UNIVERSIDADE PRESBITERIANA MACKENZIE
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA E COMPUTAÇÃO

LUCIANO NEVES DOS SANTOS JÚNIOR

AVALIAÇÃO DE TOLERÂNCIA À LATÊNCIA NA TRANSMISSÃO E RECEPÇÃO DE SERVIÇOS MPEG-H AUDIO *MULTI-STREAM* VIA *BROADBAND* E *BROADCAST*

São Paulo
2024

Luciano Neves dos Santos Júnior

Avaliação da tolerância à latência na transmissão e recepção de serviços MPEG-H Audio *multi-stream* via *broadband* e *broadcast*

Projeto de Pesquisa apresentado ao Programa de Pós-Graduação em Engenharia Elétrica e Computação da Universidade Presbiteriana Mackenzie como Requisito para Obtenção do Título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Cristiano Akamine

São Paulo
2024

S194a Santos Junior, Luciano Neves dos
Avaliação da tolerância à latência na transmissão e recepção de serviços MPEG-H Audio multi-stream via Broadband e Broadcast / Luciano Neves dos Santos Junior

4.444 KB

Dissertação (Mestrado em Engenharia Elétrica e Computação) – Universidade Presbiteriana Mackenzie, São Paulo, 2024.

Orientador: Prof. Dr. Cristiano Akamine
Bibliografia: f. 67-71
1. MPEG-H Áudio 2.TV 3.0 3. Broadcast 4.Broadband, Multi-stream. I Akamine, Cristiano, orientador II. Título

CDD 621.3

Bibliotecária responsável: Maria Gabriela Brandi Teixeira – CRB 8 / 6339

Folha de Identificação da Agência de Financiamento

Autor: Luciano Neves dos Santos Junior

Programa de Pós-Graduação *Stricto Sensu* em Engenharia Elétrica e Computação

Título do Trabalho: Avaliação da tolerância à latência na transmissão e recepção de serviços MPEG-H Audio multi-stream via broadband e broadcast

O presente trabalho foi realizado com o apoio de [1]:

☐ CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior

☐ CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico

☐ FAPESP - Fundação de Amparo à Pesquisa do Estado de São Paulo

☒ Instituto Presbiteriano Mackenzie/Isenção integral de Mensalidades e Taxas

☐ MACKPESQUISA - Fundo Mackenzie de Pesquisa

☐ Empresa/Indústria:

☒ Outro: Rede Nacional de Ensino e Pesquisa (RNP)

[1] **Observação**: caso tenha usufruído mais de um apoio ou benefício, selecione-os.

LUCIANO NEVES DOS SANTOS JÚNIOR

AVALIAÇÃO DE TOLERÂNCIA À LATÊNCIA NA TRANSMISSÃO E RECEPÇÃO DE SERVIÇOS MPEG-H AUDIO MULTI-STREAM VIA BROADBAND E BROADCAST

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Computação da Universidade Presbiteriana Mackenzie, como requisito parcial à obtenção de título de Mestre em Engenharia Elétrica e Computação.

Aprovada em 16 de agosto de 2024.

BANCA EXAMINADORA

Prof. Dr. Cristiano Akamine
Universidade Presbiteriana Mackenzie

Prof. Dr. Gustavo de Melo Valeira
Universidade Presbiteriana Mackenzie

Prof. Dr. Marcelo Ferreira Moreno
Universidade Federal de Juiz de Fora

AGRADECIMENTOS

Agradeço ao Instituto Presbiteriano Mackenzie pela concessão da minha bolsa de estudos.

Agradeço ao meu orientador, Professor Doutor Cristiano Akamine, pelos ensinamentos, apoio, confiança e paciência dedicados a mim durante esse período.

Agradeço ao Prof. Dr. Paulo Batista Lopes, pela orientação no primeiro semestre do curso e ensinamentos.

Ao Laboratório de TV Digital da Escola de Engenharia da Universidade Presbiteriana Mackenzie por ter cedido o espaço e os equipamentos para o desenvolvimento deste projeto.

Agradeço ao Instituto de Circuitos Integrados Fraunhofer, em especial aos engenheiros Adrian Murtaza e Stefan Meltzer, pelo fornecimento de arquivos de teste e por toda assistência na elaboração do projeto.

Agradeço aos meus colegas do Laboratório de TV Digital da Escola de Engenharia, Allan Seiti Sassaqui Chaubet, Cesar Augusto Diez, Fadi Jerji, George Henrique Maranhão Garcia de Oliveira e Ricardo Seriacopi Rabaça, pelas contribuições e pelo apoio neste projeto.

Agradeço também a meus familiares e amigos, pela força e incentivo durante o período.

RESUMO

A adoção do *Moving Picture Experts Group - High Efficiency Coding Part 3: 3D Audio* (MPEG-H *Audio*) como *codec* oficial para a TV 3.0 trouxe novas possibilidades a serem exploradas. Seus recursos inovadores trazem muitas opções para a nova geração de TV como a possibilidade de reprodução de três formas diferentes (baseada em canais, objetos e ambiência), personalização, interatividade e integração da transmissão de conteúdo via ar, também conhecido como *broadcast* com a transmissão de conteúdo via internet, chamada de *broadband*. Essas melhorias proporcionam uma maior flexibilidade e controle à emissora e uma melhor experiência auditiva para o usuário final. A partir disso, são necessários estudos de suas funcionalidades para o seu uso de maneira inteligente para aplicações especificas no mercado brasileiro. Tendo isso em vista, este projeto tem por objetivo realizar testes de tolerância à latência de serviços MPEG-H *Audio* com a transmissão sendo realizada via *broadcast* e *broadband*. Esse teste é possível graças à função de entrega *multi-stream*. Este recurso facilita a recepção e seleção de streams transmitidas, combinando os fluxos de acordo com a escolha do usuário. Os resultados mostraram uma eficiência da funcionalidade de entrega *multi-stream*, sendo possível receber e decodificar o conteúdo transmitido sem erros em até 0,8 segundos após a recepção e leitura do arquivo manifesto.

Palavras-chave: *MPEG-H Audio, TV 3.0, Broadcast, Broadband, Multi-stream.*

ABSTRACT

The adoption of *Moving Picture Experts Group - High Efficiency Coding Part 3: 3D Audio* (MPEG-H *Audio*) as the official codec for TV 3.0 has brought new possibilities to be explored. Its innovative features bring many options to the new generation of TV, such as the possibility of playback in three different ways (based on channels, objects and ambience), personalization, interactivity and the integration of over-the-air content transmission, also known as broadcast, with internet content transmission, known as broadband. These improvements provide greater flexibility and control for the broadcaster and a better listening experience for the end user. As a result, it is necessary to study its functionalities so that it can be used intelligently for specific applications in the Brazilian market. With this in mind, the aim of this project is to carry out latency tolerance tests on MPEG-H *Audio*services transmitted via broadcast and broadband. This test is possible thanks to the multi-stream delivery function. This feature facilitates the reception and selection of transmitted streams, combining the streams according to the user's choice. The results showed that the multi-stream delivery feature was efficient, making it possible to receive and decode the transmitted content without errors within 0.8 seconds of receiving and reading the manifest file.

Palavras-chave: *MPEG-H Audio, TV 3.0, Broadcast, Broadband, Multi-stream.*

LISTA DE ABREVIATURAS E SIGLAS

AAC	*Moving Picture Experts Group - Advanced Audio Coding*
AES	*Audio Engineering Society*
ALP	*ATSC link-layer protocol*
ATSC 1.0	*Advanced Television Systems Committee*
ATSC 3.0	*Advanced Television Systems Committee 3.0*
AU	*Access Unit*
BAT	*Bouquet Association Table*
BIT	*Broadcaster Information Table*
BRIR	*Binaural Room Impulse Responses*
CAT	*Conditional Access Table*
CDN	*Content Delivery Network*
CfP	*Call for Proposals*
DAB	*Digital Audio Broadcasting*
DASH	*Dynamic Adaptive Streaming over Hypertext Transfer Protocol(HTTP)*
DINF	*Data Information Box*
DNS	*Domain Name System*
DTMB	*Digital Terrestrial Multimedia Broadcast*
DVB	*Digital Video Broadcast*
DVB-T/T2	*Digital Video Broadcast - Terrestrial*
EIT	*Event Information Table*
ETRI	*Electronics and Telecommunications Research Institute*
FEC	*Forward Error Correction*
FIFO	*First In First Out*
FLAC	*Free Lossless Audio Codec*
FLUTE	*File Delivery over Unidirectional Transport*
Fraunhofer IIS	*Fraunhofer Institute for Integrated Circuits*
FTYP	*File Type Box*
HDLR	*Handler Reference Box*
HDR	*High Dynamic Range*
HEVC	*High-Efficiency Video Coding*
HOA	*Higher Order Ambisonics*
HTTP	*Hypertext Transfer Protocol*
IBB	*Integração Broadcast-Broadband*
IEEE	*Institute of Eletrical and Electronics Enginners*
IP	*Internet Protocol*
IPF	*Immediate Playout Frame*

ISDB-T	*Integrated Services Digital Broadcasting - Terrestrial*
ISOBMFF	*International Organization for Standardization Base Media File Format*
ITU-R	*International Telecommunication Union Radiocommunication Sector*
LCT	*Layered Coding Transport*
LDT	*Local Description Table*
LFE	*Low Frequency Effects*
MAE	*Metadata Audio Elements*
MDAT	*Media Data Box*
MDIA	*Media Box*
MHAS	*MPEG-H Audio Stream*
MINF	*Media Information Box*
MOOF	*Movie Fragment Box*
MOOV	*Movie Box*
MP3	*MPEG Audio Coding Layer 3*
MP4	*MPEG-4 Part 14*
MPD	*Media Presentation Description*
MPEG	*Moving Picture Experts Group*
MPEG-2 TS	*MPEG-2 - Parte 1 - Transport Stream*
MPEG-H *Audio*	*Moving Picture Experts Group - High Efficiency Coding Part 3: 3D Audio*
MVEX	*Movie Extends Box*
NBIT	*Network Board Information Table*
NGA	*Next Generation Audio*
NHK	*Nippon Hoso Kyokai*
NIT	*Network Information Table*
OSI	*Open Systems Interconnection*
OTA	*Over-The-Air*
OTT	*Over-The-Top*
PAT	*Program Association Table*
PCAT	*Packetized Elementary Stream Content Advisory Table*
PCR	*Program Clock Reference*
PID	*Packet Identifier*
PMT	*Program Mapping Table*
PSI	*Program Specific Information*
RAP	*Random Access Point*
ROUTE	*Real-time Object delivery over Unidirectional Transport*
RST	*Running Status Table*
RTP	*Real-time Transport Protocol*
RTSP	*Real Time Streaming Protocol*
S-TSID	*Service Transport Session Identification Description*
SBTVD-T	Sistema Brasileiro de TV Digital Terrestre

SDT	*Service Description Table*
Service ID	*Service Identification*
SI	*Service Information*
SLS	*Service Layer Signaling*
SLT	*Service List Table*
ST	*Stuffing Table*
STBL	*Sample Table Box*
STSD	*Sample Description Box*
STSZ	*Sample Size Box*
STTS	*Time-to-Sample Box*
STYP	*Segment Type Box*
TCP	*Transmission Control Protocol*
TDT	*Time and Data Table*
TFDT	*Track Fragment Decode Time Box*
TOT	*Time Offset Table*
TRAF	*Track Fragment Box*
TRAK	*Track Box*
TRUN	*Track Fragment Run Box*
TS	*Transport Stream*
TTA	*Telecommunications Technology Association*
UDP	*User Datagram Protocol*
URL	*Uniform Resource Locator*
USBD	*User Service Bundle Description*
USD	*User Service Description*
UTC	*Coordinated Universal Time*
VBAP	*Vector Base Amplitude Panning*
WAV	*Waveform Audio File Format*
WebDAV	*Web-based Distributed Authoring and Versioning*
XML	*eXtensible Markup Language*

LISTA DE FIGURAS

Figura 1 Exemplo do cabeçalho de um pacote *Transport Stream* (TS). 7

Figura 2 Sequência de apresentação dos campos de seções normais e estendidas de uma tabela de um TS. 9

Figura 3 Exemplo da tabela *Program Association Table* (PAT). 11

Figura 4 Exemplo da tabela *Program Mapping Table* (PMT). 12

Figura 5 Exemplo da tabela *Conditional Access Table* (CAT). 12

Figura 6 Pilha de Protocolo de comunicação do *Real-time Object delivery over Unidirectional Transport* (ROUTE)/*Dynamic Adaptive Streaming over Hypertext Transfer Protocol(HTTP)* (DASH) para a camada de transporte da TV 3.0. 16

Figura 7 Sessão de aquisição de serviços ROUTE. 17

Figura 8 Estrutura de comunicação do DASH. 19

Figura 9 Estrutura de um arquivo manifesto. 21

Figura 10 Exemplo de um arquivo *Media Presentation Description* (MPD). . 22

Figura 11 Exemplo de um arquivo de inicialização DASH no formato *International Organization for Standardization Base Media File Format* (ISOBMFF). 24

Figura 12 Exemplo de um segmento DASH no formato ISOBMFF. 25

Figura 13 Sistema de comunicação com codificação (a) e sem codificação (b). 26

Figura 14 Dados original (a), dados após a compressão sem perdas (b) e dados após a compressão com perdas (c). 28

Figura 15 Estrutura de um *MPEG-H Audio Stream* (MHAS). 40

Figura 16 Extração de informações do *PACTYP_AUDIOSCENEINFO* com o MPEG-H Decoder - Fraunhofer IIS (2023). 42

Figura 17 Extração de informações do *PACTYP_AUDIOSCENEINFO* com o MPEG-H Decoder - Fraunhofer IIS (2023). 43

Figura 18 Diferença de entrutura entre arquivos *MPEG-4 Part 14* (MP4), DASH com single stream e DASH com *Moving Picture Experts Group - High Efficiency Coding Part 3: 3D Audio* (MPEG-H *Audio*) multi-stream. 44

Figura 19 Exemplo de declaração de *multi-stream* no arquivo manifesto. . . . 45

Figura 20 Exemplo de codificação DASH com *multi-stream*. 45

Figura 21 Sistema de junção de MHAS na função *multi-stream*. 46

Figura 22 Exemplo de comutação de configurações de áudio. 48

Figura 23 Configuração interna do emulador de latência em rede. 49

Figura 24 Histograma comparando *ping* ao servidor DNS do Google e o perfil gaussiano do emulador. 50
Figura 25 Primeiro Cenário de comunicação representando a transmissão *broadcast*. 52
Figura 26 Segundo Cenário de comunicação representando a transmissão *broadband*. 52
Figura 27 Diagrama de comunicação entre Cenário 1. 54
Figura 28 Analise do fluxo de rede da comunicação do Cenário 1. 55
Figura 29 Diagrama de comunicação do Cenário 2. 56
Figura 30 Análise do fluxo de rede da comunicação do Cenário 2. 57
Figura 31 Cabeçalho do arquivo manifesto após as modificações. 58
Figura 32 Configuração utilizada para realização dos testes de tolerância. . . . 59
Figura 33 Exemplo de execução do *player*. 60

Sumário

1 INTRODUÇÃO **1**

1.1 Objetivos 2

1.1.1 Objetivo Geral 2

1.1.2 Objetivos Específicos 3

1.2 Justificativa 3

1.3 Metodologia 4

1.4 Estrutura e Organização do Trabalho 4

2 EVOLUÇÃO DA CAMADA DE TRANSPORTE NA TELEVISÃO DIGITAL BRASILEIRA **6**

2.1 MPEG-2 *Transport Stream* 6

2.2 ROUTE/DASH 15

2.2.1 ROUTE 16

2.2.2 DASH 18

3 ELEMENTOS DA CODIFICAÇÃO DE ÁUDIO NA ERA DIGITAL **26**

3.1 Conceitos e técnicas de compressão de áudio 26

3.2 Visão geral dos codecs de áudio 29

3.2.1 MPEG Audio Coding 29

3.2.2 MPEG Advanced Audio Coding (AAC) 31

3.2.3 Next Generation Áudio (NGA) 32

4 MPEG-H AUDIO **33**

4.1 Áudio Imersivo 34

4.1.1 Baseado em Canais 34

4.1.2 Baseado em Objetos 36

4.1.3 Baseado em Ambiência 37

4.1.4 Binaural 38

4.2 MPEG-H Audio Stream (MHAS) 39

4.3 Metadados 40

4.4 Multi-stream 43

4.5 Mecanismos de alinhamento para Pontos de Acesso 46

4.6 Perfis de Complexidade 48

5 TESTES E RESULTADOS **49**

5.1 Emulador de Latência em Rede 49

5.2 Comunicação em rede 51

5.3 Modificação do Arquivo Manifesto 57

5.4 Configuração de Testes 58

5.5 Resultados dos Testes 60

6 CONCLUSÃO **65**

6.1 Trabalhos Futuros 66

6.2 Artigos Publicados 66

REFERÊNCIAS BIBLIOGRÁFICAS **71**

1 INTRODUÇÃO

A chegada gradual da internet nas casas mundo a fora transformou a forma como o entretenimento é consumido pelo usuário final. O conteúdo passou a ser mais abrangente, tendo em vista que sua produção foi facilitada. Além disso, o tempo tornou-se um recurso ainda mais valioso nessa era digital. O acesso a uma vasta gama de conteúdos se tornou instantâneo, permitindo aos usuários escolher o que assistir, ler ou ouvir a qualquer momento, sem as limitações impostas pela programação da TV.

Para incrementar o sistema de TV brasileiro atual, foi realizado o projeto intermediário chamado "TV 2.5". Este projeto teve a finalidade de testar várias tecnologias de áudio e vídeo, a fim de facilitar a transição do sistema de TV digital brasileiro para a próxima geração. Uma das vertentes desse projeto incluiu a exploração de novas opções de *codecs* com suporte ao áudio imersivo, como o *Moving Picture Experts Group - High Efficiency Coding Part 3: 3D Audio (MPEG-H Audio)*, ao mesmo tempo em que se mantinha o *Moving Picture Experts Group - Advanced Audio Coding (AAC)* como formato de áudio principal do sistema.

A partir disso, o governo brasileiro, com o apoio do Fórum do Sistema Brasileiro de TV Digital Terrestre (SBTVD-T), busca implementar um novo sistema de televisão digital terrestre no Brasil, denominado "TV 3.0". O projeto teve início após a divulgação de uma Chamada de Propostas, do inglês *Call for Proposals (CfP)*, em julho de 2020, com o intuito de atrair novas tecnologias capazes de atender aos pré-requisitos estabelecidos pelo Fórum SBTVD-T e aceitos pelo governo (FORUM SBTVD, 2020). Como principais pré-requisitos, podem ser citados o melhor uso da faixa espectral, a *Integração Broadcast-Broadband* (IBB) e a melhoria na qualidade audiovisual.

Essa iniciativa representa um passo importante na modernização da televisão no Brasil, visando aproveitar as vantagens da era digital para oferecer uma experiência de entretenimento mais rica e personalizada aos telespectadores, alinhada com as transformações no consumo de conteúdo e entretenimento impulsionadas pela internet.

Nesse processo de modernização da transmissão no Brasil, os estudos na área de áudio desempenharam um papel fundamental na definição do novo sistema. Após uma avaliação minuciosa das opções disponíveis, o MPEG-H *Audio* foi escolhido como o *codec* oficial da

TV 3.0. Essa escolha reflete o compromisso em proporcionar uma experiência de áudio imersivo e de alta qualidade aos espectadores, alinhada com as tendências globais de entretenimento e as expectativas do público.

O MPEG-H *Audio* é uma tecnologia que oferece recursos avançados de áudio, como, por exemplo, som tridimensional, interatividade e a capacidade de personalizar a mixagem de áudio de acordo com as preferências individuais do telespectador (GREWE; MURTAZA; MELTZER, 2023). Isso significa que os espectadores poderão desfrutar de uma experiência de áudio imersivo, como se estivessem no centro da ação, além de ter o poder de ajustar o áudio de acordo com suas preferências pessoais, como realce de diálogos, efeitos sonoros ou música de fundo.

A escolha do MPEG-H *Audio* como *codec* oficial do novo sistema TV 3.0, divulgada em dezembro de 2021, é um marco importante, não apenas para o Brasil, mas também para a indústria de entretenimento em todo o mundo (FORUM SBTVD, 2021a). Isso coloca o país na vanguarda da tecnologia audiovisual, proporcionando aos espectadores uma experiência de áudio de alta qualidade que complementa o avanço das capacidades visuais oferecidas pelo sistema.

Essa abordagem inovadora e focada na qualidade de áudio reflete o compromisso do Brasil em proporcionar uma experiência de entretenimento de última geração para os cidadãos, atendendo às demandas da era digital, levando o áudio com qualidade de cinema aos lares brasileiros, de forma acessível e personalizável.

1.1 Objetivos

Esta seção indica o objetivo geral do trabalho, assim como seus objetivos específicos.

1.1.1 Objetivo Geral

Realizar um estudo sobre a tolerância de latência entre serviços MPEG-H *Audio* sendo transmitidos pelo ar, do inglês *Over-The-Air (OTA)*, e via internet, do inglês *Over-The-Top (OTT)*, e o uso das ferramentas do MPEG-H *Audio* como *multi-stream* para realizar a comutação e alinhamento do sincronismo.

1.1.2 Objetivos Específicos

- Estudar as funcionalidades do MPEG-H *Audio*, dando ênfase na entrega híbrida com *multi-stream*;

- Medir a tolerância à latência entre sinais OTA e OTT por meio do recurso *multi-stream*.

1.2 Justificativa

A escolha do MPEG-H *Audio* como o *codec* de áudio principal para TV 3.0 representa um avanço significativo para a tecnologia no país. O *codec* apresenta funcionalidades que, por meio da utilização de metadados, aprimoram a experiencia de áudio do usuário final.

O MPEG-H *Audio* possui a personalização como elemento fundamental, permitindo maior liberdade nos ajustes dos elementos de áudio. Essa capacidade de personalização abre portas para uma abordagem mais centrada no usuário final, em ambientes de transmissão de conteúdo, podendo ser personalizada de acordo com suas preferências e necessidades.

Essa adoção traz a necessidade de diversos estudos em relação à tecnologia em questão. A implementação dos pacotes do MPEG-H *Audio* na estrutura *MPEG-2 - Parte 1 - Transport Stream (MPEG-2 TS)*, atual tecnologia da camada, de transporte do SBTVD, trouxe uma vasta gama de conhecimento. Porém a adoção de um sistema com *Real-time Object delivery over Unidirectional Transport (ROUTE)/Dynamic Adaptive Streaming over Hypertext Transfer Protocol(HTTP) (DASH)* como estrutura de transporte da TV 3.0 traz novos desafios a serem investigados e implementações não convencionais para o mercado brasileiro, exigindo novas análises para as aplicações no novo sistema (FORUM SBTVD, 2021b).

O MPEG-H *Audio* traz recursos inovadores, como a entrega *multi-stream*, Ponto de Acesso Aleatório, do inglês *Random Access Point (RAP)*, e o Quadro de Reprodução Imediata, do inglês *Immediate Playout Frame (IPF)* que simplificam a transmissão, recepção e o sincronismo dos serviços de áudio. Por meio desses recursos, é possível alcançar requisitos fundamentais estabelecidos para o novo padrão de TV brasileiro. Um dos principais

requisitos é a integração *Broadcast-Broadband*. Isso traz, por exemplo, a possibilidade de enriquecimento do conteúdo OTA via OTT. O desafio para atingir isso é a diferença de latência entre os dois métodos de transmissão.

