervalo I . O cálculo é feito da seguinte forma:

$$s_I(j, q) = B \sum_{i=j}^{j+q} (P_{i+1} - P_i) \quad (3.2)$$

Onde, j é a quantidade de frames recebidos de um único fluxo (áudio ou vídeo) até o início do intervalo I e q é a quantidade recebida durante o intervalo I .

A duração total recebida até o k ésimo intervalo I é calculada da seguinte forma:

$$s(j) = B \sum_{i=0}^j (P_{i+1} - P_i) \quad (3.3)$$

Onde j representa a quantidade de frames recebidos.

Como dois pacotes não podem ser entregues ao mesmo tempo e como uma amostra de vídeo pode ter duração distinta da amostra de áudio, então a duração total já recebida s_t de áudio pode ser diferente da duração já recebida de vídeo. Como a análise leva em conta a duração disponível em buffer, então o que se tem disponível é a menor duração entre eles, uma vez que a maior duração poderá apenas um dos fluxos possuir. Sendo assim, o cálculo da duração efetiva é feito da seguinte forma:

$$s_e = \min(s_{\text{audio}}, s_{\text{video}}) \quad (3.4)$$

De posse do valor de s_e , a análise poderá ser feita levando em consideração o total de intervalos I , que é kI , onde k é a quantidade de análise já realizada. Sendo assim, kI é o tempo total passado desde o início das análises.

A cada intervalo I , a duração de dados recebida neste intervalo deve ser maior ou igual que kI , ou seja, $s_e \geq kI$. Se isto for sempre verdade, então não haverá atraso, uma vez que se tem mais duração recebida do que o tempo total em análise. Quando $s_e < kI$, houve atraso na entrega em algum intervalo de kI . Seja $\delta = s_e - kI$, ou seja, a diferença entre a duração total efetiva s_e recebida e o total de intervalos kI . Se $\delta < 0$, então houve atraso em algum momento da análise total. Sendo assim, se o ponto de reprodução $T \neq 0$, então este atraso já pode ter sido analisado e ele já está sendo amenizado pelo buffer. Sendo assim, define-se Δ , o parâmetro de análise para obtenção de T , como:

$$\Delta = \delta + T \quad (3.5)$$

Ou, pode ser escrito como:

$$\Delta = s_e - kI + T \quad (3.6)$$

Com o valor de Δ , pode-se inferir diretamente um novo valor para T . Se $\Delta < 0$, então tem-se um novo atraso que ainda não foi analisado no dimensionamento de T . Sendo assim, se $-\Delta \leq T$, então o ponto de reprodução T tem espaço suficiente para cobrir este atraso. Se $-\Delta > T$, então o T não consegue cobrir este atraso que houve. Sendo assim, $T = -\Delta$. De uma forma algébrica, para cada análise realizada no intervalo I , T pode ser obtido da seguinte forma:

$$T = \max(T, -\Delta), \forall \Delta < 0 \quad (3.7)$$

3.4 Controle do crescimento do ponto de reprodução

Através da solução mostrada, pode-se perceber que o valor T cresce ao passo em que o atraso o interfere. Sendo assim, para grandes atrasos o valor de T pode ter valores altos justamente com o propósito de cobrir o valor do atraso. Sendo assim, um controle pode

ser feito limitando o valor de T para um valor razoável à aplicação, que pode ser feito no momento da análise ou limitando o tamanho total do buffer T_{max} . Vale ressaltar que o limite do valor de T implica diretamente que se o atraso ocorrer novamente, ele não poderá ser amenizado.

3.5 Elasticidade

Uma vez que o crescimento de T leva em conta a existência de atrasos, pode-se perceber que o preenchimento de T no buffer só será realizado quando a duração em buffer for 0. Uma vez que não ocorra atraso ou que ele seja amenizado pelo valor de T , a reprodução não irá parar até se obter a duração T em buffer. Sendo assim, se a duração disponível em buffer chegar a 0, isto implica diretamente que houve atraso e este atraso é no mínimo o valor de T . Dessa forma, o controle para o decaimento do valor de T não se faz necessário.

