

Streaming Adaptativo sobre HTTP

Leandro da Silva de Melo

Instituto de Computação
Universidade Federal Fluminense (UFF) – Niterói, RJ – Brasil
leandro.dmelo@gmail.com

Abstract. *This article describes in detail the different types of adaptive video streaming over HTTP, showing the architecture of each one of them and also how it works. Information about the respective standards is presented in a simplified way, making it easier to compare their resources.*

Resumo. *Este trabalho descreve de forma detalhada os diversos tipos de streaming de vídeo adaptativo sobre HTTP, apresentando a arquitetura de cada um deles e também o seu funcionamento. Informações sobre os respectivos padrões são apresentados de forma simplificada, facilitando a comparação de seus recursos.*

1. Introdução

Por volta dos anos 90, iniciou-se a entrega de conteúdo de vídeo pela Internet. Um dos principais desafios da época além de lidar com a quantidade de dados, era lidar com o melhor esforço (*best-effort*), uma das características da Internet. A Internet por sua vez não foi projetada originalmente para ser usada com aplicações que demandem altas taxas de largura de banda, como por exemplo *streaming* de vídeo de alta qualidade. Inicialmente a maioria dos trabalhos sobre transmissão de vídeo em pacotes focaram em fornecer transmissão em tempo real com técnicas que suportassem reserva de recursos e qualidade de serviço. O Protocolo de Transporte em Tempo Real (RTP) foi proposto para oferecer suporte ao streaming em tempo real através do UDP, no entanto, esbarrava em problemas como NATs e *firewalls*. Outro ponto muito importante era a exigência de infraestrutura de rede e servidores dedicados. Isso trazia mais complexidade e acabava aumentando os custos de implantação. Geralmente por conta dos pacotes RTP não serem permitidos através de *firewalls* e pela necessidade de que o servidor gerencie as sessões de *streaming* de cada cliente separadamente, o RTP foi caindo em desuso.

Na última década, a Internet se tornou o meio padrão para a entrega de conteúdo multimídia. Em 2015, mais de 50% do pico de tráfego de download nos Estados Unidos eram de empresas como Youtube e Netflix [Cisco, 2017]. Segundo o Cisco Visual Networking Index, até 2022 cerca de 82% do tráfego total da Internet será para consumo e entrega de conteúdo. O conteúdo multimídia pode ser entregue com mais eficiência por conta do aumento da largura de banda da Internet. Vários benefícios puderam ser observados com a transição do *streaming* de vídeo em tecnologias baseadas em UDP para TCP.

O *streaming* HTTP logo se tornou um padrão comercial. Empresas como Apple, Microsoft e Adobe adotaram o HTTP para entrega de conteúdo. Cada empresa desenvolveu seu próprio formato e isso exigia que o cliente para receber o conteúdo de

cada uma das empresas deveria suportar cada um dos protocolos. Isso fez com que houvesse uma disputa para que um dos protocolos se tornasse o padrão de mercado.

Com a rápida convergência dos Sistemas multimídia como a TV tradicional, IPTV, videoconferência, Vídeo sob Demanda (VOD), os serviços ao vivo e os dispositivos móveis, nota-se que isso tem afetado diversos setores da indústria, relacionados à Internet. Padrões e protocolos para distribuição desse conteúdo passaram a ser relevantes, pois impactam na experiência do usuário (QoE).

Este trabalho tem como objetivo direcionar o leitor para um entendimento sobre os tipos de *streaming* de vídeo adaptativo sobre HTTP com suas respectivas características e arquiteturas. Além de trazer uma comparação com as principais funcionalidades facilitando o entendimento da sua aplicabilidade.

O restante do texto está estruturado da seguinte forma. A Seção 2 discute o funcionamento da adaptação dinâmica de taxa abordando os principais desafios e a experiência do usuário. Uma análise sobre a arquitetura e funcionamento do MPEG-DASH assim como suas vantagens e desvantagens. A Seção 3 discute os tipos de *streaming* que foram desenvolvidos por alguns fabricantes aumentando a disputa pelo mercado de streaming adaptativo de vídeo no mercado além de mostrar as suas respectivas arquiteturas e também o seu funcionamento. A Seção 4 faz uma comparação com todos os tipos de streaming abordado neste trabalho. A Seção 5 são as observações finais e aborda questões de possíveis pesquisas em trabalhos futuros.

2. Adaptação Dinâmica de Taxa

A adaptação dinâmica de taxa utiliza algoritmos para selecionar dinamicamente as taxas de bits de vídeo apropriadas quando as condições de rede mudam, a fim de evitar interrupções na reprodução que podem ser causadas pela falta de buffer.

O principal desafio da Adaptação Dinâmica de Taxa é equilibrar os seguintes objetivos:

- maximizar a qualidade de vídeo,
- minimizar o número de trocas de qualidade de vídeo para melhorar a experiência do usuário,
- minimizar o tempo de espera da requisição de um novo vídeo pelo usuário e o tempo em que o vídeo começa a ser reproduzido.

2.1. Experiência do Usuário

Assim como as aplicações, a entrega de conteúdo em redes IP precisa lidar com as limitações inerentes ao uso do QoS nas redes [Seufert et al., 2015]. Podem ocorrer gargalos em vários locais, no servidor, na interconexão das operadoras, no *link* de acesso do usuário final e ou na própria rede doméstica. O protocolo de transporte TCP tem um comportamento dinâmico que muitas vezes se torna imprevisível gerando impacto na qualidade de experiência do usuário, mas as tecnologias de *streaming* adaptativo permitem a suavização destes problemas. A forma mais interessante seria a produção de vários arquivos em taxas diferentes do mesmo conteúdo para distribuir aos diferentes usuários que assistem em diferentes dispositivos e em diferentes taxas de largura de banda. Apesar do destaque anterior para produção de vários arquivos, outro

fator relevante é a possibilidade de distribuí-los de forma adaptável, ou seja, quando um usuário dispõe de recursos adicionais na taxa de transferência, o *player* negocia com o servidor arquivos de melhor qualidade e caso haja uma degradação na taxa de transferência, o *player* negocia com o servidor arquivos de qualidade inferior, a fim de manter a reprodução do vídeo para o usuário. A troca ocorre sem que o usuário perceba. Isso é ótimo do ponto de vista de experiência, pois não requer nenhuma ação por parte dele.