A partir desse propósito, este trabalho tem por objetivo realizar testes de tolerância à latência entre os dois métodos de transmissão utilizando o recurso de entrega *multi-stream*, estudando os recursos disponíveis para alinhamento e sincronização do áudio com latências distintas.

1.3 Metodologia

Para a realização deste trabalho, foi realizado o estudo das funcionalidades do MPEG-H *Audio*. Esse estudo foi realizado com base em artigos científicos do Instituto de Engenheiros Elétricos e Eletrônicos, do inglês, *Institute of Eletrical and Electronics Enginners* (IEEE), Sociedade de Engenharia de Áudio, do inglês *Audio Engineering Society* (AES) e normas de padrões de televisão que já adotaram MPEG-H como *codec* de áudio, como *Advanced Television Systems Committee 3.0* (ATSC 3.0), *Digital Video Broadcast* (DVB) e SBTVD-T.

Os experimentos foram realizados com os equipamentos do Laboratório de TV Digital da Escola de Engenharia da Universidade Presbiteriana Mackenzie e fornecidos pelo Instituto de Circuitos Integrados Fraunhofer, do inglês *Fraunhofer Institute for Integrated Circuits* (Fraunhofer IIS).

1.4 Estrutura e Organização do Trabalho

O trabalho está estruturado em seis capítulos. O primeiro capítulo introduz o tema dentro do contexto no qual se encontra inserido. Também são evidenciado os objetivos do projeto, juntamente com a metodologia a ser aplicada para atingir tais objetivos e os elementos teórico-práticos que demonstram a relevância da pesquisa. No segundo capítulo, é mostrado a evolução da camada de transporte de televisão digital brasileira, definindo como funciona a camada de transporte atual do SBTVD-T com o MPEG-2 TS e a estrutura da TV 3.0, o padrão ROUTE/DASH. No terceiro capítulo, são explorados elementos de codificação de áudio na era digital, passando por conceitos e técnicas de

compressão de áudio e apresentando uma visão geral dos principais *codecs* de áudio. No quarto capítulo, são descritos os conceitos e estruturas por trás do MPEG-H *Audio*. O quinto capítulo apresenta os resultados obtidos e discussões sobre o desenvolvimento realizado e o conhecimento adquirido. O sexto e último capítulo apresenta as conclusões a cerca dos estudos e resultados obtidos neste trabalho.

2 EVOLUÇÃO DA CAMADA DE TRANSPORTE NA TELEVISÃO DIGITAL BRASILEIRA

A adoção da SBTVD-T no Brasil em 2007 mudou drasticamente a forma como o conteúdo era transportado por transicionar de um sistema analógico para um sistema digital. Essa transição para um sistema digital trouxe diversos benefícios, como uma maior qualidade de imagem e som, a possibilidade de transmissão de múltiplos programas no mesmo canal e a introdução de interatividade.

Isso porque o conteúdo de áudio e vídeo passou a ser comprimido, digitalizado e transportado em formato contêiner utilizando a tecnologia MPEG-2 TS. Isso aumentou a capacidade de envio de transmissão de dados dentro da mesma largura de banda.

A adoção do ROUTE/DASH tende a ser uma disrupção de mesma proporção por ser uma tecnologia baseada em Protocolo de Internet, do inglês *Internet Protocol* (IP). Isso proporciona uma melhora a integração do conteúdo *broadcast* e *broadband*, permitindo o enriquecimento do conteúdo de maneira mais simplificada em relação ao padrão atual.

Nesta seção serão explicadas essas tecnologias e suas aplicações na TV digital brasileira.

2.1 MPEG-2 *Transport Stream*

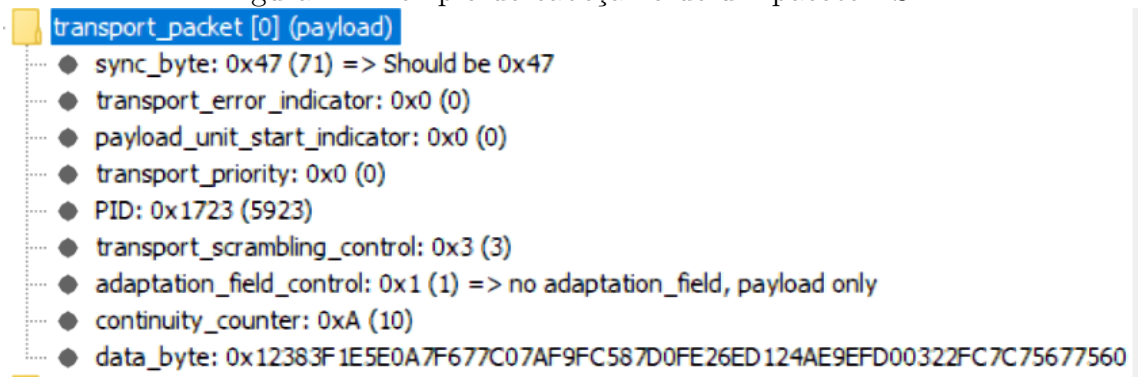
O MPEG-2 TS é um padrão de transmissão e armazenamento de dados descrito pela norma ISO/IEC - 13818-1 (2023). Esse padrão é amplamente usado na TV digital mundial, sendo adotado em sistemas como *Advanced Television Systems Committee* (ATSC 1.0), *Digital Video Broadcast - Terrestrial* (DVB-T/T2), *Digital Terrestrial Multimedia Broadcast* (DTMB), *Integrated Services Digital Broadcasting - Terrestrial* (ISDB-T) e SBTVD-T.

Um fluxo MPEG-2 TS é constituido por diversos pacotes de Fluxo de Transporte, do inglês *Transport Stream* (TS) de tamanho fixo de 188 *bytes*. Esses 188 *bytes* são divididos entre: Cabeçalho, Campo de Adaptação ou do inglês, *Adaptation Field* e os dados, também conhecidos como *Payload*. Dos 188, 4 *bytes* são destinados ao cabeçalho e os 184

bytes restantes são destinados a transmissão dos dados armazenados ou do *Adaptation Field*. Este cabeçalho possui campos que indicam informações essenciais para a identificação do pacote recebido. A Tabela 2 indica cada campo do cabeçalho de um pacote TS, sua quantidade de *bits* e função.

É possível ver na Figura 1 um exemplo do cabeçalho de um pacote TS.

Figura 1: Exemplo do cabeçalho de um pacote TS.



Fonte: Autoria Própria capturada da ferramenta DVB Inspector (2024).

No TS são encontradas as tabelas de Informação Específica do Programa, do inglês *Program Specific Information* (PSI) e Informação do Serviço, do inglês *Service Information* (SI). Essas tabelas são agrupamentos de dados que servem para enviar dados específicos ao receptor para comandar o processo de reprodução e decodificação dos pacotes (OLMEDO et al., 2016).

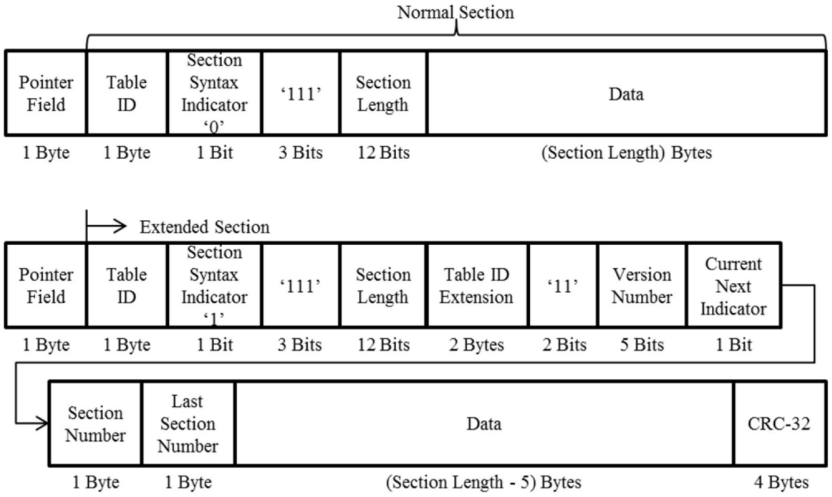
As tabelas são transmitidas no formato de seções. Existem dois tipos de seções: normais e estendidas. As seções normais possuem tabelas simples com estrutura básica. Já as estendidas são usadas em tabelas mais complexas em múltiplas seções. As Tabelas 3 e 4 descrevem cada campo das seções normais e estendidas, enquanto a Figura 2 descreve a ordem na qual esses campos devem ser descritos.

Tabela 2: Campos presentes no cabeçalho de um TS.

Campos do cabeçalho de um pacote TS	Quantidade de *bits*	Função
Sync Byte	8	Sempre com valor de 0x47. É utilizado para sincronismo de cada pacote.
Transport Error Indicator	1	Indica erro no transporte do pacote. Geralmente relacionado com erros na demodulação.
Payload Unit Start Indicator	1	Indica o inicio de um novo *Payload*. É utilizado para identificar o inicio de tabelas PSI/SI.
Transport Priority	1	Indicam a prioridade do pacote. Pacotes importantes podem ser processados antes.
PID	13	Importante para a identificação do tipo de pacote recebido.
Transport Scrambling Control	2	Indica a existência de criptografia e o tipo de criptografia utilizada.
Adaptation Field Control	2	Indica a presença do *Adaptation Field, Payload* ou ambos no pacote.
Continuity Counter	4	Valor do contador do pacote. É utilizado para conferir a continuidade dos pacotes recebidos. Uma descontinuidade neste campo pode indicar pacotes perdidos ou fora de ordem.

Fonte: Autoria Própria (2024) .

Figura 2: Sequência de apresentação dos campos de seções normais e estendidas de uma tabela de um TS.



Fonte: (VALEIRA et al., 2015).

Tabela 3: Descrição de uma seção normal de uma tabela TS.

Campos da Seção Normal	Quantidade de *bits*	Função
Table ID	8	Indica a tabela específica que a seção pertence.
Section Syntax Indicator	1	Indica o tipo de seção (0 = normal ou 1 = estendida).
Reserved Bits	3	Este campo é sempre "111" e é reservado para expansões.
Section Length	12	Indica o tamanho total da seção, incluindo cabeçalho e dados.
Data	12	Contém os dados específicos da tabela.

Fonte: Autoria Própria (2024).

As tabelas PSI contém as informações necessárias para a decodificação dos programas. Essas tabelas são: Tabela de Associação de Programas, do inglês *Program Association Table (PAT)*, Tabela de Mapeamento de Programas, do inglês *Program Mapping Table (PMT)* e Tabela de Acesso Condicional, do inglês *Conditional Access Table (CAT)*.

Tabela 4: Descrição de uma seção estendida de uma tabela TS.

Campos da Seção Estendida	Quantidade de *bits*	Função
Pointer Field	8	Indica o deslocamento do início da seção.
Table ID	8	Indica a tabela específica que a seção pertence.
Section Syntax Indicator	1	Indica o tipo de seção (0 = normal ou 1 = estendida)
Reserved Bits	3	Este campo é sempre "111" e é reservado para expansões.
Section Length	12	Indica o tamanho total da seção, incluindo cabeçalho e dados.
Table ID Extension	16	Extensão da "*Table ID*". Utilizado o valor 7 para identificar as seções adicionais.
Reserved Bits	2	Este campo é sempre "11" e é reservado para expansões.
Version Number	5	Indica a versão da tabela. Importante no caso de atualizações de uma tabela.
Current/Next Indicator	1	Indica se a seção atual esta sendo usada ou se será a próxima a ser utilizada.
Section Number	8	Indica o número da seção atual.
Last Section Number	8	Indica o número da última seção.
Data	12	Contém os dados específicos da tabela.
CRC-32	32	Contém um código de redundância cíclica (CRC) para detecção de erros na seção.

Fonte: Autoria Própria (2024).

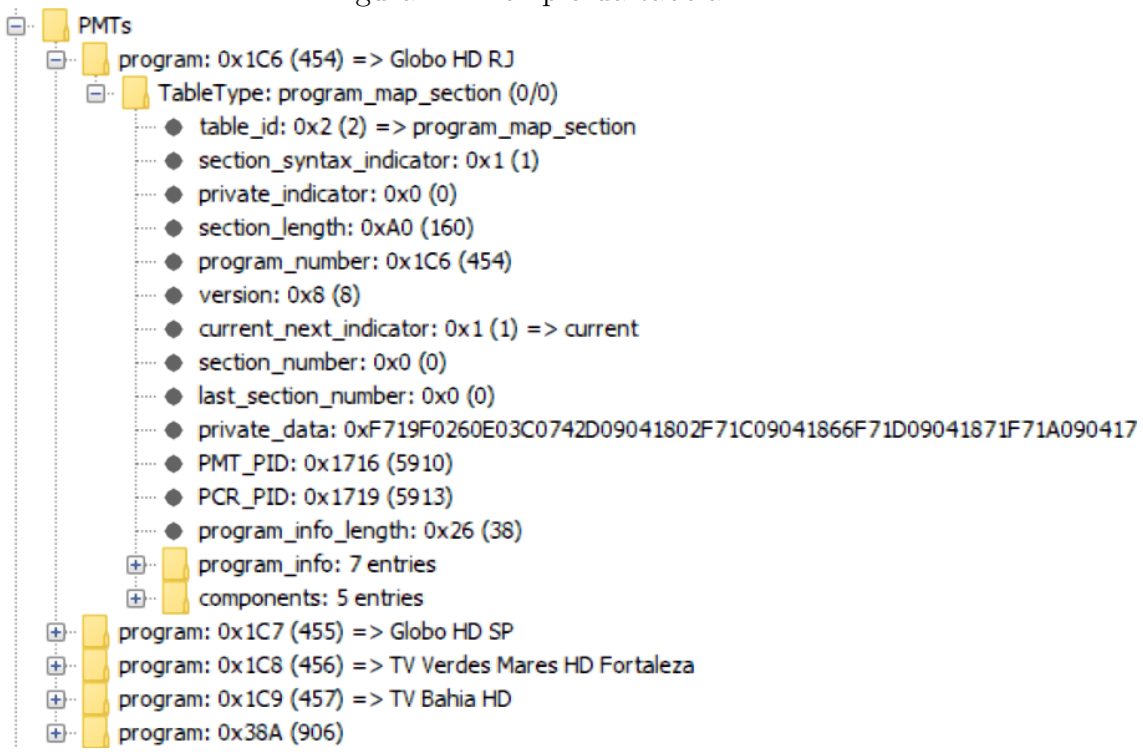
- PAT: É a tabela principal de um TS e a primeira a ser lida pelo demultiplexador. Ela informa obrigatoriamente a correspondência entre o serviços contidos no TS e seus respectivos *Packet Identifier* (PID)s para a PMT. É sempre definida pelo valor 0x0000.

Figura 3: Exemplo da tabela PAT.

Fonte: Autoria Própria capturada da ferramenta DVB Inspector (2024).

- PMT: É responsável por identificar e indicar a localização do *stream* correspondente a cada um dos serviços transmitidos e a localização da Referência de Relógio do Programa, do inglês *Program Clock Reference (PCR)*, para um serviço, como definido na ABNT - 15603-2 (2023). O PCR é responsável pela referência temporal dos elementos transmitidos. Os elemento de áudio e vídeo tem um único PCR-PID. Esse valor é utilizado para realizar a sincronização dos conteúdos de áudio e vídeo, trabalhando no controle do *buffer* de entrada. O PCR é vantajoso na compensação do *jitter* do sistema de transmissão, por exemplo.

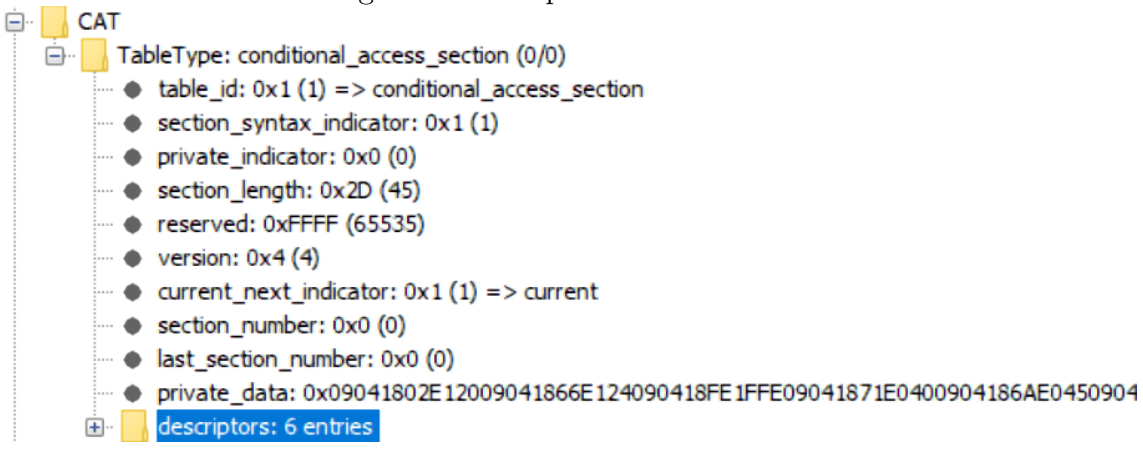
Figura 4: Exemplo da tabela PMT.



Fonte: Autoria Própria capturada da ferramenta DVB Inspector (2024).

- CAT: É responsável por descrever as informações de acesso condicional utilizados para criptografia do conteúdo. Se presente, é definido pelo valor 0x0001.

Figura 5: Exemplo da tabela CAT.



Fonte: Autoria Própria capturada da ferramenta DVB Inspector (2024).

Além da PSI, a SI descreve informações adicionais complementares referentes aos serviços e a rede de transmissão. Esses dados são estruturados em 11 tabelas possíveis:

Tabela de Associação de *Bouquet*, do inglês *Bouquet Association Table (BAT)*, Tabela de Informações da Rede, do inglês *Network Information Table (NIT)*, Tabela de Descrição de Serviços, do inglês *Service Description Table (SDT)*, Tabela de Informações de Eventos, do inglês *Event Information Table (EIT)*, Tabela de Hora e Data, do inglês *Time and Data Table (TDT)*, Tabela de Deslocamento de Tempo, do inglês *Time Offset Table (TOT)*, Tabela de Status de Execução, do inglês *Running Status Table (RST)*, Tabela de Aviso de Conteúdo de Fluxo Elementar Packetizado, do inglês *Packetized Elementary Stream Content Advisory Table (PCAT)*, Tabela de Preenchimento, do inglês *Stuffing Table (ST)*, Tabela de Informações do Radiodifusor, do inglês *Broadcaster Information Table (BIT)*, Tabela de Informações da Placa de Rede, do inglês *Network Board Information Table (NBIT)* e Tabela de Descrição Local, do inglês *Local Description Table (LDT)* (ABNT - 15603-2, 2023).

- BAT: Define os pacotes de serviços, chamados de *bouquets*, oferecidos por um provedor. Tem a função de facilitar a organização dos serviços providos por uma mesma emissora;
- NIT: Informa valores técnicos sobre a transmissão do dados como frequência, parâmetros de modulação, quantidade de fluxos disponíveis, nomenclatura e numeração dos canais, entre outros;
- SDT: Informa descrição dos serviços contidos no TS como nome, provedor e descrição dos serviços;
- EIT: contém informações sobre a grade de programação de um serviço como horários de início e fim, títulos e descrições;
- TDT: Indica a data e hora atual no formato *Coordinated Universal Time (UTC)*. É descrita em uma tabela a parte devido à constante atualização;
- TOT: Fornece informações sobre o fuso horário e os ajustes de horário de verão, permitindo a correção da hora fornecida na TDT;
- RST: Indica o *status* de execução dos serviços. Por exemplo, se o serviço está no começo ou no final;

- PCAT: Fornece informações sobre o conteúdo dos pacotes, como classificação etária, por exemplo;
- ST: Utilizada para preencher espaços vazios no TS. Pode ser usada para que o TS mantenha uma taxa de *bits* constante;
- BIT: Contém informações sobre o radiodifusor como indicadores e dados de contato;
- NBIT: Fornece informações detalhadas sobre o *hardware* e a configuração técnica da rede de transmissão;
- LDT: Contém informações sobre descrições locais de serviços.

Dentre todas as tabelas SI, as mandatórias são a NIT, SDT, EIT e TOT (VALEIRA et al., 2015).

Cada tabela possui um PID. Essa identificação é usada pelo receptor para identificar e decodificar as informações necessárias para reproduzir o conteúdo corretamente. A Tabela 5 mostra cada PID associado com as devidas tabelas.

Tabela 5: Alocação de PID para SI em um TS.

Tabela	**PID**
PAT	0x0000
PMT	Designado indiretamente pela PAT
CAT	0x0001
NIT	0x0010
SDT	0x0011
BAT	0x0011
EIT	0x0012
EIT(transmissão de televisão digital terrestre)	0x0012, 0x0026, 0x0027
RST	0x0013
TDT	0x0014
TOT	0x0014
PCAT	0x0022
BIT	0x0024
NBIT	0x0025
LDT	0x0025
ST	Exceção 0x0000, 0x0001, 0x0014
Pacotes Nulos	0x1FFF

Fonte: Adaptado da ABNT - 15603-2 (2023).

2.2 ROUTE/DASH

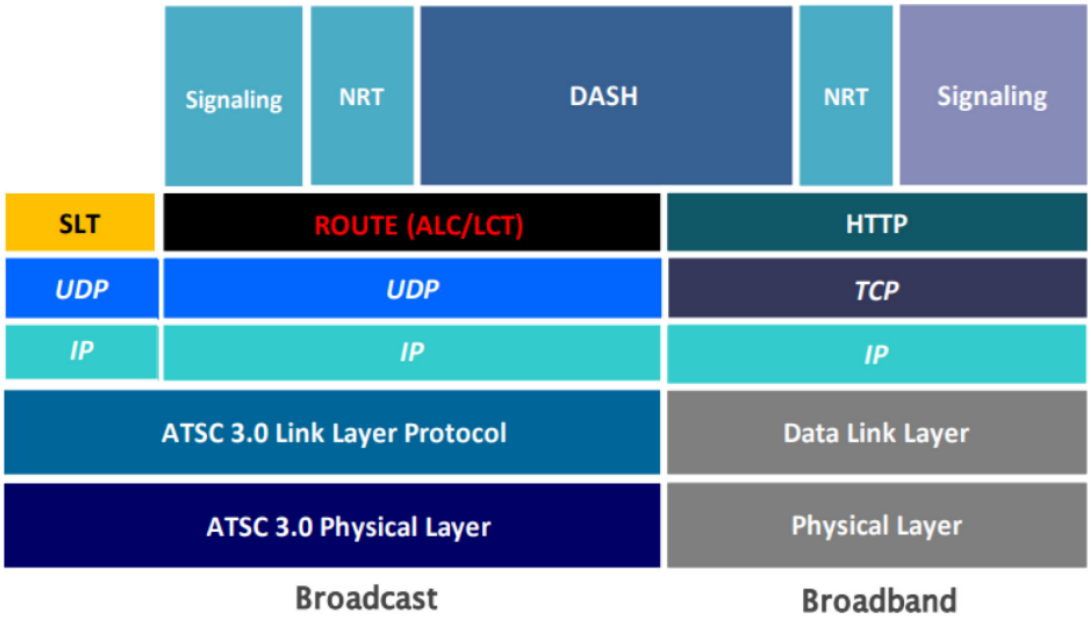
O ROUTE/DASH, padrão escolhido como estrutura de transporte da TV 3.0, é definido pela junção de duas tecnologias. O ROUTE é responsável pela estruturação do conteúdo que será entregue via camada física. Já o DASH diz respeito à forma como o conteúdo é encapsulado e organizado. Este capítulo é responsável por descrever como essas duas tecnologias funcionam.