3.6 Abordagem por algoritmo

Abaixo encontra-se uma listagem para implementação desta solução em linguagem C.

```

1  ...
2  #define MIN(a, b) (((a) < (b)) ? (a) : (b))
3  #define MAX(a, b) (((a) > (b)) ? (a) : (b))
4
5  #define INTERVAL 1000
6  #define MAX_BUFFER_TIME 30.0 //30 Segundos de crescimento
7                                //de ponto de reprodução
8
9  enum TypePacket {TP_AUDIO, TP_VIDEO};
10
11 double audDuration = 0; //
12 double vidDuration = 0;
13 double startTime = 0;
14 double bufferTime = 0; // T
15
16 void onAddPacket(TypePacket type, double duration){ //duration =
    PTS * time_base
17     switch (type){
18         case TP_AUDIO:
19             audDuration += duration;
20             break;
21         case TP_VIDEO:
22             vidDuration += duration;
23             break;

```

```

24     }
25
26     double interval = (av_gettime()-startTime) / 1000000.0;
27     double delta = MIN(audDuration, vidDuration) - interval +
        bufferTime;
28
29     if (delta < 0){//Implica que há atraso
30         bufferTime = MIN(MAX_BUFFER_TIME, MAX(bufferTime , -delta
        ));
31     }
32
33     audDuration -= interval;
34     vidDuration -= interval;
35 }
36
37 void start(){
38     startTime = av_gettime(); // Referência para intervalo
39 }
40
41 ...

```

Nesta implementação, a função *start()* é chamada quando o processo de análise é iniciado. A cada pacote demultiplexado a função *onAddPacket()* é chamada, informando como parâmetro o tipo de dado que foi demultiplexado e a duração, que é calculada através do PTS multiplicado pelo tempo base.

3.7 Perda de pacotes

Como a determinação da duração do pacote é realizada pela diferença entre o pacote e seu sucessor, então uma perda de pacote não irá afetar o cálculo de forma a deixá-lo inconsistente, pelo fato de que a diferença do pacote e seu sucessor vai continuar existindo, mesmo não sendo o sucessor o pacote esperado como sucessor.

3.8 Solução e o ambiente IPTV

Para a realidade IPTV, foram definidos empiricamente valores para T iniciais, de tal forma que levem em consideração o serviço. Para TV Linear, $T = 0,500s$ foi um valor que apresentou bom rendimento, seja na alternância entre os canais ou na amenização do atraso. Para VoD, $T = 2s$ foi um valor que apresentou bom rendimento, haja visto que não afeta tanto o conteúdo e possibilita uma amenização maior ao atraso.

4 Implementação

Para a implementação, foi utilizado a Recomendação ITU-T H.721 como base, pois nela está descrito um padrão para terminais de serviço IPTV. Essa implementação foi realizada em GNU/Linux (Stallman, 2014), mais especificamente na distribuição Xubuntu em sua versão 14.04 (Canonical Ltd., 2014). Além disso, foi utilizada como linguagem de desenvolvimento a linguagem C++ (ISO/IEC 14882, 2003), pois o desenvolvimento Orientado à Objetos para esta solução possibilita um ganho na velocidade de desenvolvimento. Como bibliotecas, foram utilizadas as seguintes bibliotecas:

- **FFMpeg (Bellard; Niedermayer, 2014):** Responsável por fazer a conexão, recepção, demultiplexação e decodificação do conteúdo. Ela possui a maioria dos padrões MPEG implementados, o que agiliza o desenvolvimento de players que utilizam esses padrões.
- **SDL 2 (Lantinga, 2014):** Responsável pela saída audiovisual do projeto. Através dela é possível utilizar elementos que podem ser graficamente acelerados, pois ela pode fazer uso de recursos da placa de vídeo para isso. Além disso, ela oferece suporte a saída de áudio. Sendo assim, ela possibilita isentar o desenvolvedor a conhecer como fazer saída audiovisual específica de cada Sistema Operacional.

