Keywords you should know. 第160回

知っておきたいキーワード

低遅延配信プロトコルLL-HLS/CMAF-ULL

岡庭大輔†

† JOCDN株式会社 技術部

"Low Latency Protocol LL-HLS & CMAF-ULL" by Daisuke Okaniwa (Technology Department, JOCDN Inc., Tokyo)

キーワード：HTTPStreaming，低遅延（Low Latency），HLS，DASH，CMAF，LL-HLS

まえがき

インターネット上でのライブ中継は，さまざまなインターネットデバイスを通じ，誰もがどこからでも手軽に視聴や配信が可能となり，かつてないほど身近で魅力的な存在となりました．そしてスポーツの試合や音楽フェスティバル，政治討論や企業の発表会，さらには個人の趣味や日常をシェアするもの等，多岐にわたるジャンルでライブ中継は活用されています．このさまざまな方面で利用，活用されているインターネット上のライブ中継の魅力をさらに増すべく，日進月歩でさまざまな試みや技術の開発がなされている中で，本稿ではライブ配信における配信遅延改善に関するアプローチであるLL（Low Latency）-HLS，CMAF-ULL（Ultra Low Latency）両規格に関して紹介します．

配信遅延とストリーミングプロトコル

映像を視聴者へ中継する上で発生する遅延の発生原因はさまざまありますが，インターネットでのライブ中継においては伝送が主な発生原因であると言って過言ではありません．それはなぜでしょうか．現在インターネットでのライブ中継のみならずVoD（Video on Demand）等々でも利用されているストリーミングプロトコルは，Webインフラを用いて配信可能なHTTP Streamingが主流になっています．HTTP Streamingはメディアデータをセグメント化しファイル的に扱えるようにすることで，擬似的にストリーミングをWebインフラ上で実現しているのですが，現在主流のHTTP Streamingプロトコルであるを例に挙げると，標準的なセグメントサイズは6～10秒程度とされているので，ライブ配信ではこのセグメントサイズ分は最低でも伝送遅延が発生してしまいます．さらにではこのセグメントファイルを不安定な通信品質環境も考慮したストリーミング再生の安定性の確保のため，3セグメント以上バッファすることも規定されています（図1中段）．そのため，最低でもセグメント長x3セグメント分の伝送遅延が仕組み上発生する構図となっています．

