

業界が挑むストリーミング形式の統合

ストリーミング配信を構成する技術は、いくつかの階層に分類することができます(表-1)。これらの技術は様々な会社から提案され、その多くは国際標準規格となっています。各々の技術を組み合わせて最適なストリーミング配信を達成するには、規格のすり合わせが必要です。現在、この中の「ストリーミング形式」に大きな動きが出てきています。

Appleが提唱しているHTTP Live Streaming(HLS)は2009年に発表されました。以降現在に至るまで、ストリーミング形式の一翼を担う重要なフォーマットです。コンテナにセグメント化されたMPEG2-TSを採用し、映像・音声データを収納した細かいファイルを連続的に配信することが特色です。また、配送プロトコルにHTTP/1.1を採用したのも大きなインパクトがありました。HLSは当初iOSを対象としたフォーマットでしたが、その後macOSやtvOS、Androidでも広く使われるようになってきました。

AppleはHLSの規格案をインターネットの標準化を推進するInternet Engineering Task Force(IETF)へ対し、Internet-Drafts(I-D)として提出しています。文書名は"draft-pantos-http-live-streaming"です。I-Dは作業中の文章という位置付けですが、RFCとして完成するものもあれば、提案だけで終わるものもあります。Appleは2009年5月1日に発行された初版以来、Roger Pantos氏の名前でこのI-Dを提出し続けており、2016年9月現在では20版を数えます。

HLSの構造は非常にシンプルです。以下に再生のための指示内容を記したファイル(マニフェストファイル)の例を示します(図-1)。

階層	技術
プレゼンテーション	HTML5, Flashなど
コーデック、メタデータ	H.264, H.265(HEVC)、AAC、WebVTT
コンテナ	MPEG2-TS, MP4
ストリーミング形式	HLS, MPEG-DASH, CMAF
配送プロトコル	HTTP/1.1
トランスポートプロトコル ネットワークプロトコル	TCP/IP

表-1 ストリーミング配信の構成

本質的なのは、URIで示されるリソースです。この例では、サーバmedia.example.com上には「first.ts」「second.ts」「third.ts」という3つのメディアファイルが設置される必要があります。このファイルは静的に配置されても、あるいは動的に配置されても構いません。そしてスキーム名を見て分かるように、このメディアファイルはhttpにより配信されることが示されています。

このマニフェストファイルもやはりWebサーバ上に設置され、再生プレイヤーに渡されます。再生プレイヤーは参考情報を読み込みつつ、URIで示されたメディアファイルに「上から順番に」アクセスしていきます。コメント行には参考情報が記されており、図-1の例だとEXT-X-TARGETDURATIONはメディアファイルの最長の長さを、EXTINFは次に続くメディアファイルの実際の長さを秒で表しています。つまり、この再生指示ファイルによって指定されているメディアは、合計で21.021秒の再生時間であることが分かります。

このI-Dの標準化活動は実施されていません。RFCとして発行するためにはIETFのワーキンググループでの議論が必要になります。しかしこのI-DはどのWGでも継続的な課題として取り上げられておらず、また議論もされていません。いわば「私教版」の扱いです。

これに対抗する意味合いもあり、MPEG-DASH(Dynamic Streaming over HTTP)がInternational Organization for Standardization(ISO)のMPEG(Moving Picture Experts Group)で規格化され、2012年にISO/IEC 23009-1として出版されています。MPEGは1988年より活動している専門家集団であり、標準規格化作業のためのワーキンググループです。

```
#EXTM3U
#EXT-X-TARGETDURATION:10
#EXTINF:9.009,
http://media.example.com/first.ts
#EXTINF:9.009,
http://media.example.com/second.ts
#EXTINF:3.003,
http://media.example.com/third.ts
#EXT-X-ENDLIST
```