2.2.1 ROUTE

O protocolo ROUTE é uma extensão da tecnologia de Entrega de Arquivos por meio de Transporte Unidirecional, do inglês *File Delivery over Unidirectional Transport* (FLUTE). Enquanto o FLUTE é mais adequado na entrega de arquivos grandes e estáticos, o ROUTE é mais eficiente para aplicações de entrega em tempo real, com capacidade de adaptação dinâmica, correção de erro com suporte a mecanismos avançados de *Forward Error Correction* (FEC) e suporte a *streaming* adaptativo.

O Protocolo da Camada de Ligação ATSC, do inglês *ATSC link-layer protocol* (ALP) é definido por uma sessão ROUTE e a sinalização da Tabela de Lista de Serviços, do inglês *Service List Table* (SLT). A ALP é transmitida via *User Datagram Protocol* (UDP) *multicast* (OLIVEIRA; VALEIRA; AKAMINE, 2024). Os conteúdos complementares, como enriquecimento de vídeo e outras opções de linguagens são enviados via OTT usando protocolos *Hypertext Transfer Protocol* (HTTP) e *Transmission Control Protocol* (TCP) (YOU; KIM; KIM, 2021). A Figura 6 ilustra essa pilha de protocolos de comunicação.

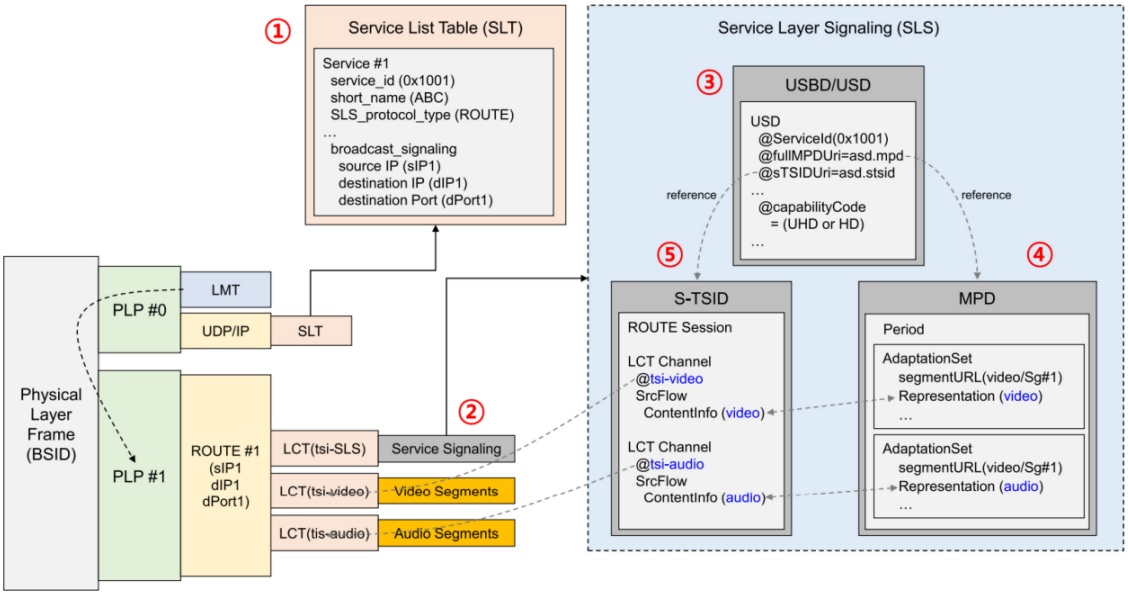
Figura 6: Pilha de Protocolo de comunicação do ROUTE/DASH para a camada de transporte da TV 3.0.



Fonte: Oliveira, Valeira e Akamine (2024).

A estrutura de aquisição dos dados de recepção de um sinal é ilustrada na Figura 7.

Figura 7: Sessão de aquisição de serviços ROUTE.



Fonte: (OLIVEIRA; VALEIRA; AKAMINE, 2024).

A recepção do conteúdo se inicia pela leitura da SLT, (1) na Figura 7. Essa tabela possui informações sobre cada canal de Transporte de Codificação em Camadas, do inglês *Layered Coding Transport* (LCT), que transporta uma *Service Layer Signaling* (SLS), (2) na Figura 7. Essas informações incluem os serviços disponíveis, incluindo IDs de serviço, nomes abreviados e informações de sinalização de transmissão, como IPs de origem e destino.

Em geral, uma seção ROUTE pode ser constituída por um ou mais canais LCT e cada canal LCT transporta um componente de conteúdo, como áudio, vídeo e legendas (YOU; KIM; KIM, 2021). A SLS pode conter algumas sinalizações, dependendo do tipo de serviço. As três sinalizações mais recorrentes são: o descritor de serviços do usuário, do inglês *User Service Description* (USD) ou o descritor do pacote de serviços ao usuário, do inglês *User Service Bundle Description* (USBD), (3) na Figura 7, o Descritor de Apresentação de Mídia, do inglês *Media Presentation Description* (MPD),(4) na Figura 7 e a Descritor da Instância de sessão de Transporte Baseada em Serviços, do inglês *Service Transport Session Identification Description* (S-TSID) ,(5) na Figura 7 (OLIVEIRA; VALEIRA; AKAMINE, 2024).

O USD indica informações gerais sobre o serviço como *Service Identification* (Service ID) e a capacidade do dispositivo. O USD é a referência para o receptor acessar as informações do serviço escolhido no MPD e o conteúdo do componente no S-TSID. Já o MPD fornece informações de identificação para os componentes de conteúdo que constituem o serviço, como a identificação do conteúdo, o tipo, formato de codificação, entre outros. Com essas informações, o dispositivo determina como decodificar e reproduzir o conteúdo. Por fim, o S-TSID fornece qual canal LCT corresponde aos segmentos DASH necessários para a decodificação (YOU; KIM; KIM, 2021).

2.2.2 DASH

O DASH foi criado em respostas a um CfP realizado pelo *Moving Picture Experts Group* (MPEG) em 2009 (SODAGAR, 2011). Esse CfP foi criado com o intuito de criar um novo padrão de *streaming* adaptativo via HTTP (SANTANA, 2023).

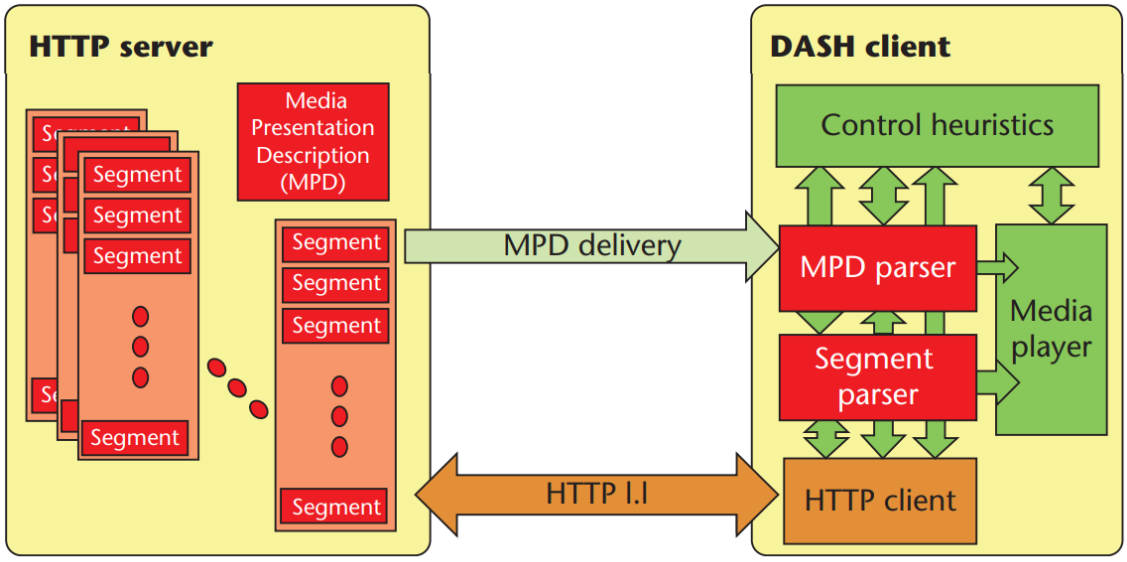
Esse padrão utiliza protocolo HTTP para realizar a transmissão dos dados. Uma das grandes vantagens do uso deste protocolo é a compatibilidade com *firewalls*. Quase todos os *firewalls* estão configurados para suportar as ligações de saída do protocolo HTTP(SODAGAR, 2011).

Este protocolo permite o trânsito na maioria dos *firewalls*, porque a grande maioria dos serviços IP *unicast* depende do HTTP (WALKER et al., 2016). Esta é uma grande vantagem em comparação com tecnologias de *streaming* anteriores ao DASH, que utilizavam o protocolo de Transporte em Tempo Real, do inglês *Real-time Transport Protocol* (RTP) ou o Protocolo de Streaming em Tempo Real, do inglês *Real Time Streaming Protocol* (RTSP) em sua maioria, tendo em vista que esses protocolos não são frequentemente permitidos por meio de *firewalls* (SODAGAR, 2011).

A utilização do PCR para sincronizar os dados recebidos fazem com que o MPEG-2 TS tenha uma desvantagem. Para que os dados estejam sincronizados, o PCR depende de uma transmissão em intervalos constantes. Isso faz com que o receptor dependa da suposição de um atraso constante (VAZ, 2024). Por ser um padrão de *streaming* adaptativo via HTTP, a utilização do DASH soluciona esse problema, sendo capaz de utilizar um sistema de *cache* e com a utilização de Coordenadas Universais de Tempo, do inglês UTC pela internet (WALKER et al., 2016).

O conteúdo de áudio e vídeo é dividido em fragmentos no formato *International Organization for Standardization Base Media File Format* (ISOBMFF) (ISO/IEC - 14496-12, 2022). Esses fragmentos são chamados de segmentos. Entre estes segmentos, também deve constar um arquivo manifesto, arquivo no formato *eXtensible Markup Language* (XML), responsável pela estrutura do conteúdo de mídia e as regras para seleção e reprodução dos segmentos. A Figura 8 representa essa comunicação.

Figura 8: Estrutura de comunicação do DASH.



Fonte: Sodagar (2011).

O arquivo manifesto de uma estrutura DASH é conhecido como Descrição de Apresentação de Mídia, do inglês MPD. Ele descreve o conteúdo dos segmentos de áudio e vídeo de forma hierárquica. O arquivo começa com um cabeçalho apresentando atributos gerais de apresentação da mídia. Alguns dos atributos, descritos conforme ISO/IEC - 23009-1 (2022), são:

- *@type*: Define o tipo de apresentação. No modo estático os segmentos estão disponíveis entre o *@availabilityStartTime* e o *@availabilityEndTime*. Por ter um início e fim pré-definido, geralmente é utilizado em serviços sob demanda.

  Já no modo dinâmico, os segmentos têm o tempo de disponibilidade diferente. O atributo *@availabilityStartTime* define o tempo inicial para disponibilidade do primeiro segmento e a partir daquele momento é calculado o tempo de disponibilidade

para os segmentos subsequentes. Caso o atributo *@minimumUpdatePeriod* esteja presente, esse valor pode ser atualizado e com isso, a base de cálculo será refeita. O modo dinâmico é geralmente utilizado para serviços em tempo real.

- *@availabilityStartTime*: No modo dinâmico, esse atributo deve estar presente. Ele é responsável por estabelecer a âncora do horário inicial do primeiro segmento disponível. No modo estático, ele especifica o horário inicial da disponibilidade dos segmentos.

- *@availabilityEndTime*: No modo dinâmico, esse atributo deve estar presente. Especifica o tempo de disponibilidade do último segmento.

- *@publishTime*: Especifica a hora em que o MPD foi criado.

- *@minBufferTime*: Especifica o menor valor de *buffer* usado para cada representação.

- *@profiles*: Indica o perfil da apresentação. Os perfis mais usados são *"urn:mpeg:dash:profile:isoff-ondemand:2011"* para serviços sob demanda e *"urn:mpeg:dash:profile:isoff-live:2011"* para serviços em tempo real.

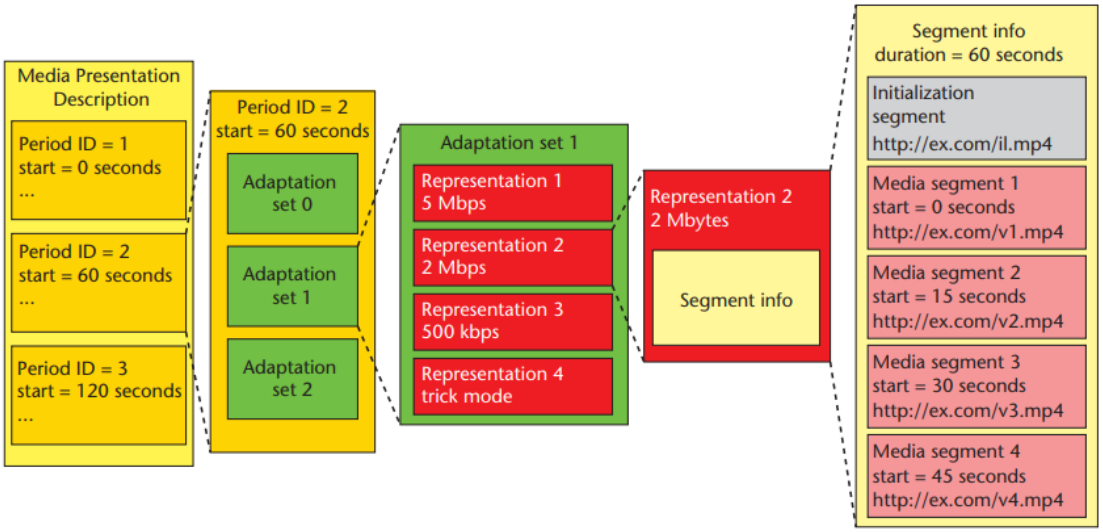
- *@mediaPresentationDuration*: Especifica a duração total da apresentação da mídia. Esse atributo deve estar presente caso o atributo *@minimumUpdatePeriod* e *Period@duration* do último período não estejam presentes.

- *@minimumUpdatePeriod*: Especifica o período mínimo de uma potencial atualização do MPD. Se não estiver presente, indica que o MPD não irá mudar. Este atributo está presente apenas no modo dinâmico.

Após o cabeçalho, são apresentados os elementos com a descrição da apresentação de mídia, começando pelo *Period IDs*. Esse elemento especifica o período de apresentação de mídia existentes naquele conteúdo. Ele pode definir um ou mais períodos de apresentação. Dentro desses períodos são definidos um ou mais *AdaptationSets*. Os *AdaptationSets*, por sua vez, podem conter uma ou mais representações, do inglês *Representations*. O *AdaptationSet* e seus *Representations* devem conter atributos e elementos que descrevem as configurações daquele conteúdo, permitindo a descrição dos segmentos e comutação sem descontinuidades. Esses atributos devem descrever o tipo do conteúdo (áudio, vídeo

ou dados), qual *codec* foi utilizado para a codificação do conteúdo, taxa de codificação, entre outros. Dentro de cada *representation*, devem conter informações específicas dos segmentos, como nome do arquivo de inicialização, padrão de nomenclatura dos segmentos e, se disponível, a *Uniform Resource Locator* (URL) do servidor de origem dos segmentos daquela representação. A Figura 9 ilustra a estrutura descrita.

Figura 9: Estrutura de um arquivo manifesto.



Fonte: Sodagar (2011).

A Figura 10 contém um exemplo de um arquivo MPD gerado no modo dinâmico. O MPD contém um período com dois *AdaptationSets*, um para o áudio e outro para vídeo.

Figura 10: Exemplo de um arquivo MPD.

```
<?xml version="1.0" encoding="UTF-8"?>
<MPD availabilityStartTime="2024-02-16T20:31:00Z" maxSegmentDuration="PT2.006S" minBufferTime="PT2S"
minimumUpdatePeriod="PT0S" profiles="urn:mpeg:dash:profile:isoff-broadcast:2015" publishTime=
"2024-02-16T20:31:00Z" timeShiftBufferDepth="PT33M20S" type="dynamic" xmlns="urn:mpeg:dash:schema:mpd:2011"
xmlns:cenc="urn:mpeg:cenc:2013" xmlns:dashif="https://dashif.org/" xmlns:scte35=
"http://www.scte.org/schemas/35/2016" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation=
"urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd http://dashif.org/guidelines/ContentProtection laurl.xsd">
   <Period id="P0" start="PT0S">
      <AdaptationSet contentType="video" id="0" maxFrameRate="60000/1000" maxHeight="1080" maxWidth="1920"
      mimeType="video/mp4" minFrameRate="60000/1000" minHeight="1080" minWidth="1920" par="16:9" segmentAlignment=
      "true" startWithSAP="1">
         <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
         <SegmentTemplate duration="480000" initialization="video-$Bandwidth$-init.mp4" media=
         "video-$Bandwidth$-$Number$.mp4v" startNumber="1" timescale="240000"/>
         <Representation bandwidth="5000000" codecs="hvc1.2.4.L123.B0" frameRate="60000/1000" height="1080" id=
         "Video1_1" sar="1:1" scanType="progressive" width="1920"/>
      </AdaptationSet>
      <AdaptationSet contentType="audio" id="1" mimeType="audio/mp4" segmentAlignment="true" startWithSAP="1">
         <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
         <SegmentTemplate duration="96000" initialization="audio-0-$Bandwidth$-init.mp4" media=
         "audio-0-$Bandwidth$-$Number$.mp4a" startNumber="1" timescale="48000"/>
         <AudioChannelConfiguration schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"
         />
         <Representation audioSamplingRate="48000" bandwidth="96000" codecs="mp4a.40.5" id="22"/>
      </AdaptationSet>
   </Period>
</MPD>
```

Fonte: Autoria Própria (2024).

O DASH utiliza uma estrutura com arquivos no formato contêiner ISOBMFF, baseado nas normas ISO/IEC - 14496-12 (2022) e ISO/IEC - 23009-1 (2022). Essa estrutura fragmenta um arquivo ISOBMFF em dois tipos: um arquivo de inicialização e segmentos de mídia. O arquivo de inicialização define informações importantes sobre temporização, estrutura e informações de mídia, como, por exemplo, qual *codec* foi utilizado na codificação. Já os segmentos de mídia contém os dados brutos de mídia.

A estrutura de um arquivo ISOBMFF é dividida no formato de objetos estruturados, chamados de *boxes*. Normalmente, esse arquivo contém 3 boxes: o *File Type Box* (FTYP), o *Movie Box* (MOOV) e o *Media Data Box* (MDAT).

O FTYP define qual o tipo do arquivo ISOBMFF e a estrutura usada. Deve ser colocado no início do arquivo. Já o MOOV carrega informações necessárias para a reprodução do conteúdo, incluindo os metadados. Por fim, o MDAT contém os dados de reprodução de mídia. Sua estrutura é descrita pelos metadados da MOOV.

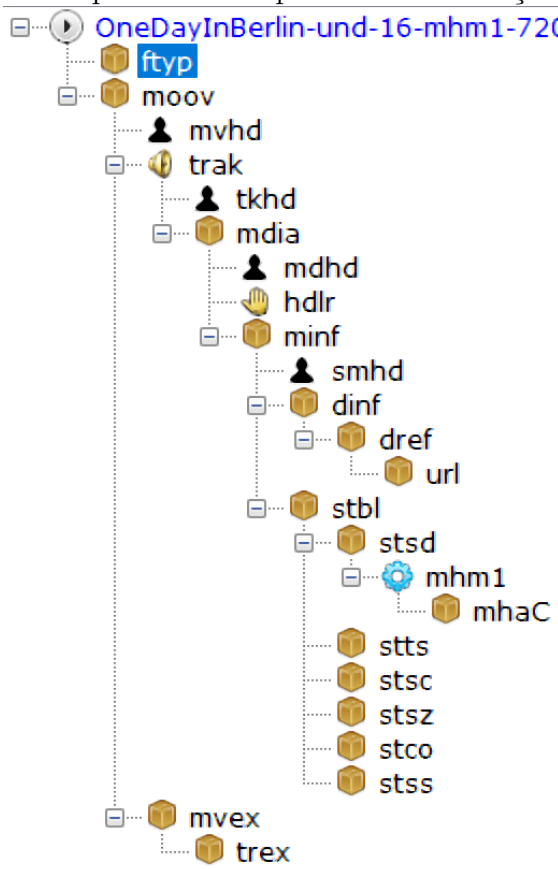
O MOOV é uma estrutura de alto nível organizada em subconjuntos. Esses subconjuntos descrevem uma estrutura de dados específicos sobre o conteúdo do arquivo. Dentro da estrutura MOOV, a estrutura *Track Box* (TRAK), que descreve as informações de uma única trilha. Múltiplas TRAKs podem estar presentes dependendo da quantidade

de trilhas presentes no arquivo. Dentro dela, o *Media Box* (MDIA) contém todas as informações necessárias para descrever a mídia em uma trilha específica. Além de seu cabeçalho, esse box é descrito pelo *Handler Reference Box* (HDLR) e pelo *Media Information Box* (MINF). O HDLR especifica o tipo de mídia enquanto e MINF contém informações específicas sobre o tipo de mídia especificado. A MINF é dividida em 2 tipos de informações: informações sobre a localização dos dados da mídia descritas na *Data Information Box* (DINF) e tabelas que descrevem a amostragem de dados na trilha, descritas na *Sample Table Box* (STBL). Dentro da STBL, é possível conferir, por exemplo, o tipo de *codec* utilizado na *Sample Description Box* (STSD), o tempo de cada amostra na *Time-to-Sample Box* (STTS), o tamanho das amostras na *Sample Size Box* (STSZ), etc.

Além da estrutura TRAK, a MOOV pode possuir um box chamado *Movie Extends Box* (MVEX), contém informações que estendem o conteúdo. Esse box é usado principalmente em arquivos de mídia que serão atualizados.

Um arquivo de inicialização deve conter apenas o conteúdo dos boxes FTYP e MOOV. O conteúdo de mídia que estaría presente no MDAT é fragmentado em nos segmentos de mídia. A Figura 11 demonstra a estrutura de um arquivo de inicialização.

