ABR in MoQ

Boulder Interim, Feb 2026

Ian, Suhas

Proposal Overview

Add **Track Groups** to indicate to the Publisher to only deliver one of the Tracks in the Group

Add **Required Bandwidth** to indicate to the Publisher how much bandwidth is needed to play the Track

Add **Epoch** and **Syncstamp** functionality to align Tracks delivery

Some Key Use Cases

SWITCH up or down in quality between two representations

Choose optimal Track of N alternative representations

Correctly prioritize sending Audio and Video

‘Fairness’ between multiple video Track Groups

Ability to prioritize a Track Group above/below others

Track Groups (aka Adaptation Sets)

Publisher indicated
Could be in the same Namespace, but not required

Track Extension Header: ALT_TRACK_GROUP

Tells Publisher to only send one Track in the group

Required Bandwidth

Indicates the expected bandwidth this Track will need

For fixed bitrate video, this might be the bitrate

For variable bitrate video, this might be the max bitrate

Extension Header: REQUIRED BANDWIDTH

Can be used in Objects to override the Track value

Epoch

Synchronizing time is difficult, Tracks with the same **Epoch** have aligned Syncstamps and/or Group/ObjectID

EPOCH is a Track Extension Header, value is an opaque sequence of bytes

WARP alignment uses Render Groups in the catalog

> "SHOULD be able to cleanly switch between time-aligned media tracks at group boundaries."

Example: Enables syncing Audio and Video delivery

Syncstamp

Called Syncstamp because might not be an absolute time

Sync delivery of Tracks within a Track Group

Syncs delivery across Track Groups - ie: Audio + Video

- Publisher knows if alternative Track is ‘live’
- Publisher can ensure no Subscription gets behind

SYNCSTAMP is an Extension Header

- Track: If present, indicates time since UNIX Epoch
- Object: Indicates time since begining of the Track

If not available, fall back to Group ID and Object ID

Selection and Prioritization

If a Track is Selected in a Group, equivalent to Forward=1

Within a Track Group, the highest (lowest numerically) priority is preferred as bandwidth permits

If more bandwidth is available and there are multiple Track Groups, pick the Subscription with the highest numerical priority value to upswitch

- This attempts to achieve some type of Fairness

Once a Subscription is Selected, Object and Subgroup Prioritization work like normal

Prioritization Example

With multiple groups of Subscriptions, how does a publisher decide which Subscription to upswitch?

Example

A VC where there's a Subscription that's the focus in higher quality, with other participants in lower quality

Proposal

Use the priority to indicate how important Track Group is. If one track group's

Relevant Issues (ABR Label)

Can we keep two subscribes in sync? #475

Sender Side ABR #259 #44

Track Switching at live edge is difficult #1101

Further Refinements

Hysteresis - Avoid switching qualities too often

Can’t switch more often than Group boundaries

But switching every Group can be a poor user experience

Proposal: Add a *MinSwitchTime* Message Parameter on Subscriptions

Quality Extension Header

There might be cases when re-using the Priority of a Subscription is suboptimal

In these cases, a QUALITY Track Extension Header could be specified by the Original Publisher

Similar to Priority, but it states a fixed quality ranking between tracks in a Track Group

When QUALITY is present, it is used for Track Selection

Top N Filters and ABR

Top N is applied first

Then ABR Track Selection

Then MoQ Priorities

Let's say I have **top 4 tracks** being chosen to be sent to a subscriber. All the tracks have **same subscriber priority**.

Also each of those tracks have 3 alternate representations:
Track A1, A2 A3, Track B1, B2, B3, Track C1, C2, C3, Track D1, D2, D3

Now, based on downlink bwe, let's consider A1, B1, C1 and D1 are being sent and they represent high quality representations.

At some point, when congestion occurs, we want subscribers to express degradation mode

One example is round-robin, where D1 → C1 → B1 → A1 → D2 → C2 → B2 → A2 ... is the order that's followed

Another example is strict-priority, where D1 → D2 → D3 → C1 → C2 → C3 → .....

The Client Can/Will Help in complex cases

For live sports, the client doesn't

If client priority is identical, then try to 'fair share' between the streams

- Numerical optimization?
- One byte per Subscription - Utility value

Greedy algorithm - Up and Down single track

- Need to do it in one pass over all Subscriptions

Theory: Ordered list of tuples containing tracks

TrackRenderSet && Adaptation Set

SWITCH is similar to this proposal

SWITCH is similar to a **Track Group** of size 2

Issue #1354 (formerly #1101) proposes a SWITCH message to enable more seamless client side ABR

RenderSet ID (100) , AdaptionRank RR, Lowest First