Para o VoD, a cada requisição que o usuário faz, o sistema envia um comando RTSP ao servidor. Sendo assim, quando PAUSE é enviado, a análise é pausada. Quando o PLAY é enviado, seja pelo fato de o usuário ter iniciado a reprodução ou feito um salto, os valores de análise são resetados e a análise é iniciada novamente, permanecendo apenas o valor de T já calculado. A permanência de T é justamente a inferência em que, se houve atrasos que geraram T na comunicação com o servidor, então estes atrasos podem voltar a ser presentes.

Para TV Linear, a cada mudança de canal a análise é reiniciada, zerando até o último valor de T obtido. Isto é feito pois a cada canal, tem-se servidores diferentes com transmissões diferentes, o que não implica que poderá haver os mesmos atrasos

anteriormente analisados.

4.1 TV Linear

O desenvolvimento do player de TV Linear, denominado Zapper, é baseado no padrão ITU-T H.750 (ITU-T H.750, 2008). Esse padrão especifica em alto-nível os padrões dos metadados de serviços IPTV, com seus respectivos elementos e protocolos de entrega. Neste player está implementado a decodificação e interpretação destes metadados, bem como funcionalidades de navegação entre canais como Próximo Canal e Canal Anterior, controle de volume, incluindo o estado mudo, uma interface que exibe informações ao usuário a respeito do canal atual, além de outras funcionalidades específicas do âmbito de TV Linear. Além disso, caso ocorra uma interrupção na comunicação entre o servidor e cliente, ele automaticamente, em intervalo de tempo pré-definido, tentará reconectar-se ao servidor.

4.2 Vídeo sobre Demanda

O desenvolvimento do player de Vídeo sobre Demanda, denominado VoDPlayer, é também baseado no padrão H.750. Neste player está implementado a decodificação destes metadados, bem como funcionalidades de controle de reprodução, como Play e Pause. Além disso, é possível realizar salto temporal por porcentagem ou por um valor pré-definido de segundos, por exemplo 10 segundos à frente, e realizar operações que aceleram a reprodução de 1 até 32 vezes, sendo que esta aceleração pode ser tanto no sentido progressivo do conteúdo como pode ser no sentido contrário. Está implementado neste player também uma interface que exibe ao usuário informações sobre o conteúdo, como nome, autor e outras informações específicas do conteúdo em reprodução.

5 Resultados obtidos

Para os testes realizados, foram utilizados vídeos na resolução 1920x1080, com a codificação do vídeo sendo H.264 e áudio AAC. Além disso, foi utilizado o MPEG-2 TS para a multiplexação dos conteúdos. Todos os vídeos testados possuíam uma incidência de frame I (frame com imagem completa (ISO/IEC 11172-2, 1993)) a cada 10 frames.

Nos gráficos aqui representados, as linhas pontilhadas representam o valor de T no instante de tempo de análise em segundos. As linhas contínuas representam o valor de duração contido no buffer neste mesmo instante de análise. Para a geração destes gráficos, foram colhidas amostras a cada segundo num intervalo de 20 minutos de reprodução (1200 segundos).

5.1 TV Linear

Para a realização destes testes, foi utilizado o VLC media player (VideoLAN, 2014) para a realização do stream, simulando a recepção de uma TV Linear. Para ambos os vídeos usados para a realização do teste, a taxa de FPS é 30.