☞

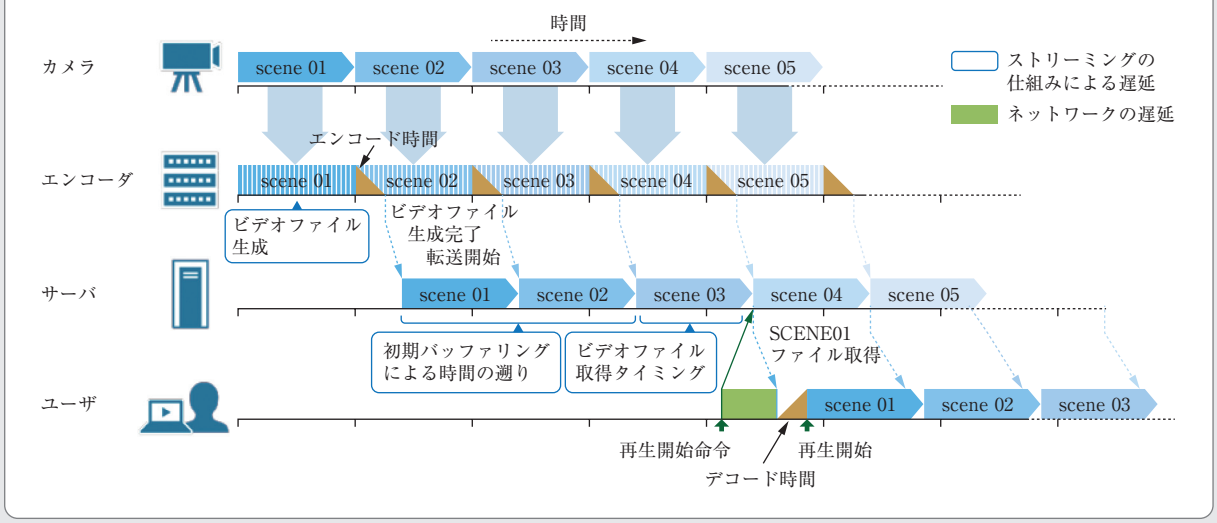

図1　動画配信における遅延発生の仕組み

ファインチューニングによる低遅延配信化とその問題点

セグメントサイズとバッファサイズが原因となり発生する伝送遅延ですが，もちろんそれらを規格の範囲内でファインチューニングすることにより，ある程度伝送遅延を抑制することは可能です．チューニングの内容としては，バッファサイズはHLSの場合3セグメント以下にすることは規格上許可されていないため，セグメントサイズを可能な限り縮めるということとなります．このアプローチで伝送遅延を大凡6～10秒程度に削減することが可能ではあるのですが，セグメントサイズを縮小化した弊害として，配信サーバに対するリクエストが急増しプロトコルオーバヘッドとサーバ負荷の増加が問題となります（図2右）．また規格上は許可されているとは言え，プレーヤーの実装的に標準的なセグメントサイズから余りにも逸脱した小さいセグメントサイズでは再生互換性に問題が生じる可能性もあります．

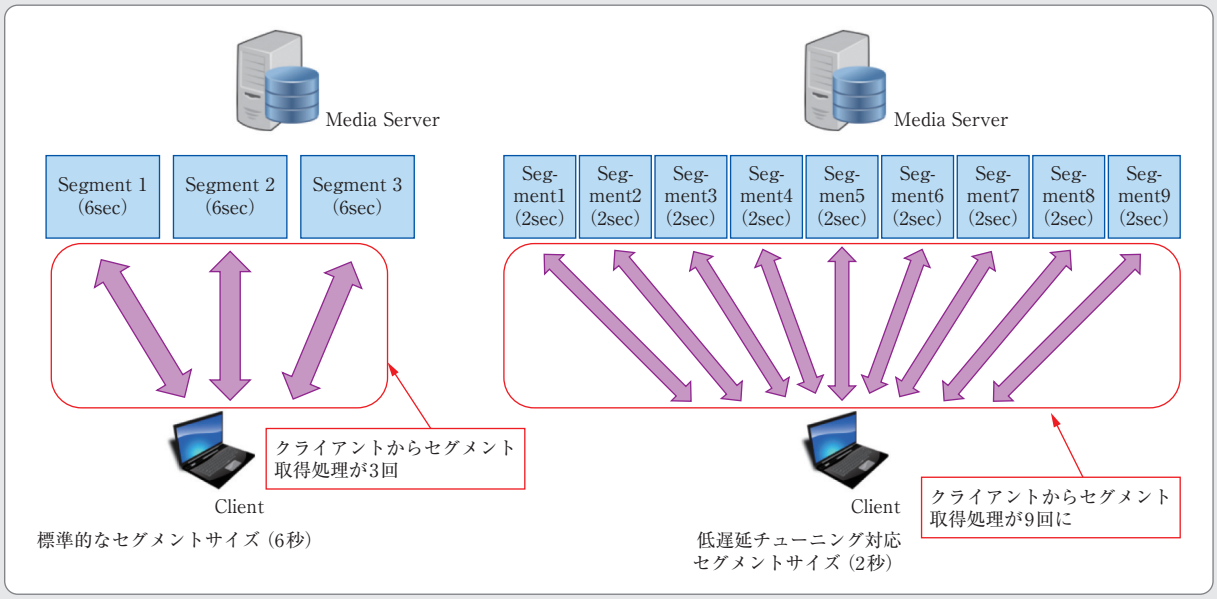

図2　セグメンテーションされた18秒間のストリームデータ取得処理回数比較（標準vs低遅延チューニング）

CMAF-ULL / LL-HLS

このように従来のHTTP StreamingプロトコルであるHLS/LL-HLSをファインチューニングすることにより起こり得る問題点を解消，および効率化することを目的に規格策定された低遅延配信対応プロトコルがCMAF-ULL，LL-HLSになります．CMAF-ULLはDASH-IF（Dynamic Adaptive Streaming over HTTP-Industry Forum）がMPEG-DASH（Moving Picture Experts Group-Dynamic Adaptive Streaming over HTTP）仕様を拡張する形で2017年に，LL-HLSはAppleがHLSの仕様拡張する形で2019年にそれぞれ規格策定を行いました．両規格ともHTTP Streamingの汎用的なWebインフラを用いた配信が可能なメリットを活かした上で，3秒前後の伝送遅延の実現を目指した規格と成っており，従来の規格から，主にメディアデータセグメントの転送単位を改良することでそれを実現しています（図3）．

