

Client-initiated server-side ABR using a new ABR Parameter

Will Law

IETF moq workgroup interim, Boulder Feb 2026

Problem Statement

Client-initiated server-side ABR functions when the server decides, at the client's request, which of a subset of tracks should be sent to the client.

1. There is a finite set of tracks (typically in the range 2-6)
2. Throughput is used as the basis for deciding which tracks to forward. Other metrics may be included in a more complex implementation.
3. The server knows no intrinsic information about the tracks. The client does know something about the tracks, via a catalog or some other out-of-band mechanism.
4. The client usually initiates & terminates this behavior.
5. The change may occur at group or object boundaries.
6. We want to use existing MOQT constructs as much as possible and not invent new messages.

General approach

1. We define a new ABR parameter which is attached to a SUBSCRIBE.
2. This parameter:
 - a. allocates the subscription to a ABR group
 - b. defines a minimum throughput necessary to deliver that subscription,
 - c. defines at which boundary the change is allowed to occur
3. For a given ABR group, the relay only forwards the track with the highest bitrate smaller than the server's estimate of throughput on the connection. If no subscriptions satisfy this constraint, then the track with the lowest bitrate is forwarded.

Define a new parameter to allocate a Subscription to an abr-group

ABR-GROUP Parameter

The ABR-GROUP parameter (Parameter Type 0xXX) MAY appear in a SUBSCRIBE or SUBSCRIBE_UPDATE message. It has 4 fields:

```
ABR-Group {  
    Type (i) = 0xXX,  
    Length (i)  
    ABR group identifier(i), // the ABR Group identifier. 0 removes the subscription from ABR.  
    Minimum throughput (i), // the minimum selection throughput, expressed in kbps,  
    [ABR Group Size (i), // the number of tracks in the group.  
    Boundary (i)] // 0 = switch at group boundaries 1 = switch at any object boundary  
}
```

Example

Catalog describes the following tracks:

example.com/broadcast/123/video-high 3000kbps

example.com/broadcast/123/video-medium 2000kbps

example.com/broadcast/123/video-low 1000kbps

example.com/broadcast/123/audio-high 256kbps

example.com/broadcast/123/audio-medium 128kbps

example.com/broadcast/123/audio-low 64kbps

SUBSCRIBE ("example.com/broadcast/123/video-high"), ABR (1,3600,3,0)

SUBSCRIBE ("example.com/broadcast/123/video-medium"), ABR (1,2400)

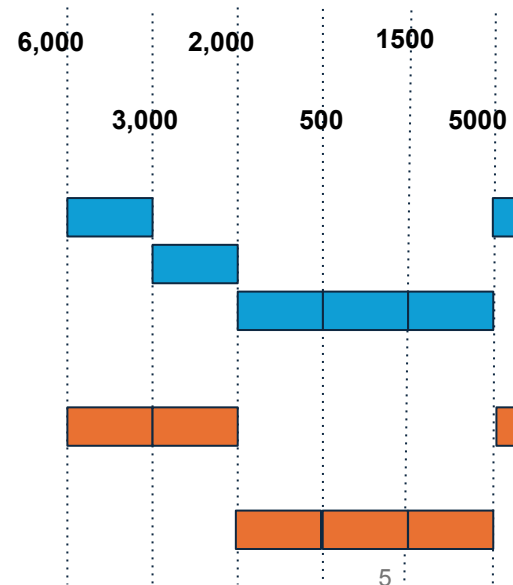
SUBSCRIBE ("example.com/broadcast/123/video-low"), ABR (1,200,)

SUBSCRIBE ("example.com/broadcast/123/audio-high"), ABR (2,4000,3, 0)

SUBSCRIBE ("example.com/broadcast/123/audio-medium"), ABR (2,3000)

SUBSCRIBE ("example.com/broadcast/123/audio-low"), ABR (2,76)

Throughput varying over time



What if we want to simplify the client API?

We can define, as part of the MOQT spec, normative **extensions** which define an **ABR group**, the **threshold bitrate** of the track and the **change boundary**. These would be set by the publisher.

The client would then subscribe to the tracks as normal, perhaps with a new bit flag in the subscription to turn **auto-abr** on or off.

This capability could exist in parallel with the client-defined **ABR parameter**.

Questions:

1. Do we need to define over what time range the throughput is calculated? Last object?, Last Group, Last N milliseconds?
2. Can we limit it to only switching at group boundaries? The definition of a group is that it is a random-access point for a track. Implementers can adjust their use of groups to comply with this scheme (i.e don't put everything in one group if you want SS ABR switching).
3. Implicit in the design is that the relay must wait until it has N objects available across the ABR group, before it releases one of them. To avoid ABR failing because one track is delayed, do we want to set a max time limit the relay is willing to wait for jitter alignment before making a selection?
4. Should the client provide the switching bitrate to the server, or the raw bitrate so a server-side algorithm can make the decision?

Thank you for your time.