図-1 HLS用マニフェストファイルの例

このグループはこれまで映像・音声圧縮フォーマットの規格化作業を主な領域としていましたが、インターネットを用いた動

画配信が広がりを見せたため、ストリーミング形式の策定にも乗り出してきました。

```
<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:mpeg:dash:schema:mpd:2011"
xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011
DASH-MPD.xsd" mediaPresentationDuration="PT0H1M6.1S"
minBufferTime="PT1.5S"
profiles="urn:mpeg:dash:profile:isoff-on-demand:2011"
type="static">
  <Period duration="PT0H1M6.1S" start="PT0S">
    <AdaptationSet>
      <Representation bandwidth="4000000"
codecs="avc1.4d401e" height="1080" id="1"
mimeType="video/mp4" width="1920">
        <BaseURL>video_4000.mp4</BaseURL>
        <SegmentBase indexRange="860-1023">
          <Initialization range="0-859" />
        </SegmentBase>
      </Representation>
      <Representation bandwidth="2400000"
codecs="avc1.4d401e" height="720" id="2"
mimeType="video/mp4" width="1280">
        <BaseURL>video_2400.mp4</BaseURL>
        <SegmentBase indexRange="859-1022">
          <Initialization range="0-858" />
        </SegmentBase>
      </Representation>
    </AdaptationSet>
    <AdaptationSet>
      <Representation bandwidth="128000" codecs="mp4a.40.2"
id="5" mimeType="audio/mp4">
        <BaseURL>audio_128.mp4</BaseURL>
        <SegmentBase indexRange="783-946">
          <Initialization range="0-782" />
        </SegmentBase>
      </Representation>
    </AdaptationSet>
  </Period>
</MPD>
```

図-2 MPDの例

MPEG-DASHの規定ではマニフェストファイルのことをMPD (Media Presentation Description) と呼びます。次にMPDの例を示します(図-2)。

XMLで構造化されているのが分かります。AdaptationSetで定義される複数のRepresentationは、クライアントが動的に切り替えて再生することを想定しています。この例ではサーバ側で4Mbpsと2.4Mbpsのビデオストリームが用意されていることが示されており、クライアントは自らの環境(回線状況やCPU消費など)に応じて最適なストリームを選択することができますようになっています。

現在広く使われているHLSやMPEG-DASH, Smooth StreamingやHTTP Dynamic Streamingは、配信プロトコルとしてHTTP/1.1を採用していたところに大きなメリットがあります。既に広く普及していたHTTP/1.1は、ストリーミング専用のプロトコルを用いるよりもスケラビリティの面で優れていました。HTTPに載せて運ばれるストリーミング形式はデータをセグメント化して、細切れにしてサーバからクライアントに配送するというアイデアは共通していました。

しかしその設計に際してHLS, MPEG-DASH, Smooth Streaming, HDSは各々マニフェストファイルとコンテナ形式を採用し、その結果すべての組み合わせが異なることになりました(表-2)。このような事態は、コンテンツ制作やCDN事業者などの現場を混乱させました。幅広くクライアントをサポートしようとする

名称	規格化	マニフェストファイル	コンテナ形式	標準化の有無
HTTP Live Streaming	Apple	m3u8 m3uの規格を独自拡張	セグメント化されたMPEG2-TS 最新版でMP4をサポート	標準化されず ただしApple以外にも広汎に利用されている
MPEG-DASH	ISO/IEC	MPD XMLで記述	MP4, MPEG2-TS (MP4が多く使われている)	国際標準 詳細定義は各所で行われている (MPEG Industry Forumなど)
Smooth Streaming	Microsoft	isml	MP4	なし
HTTP Dynamic Streaming	Adobe	f4m XMLで記述	f4f	なし

表-2 代表的なストリーミング形式

と、すべてのマニフェストファイルとコンテナ形式をあらかじめ作っておかなければなりません。仮にHLS、MPEG-DASH、Smooth Streaming、HDSを利用すると、4種類のファイルを作成し、管理しなければなりません。制作現場は工数が増大しますし、4倍のストレージを用意しなければなりません。またCDN事業者の配信システムにおいてキャッシュの効率低下を招くこととなります。再生デバイス側でも、どの方式を実装すれば良いのかは難しい問題です。実際にはSmooth StreamingとHDSが採用されることは減少しており、HLSとMPEG-DASHの2種類が選ばれることが多いと思いますが、いずれにせよこれは非効率です。

HLSは最新の20版で、大きな構造の変化がありました。MP4のサポートです。これまで指定されていたフラグメントMPEG2-TSに加え、フラグメントMP4のサポートが追加されました。またPacket AudioとWebVTTも新たなマルチメディア形式として追加されています。フラグメントMP4(fMP4)はやはりISO/IECにより標準化されている形式で、一連のデータが順序付けられた複数のファイルのことで。

このサポートにより、HLSとMPEG-DASHのメディアライブラリを共通化できる可能性が生まれました。フラグメントMP4で用意した一種類のメディアライブラリがあれば、複数のシステムに配信できるようになります。また配信事業者側の観点から言えば、キャッシュ運用の効率化が図られることとなります。HLSであってもMPEG-DASHであっても、データ容