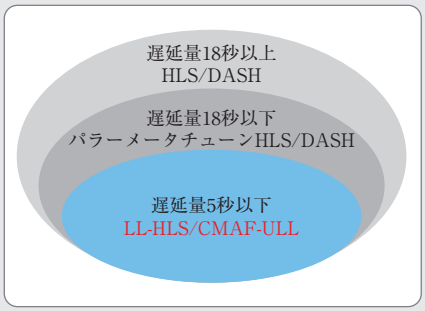

図3　HTTP Streamingプロトコル遅延量ターゲット

CMAFチャンクとパーシャルセグメント

CMAF-ULLとLL-HLSはどのようにメディアデータセグメントの伝送単位を改良しているのでしょうか．従来のプロトコルではメディアセグメントの伝送は，上述の通り，あるまとまった単位で生成された後，転送可能になるため，その分伝送遅延が確実に発生してしまうという問題がありました．このウィークポイントをCMAF-ULLではCMAFチャンク，LL-HLSではパーシャルセグメントと呼ばれるセグメント生成中に伝送可能なデータ単位を新たに設け，それを伝送することで，セグメントの伝送開始するタイミングを大幅に早め，伝送遅延の削減を実現しています．

CMAFチャンク，パーシャルセグメント，共に概念的にはほぼ同等なもので，これまで転送単位であるセグメントには必ずIDRを含める必要があったところ，その制約を設けずより小さく分割可能としたデータの単位になります（図4）．違いは後述もしますがCMAF-ULLとLL-HLSのチャンクデータの伝送方式の違いから，CMAF-ULLではセグメントをCMAFチャンク単位で逐次生成し，LL-HLSではセグメントとは別にチャンクデータを実ファイルとしてパーシャルセグメントを伝送可能としています（図5）．

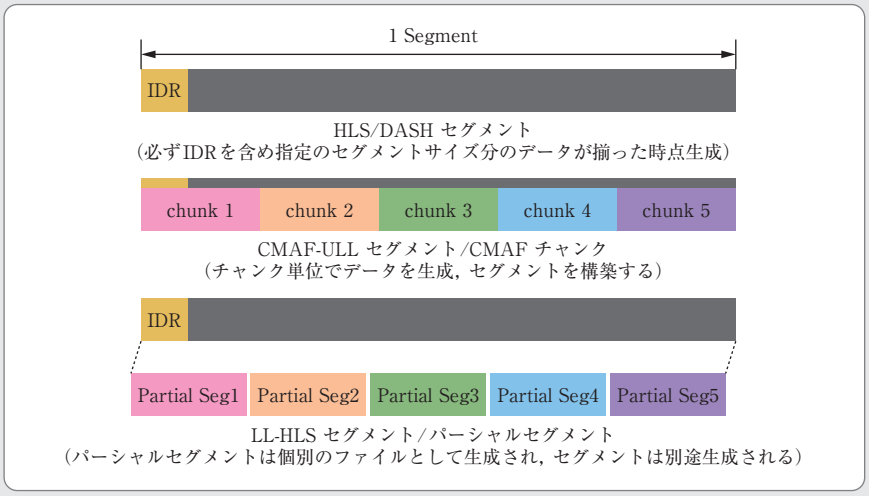

図4　HLS/DASH/CAMF/LL-HLSセグメント，チャンク構造

```
#EXTINF:6.000,
seg_1_12381_video_s5HJkfEA_llhls.m4s
#EXT-X-PROGRAM-DATE-TIME:2022-12-01T12:50:23.298+09:00
#EXT-X-PART:DURATION=1.000,URI="part_1_12382_0_video_s5HJkfEA_llhls.m4s",INDEPENDENT=YES
#EXT-X-PART:DURATION=1.000,URI="part_1_12382_1_video_s5HJkfEA_llhls.m4s",INDEPENDENT=YES
#EXT-X-PART:DURATION=1.000,URI="part_1_12382_2_video_s5HJkfEA_llhls.m4s",INDEPENDENT=YES
#EXT-X-PART:DURATION=1.000,URI="part_1_12382_3_video_s5HJkfEA_llhls.m4s",INDEPENDENT=YES
#EXT-X-PART:DURATION=1.000,URI="part_1_12382_4_video_s5HJkfEA_llhls.m4s",INDEPENDENT=YES
#EXT-X-PART:DURATION=1.000,URI="part_1_12382_5_video_s5HJkfEA_llhls.m4s",INDEPENDENT=YES
```

