MOQ Filters

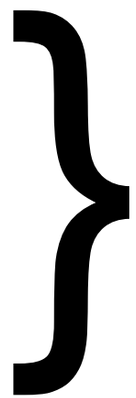
PR#1518 (was #1401)

Mo Zanaty

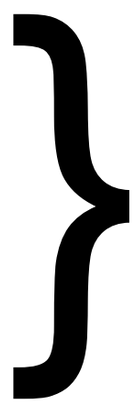
IETF MOQ Interim - May 11, 2026

Motivation - Use Cases for Filters

1. Video scrubbing keyframes: ObjectID=0
2. Video base layer: Subgroup=0
3. Multi-stream fetch: Subgroup=i
4. Specific codec/encoding: PropertyT=V
5. Critical alerts in log: Priority=0
6. Top active speakers in meetings: N=4
7. Top activity in security cameras: N=16
8. Top scoring players in online game: N=2
9. Top bid in auction/market: N=1
10. Top quality within bandwidth limit: N=1



Range Filter



Track Filter

MOQ Filter Design

PR#1518 reduced scope of #1401

- Setup options
  - Removed MAX_TRACKS_DESELECTED
  - Redefined MAX_FILTER_RANGES as **total**
- Range filters
  - **Removed Location and Group filters**
- Track selection filter
  - **Removed MaxTracksDeselected, MaxTimeSelected**
- Improved wire encoding, clarified interactions

Setup Options

MAX_FILTER_RANGES

- Limits **total** ranges in all filters for a sub/fetch
- For Subgroup, Object, Priority, Property filters
- If omitted, default is 0, no filter supported

MAX_TRACKS_SELECTED

- Limits MaxTracksSelected (top N) in Track filter
- If omitted, default is 0, no filter supported

Range Filters

Parameters in Subscribe, Subscribe_Namespace, Request_Update, Publish_Ok, Fetch:

```
SUBGROUP_FILTER { Type=0x25, Length, Set, Range… }
OBJECTID_FILTER { Type=0x26, Length, Set, Range… }
PRIORITY_FILTER { Type=0x27, Length, Set, Range… }
PROPERTY_FILTER { Type=0x28, Length, Set, Property Type, Range… }
          Range { Start, End }
```

- **AND** all filters with the **same Set**, then **OR** the results of **all Sets**.
- Length=0 (no ranges) removes that filter type in Request_Update.
- End is optional in the last range for no End.

Range Filter Examples

Video keyframes (Object ID = **0**):
ObjectID Filter { Type=0x26, Length, 0, **0, 0** }

Video base layer (Subgroup ID = **0**):
Subgroup Filter { Type=0x25, Length, 0, **0, 0** }

Video enhancement layers 1 and 3 (Subgroup IDs **1** and **3**):
Subgroup Filter { Type=0x25, Length, 0, **1, ~~1~~0, ~~3~~2, ~~3~~0** }

Property Type T, Value **>= V** (no end):
Property Filter { Type=0x28, Length, 0, T, **V** }

Property Types T1, T2, Values >= V1 **OR** V2 (no end):
Property Filter { Type=0x28, Length, 0, T1, V1 }
Property Filter { Type=0x28, Length, **1**, T2, V2 }

Remove any filter:
<any> Filter { Type=<any>, Length=**0** }

Operation	[S,E]
x==a	[a,a]
x<=a	[0,a]
x>=a	[a,-]
a<=x<=b	[a,b]
a< x< b	[a+1,b-1]
x< a	[0,a-1]
x> a	[a+1,-]
x!=a	[0,a-1,a+1,-]

Track Selection Filter

Select the top N tracks in a namespace with the highest values in a specified Property Type.

Can be combined with other Filters to limit which tracks and objects are selected, **and it always runs first before others.**

Filter Order

Filter order changed to evaluate track filter first before all other filters.

This may improve relay scalability to allow a single ingress track filter per namespace followed by egress range filters per subscriber.