Estas tecnologias acabaram permitindo que os conteúdos fossem entregues na melhor qualidade possível e podendo se adaptar na entrega para clientes que não dispõem de recursos para exibi-los. Esta facilidade permite que a maioria dos programadores de conteúdo alcancem um maior número de usuários, uma vez que eles não precisam escolher uma taxa média.

2.2. DASH

O DASH por exemplo codifica o conteúdo do vídeo em várias versões com diferentes taxas de bits discretas (taxas de representação) conforme Figura 1. Cada vídeo codificado é fragmentado em pequenos segmentos, ou *chunks*, que nada mais são que arquivos com partes discretas do conteúdo original, que pode ter vídeo, áudio, legenda, sinalizações para inserção comercial, metadados, etc. Os pequenos segmentos são alinhados com outros pequenos segmentos de outras taxas de bits numa espécie de linha do tempo do vídeo para que o cliente possa alternar suavemente entre as taxas de bits.

Chama-se de *profile* a sequência de segmentos. O tipo de codec de vídeo, áudio, resolução, além das outras características de compressão de arquivos de mídia caracterizam um *profile*. As características informadas anteriormente determinam qual será a taxa de bits de um *profile* e isso trará impacto direto na qualidade do conteúdo e na banda necessária para que o cliente consiga realizar o *download* de todos os segmentos.

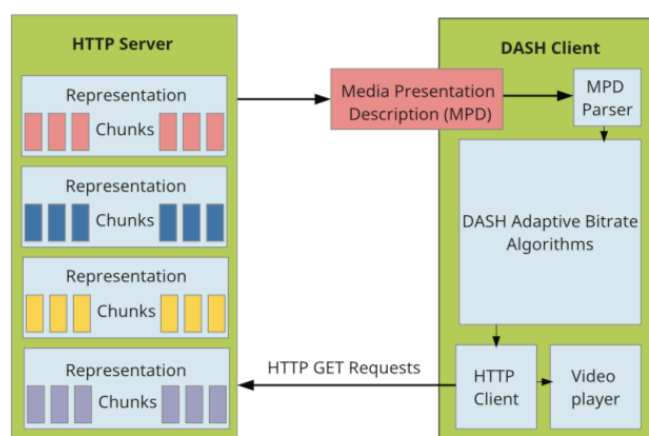


Figura 1. Arquitetura DASH

O *streaming* adaptativo permite que o cliente se adapte às condições da rede, fazendo a seleção de segmentos do *profile* que mais se encaixe à sua taxa de download disponível naquele momento conforme Figura 2. Em técnicas de adaptação baseada em taxa de

transferência Se a largura de banda disponível no cliente é de 30Mbps, mas o profile mais alto tem uma taxa de bits de 50Mbps, o cliente conseguirá exibir o conteúdo na melhor qualidade disponibilizada, porém reduzindo para um *profile* inferior, ainda assim será possível a reprodução, em uma qualidade de vídeo inferior.

É importante ressaltar que este modelo já é utilizado em transmissão tanto de conteúdos ao-vivo (*live*), como sob demanda (VoD).

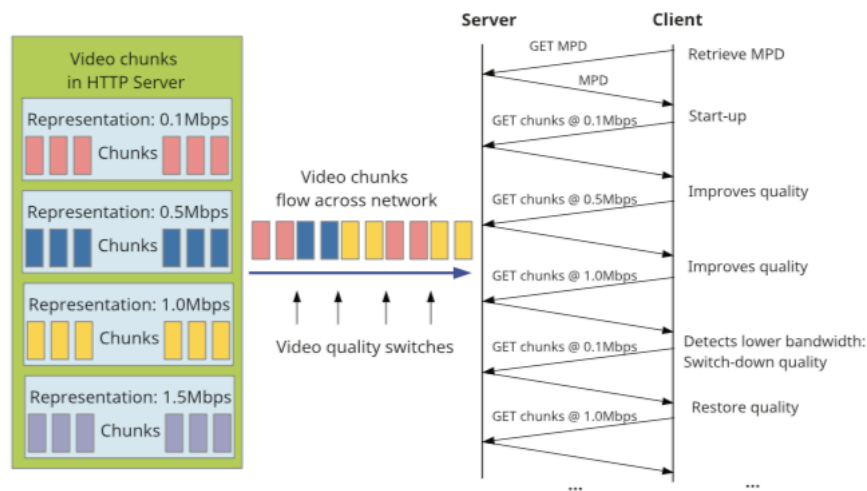


Figura 2. Linha do Tempo do Vídeo

2.3. Funcionamento

A base para entender o funcionamento do MPEG-DASH [Kua et al., 2017] é o arquivo manifesto Figura 3, conhecido também como arquivo MPD (*Media Presentation Description*). O arquivo manifesto usa o formato XML. Ele contém várias informações e parâmetros para o *streaming* de vídeo. Parâmetros como codec, segmentos, taxa de bits, resolução do vídeo, além das informações de URL e os nomes para o *streaming*. Há muitas informações neste arquivo, que é requisitado pelo *player*.