パーシャルセグメントの情報はプレイリスト上で拡張タグを用いて記述される

図5　LL-HLSプレイリストパーシャルセグメント拡張タグ

CMAF-ULLでのチャンクデータの伝送

パーシャルセグメントというチャンクファイルの実態が存在するLL-HLSでは，これまでの通りHTTPで完成したパーシャルセグメントをバルク伝送すれば良い訳ですが，CMAF-ULLではセグメントが逐次生成される形になるため，その方法で伝送してしまうと中途半端なセグメントが伝送されてしまう可能性があります．CMAF-ULLでは，この問題をHTTP/1.1でストリームデータ（事前に全体のサイズが不明なデータ）の伝送をサポートするために定義されているChunked Transfer Encodingを用いることで，チャンク単位で逐次生成されるセグメントを，チャンク単位でセグメントの生成が完了するまで逐次伝送することで，その問題を解決しています（図6）．ただし，Chunked Transfer Encodingを利用するにあたり，若干インフラ側での対応が必要になることがあるので注意が必要です．配信経路上のCDNやWebプロキシー等，HTTPを扱うコンポーネントがこのChunked Transfer Encodingに対応していないと，思わぬ視聴障害が発生する可能性があります．

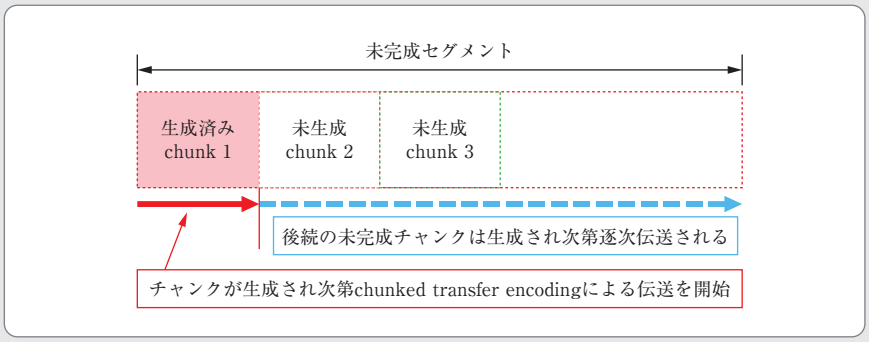

図6　チャンクデータ伝送イメージ

LL-HLSでの配信最適化

伝送するメディアデータの単位が小さくなったことによるプロトコルオーバヘッドは，CMAF-ULLではチャンクデータをChunked Transfer Encodingを用いて伝送することで，実質従来のDASHの配信とHTTPのシーケンス上はほぼ変化はありません．しかし，LL-HLSでは事情は異なりチャンクデータを個別のファイルとして扱うため，従来6秒分の映像データは1回のHTTP上の処理で済んでいたところが，パーシャルセグメントを例えば1秒で設定すると，6回のHTTP上の処理が必要となり，データプレーンのみでも6倍以上のプロトコル上のオーバヘッドが生じます．☞

