# Subscriptions

**Issue 1: Subscriber does not receive all entity updates from Orion test setup. (MAJOR/BLOCKER)**

When subscribing with no throttling enabled, there should be no loss of data for the subscriber. On the Orion test environment (https://orion-test.te st.cottech.be). However this is not what we experience.

Test setup:

We started the subscriber which notifies to an endpoint hosted in IDLab.

We are sending 50 messages total with 1 msg/s to Orion. Each of these messages contain an update for an entity where we simply increment an attribute for every new update.

When all 50 messages are sent we check the subscriber for how many updates it received from Orion. We would expect no loss of data because throttling is 0.

The result however varies from run to run but generally is something like this:
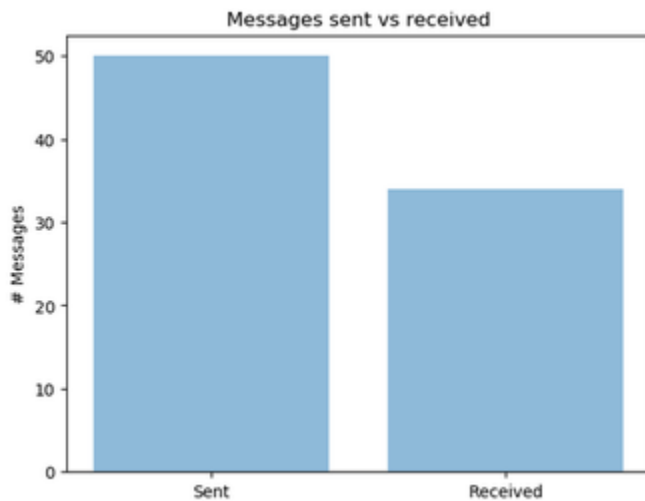


This problem does not seem to occur when using a local Orion setup.

**Cause of Issue 1**

This issue arises when there are multiple CB's. Orion has a subscription cache [1], this cache is local to every CB and holds subscriptions information. The cache is synced every X seconds (default 60) and pushes new subscription info to the other CB's.

So what is happening is the following situation (where the interval > 0) [2]:

The subscription creation request is dispatched to some of the nodes (let's assume it is node A). Node A will persist the subscription in the shared DB *and* in the local (i.e. node A) subscription cache. Note that node B doesn't get aware of the new subscription until next subscription cache refresh, which means. That is as much an interval of seconds equal to `-subCacheIval` parameter.

Let's consider now an update (on an entity/attribute which matches subscription conditions to trigger notification) arrives. Depending to which node the load balancer dispatches the update and the moment in which it arrives (with regards to subscription creation request time) it can happen one of the following cases:

- Update arrives at node B, *before* next subscription refresh (i.e. as much as `-subCacheIval` seconds). Given that node B doesn't know anything about the subscription, notification is not sent.
- Update arrives at node B, *after* next subscription refresh (i.e. as much as `-subCacheIval` seconds). Node B is aware of the new subscription as a consequence of the refresh, so the notification is sent.
- Update arrives at node A, no matter wether before or after next subscription refresh. Node A was the one creating the subscription, so its cache is up to date since the very beginning. Thus, the notification is sent.

This behavior can be confirmed by experimental testing:

In the first test we plot the test which we described in issue 1 (but with 300 messages instead of 50). We can see via that at the start (before the cache sync period ends) messages are lost because they were sent to a CB which is not aware of the subscription.
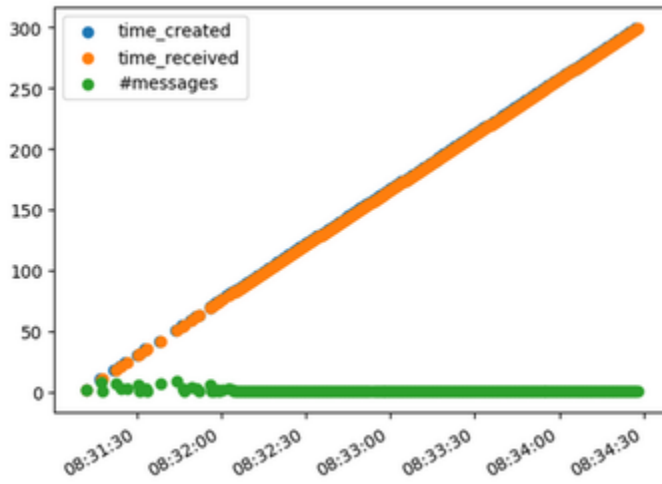


If we wait the time of the interval before sending messages, no data loss is found and all 300 messages are received.
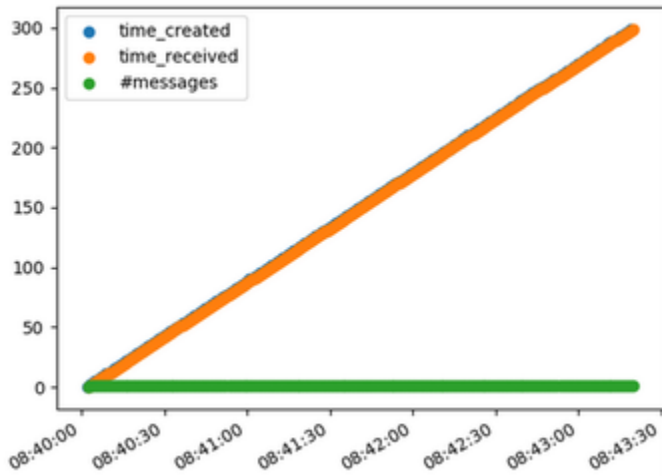


**Conclusion**

Orion will lose messages in a multi CB setup until the subscriber sync period is over, after this interval no message drops are currently visible.

Sources:

[1] https://fiware-orion.readthedocs.io/en/master/admin/perf_tuning/index.html#subscription-cache

[2] https://stackoverflow.com/questions/43857300/what-would-be-the-behavior-of-subscriptions-and-notifications-in-an-orion-load-b