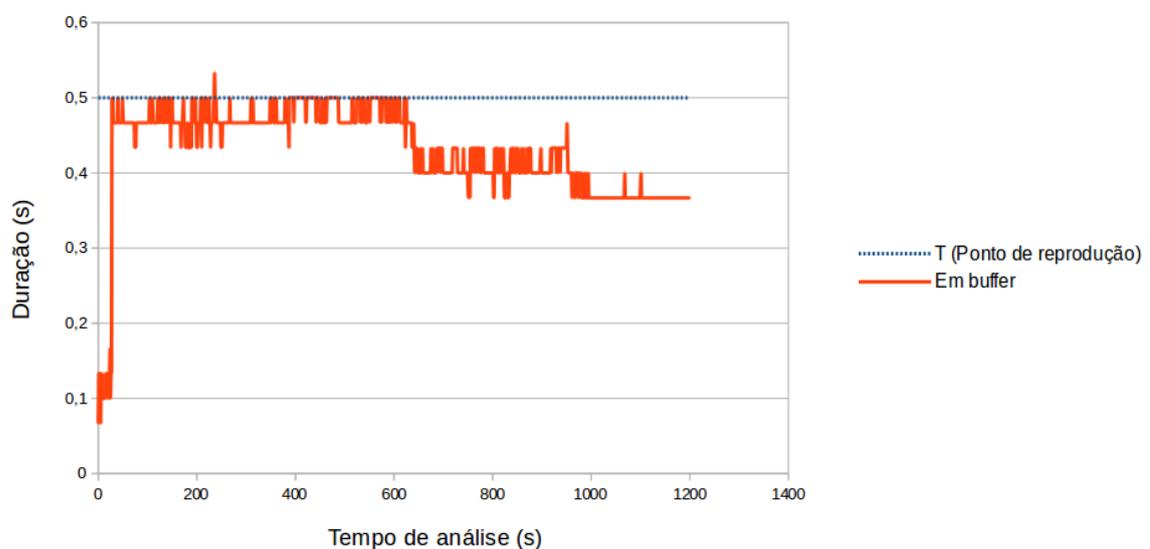


Figura 5.1: Comparação entre a duração contida no buffer e o Ponto de reprodução T para a transmissão 1

Na figura 5.1, pode-se notar o comportamento do buffer na reprodução da transmissão 1. Para este exemplo, não foram simulados atrasos. Conforme o gráfico apresentado, houve atrasos durante toda a transmissão, porém em todos eles o valor de T foi suficiente para amenizá-los.

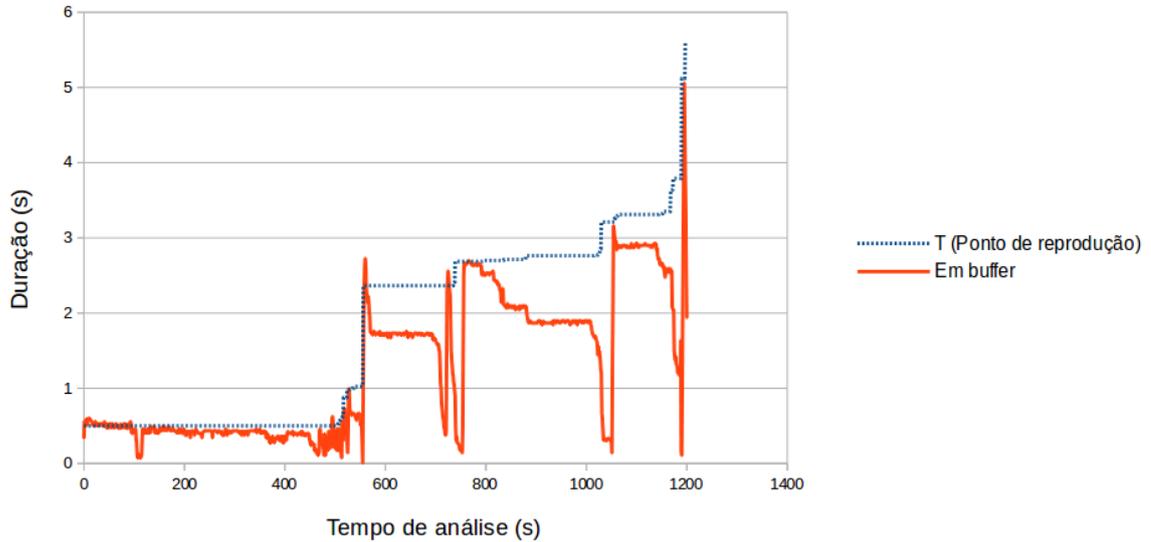


Figura 5.2: Comparação entre a duração contida no buffer e o Ponto de reprodução T para a transmissão 2

Na figura 5.2, pode-se notar o comportamento do buffer na reprodução da transmissão 2. Para este exemplo, foram simulados atrasos. Esses atrasos foram gerados da seguinte forma: a cada 200 pacotes RTP, um atraso de 50 milisegundos é inserido. Conforme o gráfico apresentado, pode-se notar a presença desses atrasos. Como ocorre de 200 em 200 pacotes, esse atraso é latente. Sendo assim, o tamanho de T tende a aumentar, ao passo que este tamanho não consiga mais amenizá-lo.