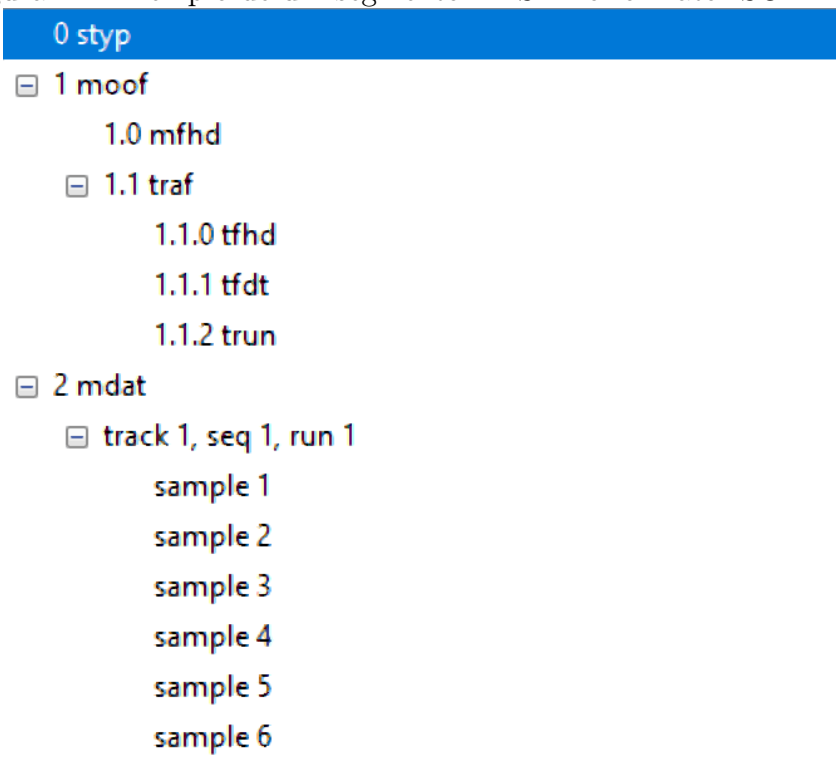
Figura 11: Exemplo de um arquivo de inicialização DASH no formato ISOBMFF.



Fonte: Autoria Própria capturada da ferramenta MP4 Inspector (2024).

Os segmentos de mídia possuem uma estrutura semelhante da descrita anteriormente. Um segmento contém 3 boxes: *Segment Type Box* (STYP), *Movie Fragment Box* (MOOF) e o MDAT. O STYP e o MOOF são análogos ao FTYP e o MOOV respectivamente, porém são usados para descrever arquivos fragmentados. Analogamente ao MOOV, o MOOF contém uma estrutura *Track Fragment Box* (TRAF). A TRAF apresenta uma estrutura simplificada, contendo, além de seu cabeçalho, os boxes *Track Fragment Decode Time Box* (TFDT) e *Track Fragment Run Box* (TRUN). O TFDT contém o tempo de decodificação para o fragmento da trilha. Este tempo é usado para sincronizar a reprodução. Já o TRUN contém informações sobre as amostras dentro do fragmento da trilha, incluindo a quantidade de amostras, tamanho, duração e outras propriedades. O box MDAT possui as amostras organizadas em trilha, sequência e execução. A Figura 12 ilustra essa estrutura.

Figura 12: Exemplo de um segmento DASH no formato ISOBMFF.



Fonte: Autoria Própria capturada da ferramenta MP4 Analyzer (2024).

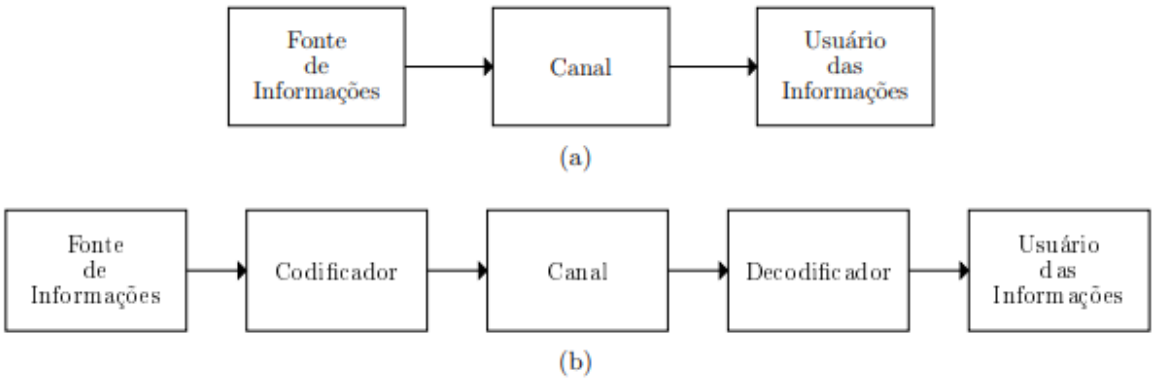
3 ELEMENTOS DA CODIFICAÇÃO DE ÁUDIO NA ERA DIGITAL

Devido a limitação de taxa de *bits*, as técnicas de compressão e codificação de áudio são importantes para a transmissão de arquivos de áudio. Isso torna possível a transmissão de arquivos de extrema qualidade, mantendo elementos específicos para a percepção do ouvido humano e descartando outros não tão essenciais. Nesta seção serão abordados fundamentos e técnicas para a realização desta compressão e da codificação do áudio, além de uma análise dos principais *codecs* de áudio predecessores ao MPEG-H *Audio*.

3.1 Conceitos e técnicas de compressão de áudio

Um *codec* (abreviação de codificador-decodificador) é definido por um algoritmo que comprime e encapsula os dados a fim de facilitar a transmissão do conteúdo. Esse processo é realizado por um codificador e consequentemente, um decodificador. Na Figura 13 é possível conferir um sistema de comunicação sem codificação (a) e um sistema de comunicação com codificação (b).

Figura 13: Sistema de comunicação com codificação (a) e sem codificação (b).



Fonte: (MOTTA, 2019).

A compressão de áudio é essencial para reduzir o tamanho dos arquivos de áudio, facilitando a transmissão. Com avanço de aplicações e tecnologias de áudio, vem a necessidade de transmitir áudio com uma alta qualidade porém mantendo a taxa de *bits* reduzida.

Isso permite que mais dados sejam transmitidos em redes com largura de banda limitada, mantendo a qualidade de percepção do áudio original.

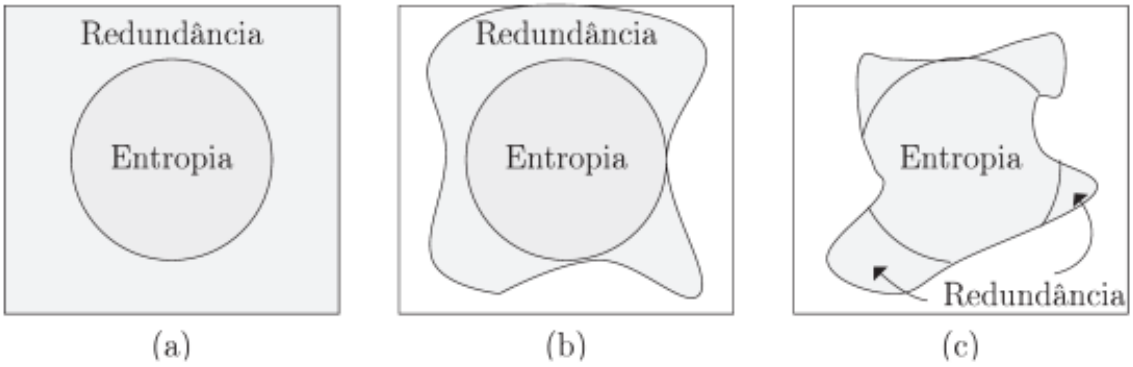
Existem dois tipos de compressão de áudio:

- Compressão com perdas: Essa compressão reduz a quantidade de dados de menor percepção ao ouvido humano. Essa compressão usa modelos perceptuais para identificar e destacar sons que estão fora da percepção humana ou que são mascarados por outros sons. É possível descartar como principais formatos que usam esse tipo de compressão o *MPEG Audio Coding Layer 3* (MP3), AAC e o MPEG-H *Audio*.

- Compressão sem perdas: Essa compressão mantém a qualidade do áudio na compressão. Todos os dados do áudio original podem ser resgatados após a descompressão do áudio. Por conta desta característica, os algoritmos de compressão sem perda, como o Codec de Áudio Sem Perdas Gratuito, do inglês *Free Lossless Audio Codec* (FLAC) e o Formato de Arquivo de Áudio em Formato de Onda, do inglês *Waveform Audio File Format* (WAV), geram arquivos de áudio maiores.

O conceito de entropia, definida por Shannon (1948) mede a quantidade mínima de bits necessária, em média, para codificar uma mensagem de uma fonte de informação de maneira eficiente. Esse conceito fornece uma base teórica para entender os limites da compressão de dados. Na compressão sem perdas, o objetivo é alcançar a taxa de compressão máxima possível sem alterar a entropia, ou seja, sem perder nenhuma informação essencial. Na compressão com perdas, parte da entropia é deliberadamente reduzida para alcançar uma taxa de compressão ainda maior, aceitando que algumas informações serão perdidas. A Figura 14 ilustra os dados originais, os dados após a compressão com e sem perdas.

Figura 14: Dados original (a), dados após a compressão sem perdas (b) e dados após a compressão com perdas (c).



Fonte: (MOTTA, 2019).

As compressões com perdas buscam usar técnicas para reduzir a perda de dados úteis através da eliminação de dados irrelevantes. Exemplos de técnicas que utilizam esse conceito são as técnicas perceptuais de compressão e o processo de quantização (MOTTA, 2019).

A quantização é o processo de mapear um grande conjunto de valores de entrada para um conjunto menor de valores de saída (SALOMON, 2007). Essa quantização pode ser feita de maneira uniforme ou não uniforme. Na quantização uniforme o sinal é dividido em intervalos iguais e cada valor de amostra é arredondado para o ponto médio do intervalo mais próximo. A quantização não uniforme utiliza intervalos de tamanhos variáveis, geralmente maiores para amplitudes maiores e menores para amplitudes menores. Isso é mais eficiente para sinais de áudio pelo fato que a percepção humana de diferenças de amplitude é logarítmica. Essa técnica inevitavelmente introduz um erro de compressão, que é a diferença entre o valor original e o valor quantizado.

As técnicas perceptuais de compressão se baseiam em modelagens psicoacústicas da estrutura auditiva do usuário a fim de encontrar limitações e com isso, descartar de maneira inteligente informações imperceptíveis. Essas técnicas são amplamente utilizadas em *codecs* de áudio tradicionais. A motivação para o desenvolvimento dessas técnicas e a aplicação delas em *codecs* de áudio foi a sua potencial aplicação em multimídias *broadcast*, porém também teve grande impacto na distribuição de áudio pela internet (SAYOOD,

2006).

Essa modelagem trabalha com os estudos das curvas de sensibilidade do ouvido humano estabelecendo filtros de duas formas: o mascaramento de frequência e o mascaramento de tempo. O mascaramento espectral, ou mascaramento de frequência, ocorre quando dois sons de frequências próximas ocorrem ao mesmo tempo. Neste caso, o som com maior amplitude tende a mascarar o de menos amplitude. A compressão de áudio com perdas usa essa propriedade para remover o som mascarado, já que sua ausência não será percebida pelo ouvinte. O mesmo conceito é utilizado de forma similar no mascaramento de tempo. Sons altos podem mascarar sons mais baixos que ocorrem imediatamente antes ou depois deles. Este efeito temporal é usado para descartar informações que não serão percebidas devido ao mascaramento causado por sons de alta amplitude em momentos próximos.

3.2 Visão geral dos codecs de áudio

Este item será dedicado a apresentar uma visão geral dos principais *codecs* de áudio que estabeleceram o legado até o surgimento de tecnologias avançadas como o MPEG-H *Audio*.

3.2.1 MPEG Audio Coding

O primeiro *codec* desenvolvido pelo grupo MPEG foi o *MPEG Audio Coding Layer I* na Alemanha no final dos anos 1980. Ele foi desenvolvido com a principal preocupação de reduzir o tamanho dos arquivos de áudio sem comprometer significativamente a qualidade. Foram propostos 14 algoritmos onde o *Layer* I emergiu como a solução de menor complexidade e facilidade na implementação. Devido à sua simplicidade e baixa complexidade, o *Layer* I foi adotado em sistemas onde a eficiência de processamento e a baixa latência eram críticas como aplicações em sistemas de transmissão digital, onde a compressão 4:1 proporcionava um bom equilíbrio entre qualidade e economia, e armazenamento digital como CDs de áudio digital (SAYOOD, 2006).

A menor eficiência da compressão e a qualidade inferior do áudio comprimido em comparação com os *Layers* subsequentes limitaram sua adoção generalizada, e conse-

quentemente seu uso diminuiu rapidamente à medida que tecnologias mais avançadas foram desenvolvidas .

Com base nos desenvolvimentos realizados no *Layer I*, foi desenvolvido o *MPEG Audio Coding Layer II*. Este foi desenvolvido no inicio dos anos 1990 para incorporar melhorias na eficiência da compressão e na qualidade do áudio, mantendo uma complexidade moderada. Tornou-se parte do padrão *MPEG-1* em 1992 e incluído no padrão *MPEG-2* posteriormente.

Devido às melhoras na eficiência de compressão e na qualidade do áudio, foi amplamente adotado em sistemas de transmissão de áudio como no Rádio Digital, do inglês *Digital Audio Broadcasting* (DAB) e na Transmissão de Televisão Digital por satélite, do inglês DVB. A capacidade de manter a compatibilidade com dispositivos que usavam *Layer* I ajudou na transição suave para tecnologias mais avançadas sem a necessidade de substituições de *hardware* significativas (PAINTER; SPANIAS, 2000).

Ainda hoje, o *Layer* II é usado em algumas aplicações específicas de transmissão devido à sua simplicidade e eficiência.

Após diversos testes, engenheiros da Fraunhofer IIS desenvolveram o MP3. O algoritmo combina técnicas avançadas de compressão com uma abordagem baseada em modelos psicoacústicos para minimizar a perda perceptível de qualidade. Essa combinação garantiu a maximização da eficiência de compressão e a qualidade do áudio. Em 1993 foi introduzido como parte do padrão MPEG-1 e posteriormente incluído no MPEG-2 (SAYOOD, 2006).

A criação do algoritmo revolucionou a forma como a música era distribuída e consumida, facilitando a distribuição digital e o armazenamento eficiente de música. O MP3 se tornou um dos formatos de áudio digital mais populares do final dos anos 1990 e início dos anos 2000, suportado por uma ampla gama de dispositivos, desde computadores a *players* de MP3 portáteis. A adoção generalizada do MP3 e sua compatibilidade universal com diversos dispositivos com *hardware* e *software* diferentes consolidou o MP3 como o formato de áudio padrão para consumidores permitindo a criação de serviços de compartilhamento de arquivos e, eventualmente, o surgimento de plataformas de *streaming* de música.

O impacto duradouro do MP3 na indústria musical e na cultura pop é inegável, solidificando sua posição como uma das inovações mais significativas na tecnologia de áudio digital.

3.2.2 MPEG Advanced Audio Coding (AAC)

O AAC foi desenvolvido e introduzido em 1997 como parte do conjunto de padrões MPEG-2 em resposta à crescente demanda por codificação de áudio de alta qualidade com taxas de bits reduzidas, especialmente para material de programa multicanal. Anteriormente, os algoritmos *MPEG Audio Coding Layer I* e *MPEG Audio Coding Layer II*, descritos na seção 3.2.1, embora eficazes, não conseguiam codificar áudio de cinco canais em taxas abaixo de 640 kb/s sem comprometer a qualidade (PAINTER; SPANIAS, 2000). A necessidade de um sistema de codificação avançado foi influenciada pelo rápido desenvolvimento da capacidade de *hardware* na época, permitindo a implementação de algoritmos mais sofisticados e eficientes.

O AAC superou as metas de design iniciais ao entregar qualidade "indistinguível" com taxas de até 320 kb/s para cinco canais de largura total (PAINTER; SPANIAS, 2000). O AAC utilizou uma abordagem modular, incorporando ferramentas tanto de padrões anteriores do MPEG quanto novas adições, configuradas de forma a permitir perfis com diferentes graus de complexidade e eficiência (SAYOOD, 2006). Essa adições incluem a introdução de ferramentas de codificação inovadoras, como adaptação de forma da janela do banco de filtros, predição de coeficientes espectrais e formas sofisticadas de codificação sem perdas.

O AAC provou ser significativamente mais eficiente que seus predecessores, oferecendo melhor qualidade de áudio, menor taxa de bits e melhor manipulação de sinais multicanais, tornando-o uma escolha superior para a compressão de áudio digital em várias aplicações. O sucesso e a eficácia do MPEG-2 AAC levaram à sua adoção como o modelo de referência para o mais recente padrão MPEG-4.

Com a sua adição ao padrão MPEG-4, o AAC se tornou um elemento fundamental em muitas tecnologias de transmissão digital, sendo adotado por exemplo no atual SBTVD-T, no sistema de TV japonês Serviço Integrado de Transmissão Digital Terrestre, do inglês ISDB-T e no DVB.

3.2.3 Next Generation Áudio (NGA)

Os chamados *codecs* de áudio da próxima geração, ou *Next Generation Audio* (NGA) são *codecs* com funcionalidades além do formato tradicional estéreo e 5.1 *surround*. Eles inovam na capacidade de oferecer experiencia imersiva, personalizada e interativa para o usuário. O conceito NGA foi introduzido para tecnologias de áudio que tendem a acompanhar os avanços de tecnologias de vídeo como resolução 8K e Alto Alcance Dinamico, do inglês *High Dynamic Range* (HDR).

Uma experiencia de áudio imersiva é atingida quando juntamos os conceitos de áudio padrão estéreo e *surround* (formatos baseados em canais) com novas técnicas de áudio baseado em objetos e em ambiência (MURTAZA; MELTZER, 2019). São usados metadados para descrever como o áudio deve ser renderizado na configuração de reprodução.

Um *codec* NGA permite que o usuário escolha entre diversas opções de personalização como a escolha de múltiplos idiomas, habilitar áudio descrição ou simplesmente ajustar as configurações para obter maior isolamento do diálogo em cena ou do ambiente de fundo.

A interação do usuário final com o áudio baseado em objetos pode ser simplificada por meio de "pré-seleções" ou *preselections* que podem ser criadas pelo radiodifusor e que podem ser transmitidas no conteúdo entregue.

4 MPEG-H AUDIO

O MPEG-H *Audio* representa uma evolução significativa nos sistema de áudio, sendo referido como *codec NGA*. Este *codec* destaca-se por proporcionar maior flexibilidade às empresas de radiodifusão, operadores e provedores de conteúdo (ETSI - TS 101 154, 2022). Sua entrada no mercado é marcada pela capacidade de oferecer uma experiência de áudio avançada e eficiente, introduzindo conceitos cruciais para o futuro do entretenimento. Características inovadoras, como áudio imersivo em três dimensões e interatividade direta com o telespectador, posicionam o MPEG-H *Audio* à frente das tecnologias de áudio, antecipando e influenciando as expectativas para o consumo de conteúdo sonoro na era digital.

O desempenho do MPEG-H *Audio* foi testado pelo grupo *MPEG*, incluindo 9 laboratórios de teste pelo mundo inteiro, sendo eles Fraunhofer IIS, Sony, *Nippon Hoso Kyokai* (NHK), Gaudio, Nokia, Orange, Qualcomm, Dolby e *Electronics and Telecommunications Research Institute* (ETRI) (GREWE; MURTAZA; MELTZER, 2023). Esse desempenho foi registrado e está disponível nos relatórios de teste do MPEG (MPEG W19407,) e (MPEG W16584,).

O MPEG-H *Audio* vem sendo amplamente adotado em sistemas ao redor do mundo, confirmando sua relevância como uma tecnologia de próxima geração. Sua adoção em padrões e normas internacionais como o ATSC 3.0 (ATSC - A/342-3, 2023), *Telecommunications Technology Association (TTA)*, DVB (ETSI - TS 101 154, 2022), no atual SBTVD-T no projeto TV 2.5 e especialmente como o principal *codec* de áudio na TV 3.0, destaca sua aceitação global. Essa adoção generalizada solidifica o papel do MPEG-H *Audio* como uma tecnologia-chave na evolução dos padrões de áudio para a próxima geração de serviços de radiodifusão.

Com o auxílio de mecânicas inteligentes, o MPEG-H *Audio* traz uma variedade de elementos que fazem com que o usuário final tenha uma melhor experiência. Desde o suporte as três principais formas de reprodução de áudio (canais, objetos e ambiência), podendo utilizá-las separadamente ou se complementando, até uma estrutura de metadados, o MPEG-H *Audio* é capaz de prover um serviço rico em possibilidades. Isso é feito por meio de ferramentas, como a entrega *multi-stream*, mecanismos de alinhamento em

pontos de acesso aleatório como o Quadro de reprodução imediata, ou IPF e o mecanismo de truncamento de áudio, entre outros.

Nas seções a seguir, serão evidenciados conceitos para o melhor entendimento de tal estrutura.

4.1 Áudio Imersivo

A forma como cada canal de áudio é tratado, pode ser representada de algumas formas diferentes, trazendo um impacto significativo na experiência auditiva e proporcionando maior imersão ao conteúdo e uma riqueza de detalhes única. Existem três abordagens principais para basear a reprodução do áudio, cada uma contribuindo de maneira distinta para a criação de uma melhor experiência sonora: baseado em canais, em objetos e em ambiência. Sendo o único formato de áudio que suporta nativamente *Higher Order Ambisonics* (HOA), MPEG-H *Audio* pode proporcionar um som envolvente utilizando uma combinação destes três formatos de reprodução de áudio bem estabelecidos (GREWE; MURTAZA; MELTZER, 2023).

4.1.1 Baseado em Canais

A abordagem clássica e mais simples no processamento de áudio é baseada em canais, em que cada canal é atribuído a um alto-falante que representa uma posição fixa no espaço. No início da era da reprodução de áudio, os sistemas eram predominantemente mono, onde um único canal de áudio era reproduzido por um único alto-falante. Isso significa que todos os sons eram combinados e transmitidos por uma única fonte.

O avanço veio com o desenvolvimento dos sistemas estéreo criados por Dower (1936). Nesse contexto, dois canais de áudio (esquerdo e direito) foram introduzidos, e os alto-falantes correspondentes foram posicionados estrategicamente para criar uma sensação de espaço e direção. Isso melhorou significativamente a experiência auditiva, especialmente em gravações musicais, onde a separação dos canais podia ser aproveitada para criar efeitos estéreos.

Com o tempo, a demanda por experiências mais imersivas exigiu uma evolução no sistema de áudio. Ao distribuir os canais entre alto-falantes frontais e traseiros e posicionar

estes em um arranjo pré-determinado, o sistema consegue oferecer ao usuário uma percepção de localização e direção no espaço sonoro. Esse posicionamento dos alto-falantes é chamado de sistema de som *surround*.

Diversas configurações são viáveis para sistemas de som *surround*, destacando-se entre elas as variantes 5.1, 7.1, 9.1 e 22.2. O número antes do ponto representa a quantidade de canais principais de áudio com alto-falantes projetados para uma ampla faixa de frequência. O número após o ponto indica a presença de um canal de efeitos para baixa frequência, do inglês *Low Frequency Effects (LFE)*, cujo propósito é transmitir apenas sons abaixo de 120 Hz (ITU-R - BS755, 2022). Este canal de baixa frequência é transmitido por um *subwoofer*, um alto-falante projetado especificamente para reproduzir faixas de frequência mais baixas. Portanto, o 5.1 possui cinco canais principais de áudio (frontais esquerdo, central e direito, além dos traseiros esquerdo e direito) e um canal de *subwoofer* para graves. O 7.1 inclui sete canais principais de áudio (frontais esquerdo, central e direito, traseiros esquerdo e direito, além de canais laterais esquerdo e direito) e um canal de *subwoofer* para graves. No caso do 9.1, são nove canais principais de áudio (frontais esquerdo, central e direito, traseiros esquerdo e direito, canais laterais esquerdo e direito, e dois canais adicionais para criar uma experiência tridimensional), acompanhados por um canal de *subwoofer* para graves. Já o 22.2 apresenta vinte e dois canais principais de áudio para uma experiência sonora altamente detalhada e imersiva, junto com dois canais de *subwoofer* para graves.