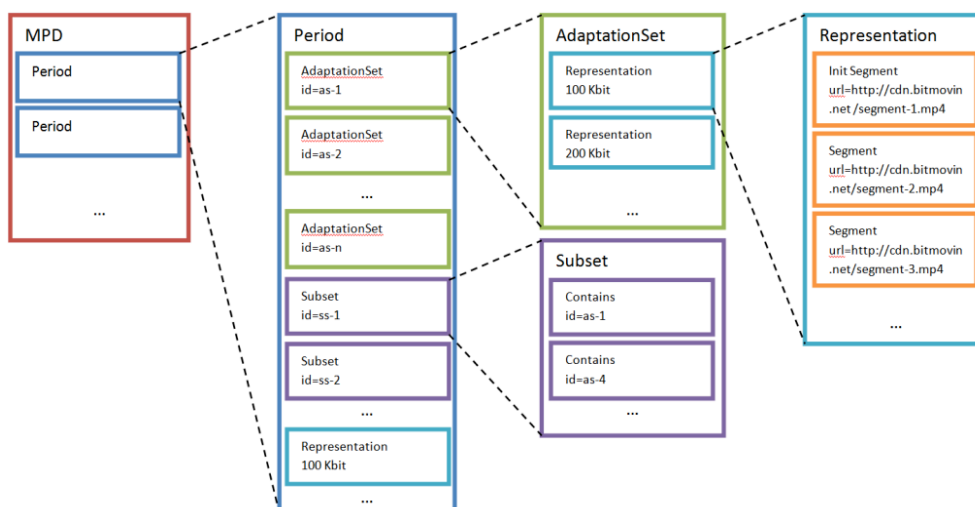


Figura 3. Estrutura do arquivo Manifesto/MPD

Seguindo uma arquitetura cliente-servidor, o player de vídeo do cliente faz uma solicitação HTTP GET, solicitando ao servidor (CDN fisicamente mais próxima do cliente) o arquivo MPD. Esta solicitação tem por finalidade obter as informações sobre a taxa de bits de mídia, resolução, etc., de cada representação, bem como as URLs dos segmentos. Toda sessão é controlada pelo cliente, que se encarrega de utilizar o algoritmo para decidir qual perfil e consequentemente qual segmento ele irá fazer o *download*. Um período irá determinar uma parte do conteúdo com início e fim. O tamanho deve variar de acordo com o tipo de *streaming* (ao vivo, VoD). Se for um conteúdo sob demanda existe uma probabilidade de que o período do segmento seja maior comparado ao *streaming* ao vivo, pois no cenário ao vivo o conteúdo é produzido na hora. O conteúdo sob demanda está armazenado previamente em algum banco de dados do servidor.

Em período Figura 4, há as configurações de adaptação, que contêm um ou mais componentes da mídia. O mais comum é ter a configuração de adaptação para o vídeo, e um para cada idioma de áudio disponível. Além disso, nestas configurações podem ser disponibilizados legendas e metadados.

As representações permitem que diferentes clientes reproduzam o mesmo conteúdo, porém com qualidades diferentes (*profiles*) [Miller et al., 2012]. Normalmente, os *players* de *streaming* adaptativo iniciam a reprodução num *profile* mais baixo, fazendo com que o início do vídeo seja no menor tempo possível.

A monitoração das métricas de taxa de transferência, tamanho do *buffer*, performance da CPU são realizadas pelo *player*.

