Updating MoQT Priorities based on Implementation Experience

Terminology and Assumptions

Using “track” to mean “subscription”

Object Priority == Subgroup Priority:

- The priority field is carried in STREAM_HEADER_SUBGROUP
- The priority of all objects in the subgroup is the same
- This is only settable by the publisher
- Ties in the same group fall back to subgroup ID

Datagrams behave like a 1 object subgroup

MoQT Draft-05 Priorities

Select the next ***track*** based on subscriber priority,

If 2+ tracks are equal, select the **track** with the highest object priority
If 2+ tracks are still equal, implementation choice

Select the next ***group*** from that track based on Asc or Desc preference

Select the next ***object*** from that group based on object priority*

*Draft-05 predates subgroups

MoQT Draft-05 Priorities

Select the next ***track*** based on subscriber priority,

If 2+ tracks are equal, select the **track** with the highest object priority

If 2+ tracks are still equal, implementation choice

Select the next ***group*** from that track based on Asc or Desc preference

Select the next ***object*** from that group based on object priority

This is computationally expensive.

It can change when receiving new data and when done sending data

Proposed Fix #1 - Explicit Publisher Subscription Priority

Don’t set the relative priority of a track implicitly by `max(track-object-priorities)`

Instead, publisher sets the relative priority of track **explicitly**

Rest of the algorithm remains the same

https://github.com/moq-wg/moq-transport/issues/585

Proposed Fix #1 - Explicit Publisher Subscription Priority

Select the next ***track*** based on subscriber priority,

~~If 2+ tracks are equal, select the **track** with the highest object priority~~

If 2+ tracks are equal, select the **track** with the highest publisher priority

If 2+ tracks are still equal, implementation choice

Select the next ***group*** from that track based on Asc or Desc preference

Select the next ***object*** from that group based on publisher priority

Proposed Fix #2 - Get rid of track priority

Select the next ***subgroup*** based on subscriber priority + subgroup priority,

If 2+ subgroups have the same priority,

- Within the same *track*,

  select the next ***group*** based on Asc or Desc preference

- Within the same *track and group*, select based on subgroup ID

- In *different tracks*: Implementation Choice

https://github.com/moq-wg/moq-transport/pull/518

How are these proposals different?

1. Retroactivity - when the priority of a track changes, does it impact data in previous groups waiting to be sent?

   Status Quo: Retroactive for increase
   Proposal 1: Retroactive for increase and decrease?
   Proposal 2: Not retroactive

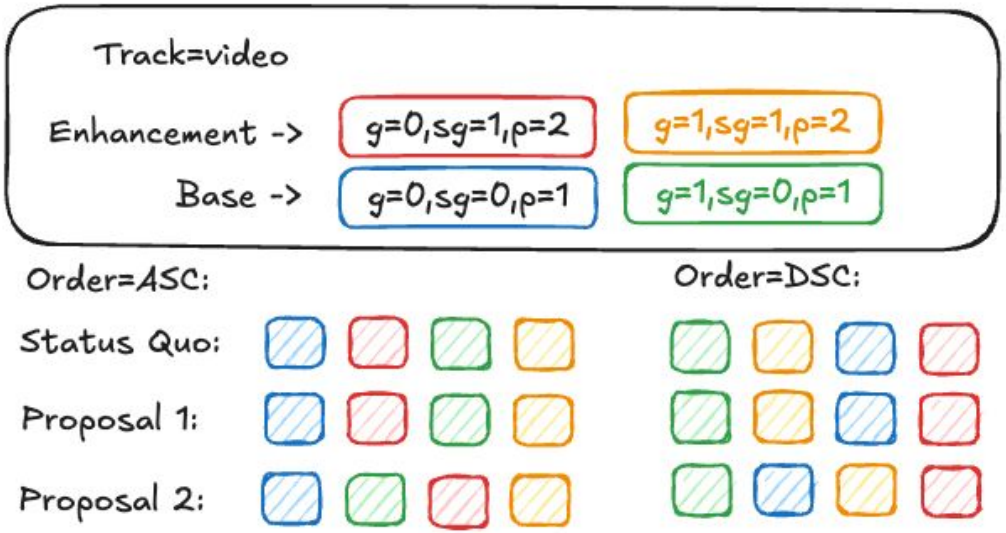
2. What are we prioritizing?