量の多い部分は動画データを格納しているフラグメントMP4ファイル群になるからです。

Appleのこのような動きは、ストリーミング形式の統一を目指す流れに合致しています。そこで提案されているのがCommon Media Application Format(CMAF)です。原案はAppleとMSにより提案され、MPEG(The Moving Picture Experts Group)で議論されています。このようなストリーミング形式の標準化作業は、IETFではなくMPEGで討議する方がより現実的です。コンテナ形式やコーデック分野のエキスパートが集まっているコミュニティがMPEGだからです。

CMAFの規格案には、サブタイトルとして次のような文言があります。

Media Application Format optimized for large scale delivery of a single encrypted, adaptable multimedia presentation to a wide range of devices; compatible with a variety of adaptive streaming, broadcast, download, and storage delivery methods

網羅的に技術をカバーしているように思えますが、CMAFはこれまでのHLSやMPEG-DASHでの成果を受け継ぐ形になっています。逆に言えば、これまでストリーミング形式で課題になってきた事項はすべてカバーされています。

名称	マニフェストファイル	コンテナ
HTTP Live Streaming	m3u8	CMAF <CMAFの内部構造> + CMAF Presentation + CMAF Selection Set (同一コンテンツの異なる要素を複数収容できる;カメラ映像、コーデック、多言語など) + CMAF Switching Set (同一コンテンツを異なるエンコード形式で複数収容できる) + CMAF Track + CMAF Fragments + CMAF Header
MPEG-DASH	MPD	

表-3 HLS、MPEG-DASHとCMAFの関係

CMAFはマニフェストファイルやプレイヤー、配送プロトコルを定義しません。HLSやMPEG-DASHがそれぞれ持つマニフェストファイルから呼び出せる形になっています。

CMAFの構造は階層化されており、複数の言語や異なるビットレートなどに対応したものとなっています(表-3)。

CMAFでメディアライブラリの作成が一本化されれば、コンテンツ作成の手間が大幅に減ります。ユーザに直接的なメリットをもたらすものではありませんが、ストリーミングメディアのさらなる普及のためには必要不可欠な作業だと受け止められているようです。

CMAFの意義は、AppleやMicrosoftが独自路線によらず、業界として大きな流れを生み出すことができるかどうかにかかっているでしょう。このようなフレームワーク、そして議論の際に培われた人的交流があれば、次の課題に対してもよりよい動きができるようになるでしょう。統一されたプロトコルやフォーマットありきではなく、いま動いているものをベースにしつつ更なる改善を目指す形は、イノベーションを起こしやすいとも言えます。

最後に、今後の課題を2つ述べます。

まず、動画視聴開始までのタイミングを短くすることです。現在のライブストリーミングは実時間より概ね30秒かそれ以上遅延が発生しています。これは複合的な要因があり、エンコードにかかる時間やネットワークへのアップロード、そしてプレ

イヤーが再生初期にバッファするセグメントの数などが影響しています。エンコーダや配信サーバ設定で簡単に変えられるものではないため、ストリームを生成する側では制御しにくいのです。しかし、やはりライブイベントはリアルタイムに近い形で再生させたいというニーズがあります。この課題は業界にも広く認識されており、将来的に改善のための提案がなされるのではないのでしょうか。

もう1つがオフライン再生です。特にモバイルでの動画視聴を考えたときに、キャリアによるダウンロード容量制限は大きな壁となっています。モバイルでは動画配信を控える方も多いでしょう。そこで、Wi-Fi接続時に一旦動画をモバイル機器にダウンロードさせてしまい、オフラインでも動画を視聴できるようにするための技術です。

GoogleはYouTubeアプリでオフライン再生するための仕組みを加えました。特定の動画を保存して、48時間までオフライン再生することができるようになっています。この機能は全ユーザには解放されておらず、主に通信インフラが発展途上の国でのみ有効とされています。またHLSもオフライン再生のための仕組みを整えました。こちらはiOSアプリで対応コードを作り込む必要があります。

いずれも一部のアプリや仕組みで対応が始まっていますが、本格的に普及が始まった段階で標準化される必要があるものです。こうした動きをフォローアップできるかどうか、業界に問われているとも言えるでしょう。



執筆者:

山本文治 (やまもと ぶんじ)

IJ 経営企画本部 配信事業推進部 シニアエンジニア。

1995年にIJメディアコミュニケーションズに入社。

2005年よりIJに勤務。主にストリーミング技術開発に従事。同技術を議論するStreams-JP Mailing Listを主催するなど、市場の発展に貢献。