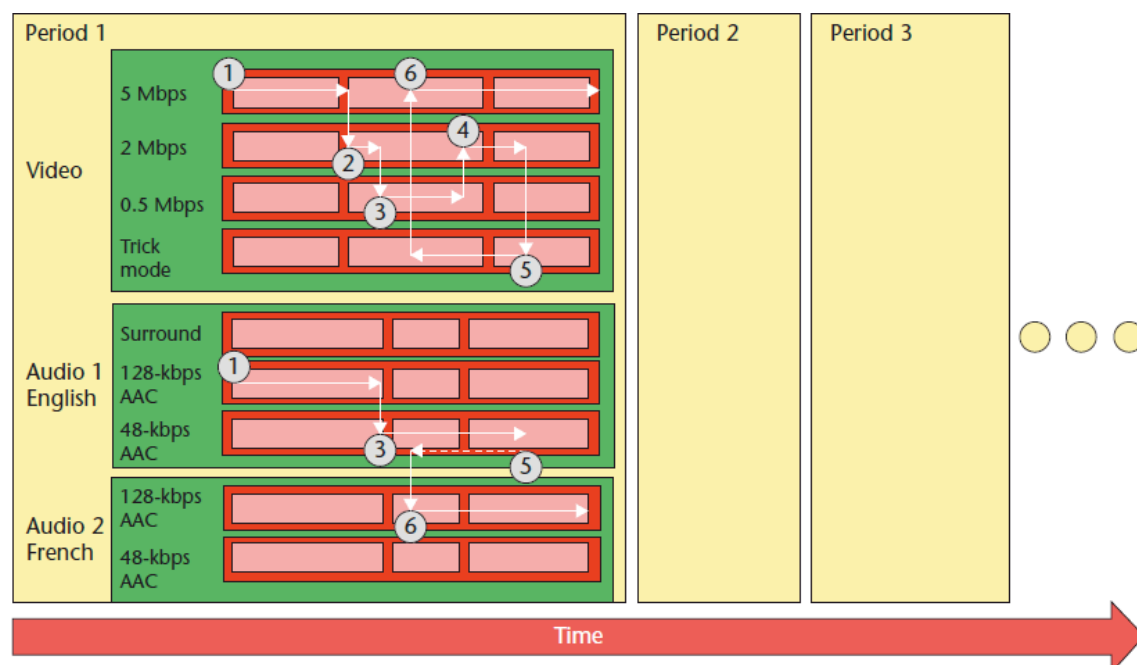


Figure 4. Variação das Taxas de Vídeo/Áudio ao longo do tempo

2.4. Vantagens

O MPEG-DASH tem várias vantagens. Uma é que ele foi o primeiro protocolo de *streaming* de padrão internacional baseado em HTTP. Várias empresas importantes de mídia participaram do seu desenvolvimento diminuindo problemas técnicos de entrega e compactação. Visa combinar todas as tecnologias e padrões em um, tornando o suporte a *streaming* contínuo em vários tipos dispositivos. Além disso, tenta reduzir as dores de cabeça técnicas e os custos de transcodificação. Agora os editores de conteúdo podem gerar um único conjunto de arquivo para codificação e *streaming* que deve ser compatível com o maior número possível de dispositivos, do móvel ao OTT (*Over-the-Top*) e em desktops por meio de *plug-ins* ou HTML5. Os consumidores não precisarão se preocupar se seus dispositivos poderão reproduzir o conteúdo que desejam assistir.

Uma grande vantagem do MPEG-DASH é o fato de ele ser agnóstico ao codec. Isso significa que os arquivos de vídeo ou de mídia enviados por meio deste protocolo podem usar uma variedade de formatos de codificação. Isso inclui padrões amplamente suportados, como o H.264, bem como formatos de vídeo de próxima geração, como HEVC / H.265 e VP10.

Desenvolvido em parceria com mais de dez empresas, sendo reconhecido pela ISO e MPEG, e com isso se tornou um padrão comercial mostrando sua interoperabilidade entre as aplicações.

O MPEG-DASH utiliza o protocolo HTTP por padrão, pois fornece alcance maior, pois o tráfego HTTP pode atravessar *firewall* na maioria dos casos facilitando sua implantação. Não requer uma infraestrutura dedicada podendo inclusive utilizar servidores HTTP comuns.

Oferece suporte a múltiplas CDNs/caches com o mesmo arquivo manifesto, garantindo flexibilidade, escalabilidade e maior tolerância a falhas.

2.5. Desvantagens

Alguns pontos que se tornam uma vantagem para o uso do MPEG-DASH acabam se tornando uma desvantagem sobre outro ponto de vista. Por não ser suportado nativamente pelo HTML5, ele acaba exigindo o uso de uma extensão. O MSE (*Media Source Extension*) é uma especificação que permite que o Javascript envie fluxos de bytes para codecs de mídia em navegadores Web que suportem vídeo e áudio HTML5. Outro ponto importante é que por ser agnóstico aos codecs de áudio/vídeo, pode ocorrer que um conteúdo num determinado codec poderá não ser suportado por um *player* específico. Clientes de *streamings* HTTP também necessitam ter um *buffer* local de um certo número de segmentos, de forma a ter uma tolerância temporal a retransmissões TCP. Tipicamente enquanto um segmento está sendo reproduzido, o outro está no cache e um terceiro está sendo baixado. O uso de um *buffer* causa, necessariamente, maior latência, já que o segmento em reprodução já foi recebido há alguns segundos (no caso de *streaming* ao vivo).

3. Tipos de *Streaming*

Com o aumento da demanda tanto na entrega e no consumo de conteúdo na Internet, vários padrões foram sendo desenvolvidos por diferentes fabricantes. Esta seção apresenta os principais tipos de streaming encontrados no mercado.

3.1. HLS

O HLS (*HTTP Live Streaming*) foi originalmente desenvolvido e disponibilizado pela Apple em 2009 justamente para coincidir com o lançamento do iPhone 3. Os modelos anteriores do iPhone tinham problemas com *streaming* de mídia on-line, principalmente porque esses dispositivos geralmente alternavam entre redes Wi-Fi e redes móveis. Antes do seu lançamento a Apple adotava o *Quicktime Streaming Server* como padrão para *streaming* de mídia. Um serviço bem robusto, mas como não utilizava portas conhecidas na camada de transporte para transferência de dados, seu protocolo RTSP constantemente era bloqueado por firewalls. A combinação destes fatores, aliada às baixas velocidades da Internet da época, acabaram levando o *Quicktime Streaming Server* ao desuso. Com as lições aprendidas com o *Quicktime Streaming Server*, a Apple melhorou o HLS (*HTTP Live Streaming*) [Rainer et al., 2016]. Ele cortava o conteúdo do vídeo em pedaços de 10 segundos e os entregava via HTTP. O uso do HTTP como protocolo padrão já conseguia eliminar os bloqueios enfrentados pelo RTSP ao atravessar algum *firewall*. Assim como o MPEG-DASH, a qualidade na entrega é dinâmica, podendo mudar de acordo com a velocidade do acesso Internet do usuário.

O HLS disponibiliza ao usuário funcionalidades de *streaming* sob demanda ou ao vivo de áudio e vídeo a partir de um servidor. Os requisitos mínimos para o uso do HLS são iOS 3.0 e computadores com Safari 4.0 ou posterior. A codificação do vídeo é feita em HEVC ou H.264 e para o áudio em AAC ou AC-3. O codec de vídeo H.264 foi desenvolvido para proporcionar o dobro da taxa de compressão comparado a outros padrões. Gera um arquivo MPEG-4 fragmentado ou um fluxo MPEG-2 *Transport Stream* (MPEG-TS). Um ponto importante do HLS é que ele realiza a correção de erros. Os três principais componentes do HLS são: servidor, distribuidor e o *software* do usuário conforme Figura 5. O *streaming* HLS é fornecido por meio de um arquivo M3U8. Este arquivo é uma lista de reprodução que contém referências ao local dos arquivos de mídia. Para transmissão ao vivo pela Internet, um arquivo M3U8 conterá uma URL, na qual o fluxo está sendo entregue.