5.2 VoD

Para a realização destes testes, foi utilizado o live555 Media Server (Live, 2014) como sendo o servidor de streaming VoD. Foi realizada a análise com a reprodução e recepção de 2 vídeos, ambos com taxa de FPS de 23,98.

Na figura 5.3, pode-se observar o comportamento do buffer na reprodução do vídeo 1. Conforme se pode ver no gráfico, em tempo de análise entre 0 e 200 segundos, momento inicial da reprodução, houve um esvaziamento de buffer maior do que a recepção

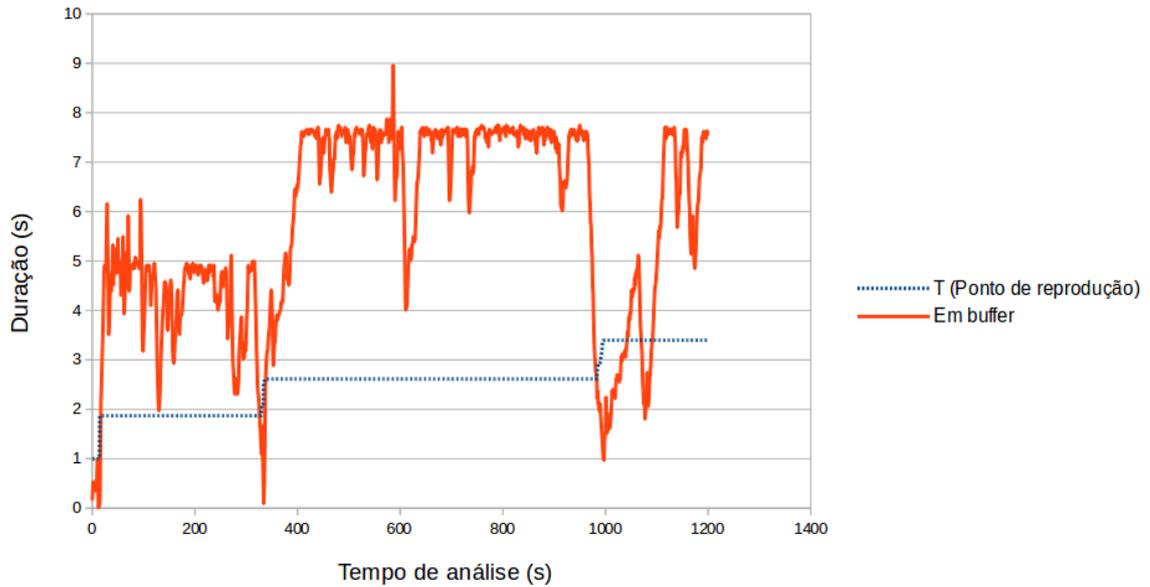


Figura 5.3: Comparação entre a duração contida no buffer e o Ponto de reprodução T para o Vídeo 1

de dados. Neste caso em específico, o valor inicial proposto para T não foi suficiente para amenizar o atraso. Sendo assim, a solução proposta implementada alterou o valor de T para um valor que cobrisse o atraso. No tempo de análise entre 200 e 400 segundos, houve outro atraso. Pode-se observar quão rápido o buffer foi esvaziado com a ocorrência deste atraso. Com isto, a solução proposta implementada alterou o valor de T para cobrir este atraso. Em tempo de análise entre 1000 e 1200 segundos, pode-se notar a ocorrência de 2 momentos críticos. No primeiro momento o tamanho de T não foi o suficiente para amenizar os atrasos ocorridos, então este valor foi alterado de forma a cobrir estes atrasos. No segundo momento neste mesmo intervalo, o tamanho de T era o suficiente para amenizar os atrasos ocorridos, sendo assim não foi feita alteração no valor de T .

