

artdaq - Idea #18205

A dropped request shouldn't imply empty fragments for subsequent requests

11/10/2017 03:20 PM - John Freeman

Status:	Closed	Start date:	11/10/2017
Priority:	Normal	Due date:	
Assignee:	Eric Flumerfelt	% Done:	0%
Category:	Additional Functionality	Estimated time:	0.00 hour
Target version:	artdaq v3_00_01		
Experiment:	-		
Description			
<p>One phenomenon observed at the CERN teststand is that it's possible for requests not to make it to a fragment generator during window mode running. This happened, e.g., in run 1000212, involving the timing fragment generator and two SSPs; if one logs onto np04-srv-009 and executes:</p>			
<pre>for i in {257..261}; do grep -E "Recieved request for sequence ID \$i" /log/pmt/pmt-3740.1-20171010053115.log; done</pre>			
<p>then it's easy to see that request #259 never made it to either SSP fragment generator:</p>			
<pre>2017-10-10 05:32:04 +0200: Recieved request for sequence ID 257 and timestamp 24091882946560 (delta: 0) 2017-10-10 05:32:04 +0200: Recieved request for sequence ID 257 and timestamp 24091882946560 (delta: 0) 2017-10-10 05:32:04 +0200: Recieved request for sequence ID 258 and timestamp 24091885043712 (delta: 1) 2017-10-10 05:32:04 +0200: Recieved request for sequence ID 258 and timestamp 24091885043712 (delta: 1) 2017-10-10 05:32:04 +0200: Recieved request for sequence ID 260 and timestamp 24091889238016 (delta: 1) 2017-10-10 05:32:04 +0200: Recieved request for sequence ID 260 and timestamp 24091889238016 (delta: 1) 2017-10-10 05:32:04 +0200: Recieved request for sequence ID 261 and timestamp 24091891335168 (delta: 2) 2017-10-10 05:32:04 +0200: Recieved request for sequence ID 261 and timestamp 24091891335168 (delta: 2)</pre>			
<p>In artdaq v2_03_03, the <code>CommandableFragmentGenerator::applyRequests</code> function will ignore all requests in its queue except the one whose sequence ID matches <code>CommandableFragmentGenerator::ev_counter()</code>. Should a request be missed, <code>applyRequests</code> will make no attempt to match incoming data fragments with the subsequent requests, and instead will basically do nothing until the timeout set by <code>missing_request_window_timeout_us_</code> is hit, at which point all requests in the queue will be handled by the sending of empty fragments downstream, even if the fragments whose timestamps match those requests, returned by <code>getNext_</code>, are available in memory. The larger <code>missing_request_window_timeout_us_</code> is, the more data gets wasted in this fashion. We should implement logic so that even if a request is missed, the data for subsequent requests is sent downstream if available.</p>			

History

#1 - 11/10/2017 03:44 PM - Eric Flumerfelt

I think it was implemented this way to help prevent out-of-sequence events from being sent to art. I don't know if this is still something we need to worry about, but any solution should be tested thoroughly.

#2 - 01/27/2018 12:39 PM - Eric Flumerfelt

- Category set to Additional Functionality
- Status changed from New to Closed
- Assignee set to Eric Flumerfelt
- Target version set to artdaq v3_00_01

We have determined that *art* does not care about sequence, and appropriate changes have been made in *artdaq*.