

# MOQT PRs & Issues 125

Alan Frindell

# #1563 Track Properties not Conveyed in Track Status

They probably should be.

Add a properties field to REQUEST\_OK, and let it be empty for the other 3 uses of it?

# #1453 Invalid Varints and #1517 non-minimal encodings

Victor observes that our new varint parser has a tri-state return:

- 1) Success
- 2) Invalid (7 byte varint, or maybe non-minimal encoding)
- 3) Not enough data present, wait for more

If we allow 7 byte varints and continue to allow non-minimal encodings, this collapses to success or need more data.

**Editors' Recommendation:** Allow 7 byte varints and non-minimal encodings

# 30 Second Editorial Bikeshed

Is a MoQ variant called: (PR #1526)

vi64

or

v64

# FILL\_TIMEOUT

How long to wait to fill a gap in a FETCH

Open PR (#1490) Question - is it:

- A) The timeout for *each* upstream FETCH issued by the relay
- B) A “budget” for all upstream fetches needed to satisfy the request;  
→ after exhausting the budget, cached objects are returned, unfilled gaps return Unknown

Example:

FETCH(0-10), cache has only even objects, each upstream FETCH takes 1s,  
FILL\_TIMEOUT=2s

A: gets all objects, total FETCH time 5s

B: gets 0-4,6,8 with 5,7,9 marked “Unknown”, total time 2s

The difference between B and (A + subscriber-side timeout) is the subscriber doesn't get objects 6 or 8 (and maybe not 4)

# #1354 Send Rate Param

See PR #1565

Applies to FETCH - in kbps

Does't apply to SUBSCRIBE because

- 1) Live is already paced at the encoded rate
- 2) Unsustainable: you can't deliver slower than live without building a queue

You probably can't use your transport pacer for this; that's for the whole connection.

This is a ceiling and not a floor. Other subscriptions/fetches can slow this down.

# #1561 New extensibility point - `.session` namespace

Reserves this tuple 0 for tracks handled by the session itself, not the application

Send `NOT_SUPPORTED` if you get one you don't understand

Register tracks or sub-namespaces in IANA

Could be advertised using Setup Options

Can be used to subscribe to information delivered using track mechanics

Examples:

- 1) Sender congestion control information (#1507)
- 2) Live trace logs (implemented in moxygen)

If you like this, should we reserve all `.*` for who-knows-what?

# Multiple Subscriptions to a Track

Primary Use Case: Preventing Relay DOS via “needle-in-a-haystack”

Subscriber 1: SUB t=huge track, range={100000,100000}-{100000,100001}

Subscriber 2: SUB t=huge track, range={200000,200000}-{200000,200001}

Relay has to either open 2 upstream sessions with narrow filters, or 1 session that is unfiltered or under-filtered.

Problem exists today but #1518 filters will create more opportunities

There may be other use cases (racy cancelling and re-subscribing)

# Multiple Subscriptions to a Track

Considerations:

Wire Deduplication or Double Work

- Do subscriptions with overlapping filters receive the object once, or once per subscription?
- If it's only once, both the publisher and subscriber both have to filter

Publisher Management - if deduplicating, multiple subscriptions is an illusion created by the publisher coalescing individual subscriptions and doing a union of the filters

The same effect could be created by expressing the union in a *single* subscription

Multiple subscriptions can have conflicting params that affect delivery

Group Order, Priority, Delivery Timeout

# Multiple Subscriptions to a Track

## Proposal 1a:

Allow multiple subscriptions, each with a unique Track Alias; no deduplication.

Subscriber/relay is responsible for minimizing overlaps/duplication and mapping correctly from upstreams to downstreams.

## Proposal 1b:

Allow multiple subscriptions, all get the same Track Alias

Publisher coalesces these as a single subscription with a union of filters

Conflicting parameters  $\overline{\setminus}_\_ (\_ \_ ) \_ /$

# Multiple Subscriptions to a Track

## Proposal 2:

- Keep single subscription per track

- Allow unions of all filter filter types in control messages

## Proposal 3:

- Status Quo

# Things we never Discuss

#667 Delivery Timeout unimplementable

Now blocking #1476 DELIVERY\_TIMEOUT is both Track and Object property

#869 Difficult to limit resource consumption of a Subscription

#1316 Possible improvements on MoQ support of VOD applications