A figura 5.4 representa o gráfico do comportamento do buffer para o vídeo 2. Entre o intervalo de tempo de análise 0 e 200 segundos pode-se notar sucessivos atrasos. Nos 100 primeiros segundos, aproximadamente, pode-se notar que o valor de T inicial não foi alterado, pois era o suficiente para amenizar estes atrasos ocorridos. Após este tempo, neste mesmo intervalo, observa-se um conjunto de atrasos críticos que T não era suficiente para amenizá-los. Sendo assim, T foi alterado de forma a se cobrir estes atrasos. Entre os intervalos de tempo de análise 400 e 600 segundos, observa-se um conjunto de atrasos críticos que levaram T a ter um aumento sucessivo. Após este intervalo, pode-se notar

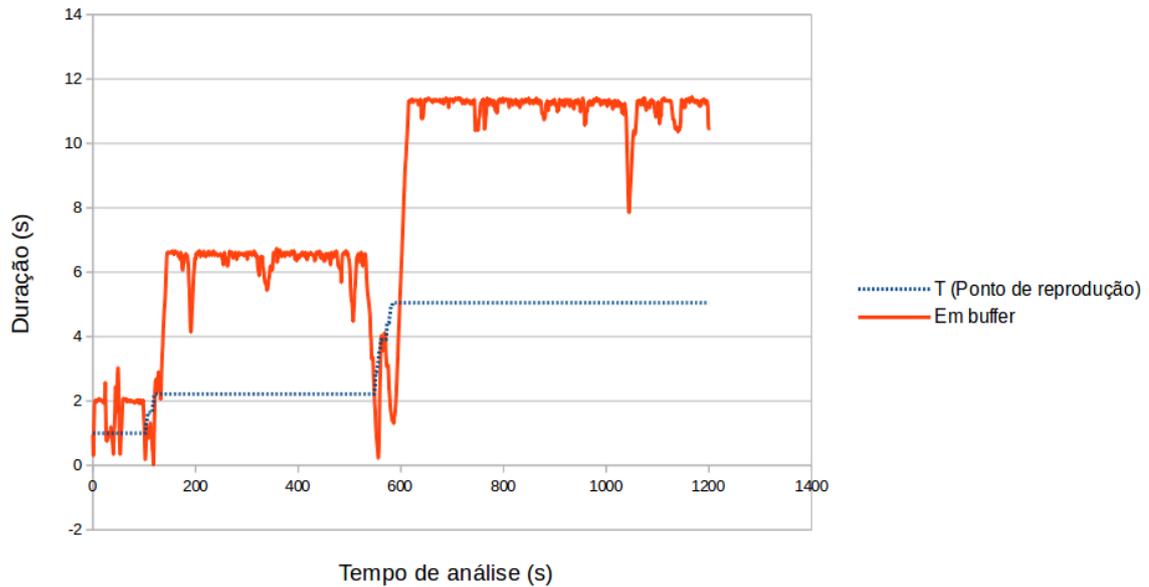


Figura 5.4: Comparação entre a duração contida no buffer e o Ponto de reprodução T para o Vídeo 2

que houve atrasos, porém em todos eles o tamanho de T era o suficiente.

Em ambos os gráficos, se uma operação de pause for enviada ao servidor, no intervalo que a recepção de conteúdo permanecer pausado, tanto o valor de T quanto a quantidade em buffer permanecerão com o seus valores inalterados. Sendo assim, a representação gráfica neste intervalo para ambos será a de uma função constante.