Essas configurações oferecem uma reprodução mais precisa do áudio em diversos contextos, desde entretenimento doméstico até ambientes profissionais como salas de cinema e estúdios de produção. O Setor de Radiocomunicação da União Internacional de Telecomunicações, do inglês *International Telecommunication Union Radiocommunication Sector (ITU-R)* recomenda um sistema com no mínimo três canais frontais e dois traseiros e um *subwoofer* como opcional (ITU-R - BS755, 2022).

O lado negativo dessa abordagem é a falta de padronização entre a produção e a configuração final do sistema, considerando que o conteúdo consumido pode ter sido produzido para um sistema 5.1, enquanto o usuário final possui um sistema estéreo. Quando isso ocorre, é necessário realizar um processo chamado *down-mixing*, que envolve a combinação ou redução dos canais de áudio para adequar o conteúdo ao sistema de reprodução

disponível. De forma análoga, no exemplo oposto, em que o conteúdo foi originalmente produzido para um sistema estéreo e o usuário final o reproduz em um sistema 5.1, é necessária a realização de um processo conhecido como *up-mixing*. Esse procedimento visa expandir o conteúdo estéreo para se adequar aos canais adicionais do sistema 5.1, proporcionando uma experiência sonora adaptada.

4.1.2 Baseado em Objetos

A abordagem baseada em objetos difere significativamente da abordagem baseada em canais, devido a sua capacidade de variar no tempo (MURTAZA; MELTZER, 2019). Nesse contexto, o termo "objetos" refere-se a pontos virtuais que representam fontes sonoras e suas características dentro de um espaço tridimensional. Diferente da abordagem baseada em canais, onde cada alto-falante representa um canal de áudio, esses objetos podem se movimentar no tempo, proporcionando uma experiência mais dinâmica e imersiva para o ouvinte.

A fim de atingir o objetivo do movimento, o sinal precisa ter um algoritmo de renderização apropriado, como, por exemplo, o *Vector Base Amplitude Panning (VBAP)* (HERRE E HILPERT, 2015). O VBAP é um método com o propósito de criar um campo sonoro tridimensional na qual o som pode ser reproduzido com a localização precisa da fonte sonora dentro espaço. Por meio de equações computacionalmente eficientes, o VBAP utiliza vetores de amplitude para distribuir o sinal da fonte sonora entre os alto-falantes disponíveis (PULKKI, 1997).

O conceito de objetos é fundamentalmente associado aos metadados, que são informações adicionais incorporadas ao conteúdo sonoro. Esses metadados descrevem não apenas a posição espacial dos objetos sonoros, mas também outras propriedades importantes, como características acústicas específicas, informações sobre a fonte sonora e dados relacionados à dinâmica do som.

Ao usar a abordagem baseada em objetos, os criadores de conteúdo têm maior flexibilidade para posicionar e movimentar fontes sonoras no ambiente auditivo. Isso não só permite uma experiência mais realista mas também oferece aos engenheiros de som a capacidade de adaptar a reprodução de áudio a diferentes configurações de alto-falantes e preferências do usuário.

4.1.3 Baseado em Ambiência

O áudio baseado em ambiência, também conhecido como *ambisonic*, se refere a uma abordagem levando em consideração a interação coletiva das fontes sonoras com o ambiente inserido. O sistema foi definido por Fellgett (1974) da seguinte forma:

> "Portanto, a mais alta fidelidade na reprodução sonora exige que a direcionalidade do som imite tanto o som direto quanto o reverberante da sala de concertos. Não se trata apenas de uma questão de um vago respingo de ecos atrasados, mas de uma relação entre direcionalidade e atraso de tempo que fornece informações específicas; essas informações são o que queremos dizer estritamente com ambiência. Os sistemas capazes de reproduzí-la serão chamados de "*ambisonic*"."

Com microfones Ambisônicos criados por Craven e Gerzon (1977), são gravadas a intensidade, direção e fase do som em todas as direções ao redor do microfone. Nessa gravação, são gerados coeficientes de harmônicas esféricas. Cada coeficiente representa uma harmônica esférica que contribui para a forma como o som é percebido em diferentes direções. Quanto maior o número da harmônica, mais refinada é a representação espacial. Por exemplo, o áudio *ambisonic* de primeira ordem é representado por quatro canais, que capturam as informações de intensidade e direção do som em todas as direções ao redor do ouvinte. Esses quatro canais são comumente rotulados como W, X, Y, e Z. A letra W representa a componente de pressão sonora omnidirecional, indicando a influência geral do som enquanto X,Y e Z representam as componentes direcionais nas três dimensões do espaço. Um exemplo prático seria o canal X representando a intensidade e a direção do som proveniente diretamente da frente, o canal Y indicaria a direção e intensidade dos sons laterais, e o canal Z seria responsável por representar a direção e intensidade do som que se origina de cima ou de baixo. O resultado é uma esfera de informações sonoras que pode ser manipulada durante a reprodução.

O sistema MPEG-H *Audio* trabalha com essa abordagem com *Ambisonics* de Ordem Superior, do inglês HOA. O HOA fornece mais coeficientes e, portanto, uma maior seletividade espacial, o que permite a reprodução de sinais de alto-falante com menos interferência, resultando em timbres com artefatos reduzidos.

4.1.4 Binaural

Uma solução avançada para áudio baseado em ambiência, especialmente otimizada para fones de ouvido e compatível com o sistema MPEG-H *Audio*, é a reprodução binaural. Essa abordagem utiliza Respostas aos Impulso Binaurais de Sala, do inglês *Binaural Room Impulse Responses (BRIR)*, para criar uma experiência sonora espacialmente envolvente.

Ao adotar a renderização binaural, os criadores de conteúdo têm a capacidade de não apenas posicionar as fontes sonoras no espaço, mas também de proporcionar uma sensação imersiva da acústica do ambiente em que a gravação ou reprodução está ocorrendo.

Essas respostas ao impulso não se limitam a simplesmente simular a direção das fontes sonoras; elas vão além, capturando nuances como reflexões, absorções e difrações do som no ambiente. Isso proporciona uma representação detalhada de como o som se propaga tridimensionalmente no espaço, criando uma experiência auditiva mais rica e autêntica.

Além disso, ao utilizar *BRIRs*, é possível modelar características cruciais do ambiente, como a reverberação. A reverberação contribui para a persistência do som após a interrupção da fonte sonora, enriquecendo ainda mais a qualidade e a naturalidade da experiência auditiva.

Dessa forma, a reprodução binaural não apenas posiciona as fontes sonoras de maneira precisa, mas também leva em conta a complexidade acústica do ambiente, proporcionando uma representação sonora tridimensional que eleva a imersão auditiva a um nível mais avançado.

Também é importante destacar que a reprodução binaural pode ser integrada com HOA. A combinação dessas técnicas permite uma reprodução sonora ainda mais precisa e imersiva. Enquanto as *BRIRs* se concentram na representação espacial ao nível dos ouvidos, o HOA complementa essa abordagem ao capturar a distribuição espacial completa do som, incluindo informações sobre direção, amplitude e fase. A utilização conjunta dessas técnicas permite a criação de experiências auditivas altamente realistas, levando em conta não apenas a localização precisa das fontes sonoras em relação ao ouvinte, mas também a complexidade acústica completa do ambiente. Essa sinergia entre binaural e HOA eleva a qualidade e a fidelidade sonora, sendo especialmente benéfica em aplicações que exigem um alto grau de realismo.

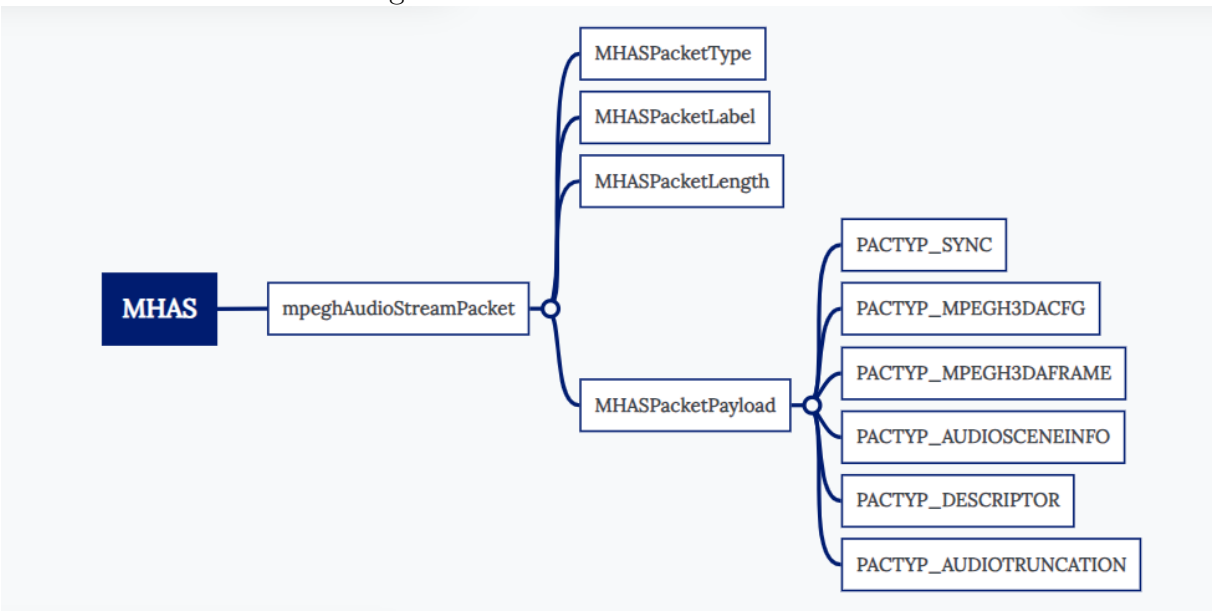
4.2 MPEG-H Audio Stream (MHAS)

Um fluxo MPEG-H *Audio*, do inglês *MPEG-H Audio Stream* (MHAS), é um formato de fluxo de bytes autônomo, flexível e extensível para transportar dados de áudio *MPEG-H* usando uma abordagem de pacotes. Ele precede o encapsulamento em MPEG-2 TS ou ISOBMFF (BLEIDT et al., 2017).

A Figura 15 exemplifica essa estrutura. Um MHAS é dividido em 4 tipos de pacotes: *MHASPacketType*, *MHASPacketLabel*, *MHASPacketLength* e *MHASPacketPayload*. O três primeiros constituem um cabeçalho do fluxo, indicando informações importantes para decodificação do áudio como tipo dos pacotes presentes, tamanho dos pacotes e indicações de agrupamento. Já o *MHASPacketPayload* fornece o conteúdo de áudio e pacotes com informações adicionais sobre esse conteúdo. Os principais pacotes são:

- *PACTYP_SYNC*: Contém uma mensagem fixa para sincronia dos fluxos;
- *PACTYP_MPEGH3DACFG*: Apresenta a estrutura da configuração de áudio;
- *PACTYP_MPEGH3DAFRAME*: Contém o conteúdo dos elementos de áudio presentes no fluxo;
- *PACTYP_AUDIOSCENEINFO*: Contém a estrutura de metadados estáticos com a função de descrever o cenário de áudio presente no conteúdo. É descrito mais afundo na seção 4.3.
- *PACTYP_DESCRIPTOR*: É usado para incorporar descritores MPEG-2 TS no fluxo MPEG-H *Audio*;
- *PACTYP_AUDIOTRUNCATION*: Apresenta informações no caso de um possível truncamento das amostras de áudio. A configurações desse pacote é detalhada na seção 4.5.

Figura 15: Estrutura de um MHAS.



Fonte: Autoria Propria.

Essa estrutura permite que o sistema tenha informações importantes para o gerenciamento do conteúdo como por exemplo, o encapsulamento em multi-stream, referente a divisão do conteúdo de um cenário de áudio em múltiplos arquivos e a possibilidade de mudanças de configurações sem perda.

4.3 Metadados

Como explicado por Herre et al. (2015), os metadados descrevem as propriedades de cada componente de áudio, incluindo parâmetros descritivos, informativos e de controle, criados durante a produção pelo criador de conteúdo. No sistema MPEG-H *Audio*, os metadados são descritos como *Metadata Audio Elements* (MAE), estabelecendo padrões e diretrizes específicas para a utilização dessas informações no contexto do sistema. O conjunto de MAE pode ser definido em quatro tipos, são eles os metadados descritivos, de controle, posicionais e estruturais (ISO/IEC - 23008-3, 2022). Os metadados descritivos informam sobre a presença de elementos de alto nível e suas propriedades. Os metadados de controle contém as interações definidas pela emissora. Os metadados posicionais são relacionados às opções de reprodução do elemento de áudio. Já os metadados estruturais têm a função de realizar o agrupamento e combinação dos elementos de áudio.

No sistema MPEG-H *Audio*, os metadados desempenham um papel crucial na melhoria da experiência do usuário.

Metadados podem ser divididos entre estáticos e dinâmicos. Os estáticos são aqueles que são constantes durante a duração do programa, como, por exemplo, textos utilizados para cada predefinição de áudio ou nos objetos de áudio com que o espectador pode interagir. Já os dinâmicos são aqueles que mudam ao longo do tempo de acordo com a renderização do áudio. Os metadados dinâmicos podem incluir informações em tempo real sobre a posição dos objetos de áudio em um ambiente tridimensional.

Dentro de uma MHAS, os metadados estáticos são definidos na seção de informações de cena de áudio, ou "*AudioSceneInfo*", presente no pacote "*PACTYP_AUDIOSCENEINFO*". Já os metadados dinâmicos são divididos em grupos com conteúdos diferentes, com cada grupo tendo seus elementos de áudio específicos. Também são definidos grupos de troca, na qual reúnem diversos grupos mutualmente exclusivos e *preset*, configuração pré-definida com combinação de grupos e objetos disponíveis. O termo "elemento de áudio" é usado para se referir a cada faixa de áudio acompanhada de metadados específicos para sua renderização.

Utilizando o MPEG-H Decoder - Fraunhofer IIS (2023), é possível extrair informações da "*AudioSceneInfo*" como os idiomas disponíveis e os *presets* definidos. Essa extração gera um arquivo xml com essas informações, como é possível ver nas Figuras 16 e 17.

Figura 16: Extração de informações do *PACTYP_AUDIOSCENEINFO* com o MPEG-H Decoder - Fraunhofer IIS (2023).

Fonte: Adaptado e traduzido de Bleidt et al. (2017).

Figura 17: Extração de informações do *PACTYP_AUDIOSCENEINFO* com o MPEG-H Decoder - Fraunhofer IIS (2023).

```
<audioElementSwitch id="0" isAvailable="true" isActionAllowed="true">
<audioElements>
<audioElement id="1" isActive="true" isDefault="true" isAvailable="true">
<kind table="ContentKindTable" code="10"/>
<customKind langCode="eng">
<description langCode="eng">English</description>
</customKind>
</audioElement>
<audioElement id="2" isActive="false" isDefault="false" isAvailable="true">
<kind table="ContentKindTable" code="10"/>
<customKind langCode="ger">
<description langCode="eng">German</description>
</customKind>
</audioElement>
<audioElement id="3" isActive="false" isDefault="false" isAvailable="true">
<kind table="ContentKindTable" code="10"/>
<customKind langCode="fre">
<description langCode="eng">French</description>
</customKind>
</audioElement>
<audioElement id="4" isActive="false" isDefault="false" isAvailable="true">
<kind table="ContentKindTable" code="10"/>
<customKind langCode="ita">
<description langCode="eng">Italian</description>
</customKind>
</audioElement>
<audioElement id="5" isActive="false" isDefault="false" isAvailable="true">
<kind table="ContentKindTable" code="10"/>
<customKind langCode="rus">
<description langCode="eng">Russian</description>
</customKind>
</audioElement>
</audioElements>
<customKind>
<description langCode="eng">Language</description>
</customKind>
</audioElementSwitch>
</AudioSceneConfig>
```

Fonte: Adaptado e traduzido de Bleidt et al. (2017).

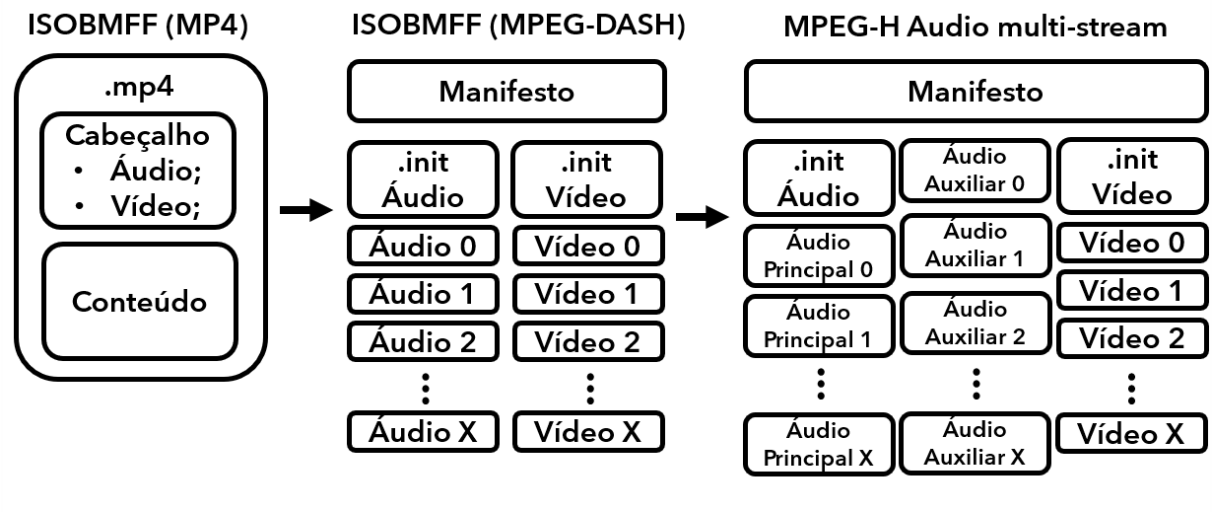
4.4 Multi-stream

A função de entrega *multi-stream* revela-se altamente vantajosa para a transmissão e recepção de áudio nas próximas gerações, uma vez que possibilita a integração entre os domínios *broadcast* e *broadband*. Um MHAS pode ser codificado em arquivos ISOBMFF como *single* ou *multi-stream*.

Como *single-stream*, o conteúdo do áudio e seus metadados serão encapsulados como um único fluxo MHAS. Já como *multi-stream*, os elementos e metadados que formam

uma cena de áudio são distribuídos em múltiplos MHAS. O MHAS principal deve conter a conteúdo principal e a descrição de todos os MHAS auxiliares. A Figura 18 ilustra a diferença entre formatos ISOBMFF *MPEG-4 Part 14* (MP4), DASH com áudio single-stream e DASH com MPEG-H *Audio* multi-stream.

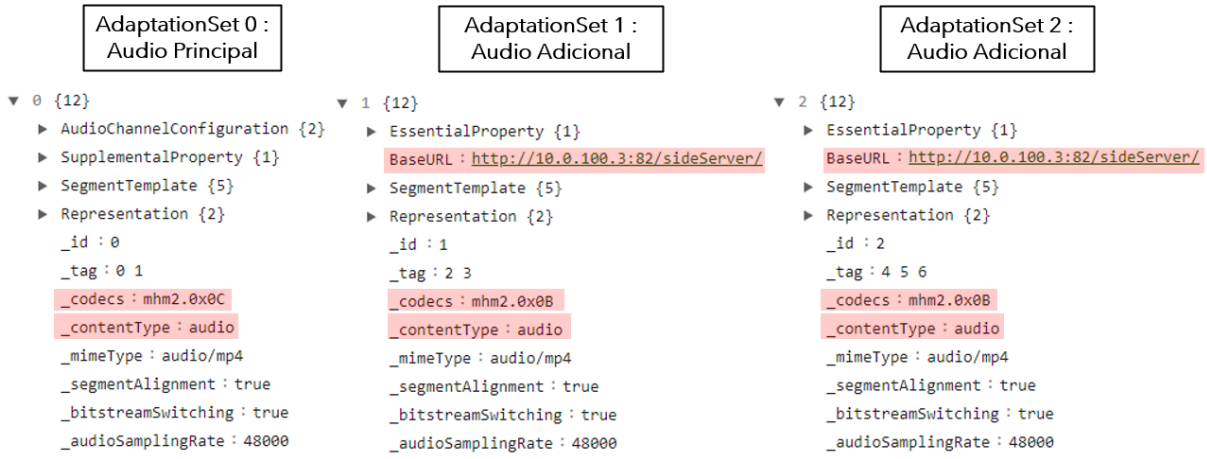
Figura 18: Diferença de entrutura entre arquivos MP4, DASH com single stream e DASH com MPEG-H *Audio* multi-stream.



Fonte: Autoria Própria (2024).

Em relação as sinalizações, o MPEG-H *Audio* quando codificado como single-stream é sinalizado como "mhm1". Já quando codificado como multi-stream, é sinalizado como "mhm2", indicando que seu conteúdo foi dividido em múltiplos fluxos. A Figura 19 demonstra como os arquivos multi-stream são declarados em um arquivo manifesto.

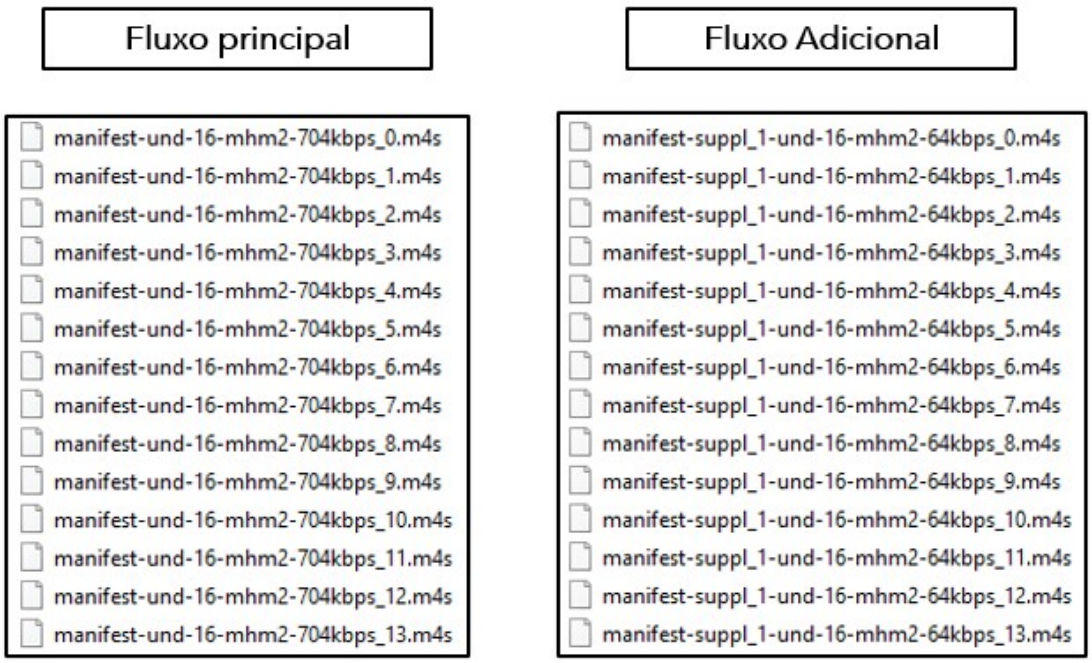
Figura 19: Exemplo de declaração de *multi-stream* no arquivo manifesto.