   Proposal 1: Tracks relative to each other
   Subgroups relative to each other in the same group
   Proposal 2: All subgroups of all tracks
   Status Quo: Subgroups relative to each other in the same group, which implicitly prioritizes tracks

3. What group orders are possible?

   Status Quo/Proposal 1: Ascending or Descending only
   Proposal 2: 8-6-7-5-3-0-9

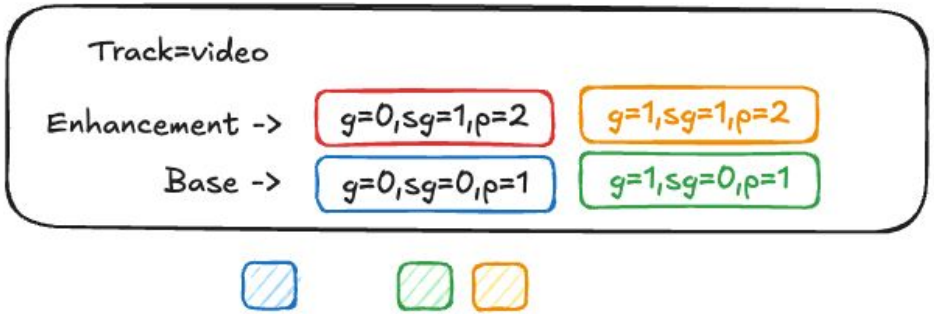
Example



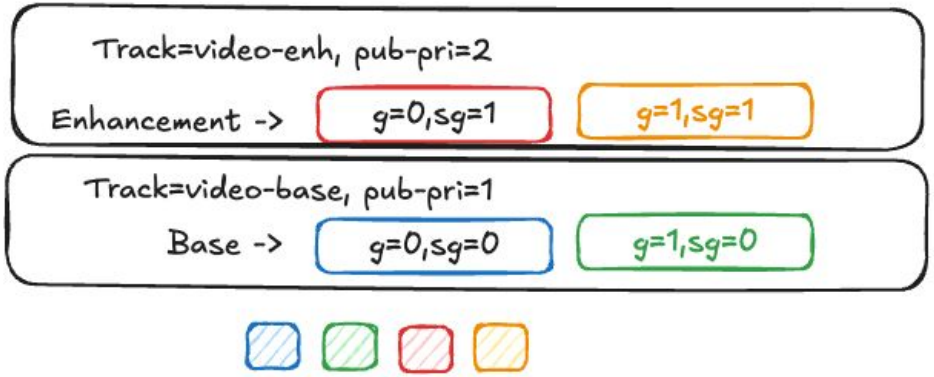
Either proposal 1 or prospal 2 order could be valid

Prioritize g=1 base with Proposal #1

Use delivery timeout

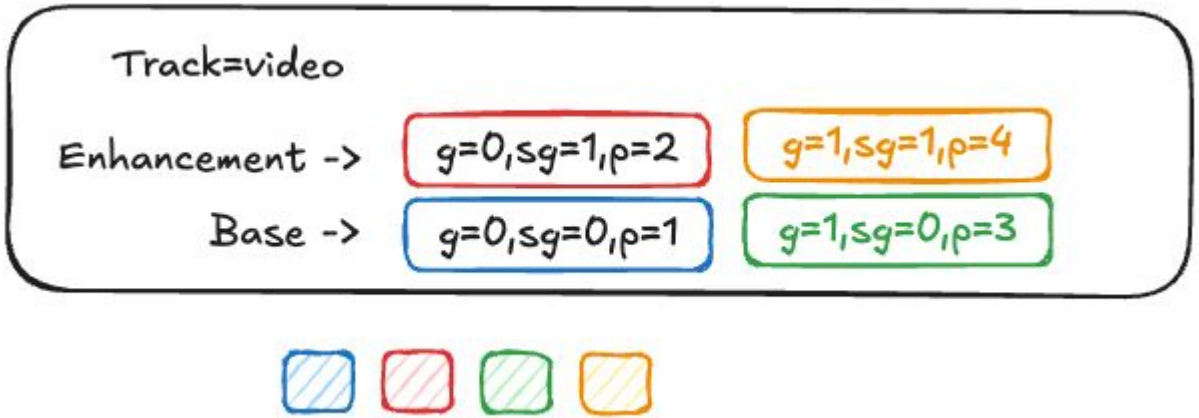


Represent layers with tracks instead of subgroups



Prioritize g=0 enhancement with Proposal #2

Use unique priorities for every subgroup



If you have N prioritized tracks, with M prioritized layers per track, you need M*N priorities per group

Discussion

1. Should we keep the status quo?
2. If not, Proposal #1 or Proposal #2