

Fixing HOLB

New Subscription Filter: Rewind

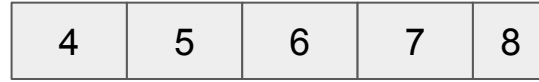
- One value (PreviousGroups): number of groups to go back (0 = start of current group) = “Please send me N old groups as if they were new, if you have them in cache”
- The Publisher MAY reply with a LargestObject < than the first group it has in cache (if some objects are missing that’s OK) – this is a “lie”
 - MUST NOT if not in cache
 - SHOULD NOT if the client is DoSing itself
 - Might Not if the Group is about to end
 - Do not honor $N > 0$ if groups are non-increasing
- All objects > the reported LargestObject are delivered as subgroup streams.
- Is compatible with Joining Fetch, like Largest Object

Required changes

- Add a Subscription Filter
- No subscriber changes at all beyond adding the filter – is indistinguishable from a track where LargestObject is the true live edge
- Publisher logic on when to report a lower LargestObject
 - this is best effort, optional to implement
 - if the publisher does nothing or doesn't cache, PreviousGroups = LargestObject

Examples - N objects per group

Cache



PreviousGroup = 2

LargestObject = 5,N



Relative Joining Fetch = 2

Upstream FETCH 3,1..N



3, 4, 5 in FETCH stream

Examples - N objects per group

Cache



PreviousGroup = 6



LargestObject = 3,N



Relative Joining Fetch = 2



Upstream FETCH 2..3



2, 3 in FETCH stream

RECEIVE*

SUBSCRIBE + FETCH properties without the verbiage

* Paint this bikeshed.

Format

```
RECEIVE Message {  
  Type (i) = TBD,  
  Length (16),  
  Request ID (i),  
  Track Namespace (..),  
  Track Name Length (i),  
  Track Name (..),  
  Number of Parameters (i),  
  Parameters (..) ...  
}
```

Parameters

The same as SUBSCRIBE, except for Subscription Filters

AbsoluteStart: Location -> end of track

AbsoluteRange: Location -> Location

Future Group: Start at group boundary, n groups into the future (0 = “Largest Object”)

Past Group: Start at group boundary, n groups in the past (0 = “Current Group, 0”)

No parameter: give me everything – equivalent to AbsoluteStart(0,0)

Publisher Actions

Send LargestObject parameter in RECEIVE_OK

Anything > largest object is sent in subgroups/datagrams

Anything < largest object is send in a single stream like FETCH

So all “FETCH” are Joining, unless the End Location < LargestObject

The Publisher MAY....

Send a LargestObject parameter with a value less than the “true” value (or omit it), if it has at least some of the required objects in cache (i.e. it is cached from an existing RETRIEVE).*

If not the true LargestObject, it MUST have object ID UINT64_MAX (or spell it NextObject) only begins on group boundaries

this allows the current group to be delivered in streams.

* Fallback proposal – can only do start of the current group or omit it entirely

The Subscriber...

- For a given track, **MUST** have no more than one total RECEIVE for a given track with a subscription filter that exceeds the highest Location the publisher has
 - i.e. there will not be duplicate Subgroup streams/datagrams
 - All filters other than AbsoluteRange always meet this criteria
 - if you don't know the highest location, don't send more than one!
- Once it knows the live edge, **MAY** send any number of RECEIVES with Absolute Range filters as long with $EndLocation < LiveEdge$
 - this allows retrieval of past objects that are not in cache

Non-increasing Group IDs

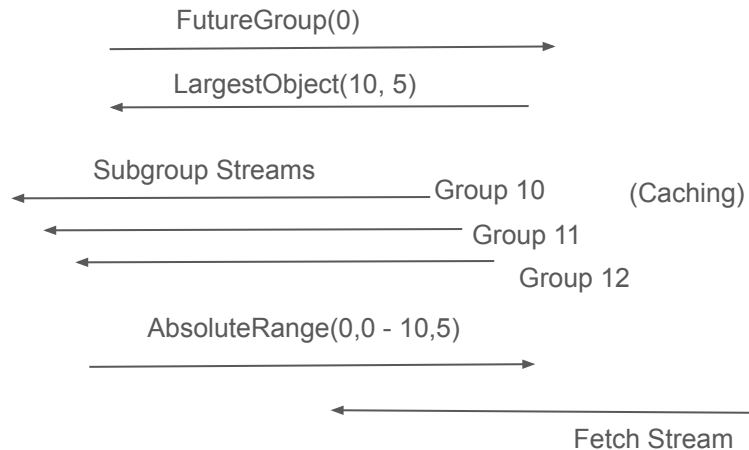
Add a track extension to let relays know what is happening (FUTURE_ONLY = 0, 1, or 2)