Track Selection Filter

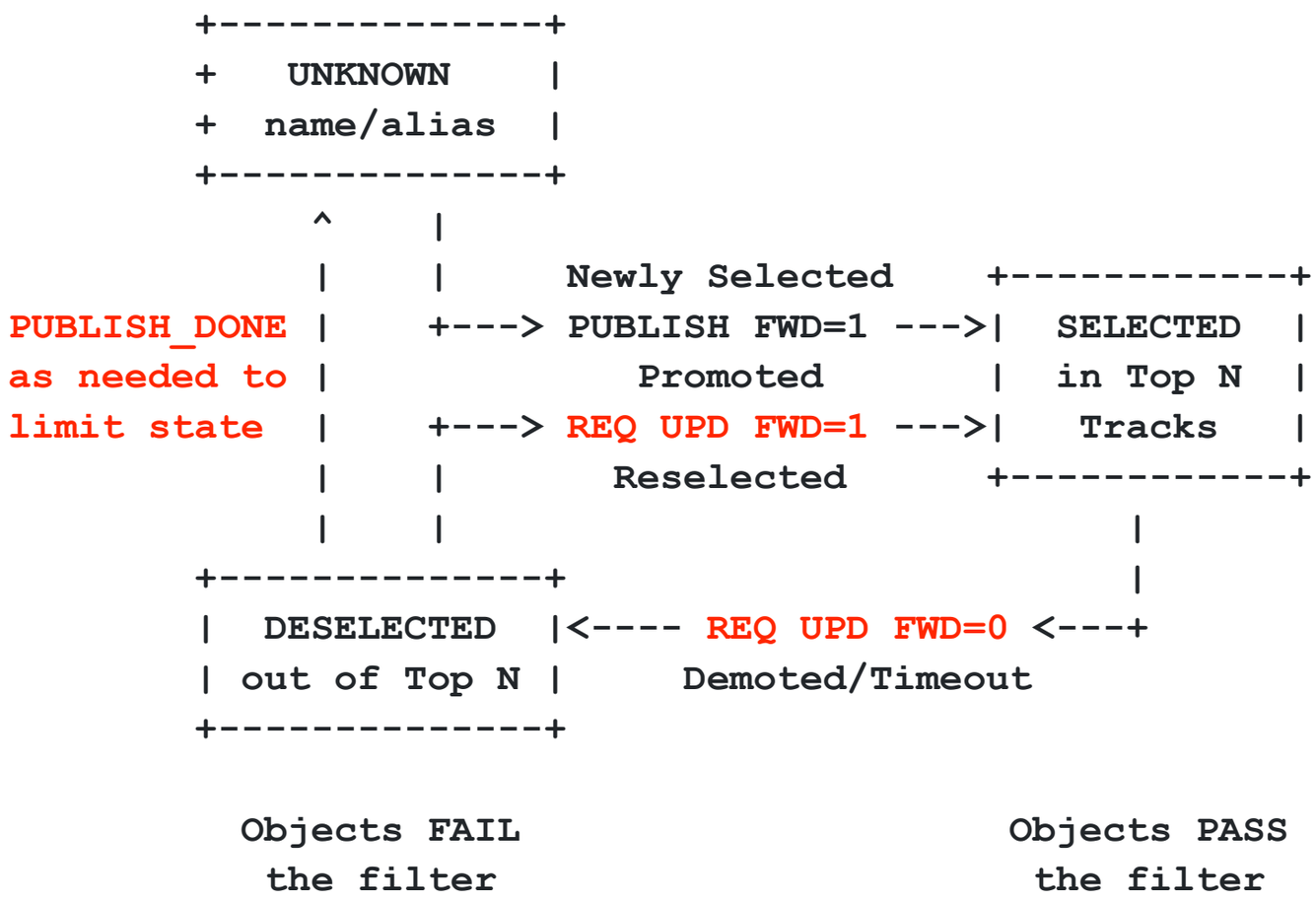
Parameter in Subscribe_Namespace, Request_Update:

```
Track Filter { Type=0x29, Length, Property Type,
               MaxTracksSelected }
```

Removed MaxTracksDeselected. Allow relays to purge old state via PUBLISH_DONE upon local decision.

Removed MaxTimeSelected. Allow publishers to specify it as a Track Property, so it’s uniform for all subscribers.

Track Selection States



Other changes

- Delta encoded range values
- Property filter uses a parameter per type
- Clarified interactions with subgroup gaps
- Added details of track filter selection, deselection, reselection process with guidance for relay protection
  - Signal de/reselection via forward=0/1

Relay Resource Protection in Large Namespaces

Relays SHOULD aggregate and propagate filters upstream on subscriptions, especially namespace subscriptions including track selection filters, to conserve and protect their resources from excessive load. They MAY also impose limits on the number of publishers in a namespace, by rejecting or closing namespace subscriptions with the error NAMESPACE_TOO_LARGE, or CONFLICTING_FILTERS if a track selection filter has non-uniform parameter values across a large number of subscribers, or PREFIX_OVERLAP if different subscribers force an aggregated upstream subscription to overlap.

Track Filter Implementations

Each ~700-1500 lines (excluding tests)

- quicr/laps
  - Live demo at NAB in LV in April
- openmoq/moqx
  - Alan concerned about ranking code complexity
- cloudflare/moq-rs
  - Suhas created PR while codebase is still updating to draft -16

Next Steps

- Merge into MOQT?
- Create a new extension draft?
- Simplify? More implementation experience?
  - Top N < 256 ?
  - Single selector property ?

Backup - All Issues

#2 Relay amplification concerns with top-n
#3 Are subscriber chosen max-deselected and timeout required?
#4 Do we need to signal selected -> deselected transition?
#5 Updating filters should update the joining point
#6 Track selection filters with a "WHERE clause"
#7 Fit for purpose - will a real application be able to use top-n filters?
#8 Evaluating top-n metrics only at the beginning of a subgroup
#9 Relay amplification concerns with disjoint Filters
#10 Objects filters and Subgroup Streams
#11 Can I express "All Tracks with property X" as well as "All Objects in all Tracks in a NS with property X"?
#12 Top-N based on leader track within subscribe namespace
#13 Does MaxTimeSelected refresh on objects without an Object Property but do have the Track Property
#14 Tie breakers
#15 Does reordering objects affect top-N algorithm
#16 Do your own tracks count in the top-N?
#17 Limit to 1 property per SUB_NS prefix