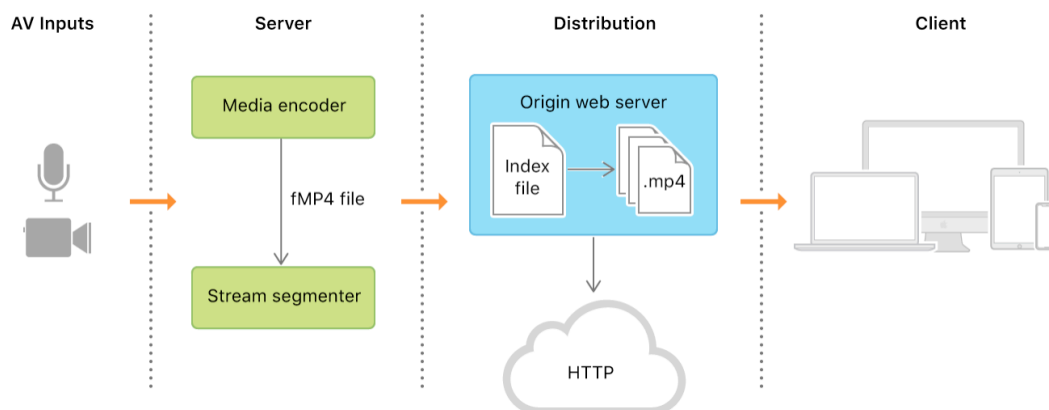


Figura 5. Estrutura do HLS (HTTP Live Streaming)

3.2. MSS

Quando a *Microsoft* anunciou o *Internet Information Server* (IIS) 7.0, ela mencionou um novo recurso de *streaming* HTTP adaptável chamado *Microsoft Smooth Streaming* (MSS) [Gouta et al., 2013], visando fazer a entrega de conteúdo em alta definição (HD) aos clientes. Um ponto importante é que o cliente é baseado no *Silverlight*. De forma automática ele detecta as condições de largura de banda disponíveis, utilização da CPU e a resolução da janela de reprodução pra decidir qual representação do conteúdo se encaixaria melhor. Estas análises são realizadas de forma contínua evitando assim uma parada na reprodução. A primeira demonstração do seu sistema foi nos Jogos Olímpicos de Pequim em 2008. Em 2010, durante os Jogos Olímpicos de Inverno em Vancouver, a emissora de TV NBC Sports usou o *Smooth Streaming* para transmitir o evento alcançando aproximadamente 15,8 milhões de usuários chegando a um total de 50,5 milhões de *streamings* com resolução 720p. O total de tráfego gerado foi de 3,6 Petabytes. Atualmente a *Microsoft* utiliza o MSS na plataforma do *Xbox One*.

O MPEG-4 parte 14 (ISO / IEC 14496-12) foi o padrão adotado para os arquivos de mídia do *Microsoft Smooth Streaming*. O MP4 por ser baseado no IBMFF (ISO *Base Media File Format*) é organizado usando caixas como unidades de dados, metadados além de oferecer várias possibilidades para organização destas caixas num arquivo. O MP4 permite a divisão dos metadados e dados de mídia em vários fragmentos. Cada fragmento é constituído por um metadado e um bloco de dados, que são chamados no streaming suave de MP4 fragmentado (fMP4). Por isso conseguimos armazenar segmentos de mídia separados que correspondem a um ou mais grupo de imagens (GOP), usando uma caixa de fragmento de filme (moof) e uma caixa de dados de mídia (mdat) conforme representado na Figura 6.

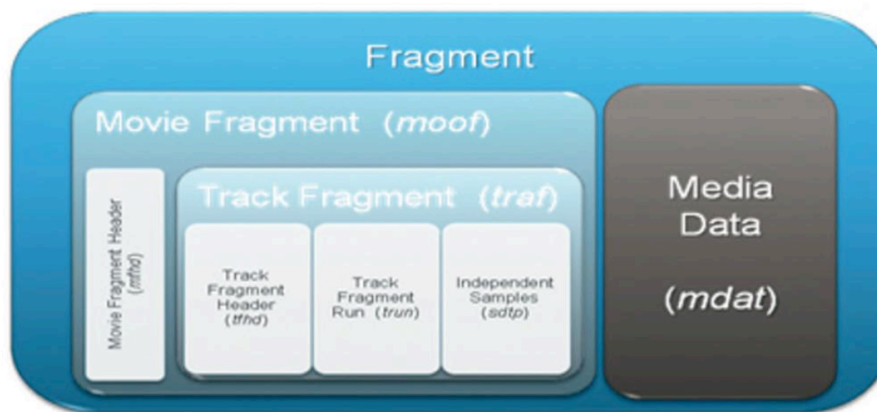


Figura 6. Formato do Fragmento do MSS

O tempo de início da reprodução é curto uma vez que a quantidade de informação no início do arquivo é pequena. O cliente recebe pequenos fragmentos do arquivo MP4 de aproximadamente 2 segundos de duração. Uma checagem é realizada pra saber se o arquivo chegou íntegro ao usuário e, quando este arquivo não chega íntegro, a qualidade do vídeo é reduzida nas próximas partições do vídeo. Quando houver condições

melhores na largura de banda, o cliente voltará a receber partições do vídeo em resoluções melhores. A Figura 7 representa a arquitetura do MSS.