If the value is 1 or 2:

- AbsoluteStart/AbsoluteRange filters are ignored
- FutureGroup filter: publishing will begin with the next group, regardless of group ID
- PastGroup filter: publisher will deliver in subgroups/datagrams the beginning of the group, if in cache (value = 1). Otherwise, will proceed as if no filter (value = 2).
- Any second RETRIEVE for that track is ignored
- the publisher delivers any newly arriving objects via subgroup stream/datagram, regardless of location

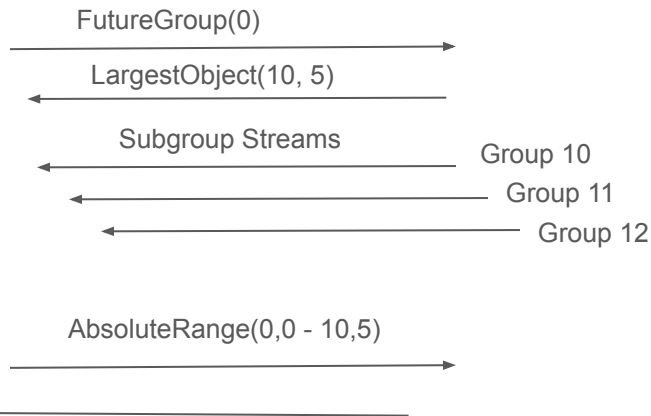
Example

Subscriber 1

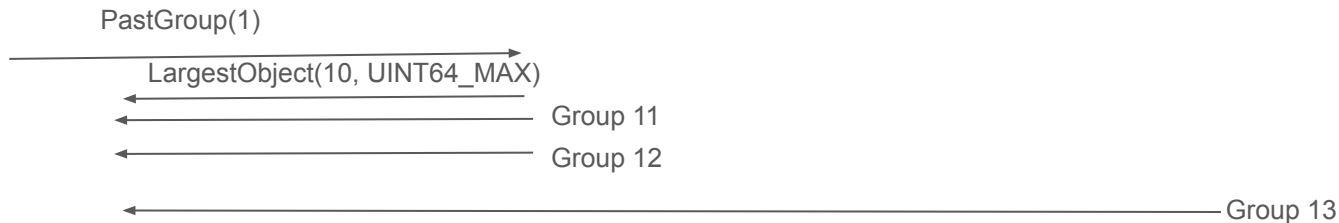


Relay

Publisher

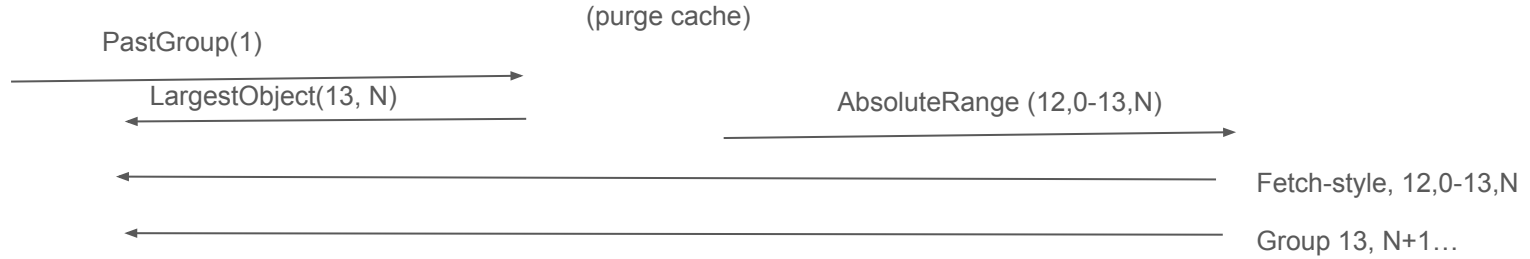


Subscriber 2



Example Cont'd

Subscriber 3



Publisher Pseudocode (ignoring MAYs)

switch (filter_type)

case AbsoluteStart:

SUBSCRIBE logic plus FETCH logic if Start Location < Live Edge

case AbsoluteRange:

SUBSCRIBE logic if End > Live Edge, plus FETCH logic if Start Location < Live Edge

case PastGroup:

SUBSCRIBE+Joining FETCH code path

case Future Group:

SUBSCRIBE code path