Fonte: Autoria Própria (2024).

Os segmentos DASH de áudio são gerados de forma separada de acordo com seu conteúdo. A Figura 20 exemplifica essa separação, onde os arquivos do lado esquerdo contém os elementos de áudio com o som ambiente e um idioma principal, e os arquivos da direita contém elementos de áudio de um idioma adicional.

Figura 20: Exemplo de codificação DASH com *multi-stream*.

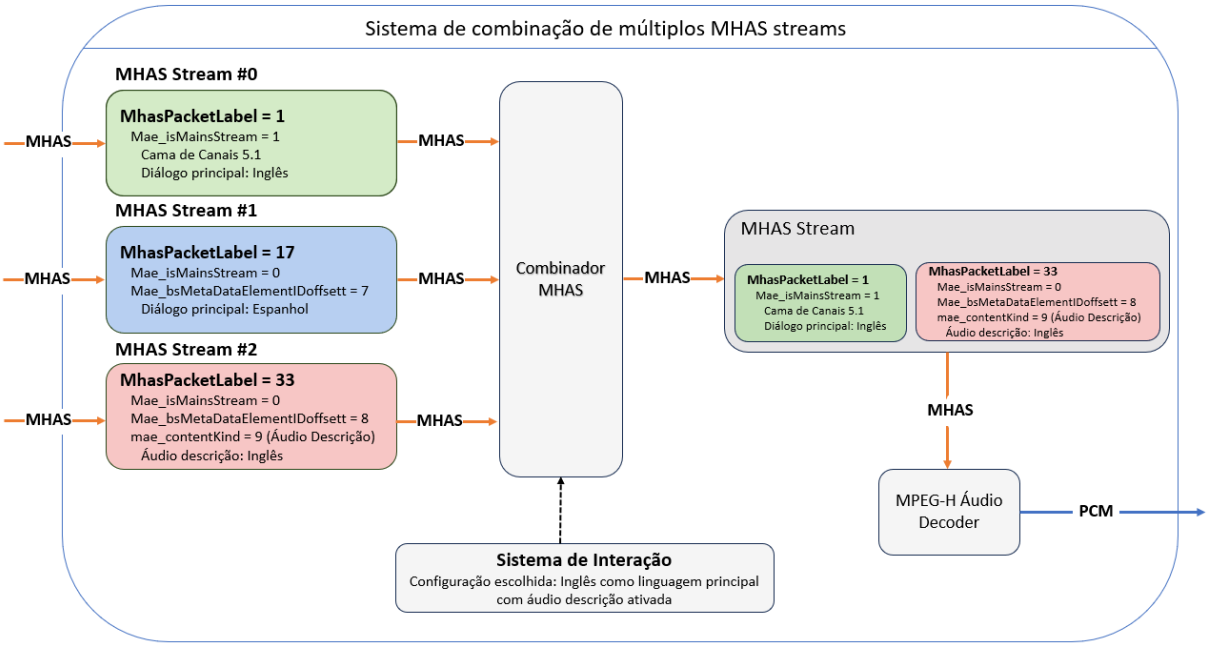


Fonte: Autoria Própria (2024).

Dessa forma, o receptor consegue realizar a gestão dos pacotes independente da forma de transmissão, OTA ou OTT. Dentre as diversas opções de MHAS recebidos, o *decoder* deve unificar as informações em um MHAS final de acordo com a escolha do usuário. Ambos os fluxos, principal e auxiliares, são criados no mesmo codificador totalmente alinhados e sem defasagem, permitindo um controle mais eficaz na sincronização do conteúdo principal e auxiliar.

O MHAS principal deve ser sinalizado nos metadados pela variável *"mae_isMainStream"* igual a 1 e consequentemente, todas as outras MHAS adicionais, iguais a 0. A escolha do usuário é relacionada com a variável *"mae_bsMetaDataElementIDoffset"*, no qual cada MHAS adicional tem um valor específico da variável. Quando selecionado, o MHAS em questão é unificado com o principal em um único fluxo contendo o serviço de áudio escolhido. Esse processo de unificação pode ser observado na Figura 21.

Figura 21: Sistema de junção de MHAS na função *multi-stream*.



Fonte: Adaptado e traduzido de ATSC - A/342-3 (2023).

4.5 Mecanismos de alinhamento para Pontos de Acesso

Para que a transmissão e recepção do sinal ocorra sempre sincronizada e sem perdas, o MPEG-H *Audio* trabalha com a conceito de RAP. Os RAPs, já presentes no MPEG-2 TS,

são importantes pois são nesses momentos em que ocorrem as mudanças de configuração. No começo de cada fragmento ISOBMFF começa com um RAP. Cada RAP consiste na transmissão de um pacote PACTYP_MPEGH3DACFG, *PACTYP_AUDIOSCENEINFO* e *PACTYP_MPEGH3DAFRAME*.

Neste caso, o pacote "*PACTYP_MPEGH3DAFRAME*" do RAP deve conter informações do último quadro de áudio. Esse sistema é chamado de Quadro de reprodução imediata ou, do inglês *Immediate Playout Frame*. Isso possibilita a troca de configurações de áudio sem perda do conteúdo exibido. A existência desse conteúdo é indicada pela estrutura de *AudioPreRoll* no pacote *PACTYP_MPEGH3DACFG*.

Outro sistema importante que acontece em um RAP é o mecanismo de truncamento de áudio. Esse sistema consiste na utilização do pacote "*PACTYP_AUDIOTRUNCATION*" após o ultimo quadro do conteúdo. Neste pacote, é possível a identificação de amostras que o receptor pode descartar para alinhar áudio e vídeo. O descarte pode ocorrer no começo ou no final do fluxo. Caso a variável *truncFromBegin* seja igual a 1, o descarte acontece no inicio. Esse fluxo deve conter um pacote *PACTYP_MPEGH3DACFG* adicional e o descarte deve ser realizado entre esse pacote e o próximo *PACTYP_MPEGH3DAFRAME*. Caso a variável *truncFromBegin* seja igual a 0, o descarte acontece no final, sendo realizado antes do próximo *PACTYP_MPEGH3DAFRAME* presente.

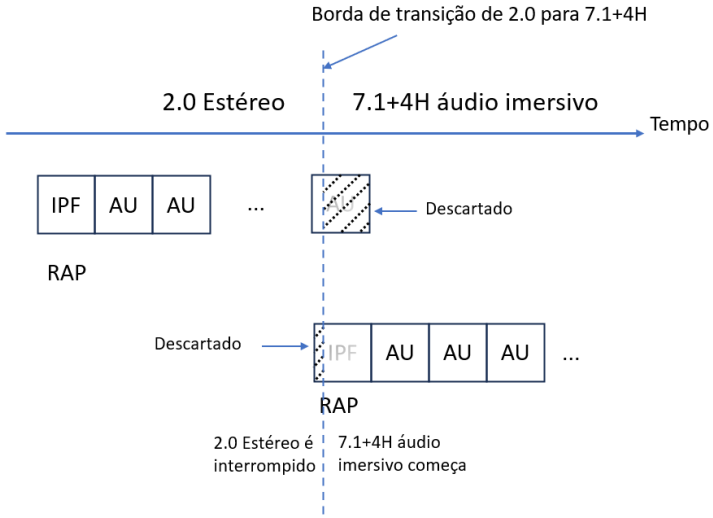
Dessa forma, a realização do descarte de parte das amostras permite realizar o alinhamento com os pacotes de vídeo, fornecendo uma melhor transição entre conteúdos.

Existem alguns cenários que esses sistemas são necessários. Por exemplo, quando ocorre a troca de conteúdo de um programa para uma propaganda é necessário o truncamento dos pacotes a fim de alinhar áudio e vídeo novamente. Isso acontece, pois o número de quadros de áudio e vídeo não são os mesmos. Enquanto o MPEG-H *Audio* usa 1024 amostras por quadro com 48 khz de frequência de amostragem, é comum os *codecs* de vídeo usarem 50 ou 59,97 quadros por segundo.

Isso também ocorre na troca entre configurações de áudio. No Figura 22 é possível analisar a comutação entre um canal de áudio estéreo 2.0 e um canal de áudio 7.1+4H imersivo. São descartados os últimos pacotes da unidade de acesso, do inglês *Access Unit* (AU), do canal estéreo 2.0 e os pacotes iniciais do canal 7.1+4H. Dessa forma, é obtida uma amostra de áudio completa sem perdas com uma transição suave sem interferência

no conteúdo do usuário.

Figura 22: Exemplo de comutação de configurações de áudio.



Fonte: Adaptado e traduzido de Bleidt et al. (2017).

4.6 Perfis de Complexidade

O MPEG-H *Audio* realiza a entrega dos dados de acordo com a estimativa de complexidade de processamento do decodificador por meio de perfis de complexidade. Cada perfil contém níveis de complexidade diferentes, apresentando mais ou menos recursos. Existem 3 perfis de complexidade: perfil elevado, perfil de baixa complexidade e perfil básico.

O perfil elevado contém todas as funcionalidades disponíveis para baixa e alta taxa de *bits*. Ele permite renderização em todos os cenários de reprodução. Esse perfil é um superconjunto do perfil de baixa complexidade. Já o perfil de baixa complexidade fornece funcionalidades para radiodifusão e *streaming* com complexidade de decodificação reduzida. Por fim, o perfil básico é um subconjunto do perfil de baixa complexidade, oferecendo suporte a formato baseado em canais e em objetos (ISO/IEC - 23008-3, 2022).

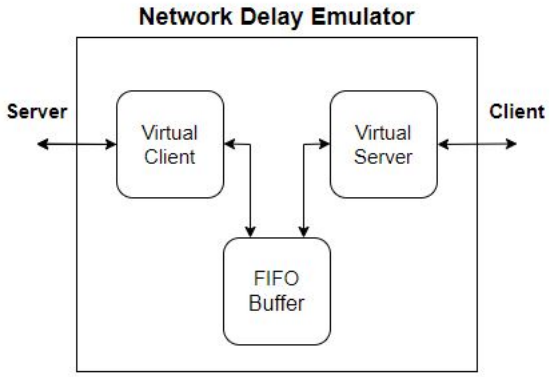
5 TESTES E RESULTADOS

Para a realização dos testes, foram necessários modificações, desenvolvimentos e estudos de algumas ferramentas essenciais como o emulador de latência em rede, e a forma como o emulador realiza a comunicação com o *player* e com o *Webserver*, entre outros.

5.1 Emulador de Latência em Rede

O emulador de latência em rede é um código feito na linguagem C++, elaborado com o objetivo de atrasar pacotes em rede. A Figura 23 demonstra o modo de operação do emulador. A comunicação é realizada por meio de um cliente virtual e um servidor virtual. Quando o cliente envia uma solicitação de comunicação, o servidor virtual recebe essa comunicação, armazena em um *buffer* com sistema *First In First Out* (FIFO). Esse *buffer* armazena os pacotes no tempo pré determinado e em seguida, libera para o cliente virtual se comunicar com o servidor. Esse processo é válido para as duas vias, cliente-servidor ou servidor-cliente.

Figura 23: Configuração interna do emulador de latência em rede.



Fonte: (JUNIOR et al., 2024).

O emulador tem três tipos possíveis de atraso que podem ser utilizados: o fixo, o uniforme aleatório e o gaussiano. O atraso fixo retarda os pacotes com uma distribuição com tempo fixo. Se, por exemplo, o valor de latência inserido for igual a 1 segundo, o emulador irá liberar os pacotes após 1 segundo. Já no atraso uniforme aleatório, é inserido um valor de máximo e mínimo de retardo e o emulador libera os pacotes em um valor de

latência aleatória dentro do intervalo estipulado. O perfil gaussiano por outro lado utiliza os conceitos do método *Box-Muller* para realizar uma distribuição de tempo gaussiana para liberar os pacotes (GHAZEL et al., 2001). A amplitude da variação gaussiana é definida por metade do valor máximo (max) dividido pela raiz quadrada de 2, como mostra a Equação 1.

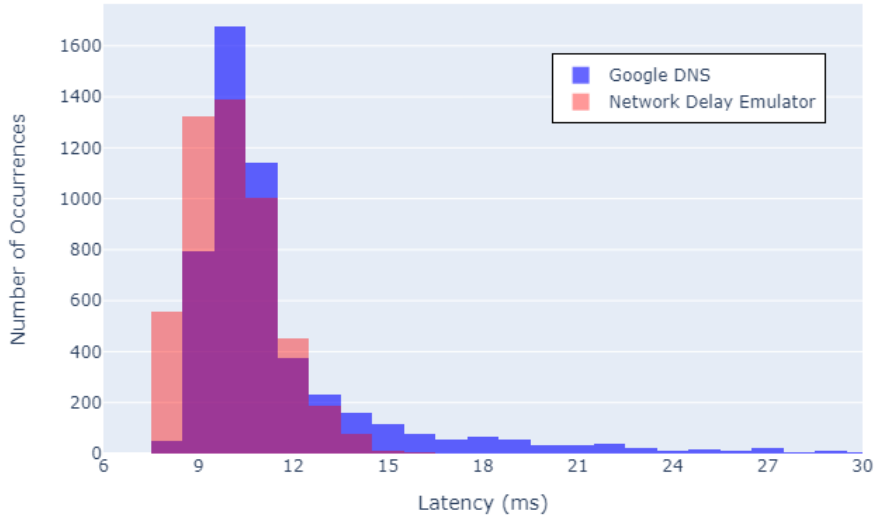
$$\text{amplitude} = \frac{\text{max}}{2\sqrt{2}} \tag{1}$$

O atraso é calculado multiplicando o valor de amplitude calculado na Equação 1 com um valor Y, onde Y é uma variável aleatória com uma distribuição gaussiana com média zero e um desvio padrão de 1. O resultado desta multiplicação é adicionado ao valor mínimo introduzido. A Equação 2 demonstra este cálculo.

$$\text{atraso} = \text{min} + |\text{Y} \times \text{amplitude}| \tag{2}$$

Na Figura 24 mostra um comparativo entre o tempo de latência de um *ping* a um servidor *Domain Name System* (DNS) do Google (8.8.8.8) e o perfil de atraso gaussiano do emulador. Foram geradas 5000 amostras para cada Cenário descrito.

Figura 24: Histograma comparando *ping* ao servidor DNS do Google e o perfil gaussiano do emulador.



Fonte: (JUNIOR et al., 2024).

Os testes foram realizados considerando o perfil gaussiano para representar o atraso em rede.

Além do histograma, é possível calcular o *jitter* médio do emulador de atraso com as 5000 amostras coletadas. O *jitter* é uma medida de variabilidade na latência de chegada dos pacotes. Para realizar essa medida, é necessário calcular a diferença de latência entre pacotes consecutivos. Isso é realizado calculando "Dif", onde "Dif" é a diferença de latência entre o pacote i e o pacote i+1, indicado na equação :

$$\text{Dif} = Latência_{i+1} - Latência_i \tag{3}$$

Em seguida, calcula-se o *jitter* realizando o somatório dos valores de diferença de latência dos pacotes em módulo, dividindo por "n-1", onde n é o número total de pacotes.

$$Jitter = \frac{\sum_{i=1}^{n-1} |Dif|}{n-1} \tag{4}$$

Realizando esta conta para as 5000 amostras com uma latência de 8,3 ms, o valor de *jitter* médio calculado é igual a 1,5 ms, possuindo um *jitter* de 13,6% do valor da latência. O mesmo calculo foi realizado para as 5000 amostras do servidor da Google e o *jitter* resultante foi de 1,31 ms, possuindo um *jitter* de 12% do valor da latência.

5.2 Comunicação em rede

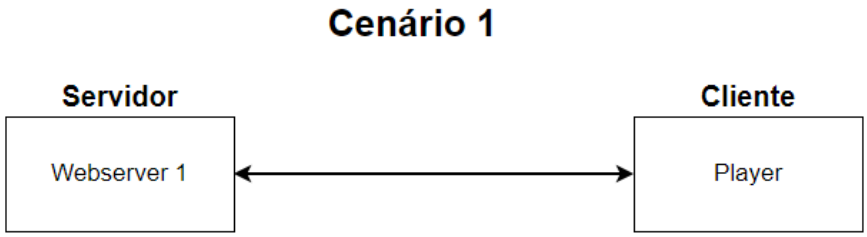
A comunicação em redes de computadores envolve a utilização de diferentes protocolos, cada um com suas funções e camadas de operação específicas. Dois dos principais protocolos utilizados são o TCP e o HTTP. O TCP opera na camada de transporte do modelo Interconexão de Sistemas Abertos, do inglês *Open Systems Interconnection* (OSI), sendo responsável por estabelecer e manter a conexão entre os sistemas, conforme definido por Postel (1981). Já o HTTP opera na camada de aplicação, sendo responsável pela formatação e transmissão de dados entre clientes e servidores na web, conforme especificado por Nielsen et al. (1999).

Neste trabalho, a comunicação em rede aconteceu em dois cenários diferentes. Em

ambos os cenários, a comunicação é proposta pela aplicação do player cedida pela Fraunhofer IIS. Os *Webservers* e o emulador de latência respondem as requisições propostas pelo player.

No primeiro cenário, é realizada a comunicação entre o *Webserver* 1, um servidor *Web-based Distributed Authoring and Versioning* (WebDAV) configurado com os arquivos DASH principais de áudio, vídeo e o manifesto, e o *player*. Este Cenário representa a transmissão OTA e é apresentado na Figura 25.

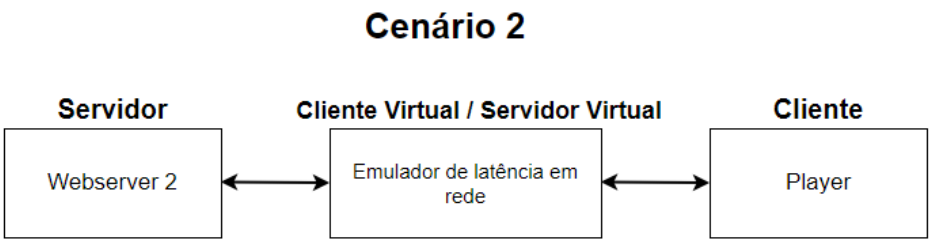
Figura 25: Primeiro Cenário de comunicação representando a transmissão *broadcast*.



Fonte: Autoria Própria (2024).

Já o segundo Cenário representa a comunicação OTT. Esse Cenário consiste na comunicação entre o *Webserver* 2, um servidor WebDAV configurado com os arquivos DASH com conteúdo MPEG-H *Audio* auxiliares e o *player*, porém, com a intermediação do emulador de latência entre eles. Esse Cenário é ilustrado na Figura 26.

Figura 26: Segundo Cenário de comunicação representando a transmissão *broadband*.



Fonte: Autoria Própria (2024).

A comunicação no Cenário 1 é iniciada com o estabelecimento de uma conexão

confiável usando o TCP, seguido pela troca de dados por meio do HTTP. Esta sequência garante que os pacotes de dados sejam entregues corretamente e que a comunicação ocorra de maneira estruturada e eficiente.

O processo começa com um *handshake* de três vias, do inglês "*three-way handshake*". Esse processo é crucial para a sincronização entre o sistema cliente-servidor, a fim de estabelecer uma conexão confiável e segura entre eles. O processo ocorre em três etapas:

1. SYN: O cliente envia um segmento SYN (*synchronize*) para o servidor para demonstrar o interesse em estabelecer uma conexão;
2. SYN-ACK: O servidor responde com um segmento SYN-ACK (*synchronize-acknowledge*), indicando que recebeu a mensagem do cliente e está disposto a sincronizar;
3. ACK: O cliente envia um segmento ACK (*acknowledge*) de volta ao servidor, confirmando a recepção da mensagem SYN-ACK do servidor.

Este processo de *handshake* de três vias utiliza o TCP, conforme definido por Postel (1981), que é um protocolo de transporte responsável por garantir a entrega confiável de pacotes entre o cliente e o servidor. O TCP é escolhido para esta etapa porque ele oferece mecanismos de controle de fluxo, detecção de erros e retransmissão de pacotes perdidos, essenciais para estabelecer uma conexão estável.

Após estabelecer a comunicação por meio do *handshake* de três vias, a troca de dados ocorre utilizando o HTTP. O HTTP é um protocolo de aplicação utilizado para a transferência de documentos da *web* e outros recursos, conforme especificado por Nielsen et al. (1999). A requisição "HEAD" solicita que o servidor envie apenas as linhas de cabeçalho do conteúdo indicado, sem o corpo da mensagem.

Quando o servidor recebe essa requisição, ele responde com uma mensagem contendo o cabeçalho HTTP, incluindo um código de *status*, como o 200 (OK), que indica que a requisição foi bem-sucedida.

Após a requisição HTTP inicial e a resposta do servidor, ocorre o encerramento da conexão inicial entre o cliente e o servidor. Esse encerramento é realizado por meio de uma troca de segmentos "FIN/ACK". Primeiro, o cliente envia um segmento "FIN/ACK"

ao servidor para iniciar o processo de encerramento da conexão. Em resposta, o servidor envia seu próprio segmento "FIN/ACK", completando o encerramento da conexão. Esse procedimento garante que ambos os lados reconheçam que a comunicação atual foi finalizada corretamente.

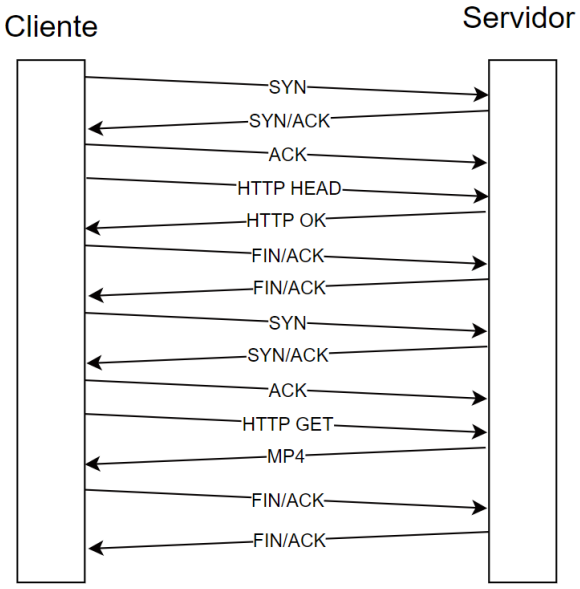
Com a conexão inicial encerrada, o cliente e o servidor estabelecem uma nova conexão iniciada pelo cliente. Este novo estabelecimento de conexão segue novamente o processo de *handshake* de três vias.

Na sequência, o cliente envia uma nova requisição HTTP, desta vez utilizando o método "GET" para solicitar um arquivo MP4 do servidor. O servidor responde a essa requisição enviando o conteúdo do arquivo MP4 solicitado.

Por fim, após a transferência do arquivo MP4, a conexão entre o cliente e o servidor é encerrada novamente a partir da troca de segmentos "FIN/ACK". O cliente inicia o encerramento enviando um segmento "FIN/ACK", e o servidor responde com seu próprio segmento "FIN/ACK".

