# Robotics Service Bus - Bug #1412
## Performance degradation using RPC calls from Python (master) to rsbag 0.7 and master

02/13/2013 02:20 PM - J. Wienke

| | | | | |
|---|---|---|---|---|
| **Status:** | Resolved | | **Start date:** | 02/13/2013 |
| **Priority:** | Normal | | **Due date:** | |
| **Assignee:** | J. Moringen | | **% Done:** | 100% |
| **Category:** | Common Lisp | | **Estimated time:** | 0.00 hour |
| **Target version:** | rsb-0.7 | | | |

**Description**

A roundtrip for calling emitandnext from python to a running rsbag instance with remote control interface takes approximately 0.06 seconds, which is much more than the same call using the spread transport. CPU load is not very high, seems to be some timeout / TCP_NODELAY thing.

These are some of the real times for the call controlServer.emitandnext.async().get(5):

    Emit delay: 0.0642349720001
    Processing 30 of 14968 events
    Emit delay: 0.0642948150635
    Processing 31 of 14968 events
    Emit delay: 0.0642681121826
    Processing 32 of 14968 events
    Emit delay: 0.0644030570984
    Processing 33 of 14968 events
    Emit delay: 0.0640850067139
    Processing 34 of 14968 events
    Emit delay: 0.0640649795532

**Associated revisions**

**Revision 0d59f5aa - 02/13/2013 04:33 PM - J. Moringen**

Fixed missing TCP_NODELAY in src/transport/socket/bus-connection.lisp

refs #1412

When operating in bus server mode, TCP connections to bus clients
never used TCP_NODELAY, no matter whether the option had been
specified for the respective connectors or not.

Simply set (or unset) TCP_NODELAY for all socket objects before their
use within a bus-connection.

  - src/transport/socket/bus-connection.lisp (header): updated copyright
      (shared-initialize :after bus-connection t): set TCP_NODELAY option
      before socket object is used

**Revision adf268b8 - 02/13/2013 04:33 PM - J. Moringen**

Use TCP_NODELAY by default in src/transport/socket/connector.lisp

fixes #1412

Use TCP_NODELAY by default since this tradeoff seems more reasonable
than the alternative.

  - src/transport/socket/connector.lisp (header): updated copyright
    (connector::nodelay?): changed initform nil -> t


**Revision 7c8d7354 - 02/13/2013 05:11 PM - J. Moringen**

Backport: Fixed missing TCP_NODELAY in src/transport/socket/bus-connection.lisp

refs #1412

When operating in bus server mode, TCP connections to bus clients
never used TCP_NODELAY, no matter whether the option had been
specified for the respective connectors or not.

Simply set (or unset) TCP_NODELAY for all socket objects before their
use within a bus-connection.

  - src/transport/socket/bus-connection.lisp (header): updated copyright
    (shared-initialize :after bus-connection t): set TCP_NODELAY option
    before socket object is used


**Revision df73fd3e - 02/13/2013 05:11 PM - J. Moringen**

Backport: Use TCP_NODELAY by default in src/transport/socket/connector.lisp

refs #1412

Use TCP_NODELAY by default since this tradeoff seems more reasonable
than the alternative.

  - src/transport/socket/connector.lisp (header): updated copyright
    (connector::nodelay?): changed initform nil -> t


**Revision 3e2181e3 - 02/13/2013 05:33 PM - J. Moringen**

Added RSB_TRANSPORT_SOCKET_NODELAY in common.rst

refs #1412

  - common.rst: added description of RSB_TRANSPORT_SOCKET_NODELAY
    variable


**History**

**#1 - 02/13/2013 04:14 PM - J. Moringen**

*- Category set to Common Lisp*

*- Status changed from New to In Progress*

*- % Done changed from 0 to 30*


**#2 - 02/13/2013 04:22 PM - J. Moringen**

*- Subject changed from Performance degradation using RPC calls from python (master) to rsbag 0.7 and master to Performance degradation using RPC calls from Python (master) to rsbag 0.7 and master*


The problem was that the TCP_NODELAY option was not applied when operating in bus server mode.

This problem may affect other implementations.


**#3 - 02/13/2013 05:32 PM - J. Moringen**

*- Status changed from In Progress to Resolved*

*- % Done changed from 30 to 100*


Applied in changeset rsb-cl|commit:adf268b8d2c30f01d53b34284c0c78580bfceee9.