5.2.1 Armazenamento físico

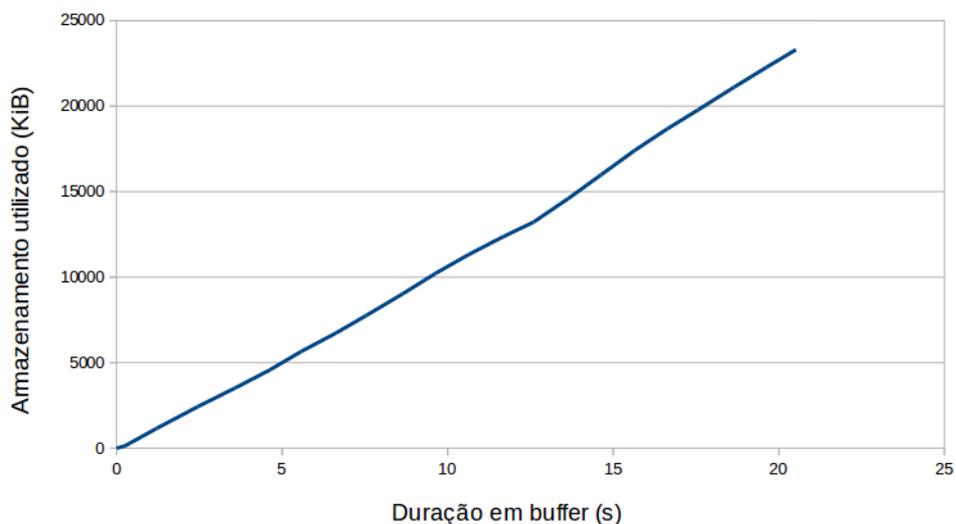


Figura 5.5: Espaço utilizado para cada duração em buffer

A figura 5.5 representa como está relacionado o espaço físico utilizado em KiB para armazenar o conteúdo recebido e a duração deste conteúdo. Como o conteúdo contém áudio e vídeo, o armazenamento utilizado é a soma dos espaços utilizados pelos elementos de mídia.

6 Considerações Finais

A percepção do usuário com relação ao uso do serviço IPTV é de grande valia. Sendo assim, tanto para serviços de TV Linear quanto para os serviços de Vídeo sobre Demanda, a qualidade de experiência (QoE) é um desafio na provisão destes serviços.

Desta maneira, o presente trabalho busca propor uma solução a gerência de buffer para players de vídeo IPTV, tanto para serviços de vídeo sobre demanda (VoD) quanto para serviços de TV Linear, melhorando a qualidade de experiência. Como esta gerência ocorre de forma automatizada e periódica, o valor obtido é conciso com relação aos atrasos que ocorrerem.

Espera-se com este trabalho que a percepção dos usuários destes serviços com relação ao atraso seja reduzida, melhorando assim a qualidade de experiência dos serviços IPTV.

Uma proposta de trabalho futuro é a análise do comportamento da solução proposta para os outros formatos estabelecido no padrão ITU-T (ITU-T H.721, 2009), como por exemplo outros formatos de codificação de áudio e vídeo, bem como o comportamento da biblioteca FFmpeg a estes formatos. Outra proposta é a análise da solução proposta para dispositivos móveis como celular.

Referências Bibliográficas

- Barbosa, S. D. J.; Soares, L. F. G. **Fundamentos, Padrões, Autoria Declarativa e Usabilidade**. Rio de Janeiro, RJ: PUC–Rio, 2008, 105–174p.
- Batista, C. E. C. F.; Moreno, M. F. ; Soares, L. F. G. **Recuperação e tolerância a falhas em apresentações de aplicações hipermídia distribuídas**. In: XVII Simpósio Brasileiro de Sistemas Multimídia e Web, 2011.
- Begazo, D. C. **Avaliação objetiva e subjetiva de qualidade de vídeo via rede ip com variação de atraso**. 2012. Dissertação de Mestrado - Escola Politécnica da Universidade de São Paulo.
- Bellard, F.; Niedermayer, M. **Ffmpeg site**. <https://www.ffmpeg.org/>, 2014. Acesso em 05 de Novembro de 2014.
- Bhadada, R.; Sharma, K. L. Evaluation and analysis of buffer requirements for streamed video data in video on demand applications. **IETE Journal of Research**, v.56, sep 2010.
- Calado, I. A. A. R. **Infraestrutura para transmissão de conteúdo multimídia com suporte à adaptação de fluxos**. 2010. Dissertação de Mestrado - Universidade Federal de Campina Grande.
- Canonical. **Xubuntu**. <http://xubuntu.org/>, 2014. Acesso em 11 de Novembro de 2014.
- Cavendish, S. A. **Algoritmo de ajuste elástico para vídeo em fluxos mpeg-2**. 2005. Dissertação de Mestrado - Pontifícia Universidade Católica do Rio de Janeiro.
- Duque, L. H. **Avaliação da qualidade de vídeo em redes iptv com acesso baseado em adsl**. 2008. Dissertação de Mestrado - Universidade de Brasília.
- IETF. **User datagram protocol**. RFC 768 (INTERNET STANDARD), aug 1980.
- IETF. **Transmission control protocol**. RFC 793 (INTERNET STANDARD), sep 1981. Updated by RFCs 1122, 3168, 6093, 6528.
- IETF. **Real time streaming protocol (rtsp)**. RFC 2326 (Proposed Standard), apr 1998.
- IETF. **Rtp: A transport protocol for real-time applications**. RFC 3550 (INTERNET STANDARD), jul 2003. Updated by RFCs 5506, 5761, 6051, 6222, 7022, 7160, 7164.
- IETF. **A proposed media delivery index (mdi)**. RFC 4445 (Informational), apr 2006.
- ISO/IEC. **Information technology - coding of moving pictures and associated audio for digital storage media at up to about 1,5 mbit/s - part 2: Video**. Standard 11172-2, International Organization for Standardization/International Electrotechnical Commission, 1993.