Esse processo de comunicação é realizado repetidas vezes a fim de continuar o envio dos segmentos para o cliente. A Figura 27 ilustra esta comunicação por meio de um diagrama.

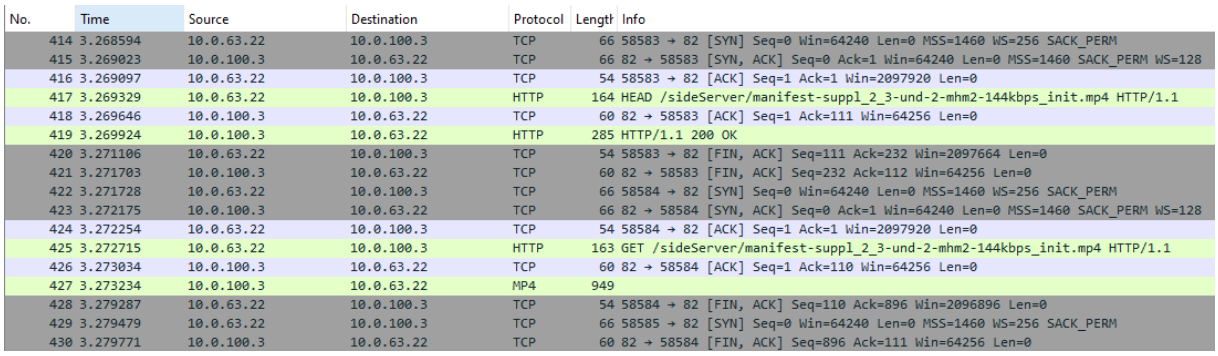
Figura 27: Diagrama de comunicação entre Cenário 1.



Fonte: Autoria Própria (2024).

Esse fluxo foi analisado através do *software* de analise de fluxo de rede *Wireshark* e é apresentado na Figura 28.

Figura 28: Analise do fluxo de rede da comunicação do Cenário 1.



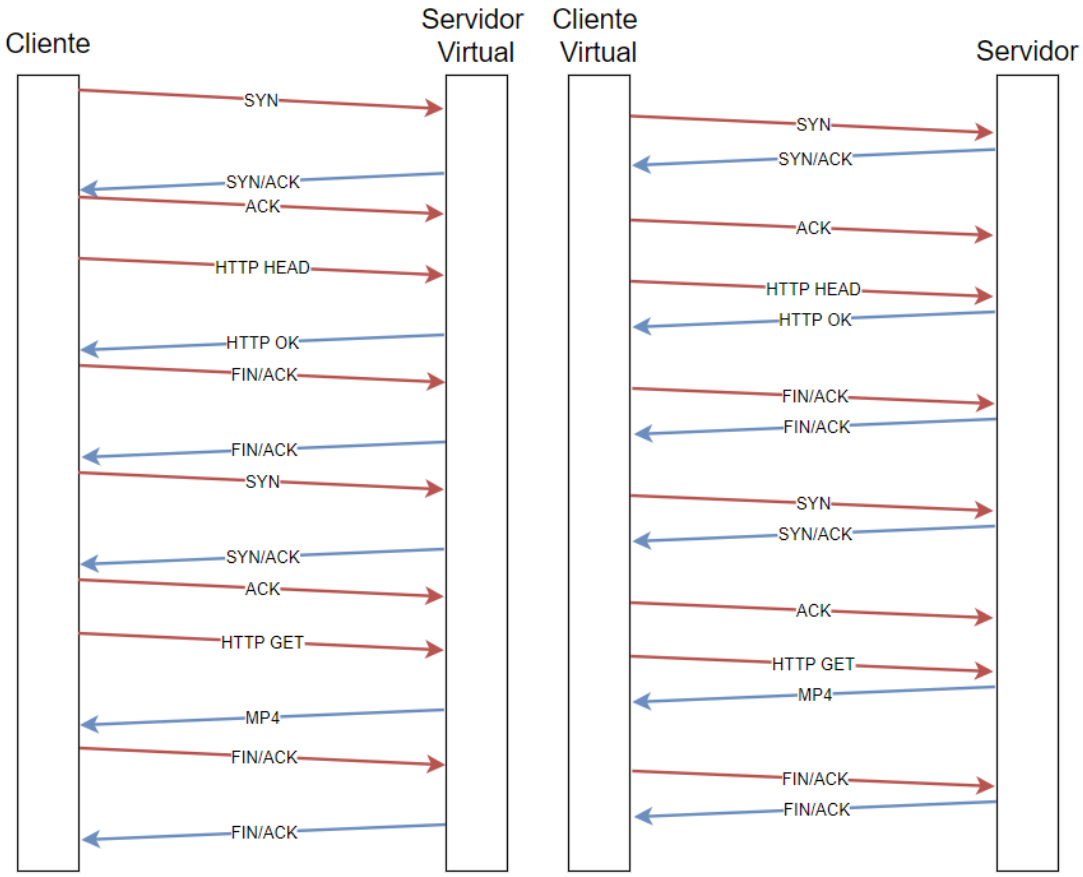
No.	Time	Source	Destination	Protocol	Length	Info
414	3.268594	10.0.63.22	10.0.100.3	TCP	66	58583 → 82 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
415	3.269023	10.0.100.3	10.0.63.22	TCP	66	82 → 58583 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
416	3.269097	10.0.63.22	10.0.100.3	TCP	54	58583 → 82 [ACK] Seq=1 Ack=1 Win=2097920 Len=0
417	3.269329	10.0.63.22	10.0.100.3	HTTP	164	HEAD /sideServer/manifest-suppl_2_3-und-2-mhm2-144kbps_init.mp4 HTTP/1.1
418	3.269646	10.0.100.3	10.0.63.22	TCP	60	82 → 58583 [ACK] Seq=1 Ack=111 Win=64256 Len=0
419	3.269924	10.0.100.3	10.0.63.22	HTTP	285	HTTP/1.1 200 OK
420	3.271106	10.0.63.22	10.0.100.3	TCP	54	58583 → 82 [FIN, ACK] Seq=111 Ack=232 Win=2097664 Len=0
421	3.271703	10.0.100.3	10.0.63.22	TCP	60	82 → 58583 [FIN, ACK] Seq=232 Ack=112 Win=64256 Len=0
422	3.271728	10.0.63.22	10.0.100.3	TCP	66	58584 → 82 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
423	3.272175	10.0.100.3	10.0.63.22	TCP	66	82 → 58584 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
424	3.272254	10.0.63.22	10.0.100.3	TCP	54	58584 → 82 [ACK] Seq=1 Ack=1 Win=2097920 Len=0
425	3.272715	10.0.63.22	10.0.100.3	HTTP	163	GET /sideServer/manifest-suppl_2_3-und-2-mhm2-144kbps_init.mp4 HTTP/1.1
426	3.273034	10.0.100.3	10.0.63.22	TCP	60	82 → 58584 [ACK] Seq=1 Ack=110 Win=64256 Len=0
427	3.273234	10.0.100.3	10.0.63.22	MP4	949	
428	3.279287	10.0.63.22	10.0.100.3	TCP	54	58584 → 82 [FIN, ACK] Seq=110 Ack=896 Win=2096896 Len=0
429	3.279479	10.0.63.22	10.0.100.3	TCP	66	58585 → 82 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
430	3.279771	10.0.100.3	10.0.63.22	TCP	60	82 → 58584 [FIN, ACK] Seq=896 Ack=111 Win=64256 Len=0

Fonte: (JUNIOR et al., 2024).

A comunicação do Cenário 2 ocorre de forma semelhante ao Cenário 1. Toda a comunicação entre *player* e servidor virtual do emulador é espelhada para o cliente virtual do emulador e o *Webserver* 2. Dessa forma, evidencia-se que a influencia do emulador de latência é aplicada apenas no atraso, mantendo a ordem de comunicação que seria estabelecida entre cliente-servidor.

Essa comunicação é ilustrada no diagrama da Figura 29. As setas vermelhas do diagrama indicam a comunicação enviada pelo *player* e recebidas pelo *Webserver* 2. Já as setas azuis indicam as respostas enviadas do *Webserver* 2 e recebidas pelo *player*.

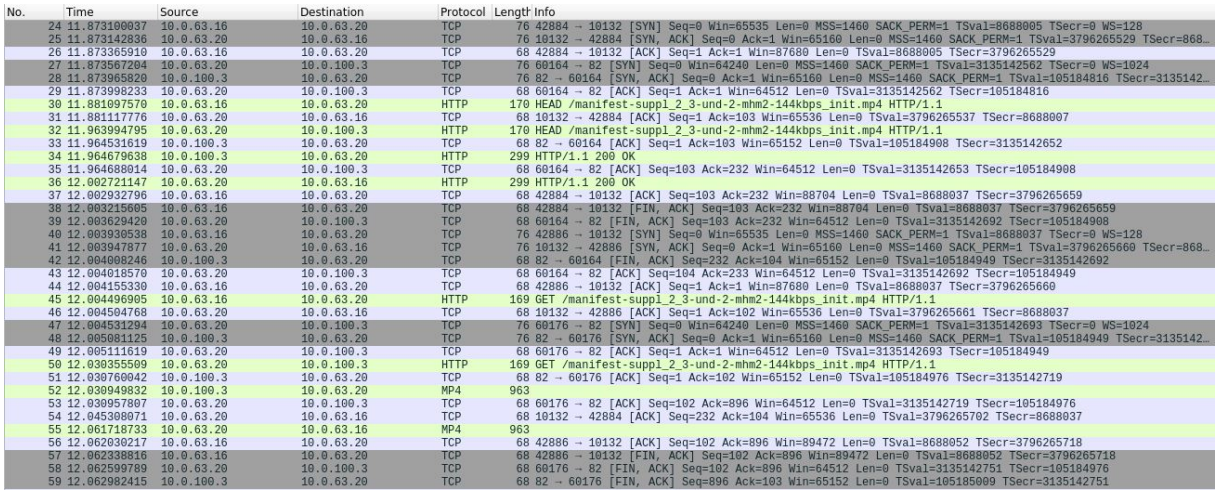
Figura 29: Diagrama de comunicação do Cenário 2.



Fonte: Autoria Própria (2024).

Esse fluxo foi analisado usando o *software* de análise de fluxo de rede *Wireshark* e é apresentado na Figura 30.

Figura 30: Análise do fluxo de rede da comunicação do Cenário 2.



No.	Time	Source	Destination	Protocol	Length	Info
24	11.873100037	10.0.63.16	10.0.63.20	TCP	76	42884 → 10132 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1 TSval=8688005 TSecr=0 WS=128
25	11.873142836	10.0.63.20	10.0.63.16	TCP	76	10132 → 42884 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=3796265529 TSecr=868…
26	11.873365910	10.0.63.16	10.0.63.20	TCP	68	42884 → 10132 [ACK] Seq=1 Ack=1 Win=87680 Len=0 TSval=8688005 TSecr=3796265529
27	11.873567204	10.0.63.20	10.0.100.3	TCP	76	60164 → 82 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3135142562 TSecr=0 WS=1024
28	11.873965820	10.0.100.3	10.0.63.20	TCP	76	82 → 60164 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=105184816 TSecr=3135142…
29	11.873998233	10.0.63.20	10.0.100.3	TCP	68	60164 → 82 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=3135142562 TSecr=105184816
30	11.881097570	10.0.63.16	10.0.63.20	HTTP	170	HEAD /manifest-suppl_2_3-und-2-mhm2-144kbps_init.mp4 HTTP/1.1
31	11.881117776	10.0.63.20	10.0.63.16	TCP	68	10132 → 42884 [ACK] Seq=1 Ack=103 Win=65536 Len=0 TSval=3796265537 TSecr=8688007
32	11.963994795	10.0.63.20	10.0.100.3	HTTP	170	HEAD /manifest-suppl_2_3-und-2-mhm2-144kbps_init.mp4 HTTP/1.1
33	11.964531619	10.0.100.3	10.0.63.20	TCP	68	82 → 60164 [ACK] Seq=1 Ack=103 Win=65152 Len=0 TSval=105184908 TSecr=3135142652
34	11.964679638	10.0.100.3	10.0.63.20	HTTP	299	HTTP/1.1 200 OK
35	11.964688014	10.0.63.20	10.0.100.3	TCP	68	60164 → 82 [ACK] Seq=103 Ack=232 Win=64512 Len=0 TSval=3135142653 TSecr=105184908
36	12.002721147	10.0.63.20	10.0.63.16	HTTP	299	HTTP/1.1 200 OK
37	12.002932796	10.0.63.16	10.0.63.20	TCP	68	42884 → 10132 [ACK] Seq=103 Ack=232 Win=88704 Len=0 TSval=8688037 TSecr=3796265659
38	12.003215605	10.0.63.16	10.0.63.20	TCP	68	42884 → 10132 [FIN, ACK] Seq=103 Ack=232 Win=88704 Len=0 TSval=8688037 TSecr=3796265659
39	12.003629420	10.0.63.20	10.0.100.3	TCP	68	60164 → 82 [FIN, ACK] Seq=103 Ack=232 Win=64512 Len=0 TSval=3135142692 TSecr=105184908
40	12.003930538	10.0.63.16	10.0.63.20	TCP	76	42886 → 10132 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1 TSval=8688037 TSecr=0 WS=128
41	12.003947877	10.0.63.20	10.0.63.16	TCP	76	10132 → 42886 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=3796265660 TSecr=868…
42	12.004008246	10.0.100.3	10.0.63.20	TCP	68	82 → 60164 [FIN, ACK] Seq=232 Ack=104 Win=65152 Len=0 TSval=105184949 TSecr=3135142692
43	12.004018570	10.0.63.20	10.0.100.3	TCP	68	60164 → 82 [ACK] Seq=104 Ack=233 Win=64512 Len=0 TSval=3135142692 TSecr=105184949
44	12.004155330	10.0.63.16	10.0.63.20	TCP	68	42886 → 10132 [ACK] Seq=1 Ack=1 Win=87680 Len=0 TSval=8688037 TSecr=3796265660
45	12.004496905	10.0.63.16	10.0.63.20	HTTP	169	GET /manifest-suppl_2_3-und-2-mhm2-144kbps_init.mp4 HTTP/1.1
46	12.004504768	10.0.63.20	10.0.63.16	TCP	68	10132 → 42886 [ACK] Seq=1 Ack=102 Win=65536 Len=0 TSval=3796265661 TSecr=8688037
47	12.004531294	10.0.63.20	10.0.100.3	TCP	76	60176 → 82 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3135142693 TSecr=0 WS=1024
48	12.005081125	10.0.100.3	10.0.63.20	TCP	76	82 → 60176 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=105184949 TSecr=3135142…
49	12.005111619	10.0.63.20	10.0.100.3	TCP	68	60176 → 82 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=3135142693 TSecr=105184949
50	12.030355509	10.0.63.20	10.0.100.3	HTTP	169	GET /manifest-suppl_2_3-und-2-mhm2-144kbps_init.mp4 HTTP/1.1
51	12.030760042	10.0.100.3	10.0.63.20	TCP	68	82 → 60176 [ACK] Seq=1 Ack=102 Win=65152 Len=0 TSval=105184976 TSecr=3135142719
52	12.030949832	10.0.100.3	10.0.63.20	MP4	963	
53	12.030957807	10.0.63.20	10.0.100.3	TCP	68	60176 → 82 [ACK] Seq=102 Ack=896 Win=64512 Len=0 TSval=3135142719 TSecr=105184976
54	12.045308071	10.0.63.20	10.0.63.16	TCP	68	10132 → 42884 [ACK] Seq=232 Ack=104 Win=65536 Len=0 TSval=3796265702 TSecr=8688037
55	12.061718733	10.0.63.20	10.0.63.16	MP4	963	
56	12.062030217	10.0.63.16	10.0.63.20	TCP	68	42886 → 10132 [ACK] Seq=102 Ack=896 Win=89472 Len=0 TSval=8688052 TSecr=3796265718
57	12.062338816	10.0.63.16	10.0.63.20	TCP	68	42886 → 10132 [FIN, ACK] Seq=102 Ack=896 Win=89472 Len=0 TSval=8688052 TSecr=3796265718
58	12.062599789	10.0.63.20	10.0.100.3	TCP	68	60176 → 82 [FIN, ACK] Seq=102 Ack=896 Win=64512 Len=0 TSval=3135142751 TSecr=105184976
59	12.062982415	10.0.100.3	10.0.63.20	TCP	68	82 → 60176 [FIN, ACK] Seq=896 Ack=103 Win=65152 Len=0 TSval=105185009 TSecr=3135142751

Fonte: (JUNIOR et al., 2024).

5.3 Modificação do Arquivo Manifesto

Para a realização dos testes, foi necessária a configuração do arquivo manifesto no modo dinâmico, considerando que o escopo do projeto prevê a decodificação de conteúdos transmitidos em tempo real. Esse modo garante que o *player* decodifique os segmentos DASH a partir de uma variável de tempo, ao contrário do modo estático. No modo estático o *player* decodifica o conteúdo a partir do segmento indicado no atributo "*StartNumber*".

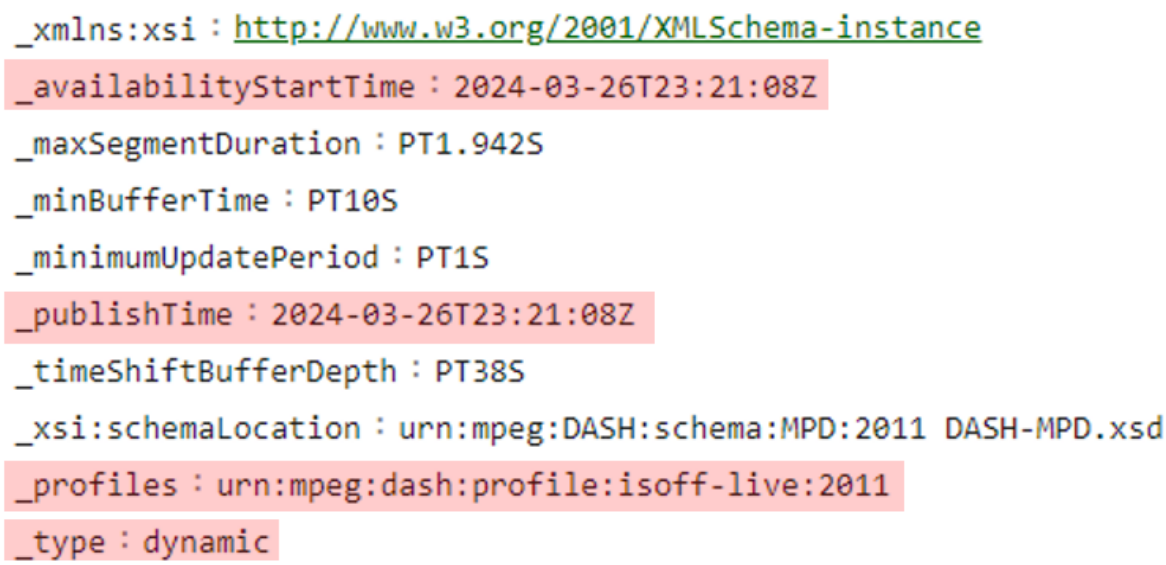
Tendo em vista que os arquivos fornecidos estavam inicialmente no modo estático, foram necessárias alterações de alguns atributos do cabeçalho do manifesto. O principal foi o atributos "mode", que indica o modo de solicitação dos segmentos, de estático para dinâmico.

Em seguida, foi necessária a adição de dois atributos importantes para a reprodução do conteúdo no modo dinâmico: o "*AvailabilityStartTime*" e o "*PublishTime*". O "*AvailabilityStartTime*" especifica o momento inicial que os segmentos estarão disponíveis. Já o "*PublishTime*" indica o momento que o arquivo manifesto foi criado.

Outra modificação realizada foi a remoção do atributos "*MediaPresentationDuration*". Esse atributos indica o tamanho total da mídia transmitida. No caso de uma transmissão

em tempo real, esse valor não é definido. A Figura 31 ilustra a configuração do cabeçalho do arquivo manifesto após as modificações realizadas.

Figura 31: Cabeçalho do arquivo manifesto após as modificações.



Fonte: Autoria Própria (2024).

Além das modificações mencionadas, foi desenvolvido um código na linguagem *Python* para a atualização do campo "*AvailabilityStartTime*" de forma automática. Este código realiza a modificação do arquivo XML e envia o arquivo editado para o *Webserver* 1.

Dessa forma, é possível atualizar o valor do atributos "*AvailabilityStartTime*" e consequentemente, a âncora de início de solicitação dos segmentos DASH. Foi necessário realizar o desenvolvimento supracitado para permitir que um conteúdo, configurado para DASH dinâmico, pudesse ser transmitido diversas vezes.

5.4 Configuração de Testes

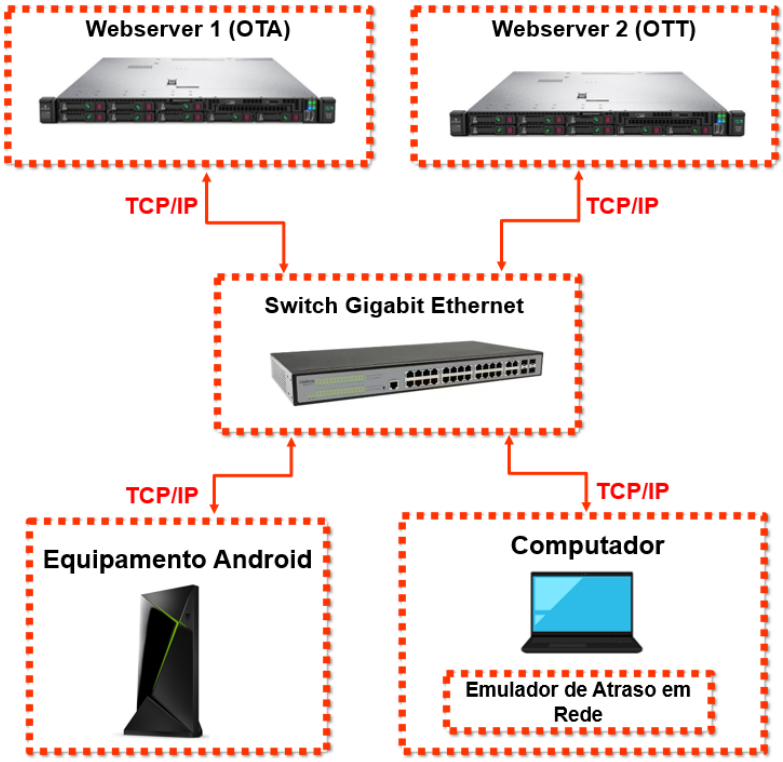
A Figura 32 ilustra a configuração utilizada para a realização dos testes. O *Webserver* 1 simula os arquivos recebidos e demodulados via OTA. Para os testes, foi considerado o momento seguinte à demultiplexação dos arquivos recebidos, tendo os segmentos DASH com Codificação de Vídeo de Alta Eficiência, do inglês *High-Efficiency Video Coding*

(HEVC), os segmentos principais MHAS juntamente do arquivo manifesto referente ao conteúdo testado. Os arquivos foram pré-codificados considerando uma transmissão *multi-stream* e encapsulados em arquivos baseados no formato de mídia padronizado pela Organização Internacional de Normalização, do inglês ISOBMFF. Cada segmento foi gerado com 2 segundos de duração. O arquivo manifesto foi modificado para o modo dinâmico e contém a variável "minBufferTime" igual a 4 segundos.

