Robotics Service Bus - Bug #1335
Socket transport in Python silently drops events

01/16/2013 04:26 PM - D. Klotz

Status:	New	**Start date:**	01/16/2013
Priority:	Normal	**Due date:**	
Assignee:	J. Moringen	**% Done:**	0%
Category:	Python	**Estimated time:**	0.00 hour
Target version:			

Description

It seems that under certain circumstances, the socket transport in rsb-python can silently drop / not deliver / not correctly receive some events that are sent. A prerequisite to reproduce this so far has been that both a listener and an informer exist in the python process and the events have to be arriving very fast so the socket blocks(?).

I have attached a more or less simple test case (which also needs RST), which is a stripped down version of what I'm actually doing in the code where I noticed the problem. The log file /vol/humavips/studies/Y2_Vernissage/corpus/stereo-slots/12-stereo2/faceresults.tide contains 6018 events on the scope /nao/identities, as can be seen in this output from bag-info:

    Channel "/nao/identities/:.rst.vision.HeadObjects"
      Type   : (RSB-EVENT-0.7 .rst.vision.HeadObjects)
      [...]
      Events : 6,018

If you start the attached test case (with just the socket transport enabled in the rsb.conf) and pipe it this data using e.g. bag-play --replay-strategy 'fixed-rate :rate 3000' [...]/faceresults.tide socket:/ it should report in the end that is has received all 6018 events. But each time I start it here it reports back a different, lower number, even long after bag-play has finished:

    dklotz@rhodenit:[~]python rsbtest.py
    No handlers could be found for logger "rstsandbox"
    Events received so far: 0
    ...
    Events received so far: 5715
    Events received so far: 5715
    Events received so far: 5715

The default strategy in RSB is sending "RELIABLE" messages, right? So in this case, something like this should definitely not happen (if e.g. some buffers fill up and we simply can't process the data fast enough, it should probably die with an error message, not silently drop information).

What makes this even stranger is that this only happens when the informer is used in the test code, without it, all events are received fine, even with bag-play set to this high rate.

History

#1 - 01/16/2013 04:27 PM - D. Klotz

- Description updated

Files

rsbtest.py	1.46 KB	01/16/2013	D. Klotz