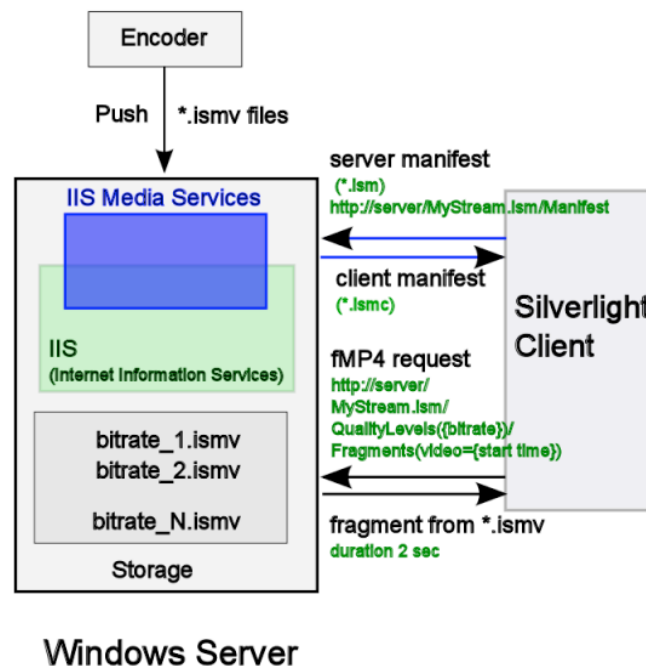


Figura 7. Arquitetura Microsoft Smooth Streaming

3.3. Adobe HDS

O HDS (HTTP *Dynamic Streaming*) é o formato de *streaming* desenvolvido pela Adobe. Foi projetado para ter compatibilidade com o *plugin* de vídeo *Flash* que é utilizado em navegadores. Atualmente na versão 3.0, o HDS apresenta vários recursos novos para trabalhar junto com o Adobe *Primetime* incluindo *closed captions*, sinalização avançada para publicidade e ainda ganhou uma melhoria no suporte a falhas e a tolerância a falhas.

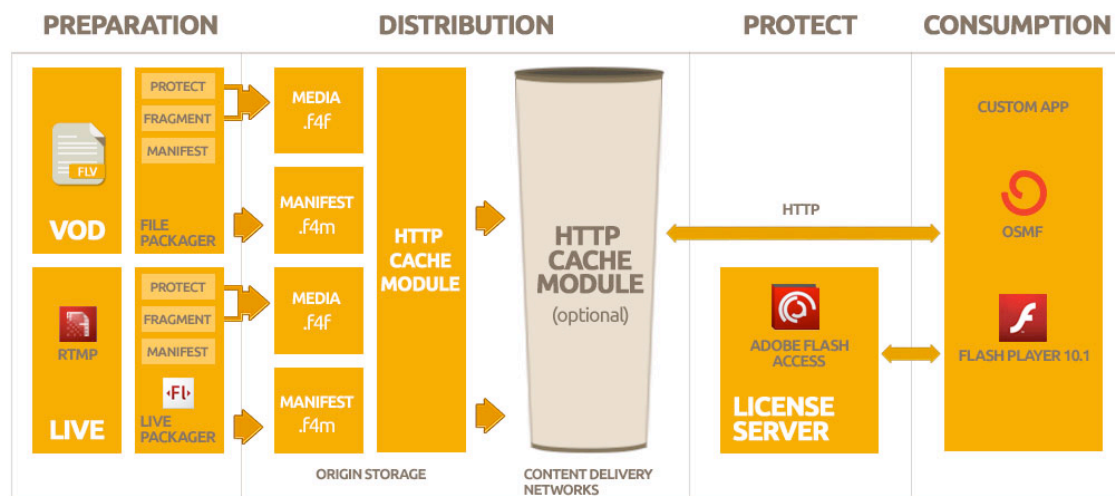


Figura 8. Arquitetura HDS

A Figura 8 representa a arquitetura do HDS. O HDS suporta tanto conteúdo ao vivo, onde é necessário fornecer a entrada da mídia através do protocolo RTMP, caso contrário, a mídia deve estar pré-gravada como o VoD e armazenada em um servidor que pode ser acessado diretamente. São criados basicamente três arquivos nomeados como arquivo de segmento (*Segment File*), arquivo de manifesto (*Manifest File*) e arquivo de índice (*Index File*).

Os arquivos de segmento utilizam a extensão (.f4f). Para cada arquivo de mídia, haverá um arquivo de vários segmentos de acordo com sua taxa de dados. Cada arquivo de segmento contém muitos fragmentos. URLs são usadas para localizar esses fragmentos.

O arquivo de manifesto utiliza a extensão (.f4m). O arquivo de manifesto contém as informações da taxa de dados e sua resolução com as informações de codec do arquivo de mídia.

O arquivo de índice utiliza a extensão (.f4x). Este arquivo de índice contém a localização do fragmento específico. Esses arquivos contêm pequenos pedaços e sua URL. Após a preparação destes arquivos eles são armazenados na cadeia de distribuição para que em seguida fiquem aptos para serem entregues quando requisitados pelos *players* de reprodução dos clientes (*Adobe Flash Player* ou *Adobe Air*).

4. Comparação HDS, HLS, MSS e MPEG-DASH

A Tabela 1 mostra uma comparação de recursos das tecnologias proprietárias de *streaming* adaptável: Adobe HTTP *Dynamic Streaming* (HDS), Apple HTTP *Live Streaming* (HLS), Microsoft *Smooth Streaming* (MSS) e o *Dynamic Adaptive Streaming* sobre HTTP (MPEG-DASH). A tabela é apenas um comparativo de recursos em cada um dos modelos de *streaming* adaptativo.

Tabela 1. Comparação das funcionalidades.

