ABR in MoQ

IETF in Montreal, November 2025
Ian Swett

Observations

Both fixed quality ABR and adaptive encoding are typically last hop today

- So WARP or another extension could improve ABR

The existing design is OK for Client Side ABR

- No support for Sender/Server-Side ABR

YouTube ABR dev thought the MoQ name was confusing:
"I wish they would rename the RFC to something like PSOQ (Pub/Sub Over Quic) because it's really just a pub/sub protocol."

What belongs where? MoQT vs WARP

When going through issues, it’s worth asking where the right place is?

- ie: Is it sufficiently media-specific or generally useful?
- If we don’t do it in MoQT, will it be painful in WARP?

Client Side ABR

Track switching in the future is quite good

- Can use another Track & priorities to probe for BW

There are cases you get extra Objects for ~1 RTT

An Approach: (from #1101, which has nice diagrams)

- Existing Subscription A
- SUBSCRIBE B next_group
- SUBSCRIBE_OK returns Largest Location
- Send SUBSCRIBE_UPDATE to end A at end of
  - Largest Location.Group

Server Side ABR

High-level Requirements

Server needs to know which Tracks are possible

Ie: Don’t consider AV1 if the client doesn’t

Server needs to know expected bitrate of Tracks

Higher quality could be 1.25x or 2x or 3x bitrate

Server needs to know when it can switch between Tracks and how to align them

Temporal Scalability Changes the Approach

When temporal Scalability is present, can use priorities to run much closer to full bandwidth utilization

When not, even a temporary bandwidth drop causes a rebuffer

As a result, Meet devs indicated the current MoQ design was fairly close, YouTube devs did not

Relevant Issues (ABR Label)

Bandwidth Estimation/Discovery

- Allow padding streams #534
  - Also: Can we ask the relay to probe? #370
- Should a priority indicate Less than Best Effort #471

Switching Tracks and Sync

- Can we keep two subscribes in sync? #475
- Sender Side ABR #259 #44
- Track Switching at live edge is difficult #1101

Server Side ABR Challenges

Today’s Priorities don’t achieve the desired result:

- Want one of 320p, 480p, 720p, 1080p, etc, but not all
- One suggestion on #259 was ‘group=1’ as a Param
  - in multiple SUBSCRIBEs to indicate this
- Average Track bitrate is very helpful

Do we want this in MoQT? WARP?

- Somewhat *similar* to SUBSCRIPTION_FILTER…

Server Side ABR Challenges

Keeping Subscribes in Sync #475 (Timestamps)

- WARP suggests approaches using Group IDs
- Do we also want a ‘Publish Timestamp’ Object
  - Extension to keep Tracks in sync?
- Could be useful for keeping Audio & Video in sync

Timestamps seem more generally useful

- Again, timestamps are only for MoQT delivery
- NOT media timestamps

Bandwidth Discovery (Client ABR)

Allow padding streams #534
Can we ask the relay to probe? #370

Today, a Subscriber can ‘probe’ by requesting another Track at a lower priority - Could be next higher quality

Given that, do we need more?

1. Could add a well-known padding Track
2. Could add a Parameter to request probing on Sub
3. Could add a control stream to request probing
4. Do nothing

If we do something, does it need to be in MoQT?

Bandwidth Discovery (Client ABR)

Should a priority indicate Less than Best Effort #471

Could make probing safer

1. Specify one (or more) priorities as LBE (ie: 255)
2. Specify a 'headroom' param in SUBSCRIBE
    a. Subscription would never consume the last X bytes or Y% of Congestion Window
3. Do nothing

Switching Challenges (Client Side ABR)

#1101 Discusses challenges switching Tracks

- If SUBSCRIBE_UPDATE arrives late, can get extra Objects from A
  - Want to prioritize B in that case
- If A is getting behind, still need to get full Group
  - Want to prioritize A in that case to avoid rebuffers

Editor's Proposal: By sending the SUBSCRIBE+SUBSCRIBE_UPDATE at the same time with an explicit Group ID in the future, prioritization is easy

Alternative: Add a SWITCH message