O *Webserver* 2 contém os segmentos MHAS auxiliares simulando um ambiente OTT. Para inserir o aumento de atraso entre o *Webserver* e o *player*, foi utilizado um emulador de latência em rede, conforme Seção 5.1. Por meio de uma relação cliente-servidor, o emulador consegue atrasar os pacotes de uma conexão *ethernet*. Testes similares foram propostos em Forum SBTVD (2021b) e Chaubet et al. (2021).

O equipamento *android* possui uma aplicação concedida pela Fraunhofer IIS capaz de receber e decodificar arquivos no formato *multi-stream* utilizada como *player* de áudio e vídeo.

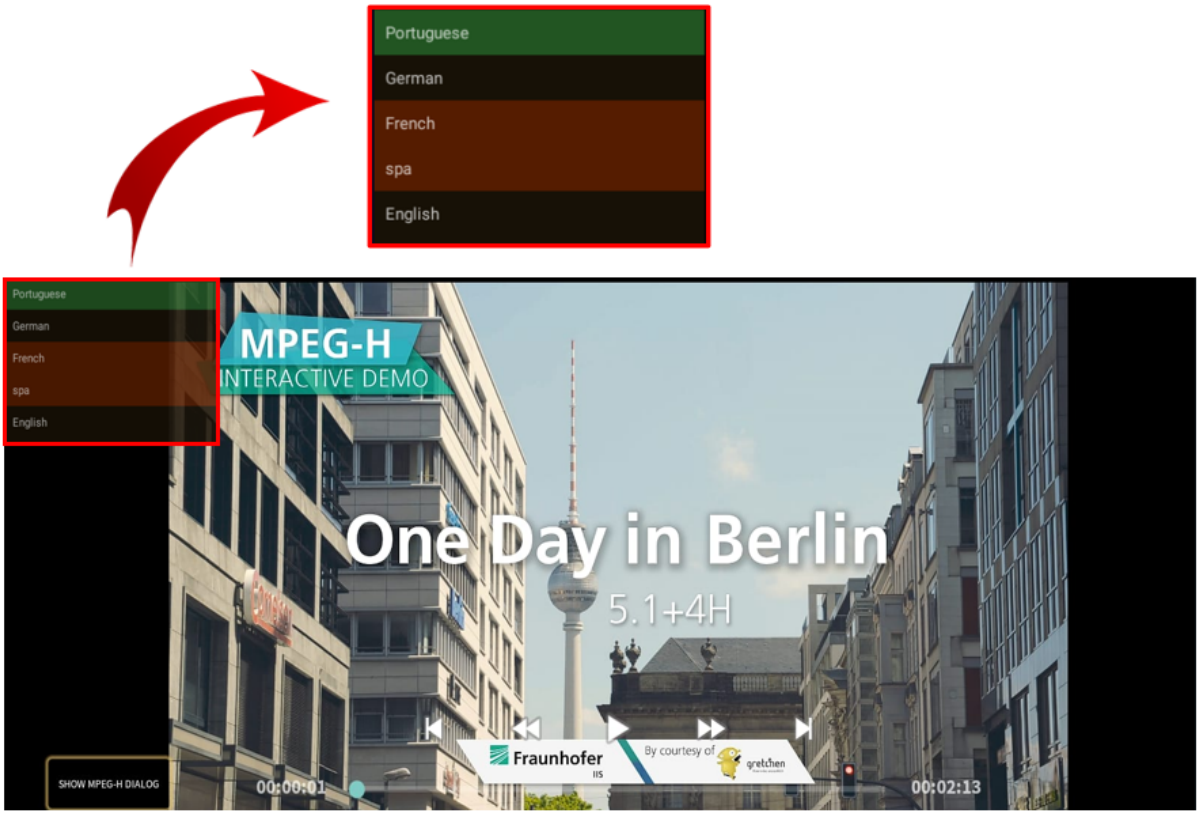
Figura 32: Configuração utilizada para realização dos testes de tolerância.



Fonte: Adaptado e traduzido de (JUNIOR et al., 2024).

O *player* lê o arquivo manifesto e abre a predefinição principal e de acordo com a interação com o usuário, a aplicação pode solicitar os segmentos contidos no *Webserver* 2, definindo a nova predefinição escolhida. O *player* confere se os arquivos auxiliares estão disponíveis ou não, indicando em verde caso esteja disponível e selecionado, em vermelho caso não estejam e em preto caso estejam disponíveis. A Figura 33 ilustra o comportamento descrito com o idioma português disponível e selecionado, os idiomas alemão (*German*) e Inglês (*English*) disponíveis e os idiomas francês (*French*) e espanhol (*spa*) não disponíveis.

Figura 33: Exemplo de execução do *player*.



Fonte: Autoria Própria (2024).

5.5 Resultados dos Testes

Foram realizados testes subjetivos a fim de avaliar o comportamento do receptor com a inserção de um atraso no sistema. O conteúdo do receptor foi avaliado em um intervalo de 2 minutos. Foram realizadas 10 tentativas para cada valor de atraso e, caso o conteúdo

seja decodificado sem erros, foram realizadas novas tentativas com um maior valor de atraso.

Durante os testes, notou-se três tipos de comportamento do receptor: instabilidade na decodificação do vídeo, falha do sincronismo entre o conteúdo OTA e OTT e indisponibilidade do conteúdo de áudio transmitido via OTT.

As Tabelas 6, 7 e 8 mostram os resultados dos testes de tolerância a latência para cada tipo de comportamento identificado. Caso a recepção do conteúdo não apresente nenhum dos erros constatados, é descrito como "OK". Caso apresente algum erro, é descrito como "NOK".

Tabela 6: Resultados dos testes de tolerância a latência o erro de instabilidade na decodificação do vídeo.

Atraso	**Status Video (OTA)**									
Total (ms)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
100	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
300	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
500	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
800	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
900	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
1000	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
1100	NOK	OK	OK	OK	OK	OK	OK	OK	OK	OK
1200	OK	NOK	OK	OK	NOK	OK	OK	OK	OK	OK
1300	OK	NOK	NOK	OK	OK	OK	OK	OK	OK	OK
1400	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	OK
1500	NOK	NOK	NOK	NOK	NOK	NOK	OK	NOK	NOK	NOK
1600	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK

Fonte: Autoria Própria.

Tabela 7: Resultados dos testes de tolerância a latência para o erro de lipSync entre OTA/OTT.

Atraso	**Erro de LipSync entre OTA/OTT**									
Total (ms)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
100	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
300	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
500	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
800	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
900	OK	OK	NOK	OK	OK	OK	OK	OK	OK	OK
1000	OK	NOK	OK	OK	OK	OK	OK	OK	OK	OK
1100	NOK	OK	OK	NOK	NOK	OK	OK	OK	OK	OK
1200	OK	OK	OK	NOK	OK	NOK	OK	OK	OK	OK
1300	OK	OK	OK	NOK	OK	NOK	OK	OK	OK	OK
1400	NOK	OK	OK	NOK	OK	OK	NOK	OK	OK	OK
1500	OK	OK	OK	NOK	OK	OK	OK	NOK	OK	OK
1600	NOK	OK	NOK	NOK	NOK	NOK	OK	OK	NOK	NOK

Fonte: Autoria Própria.

Tabela 8: Resultados dos testes de tolerância a latência para a indisponibilidade do conteúdo de áudio adicional OTT.

Atraso	**Disponibilidade do Áudio Adicional**									
Total (ms)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
100	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
300	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
500	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
800	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
900	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
1000	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
1100	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
1200	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
1300	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
1400	OK	OK	OK	OK	OK	OK	OK	OK	OK	NOK
1500	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
1600	OK	OK	OK	OK	OK	OK	OK	OK	NOK	OK

Fonte: Autoria Própria.

Para a avaliação dos resultados, foi utilizado o método de intervalo de confiança. O

intervalo de confiança fornece uma previsão da incerteza de um valor estatístico calculado a partir de uma amostra. Para realizar esse calculo, primeiramente é calculada a proporção de sucesso das amostras (p). Esse calculo é feito com a divisão do numero de sucessos (s) pelo valor total de amostras (n) como mostra na Equação 5.

$$p = \frac{s}{n} \tag{5}$$

Em seguida, é calculado o erro padrão da proporção amostral (SE). Esse calculo é feito usando a Equação 6 onde (p) é a proporção da amostra e (n) e o valor total de amostras.

$$SE = \sqrt{\frac{p \times (1-p)}{n}} \tag{6}$$

Com esse valor de erro, é possível calcular o intervalo de confiança de cada valor de atraso. Para chegar nesse intervalo, é usada Equação 7, na qual (p) é a proporção de sucesso, (z) é o valor crítico fixo utilizado para calcular o intervalo de confiança com 95 %, neste caso, 1,96 e (SE) é o erro da proporção.

$$\text{IC} = p \pm z \times SE \tag{7}$$

Para a realização do calculo da probabilidade de cada valor de atraso, foram considerados os 3 tipos de erros possíveis. Considerando 3 comportamentos possíveis e 10 tentativas para cada, o total de amostras é igual a 30 para cada valor de atraso aplicado.

Pode ser observado que até 800 ms o *player* foi capaz de decodificar e reproduzir o conteúdo de áudio e vídeo sem erros, consequentemente, apresenta um intervalo de confiança de 100%. Com um intervalo de confiança de 95% das vezes, a chance do *player* decodificar e reproduzir o áudio e video sem erros é :

- Para valores de atraso de 900 e 1000 ms, o intervalo esta entre 90,24% e 100%.
- Para valores de atraso de 1100, 1200 e 1300 ms, o intervalo esta entre 74,50% e 98,80%.
- Para valores de atraso de 1400 e 1500 ms, o intervalo esta entre 38,93% e 74,40%.

- Para o valor de atraso de 1600, o intervalo esta entre 22,47% e 57,50%.

6 CONCLUSÃO

A adoção de novas tecnologias para a TV 3.0 abre espaço para estudos de implementação em um sistema completamente novo. Cada tecnologia possui características únicas a serem exploradas. Neste trabalho foi apresentado um estudo sobre o MPEG-H *Audio* e conceitos importantes para entender o contexto de sua utilização e implementação.

O MPEG-H *Audio* é um *codec* de áudio consolidado no mundo e vem mostrando evolução nos últimos anos com funcionalidades importantes para as atuais aplicações existentes. Dentre essas funcionalidades, é possível destacar o uso de metadados, a entrega *multi-stream* e os mecanismos de alinhamento entre os pontos de acesso aleatório, como o mecanismo de truncamento de áudio e o IPF. Essas funcionalidades garantem um dos principais objetivos da TV 3.0: integração *broadcast-broadband*.

Com o objetivo de testar essas funcionalidades e garantir maiores dados para implementação da tecnologia no sistema, foi realizado um estudo sobre a tolerância do *player* a latência em um cenário onde o conteúdo pré-gravado é transmitido parte via *broadcast* e parte via *broadband*.

Para realizar essa avaliação, foi criada uma configuração com dois *Webservers* simulando os dois métodos de transmissão, um emulador de latência em rede e a utilização de um equipamento com sistema *android* rodando uma aplicação com os recursos necessários para realizar a decodificação proposta. O *player* não possui nenhum mecanismo de compensação de latência, se tratando de um aplicação não comercial.

É importante destacar que a implementação do receptor tem um impacto direto na decodificação do conteúdo. Cada implementação implica em consequências diversas na decodificação e com o acréscimo de um atraso, é esperado que cada aplicação mostre comportamentos diversos.

Foi concluído que, considerando a implementação do *player* e o perfil de atraso utilizado, o *player* suportou uma tolerância a latência de até 800 ms aproximadamente entre a leitura do arquivo manifesto e a recepção e reprodução dos segmentos auxiliares transmitidos via OTT sem nenhum erro. Com o aumento do valor de atraso, a probabilidade de acontecer erros aumenta. Esta probabilidade foi calculada usando um intervalo de confiança de 95%. Para valores de atraso de 900 e 1000 ms, a probabilidade de ocorrer

um erro está entre 90,24% e 100%. Para valores de atraso de 1100, 1200 e 1300 ms, a probabilidade de ocorrer um erro está entre 74,50% e 98,80%. Para valores de atraso de 1400 e 1500 ms, a probabilidade de ocorrer um erro está entre 38,93% e 74,40%. Para o valor de atraso de 1600, a probabilidade de ocorrer um erro está entre 22,47% e 57,50%.

Isso mostra que a entrega híbrida com arquivos multi-stream é muito promissora visto a facilidade de integração de dois meios de entrega, OTA e OTT. Essa função reforça o requisito de integração entre o broadcast e broadband e facilita aspectos de interatividade e acessibilidade, facilitando a entrega de conteúdos adicionais.

Os resultados obtidos neste estudo indicam que o *player* pode ser utilizado em cenários com conteúdo pré-gravado porém são necessárias avaliações adicionais para utilização em cenários onde o conteúdo é gerado em tempo real.

6.1 Trabalhos Futuros

- Realizar melhorias na implementação do *player* utilizado para diminuir a tolerância a latência;
- Realizar testes adicionais integrando as tecnologias de camada física;
- Realizar testes incluindo o cenário de conteúdo gerado em tempo real, adicionando analises de atraso da camada física e da replicação na rede de entrega de conteúdo, do inglês *Content Delivery Network* (CDN);

6.2 Artigos Publicados

- Evaluation of the latency tolerance between OTA and OTT in multi-stream MPEG-H Audio delivery, IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), 19 a 21 de Junho de 2024, Toronto, Canada. Autores: Luciano Neves dos Santos Júnior, George Henrique Maranhao Garcia de Oliveira, Adrian Murtaza, Fadi Jerji e Cristiano Akamine.

REFERÊNCIAS BIBLIOGRÁFICAS

ADVANCED TELEVISION SYSTEMS COMMITTEE. *ATSC Standard: A/342:2021 Part 3, MPEG-H System*. [S.l.], 2023. Disponível em: <https://www.atsc.org/atsc-documents/a342-part-32017-mpeg-h-system/>.

ASSOCIAçãO BRASILEIRA DE NORMAS TéCNICAS. *Televisão digital terrestre – Multiplexação e serviços de informação (SI) Sintaxes e definições da informação básica de SI*. [S.l.], 2023.

BLEIDT, R. L. et al. Development of the mpeg-h tv audio system for atsc 3.0. *IEEE Transactions on Broadcasting*, v. 63, n. 1, p. 202–236, 2017.

CHAUBET, A. S. S. et al. Latency comparison of mmt and route/dash for the transport layer of the tv 3.0 project. In: *2021 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*. [S.l.: s.n.], 2021. p. 1–6.

Peter Graham Craven e Michael Anthony Gerzon. *Coincident microphone simulation covering three dimensional space and yielding various directional outputs*. ago. 1977. 4042779. Disponível em: <https://patents.google.com/patent/US4042779A/en>.

Blumlein Alan Dower. *Sound-transmission, sound-recording, and sound-reproducing system*. nov. 1936. 2062275. Disponível em: <https://patents.google.com/patent-/US2062275A/en>.

EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. *Digital Video Broadcasting (DVB); Specification for the use of Video and Audio Coding in Broadcast and Broadband Applications*. [S.l.], 2022. Disponível em: <https://www.etsi.org/deliver-/etsi_ts/101100_101199/101154/02.04.01_60/ts_101154v020401p.pdf>.

FELLGETT, P. B. Ambisonic reproduction of directionality in surround-sound systems. *Nature*, 1974.

FORUM SBTVD. *Call for Proposals: TV 3.0 Project*. [S.l.], 2020. Disponível em: <https://forumsbtvd.org.br/wp-content/uploads/2020/07/SBTVDTV-3-0-CfP.pdf>.

FORUM SBTVD. *Testing and Evaluation Report: TV 3.0 Project - Audio Coding.* [S.l.], 2021. Disponível em: <https://forumsbtvd.org.br/wp-content/uploads/2020/07-/SBTVDTV-3-0-CfP.pdf>.

FORUM SBTVD. *Testing and Evaluation Report: TV 3.0 Project - Transport Layer.* [S.l.], 2021. Disponível em: <https://forumsbtvd.org.br/wp-content/uploads/2021/12-/SBTVD-TV_3_0-TL-Report.pdf>.

GHAZEL, A. et al. Design and performance analysis of a high speed awgn communication channel emulator. In: *2001 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (IEEE Cat. No.01CH37233).* [S.l.: s.n.], 2001. v. 2, p. 374–377 vol.2.

GREWE, Y.; MURTAZA, A.; MELTZER, S. Mpeg-h audio system for sbtvd tv 3.0 call for proposals. *SET INTERNATIONAL JOURNAL OF BROADCAST ENGINEERING*, v. 7, Mar. 2023. Disponível em: <https://revistas.set.org.br/ijbe/article/view/219>.

HERRE, J. et al. Mpeg-h 3d audio—the new standard for coding of immersive spatial audio. *IEEE Journal of Selected Topics in Signal Processing*, v. 9, n. 5, p. 770–779, 2015.

INTERNATIONAL ORGANIZATION FOR STANDARDISATION/INTERNATIONAL ELECTROTECHNICAL COMMISSION. *ISO/IEC 14496-12 - Information technology — Coding of audiovisual objects — Part 12: ISO base media file format.* [S.l.], 2022. Disponível em: <https://www.iso.org/standard/83102.html>.

INTERNATIONAL ORGANIZATION FOR STANDARDISATION/INTERNATIONAL ELECTROTECHNICAL COMMISSION. *ISO/IEC 23008-3:2022 - Information technology — High efficiency coding and media delivery in heterogeneous environments — Part 3: 3D audio.* [S.l.], 2022. Disponível em: <https://www.iso.org/obp/ui/en-/iso:std:iso-iec:23008:-3:ed-3:v1:en>.

INTERNATIONAL ORGANIZATION FOR STANDARDISATION/INTERNATIONAL ELECTROTECHNICAL COMMISSION. *ISO/IEC 23009-1 - Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats.* [S.l.], 2022. Disponível em: <https://www.iso.org-/standard/83314.html>.

INTERNATIONAL ORGANIZATION FOR STANDARDISATION/INTERNATIONAL ELECTROTECHNICAL COMMISSION. *ISO/IEC 13818-1:2023 - Information technology — Generic coding of moving pictures and associated audio information Part 1: Systems*. [S.l.], 2023. Disponível em: <https://www.iso.org/standard/87619.html>.

JUNIOR, L. N. dos S. et al. Evaluation of the latency tolerance between ota and ott in multi-stream mpeg-h audio delivery. In: *2021 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*. [S.l.: s.n.], 2024. p. 1–5.

MOTTA, L. L. *Um Método de Compressão de Áudio Baseado na Decomposição do Sinal em Sub-bandas Wavelet e Codificação dos Coeficientes Wavelet Mais Expressivos*. 90 p. Dissertação (Mestrado) — Universidade Estadual de Campinas, Campinas, 2019.

MOVING PICTURE EXPERTS GROUP. *MPEG-H 3D Audio Baseline Profile Verification Test Report*. [S.l.]. Disponível em: <https://www.mpegstandards.org-/wpcontent/uploads/2020/07/w19407.zip>.

MOVING PICTURE EXPERTS GROUP. *MPEG-H 3D Audio Verification Test Report*. [S.l.]. Disponível em: <https://mpeg.chiariglione.org/standards/mpeg-h/3d-audio-/mpeg-h-3d-audio-verification-test-report>.

MPEG-H Decoder - Fraunhofer IIS. *MPEG-H Decoder Software*. 2023. Disponível em: <https://github.com/Fraunhofer-IIS/mpeghdec>. Acesso em: 27 dec. 2023.

MURTAZA, A.; MELTZER, S. The use of mpeg-h audio in broadcast. In: *NAB Broadcast Engineering and Information Technology Conference Proceedings 2019*. [S.l.: s.n.], 2019.

NIELSEN, H. et al. *Hypertext Transfer Protocol – HTTP/1.1*. RFC Editor, jun. 1999. RFC 2616. (Request for Comments, 2616). Disponível em: <https://www.rfc-editor.org-/info/rfc2616>.

OLIVEIRA, G. H. M. G. de; VALEIRA, G. d. M.; AKAMINE, C. A proposal to use route/dash in the advanced isdb-t. *IEEE Transactions on Broadcasting*, p. 1–10, 2024.

OLMEDO, G. et al. Mpeg-2 transport stream analyzer for digital television. In: *2016 35th International Conference of the Chilean Computer Science Society (SCCC)*. [S.l.: s.n.], 2016. p. 1–9.

PAINTER, T.; SPANIAS, A. Perceptual coding of digital audio. *Proceedings of the IEEE*, v. 88, n. 4, p. 451–515, 2000.

POSTEL, J. *Transmission Control Protocol: DARPA Internet Program Protocol Specification*. set. 1981. Request for Comments 793, Internet Engineering Task Force. Disponível em: <https://www.rfc-editor.org/rfc/rfc793.txt>.

PULKKI, V. Virtual sound source positioning using vector base amplitude panning. *Journal of the Audio Engineering Society*, Audio Engineering Society, v. 45, n. 6, p. 456–466, 1997. ISSN 1549-4950.

RADIOCOMMUNICATION SECTOR OF INTERNATIONAL TELECOMMUNICATION UNION. *Multichannel stereophonic sound system with and without accompanying picture*. [S.l.], 2022. Disponível em: <https://www.itu.int/rec/R-REC-BS.775-4-202212-I/en>.

SALOMON, D. *Data compression - The Complete Reference, 4th Edition*. [S.l.: s.n.], 2007. ISBN 978-1-84628-602-5.

SANTANA, M. A. *Desenvolvimento de um Analisador de Fluxo para o Padrão ROUTE-DASH*. 77 p. Dissertação (Mestrado) — Universidade Presbiteriana Mackenzie, São Paulo, 2023.

SAYOOD, K. 16 - audio coding. In: SAYOOD, K. (Ed.). *Introduction to Data Compression (Third Edition)*. Third edition. Burlington: Morgan Kaufmann, 2006, (The Morgan Kaufmann Series in Multimedia Information and Systems). p. 515–536.

SHANNON, C. E. A mathematical theory of communication. *The Bell System Technical Journal*, v. 27, n. 3, p. 379–423, 1948.

SODAGAR, I. The mpeg-dash standard for multimedia streaming over the internet. *IEEE MultiMedia*, v. 18, n. 4, p. 62–67, 2011.

VALEIRA, G. de M. et al. Transport stream analysis of a isdb-t signal using an embedded and reconfigurable system. *IEEE Transactions on Broadcasting*, v. 61, n. 1, p. 30–38, 2015.

VAZ, R. A. *TESTES DE LATÊNCIA E ESCALABILIDADE DE VÍDEO POR BROADBAND E BROADCAST UTILIZANDO-SE O SISTEMA ATSC 3.0*. 157 p. Tese (Doutorado em Engenharia Elétrica e Computação) — Programa de Pós-Graduação em Engenharia Elétrica e Computação da Universidade Presbiteriana Mackenzie, São Paulo, 2024.

WALKER, G. K. et al. Route/dash ip streaming-based system for delivery of broadcast, broadband, and hybrid services. *IEEE Transactions on Broadcasting*, v. 62, n. 1, p. 328–337, 2016.

YOU, D.; KIM, S.-H.; KIM, D. H. Atsc 3.0 route/dash signaling for immersive media: New perspectives and examples. *IEEE Access*, v. 9, p. 164503–164509, 2021.