Funcionalidades	Adobe HDS	Apple HLS	MS <i>Smooth Streaming</i>	MPEG-DASH
Implantação em Servidores HTTP comuns	Não	Sim	Não	Sim
Padrão Internacional (ISO/IEC MPEG)	Não	Não	Não	Sim
Vários canais de áudio	Não	Sim	Sim	Sim
Gerenciamento de Direitos Digitais	Sim	Sim	Sim	Sim
Legendas e <i>Closed Captions</i>	Sim	Sim	Sim	Sim
Inserção eficiente de anúncios	Não	Não	Não	Sim
Troca rápida de canal	Sim	Não	Sim	Sim
Suporta múltiplas CDNs em paralelo	Não	Não	Não	Sim
Suporte ao HTML5	Não	Não	Não	Sim
Suporte em HbbTV (versão 1.5)	Não	Não	Não	Sim
Pronto para HEVC (UHD/4K)	Não	Não	Não	Sim
Agnóstico aos codecs de vídeo	Não	Não	Não	Sim
Agnóstico aos codecs de áudio	Não	Não	Não	Sim
Segmentos ISO-BMFF	Sim	Não	Sim	Sim
Segmentos MPEG-2 TS	Não	Sim	Não	Sim
Extensões no Formato do Segmento além	Não	Não	Não	Sim

do MPEG				
Suporte a conteúdo multiplexado (Áudio e Vídeo)	Sim	Sim	Não	Sim
Suporte a conteúdo não multiplexado (Áudio e Vídeo separados)	Não	Sim	Sim	Sim
Definição de métricas de qualidade	Não	Não	Não	Sim
Relatório e Registro de Cliente	Não	Não	Não	Sim
Tolerância a falha de cliente	Não	Não	Não	Sim
Aumenta ou diminui a qualidade durante o <i>streaming</i>	Não	Não	Não	Sim
Múltiplas Visualizações de Vídeo	Não	Não	Não	Sim

O MPEG-DASH e o Apple HLS podem ser usados com servidores HTTP comuns, como Apache, Nginx, IIS, etc. A Adobe e a Microsoft utilizam mecanismos no lado do servidor [Biaz et al., 2008] precisando assim de recursos adicionais.

O MPEG-DASH é um padrão internacional, ratificado em 2012 e atualmente adotado pelo YouTube, Netflix, etc. Empresas como Apple, Samsung, Akamai, Cisco, Dolby, Ericsson, Harmonic, Qualcomm, Netflix e Intel contribuíram para sua padronização.

A troca entre vários canais de áudio é especialmente importante para o conteúdo em vários idiomas. O MPEG-DASH suporta esse recurso, assim como o Apple HLS e o *Microsoft Smooth Streaming*. O MPEG *Common Encryption* (CENC) permite que o conteúdo criptografado uma vez seja compatível com vários sistemas DRM. Isso só é possível, pois quase todos os sistemas DRM oferecem suporte ao método de criptografia AES.

É possível realizar a inserção de anúncios em todos os formatos através da substituição de blocos. Isso significa que algumas partes do vídeo original serão simplesmente substituídas por partes que contém o anúncio. O MPEG-DASH permite a inserção de anúncios de maneira mais eficiente através de uma interface padronizada. Por este motivo servidores HTTP comuns podem ser usados sem necessidade recursos adicionais para redirecionar as solicitações de partes específicas para as que contêm o anúncio.

MPEG-DASH, MSS e Adobe HDS podem usar segmentos baseados em ISO *Base Media Format* (ISOBMFF), por exemplo, MP4, que sobrecarregam menos do que os segmentos MPEG-2 TS usados pela Apple HLS. O MPEG-DASH e o Apple HLS podem usar segmentos no formato MPEG-2 TS que sobrecarregam mais que os segmentos baseados no ISO *Base Media Format* (ISOBMFF). O MPEG-2 TS consiste em pacotes com tamanho de 188 bytes, em que cada pacote possui cabeçalhos com um tamanho variável de 4 a 12 bytes. Portanto, a sobrecarga causada por esses cabeçalhos aumenta proporcionalmente ao tamanho do segmento, o que significa que a sobrecarga relativa não tende a zero com o aumento da taxa de bits. Além disso, cada fluxo TS MPEG-2 consiste em vários outros dados além de áudio e vídeo, como as tabelas de associação de programas que introduzem sobrecarga adicional. Os fluxos de áudio e vídeo são encapsulados no *Packetized Elementary Streams* (PES), que introduz uma sobrecarga extra por amostra de áudio ou quadro de vídeo.

O MPEG-DASH [Stockhammer et al., 2011] é agnóstico aos codecs de vídeo e pode ser usado com conteúdo HEVC e 4K codificado com H.265/HEVC ou H.264/AVC. O MPEG-DASH também suporta formatos além do MPEG, como o WebM. O Adobe HDS, o Apple HLS v6 e o MPEG-DASH podem ser usados com conteúdo de áudio e vídeo multiplexado ou seja empacotados num mesmo segmento. O Apple HLS v8, MSS e MPEG-DASH pode ser usado com conteúdo de áudio e vídeo separado. O MPEG-DASH define métricas de qualidade que permitem ao cliente enviar relatórios específicos sobre comportamento de uso, início da reprodução, parada para um servidor definido.

O MPEG-DASH tem mais flexibilidade e conta com uma forma melhorada na hora de sinalizar os mecanismos de *failover* de CDNs do cliente. É possível que um cliente escolha vários locais diferentes para baixar um segmento através das URLs.

O MPEG-DASH permite, através de períodos, a inserção e exclusão dinâmicas de representações de qualidade durante o *streaming*, com entrega sem interrupções no lado do cliente.