- ISO/IEC. **Information technology - generic coding of moving pictures and associated audio information: Systems**. Standard 13818-1, International Organization for Standardization/International Electrotechnical Commission, Geneva, 2000.
- ISO/IEC. **Programming languages - c++**. Standard 14882, International Organization for Standardization/International Electrotechnical Commission, USA, 2003.
- ITU-T. **Iptv functional architecture**. Recommendation Y.1910, International Telecommunication Union, Geneva, sep 2008.
- ITU-T. **High-level specification of metadata for iptv services**. Recommendation H.721, International Telecommunication Union, Geneva, oct 2008.
- ITU-T. **Requirements for the support of iptv services**. Recommendation Y.1901, International Telecommunication Union, Geneva, jan 2009.
- ITU-T. **Iptv terminal devices: Basic model**. Recommendation H.721, International Telecommunication Union, Geneva, mar 2009.
- Inamori, M.; Naganuma, J. ; Endo, M. **A memory-based architecture for mpeg2 system protocol lsi's**. In: IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, volume 7, p. 339–344. IEEE, sep 1999.
- Lantinga, S. **Simple directmedia layer**. <https://www.libsd1.org/>, 2014. Acesso em 05 de Novembro de 2014.
- Live. **Live555 media server**. <http://www.live555.com/mediaServer/>, 2014. Acesso em 11 de Novembro de 2014.
- Medeiros, L. **Módulo ip de um demultiplexador para o subsistema fluxo de transporte - mpeg-2 ts - aderente ao sistema brasileiro de tv digital**. 2010. Dissertação de Mestrado - Universidade Federal da Paraíba.
- Mokarzel, M. P. **Redução do tempo de zapping em serviços iptv sobre redes gpon utilizando vídeos escaláveis**. 2010. Dissertação de Mestrado - Universidade de São Paulo.
- Muezerie, A. **Tráfego multimídia com qos em redes deflexivas e redes híbridas: simulação e implementação**. 2005. Dissertação de Mestrado - Universidade de São Paulo.
- Soares, L. F. G.; Moreno, M. F.; Costa, R. M. R. ; Moreno, M. F. **Towards the convergence of digital tv systems**. In: Journal of Internet Services and Applications, p. 69–79, Rio de Janeiro, 2010. Springer-Verlag.
- Stallman, R. **Free software foundation site**. <https://www.gnu.org/gnu/linux-and-gnu.en.html>, 2014. Acesso em 11 de Novembro de 2014.
- Tanenbaum, A. S.; Wetherall, D. **Redes de computadores**. 5. ed., Pearson Education do Brasil, 2011.
- VideoLAN. **Vlc media player**. <http://www.videolan.org/vlc/>, 2014. Acesso em 11 de Novembro de 2014.
- Zapater, M. N. **Análise técnico-econômica da introdução de serviços iptv por operadoras de telecomunicações**. 2007. Dissertação de Mestrado - Escola Politécnica da Universidade de São Paulo.