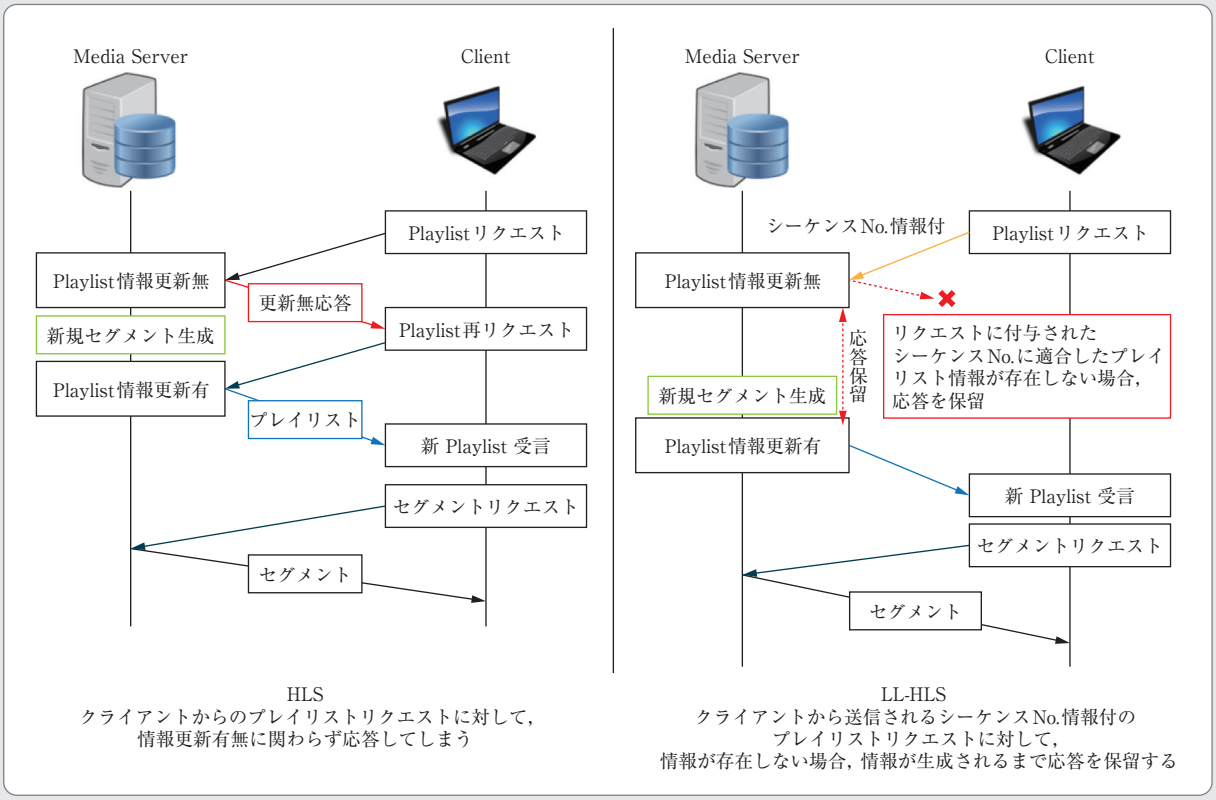

図7　HLS/LL-HLSプレイリスト取得シーケンス比較

☞ またコントロールプレーンにおいてもプレイリストの取得回数の増加，パーシャルセグメントの情報を含んだことによるプレイリストのデータサイズの増加等，従来のHLSのパラダイム内で収めようとすると，オーバヘッドがパフォーマンス上の問題になりかねないレベルとなることが想定されます．Appleはこれらのオーバヘッドの要因に対して従来のHLSの規格を改良することで，パーシャルセグメントを扱う低遅延配信に対応したプロトコルの最適化を行っています．最適化の内容としては，基本的なところではトランスポートプロトコルであるHTTPのバージョンをHTTP/2の利用を推奨することで，HTTPレベルでのプロトコルオーバヘッドの最小化を図っており，また従来のHLSにおいても問題として見られていたプレイリストの取得動作に関しても（図7左），プレイリストの更新が行われない限り，無駄なプレイリストの配信が行われないように，視聴クライアントからリクエストは一時的に保留されるような最適化がなされています（図7右）．これら最適化を行った場合でもLL-HLSのプロトコルオーバヘッドは目に余るところはありますが，最適化されなかった場合，そもそもLL-HLS自体が上手く機能しないことも考えられるため，必要かつ効果的なものであると言えるでしょう．

CMAF-ULL vs LL-HLS

低遅延配信を行う上で，よく「CMAF-ULLとLL-HLSどちらが優れているのか」というお問合せを受けることがあります．どちらを採用した場合でも実現できるものとしてはほぼ同等となるため，結論としてはどちらも同じということになってしまうのですが，CMAF-ULLにはchunked transfer encodingを利用したオーバヘッドの少ないチャンクデータの伝送，LL-HLSにはインフラの互換性を重視した伝送やAppleデバイスの再生環境のネイティブサポート等，両規格とも優れた面が存在するため，どちらが優れているというより，利用シーンに応じた選択が必要ということになるかと思われます．

図8. CMAF-ULL/LL-HLS配信環境をサポートするOSS

むすび

低遅延配信には今回解説させて頂きましたCMAF-ULL, LL-HLS以外にも，それを実現する技術は存在しますが，CMAF-ULL, LL-HLSはこれまでDASH, HLSで利用してきたインフラを利用することができ，再生環境も後方互換性が確保されているというメリットがあります．また標準規格であるため，その利用にコストも特別何かが掛かるということはありません．インターネット中継での視聴体験をさらに魅力的にするためにCMAF-ULL, LL-HLSによる低遅延化の導入の敷居は低いと言えるのではないでしょうか．

（2024年9月26日受付）

参 考 文 献

1）draft-pantos-hls-rfc8216bis（HLS2，https://datatracker.ietf.org/doc/html/draft-pantos-hls-rfc8216bis
2）Low-Latency Modes for DASH，https://dash-industry-forum.github.io/docs/CR-Low-Latency-Live-r8.pdf

岡庭 大輔（おかにわ だいすけ）
大手配信サイトの立ち上げおよび運用に従事後，2008年，（株）インターネットイニシアティブ（IIJ）入社．映像配信関連のサービス開発，運用，技術検証などに携わる．2020年よりCDNサービスを提供する子会社JOCDN（（株））へ出向．CDNサービスの運用，サポートをメインに，新機能の検討や検証に従事．