5. Conclusão

É impressionante perceber que os *streamings* de vídeo adaptativo sobre HTTP funcionam muito bem quando se trata de grandes eventos ao vivo com milhões de espectadores. Tanto o HLS quanto o MPEG-DASH são protocolos que contam com recursos bem parecidos trazendo uma certa robustez e confiabilidade durante seu uso. O MPEG-DASH vem ganhando cada vez mais espaço no mercado principalmente por vir com algumas vantagens. Ele foi o primeiro protocolo internacional padrão de *streaming* baseado em HTTP. Esse recurso ajudou a acelerar o processo de adoção.

Uma grande vantagem do MPEG-DASH é que por ser um protocolo agnóstico ao codec permite que os arquivos de vídeo ou das mídias enviadas via MPEG-DASH podem utilizar uma variedade de formatos de codificação. Esses formatos de codificação incluem padrões que são suportados em sua grande maioria, como H.264 (como no protocolo de *streaming* HLS), bem como formatos de vídeo de última geração, como HEVC / H.265 e VP10. O MPEG-DASH, como os outros protocolos abordados neste trabalho, usa servidores HTTP padrão reduzindo o custo e a dificuldade técnica da implementação, quando comparado com métodos como o MSS e o Adobe HDS.

Por ser um padrão aberto, MPEG-DASH abre possibilidades para o desenvolvimento de melhorias. Apesar de não ser suportado nativamente via HTML5, os players podem ser implementados via *Javascript* e ou através de extensões de fonte de mídia. O HLS leva vantagem por ter chegado primeiro ao mercado mas ambos seguem disputando o mercado de igual pra igual. Com o aumento do consumo de vídeo é bem provável que a concorrência aumente. Bom para os clientes que poderão contar com diferentes tecnologias aumentando assim a compatibilidade com seus respectivos dispositivos.

6. Referências

J. Kua, G. Armitage and P. Branch, "A Survey of Rate Adaptation Techniques for Dynamic Adaptive Streaming Over HTTP," in IEEE Communications Surveys &

- Tutorials, vol. 19, no. 3, pp. 1842-1866, thirdquarter 2017, doi: 10.1109/COMST.2017.2685630.
- Stockhammer, Thomas & Sodagar, Iraj. (2012). MPEG DASH: The Enabler Standard for Video Delivery Over the Internet. SMPTE Motion Imaging Journal. 121. 40-46. 10.5594/j18188.
- Stockhammer, Thomas. (2011). Dynamic adaptive streaming over HTTP: Standards and design principles. IEEE International Conference on Multimedia Systems. 133-144. 10.1145/1943552.1943572.
- S. Biaz and S. Wu, "Rate adaptation algorithms for IEEE 802.11 networks: A survey and comparison," 2008 IEEE Symposium on Computers and Communications, Marrakech, 2008, pp. 130-136, doi: 10.1109/ISCC.2008.4625680.
- Cisco. Cisco visual networking index: Global mobile data traffic forecast update, 2016 - 2021. Tech. rep., Cisco, 2017.
- Akhshabi, S.; Begen, A. C.; Dovrolis, C. An experimental evaluation of rateadaptation algorithms in adaptive streaming over http. In Proceedings of the second annual ACM conference on Multimedia systems (2011), ACM, pp. 157–168.
- Dobrian, F.; Sekar, V.; Awan, A.; Stoica, I.; Joseph, D.; Ganjam, A.; Zhan, J.; Zhang, H. Understanding the impact of video quality on user engagement. In ACM SIGCOMM Computer Communication Review (2011), vol. 41, ACM, pp. 362–373.
- Huang, T.-Y.; Johari, R.; McKeown, N.; Trunnell, M.; Watson, M. A buffer-based approach to rate adaptation: Evidence from a large video streaming service. ACM SIGCOMM Computer Communication Review 44, 4 (2015), 187–198.
- ISO. Information technology, 23009-1:2012 - mpeg dash specification. ISO.
- Li, Z.; Zhu, X.; Gahm, J.; Pan, R.; Hu, H.; Begen, A. C.; Oran, D. Probe and adapt: Rate adaptation for http video streaming at scale. arXiv preprint arXiv:1305.0510 (2013).
- Liu, C.; Bouazizi, I.; Gabbouj, M. Rate adaptation for adaptive http streaming. In Proceedings of the second annual ACM conference on Multimedia systems (2011), ACM, pp. 169–174.
- Seufert, M.; Egger, S.; Slanina, M.; Zinner, T.; Hoßfeld, T.; Tran-Gia, P. A survey on quality of experience of http adaptive streaming. IEEE Communications Surveys & Tutorials 17, 1 (2015), 469–492.
- K. Miller, E. Quacchio, G. Gennari and A. Wolisz, "Adaptation algorithm for adaptive streaming over HTTP," 2012 19th International Packet Video Workshop (PV), Munich, 2012, pp. 173-178, doi: 10.1109/PV.2012.6229732.
- B. Rainer, D. Posch and H. Hellwagner, "Investigating the Performance of Pull-Based Dynamic Adaptive Streaming in NDN," in IEEE Journal on Selected Areas in Communications, vol. 34, no. 8, pp. 2130-2140, Aug. 2016, doi: 10.1109/JSAC.2016.2577365.
- A. Gouta, D. Hong, A. Kermarrec and Y. Lelouedec, "HTTP Adaptive Streaming in Mobile Networks: Characteristics and Caching Opportunities," 2013 IEEE 21st International Symposium on Modelling, Analysis and Simulation of Computer and

Telecommunication Systems, San Francisco, CA, 2013, pp. 90-100, doi:
10.1109/MASCOTS.2013.17.