# PERFORMANCE ASSESSMENT OF SOFTWARE INTERFACES FOR THE SOFTWARE COMMUNICATIONS ARCHITECTURE (SCA)

Michael Gudaitis (L-3 Communications, Rome, NY, USA, mike.gudaitis@L-3com.com)
Curtis Nelson (L-3 Communications, Rome, NY, USA, curtis.nelson@L-3com.com)

## ABSTRACT

This paper discusses results of performance measurements of the Software Communications Architecture (SCA) and the Packet Application Program Interface (API). A test application is developed to assess the performance of interprocess communication in an SCA v2.2.2 compliant environment. The operating environment is based on the OrcaCF using three different Object Request Brokers (ORBs), TAO, OmniORB, and ORBexpress, to measure the effect the ORB has on timing and latency of data flow using the Packet API. The target environment is x86 computers running Linux in a dual node configuration interconnected through an Ethernet switch. Various packet sizes are used in order to assess the effect that packet size has on efficient data exchange. The hypothesis is that the lower bound on the data transfer time is constrained and dominated by TCP, which is the default protocol used by ORBs to communicate among distributed objects. Therefore, a non-SCA non-CORBA application is included in the performance measurement to quantify the lower bound of transport time using TCP as the transport protocol between distributed software components.

## 1. INTRODUCTION

The Joint Tactical Radio System (JTRS) Joint Program Executive Office (JPEO) has specified the Software Communications Architecture (SCA) for its software defined radios. The JPEO released version 2.2.2 of the SCA specification [1] in June 2006. In April 2007, the JPEO publicly released a number of APIs, including the Packet API [2] for use as the standard interface for transferring information between components. The SCA specification mandates the use of Common Object Request Broker Architecture (CORBA) middleware [2], which uses the Internet Inter-ORB Protocol (IIOP) as the standard interconnect mechanism between software components. IIOP is built upon TCP as the default connection protocol.

The hardware architecture of a JTRS radio often requires multiple processing nodes interconnected with Ethernet using TCP as the default transport protocol. The SCA adds some overhead in addition to that required from raw TCP data transfers. This paper describes the results of testing to quantify the time required to send and receive a packet of data using two processing nodes running an SCA-compliant application composed of an SCA software Resource on one node interconnected to an SCA software Device on the second node. Data is transferred between SCA software components through software Ports that implement the Packet API. A non-SCA application is used to measure TCP performance to compare with the SCA performance. Three different ORBs are used in the SCA testing to compare the performance of the ORBs.

## 2. TEST DESCRIPTION

Two x86 processing nodes were connected to one of three different Ethernet switches:

- 10Mb Switch: 3Com® 24 port SuperStack II Switch 1000.
- 100Mb Switch: Netgear® 5 port Fast Ethernet Switch, model FS105.
- 1000Mb Switch: Netgear® ProSafe 8 port Gigabit Switch, model GS108.

The two processing nodes were as follows:

- Node 1 was a Dell Precision 380N workstation, P4 Processor 640, 3.20GHz, 2MB L2800MHz; with 2GB, 533 MHz DDR2 ECC SDRAM. The operating system was Linux CentOS 4.4.
- Node 2 was a Dell Latitude D820 laptop, Intel Core Duo T2400, 1.83GHz, 667MHz, 2M L2 Cache; with 2GB, 667 MHz DDR2 SDRAM. The operating system was Linux Fedora Core 5.

The OrcaCF [4] was used for the SCA Operating Environment. The OrcaCF includes a Sample Application that was modified to perform the SCA-based interconnect tests. The Sample Application consisted of an SCA Resource with Ports connected to an SCA Device. The Ports implemented the Packet API, which was implemented using the 'deque' double-ended queue C++ standard template library. Three different ORBs were used for each set of tests. The ORBs were ORBexpress [5] from OIS, omniORB

[6], and TAO [7]. The tests consisted of sending an SCA Packet from one node to the other and measuring the time required to send and receive the Packet. The Packet payload size was varied from two bytes to 262144 bytes (i.e. 256 KB) by powers of 2. Each Packet size was sent 100 times with each ORB, and with each of three different Ethernet switches (10Mb, 100Mb, and 1000Mb). The network connection had other traffic on it to more closely represent the loading of an actual system. Several test runs were conducted on different days at different times of the day, and the results were averaged to obtain performance measures that could be described as typical for this hardware/software test configuration.

TCP performance was tested using two different TCP test utilities: ttcp [8] and nuttcp [9]. These utilities test the 'raw' TCP performance, i.e. they measure the time required to send and receive a specified number of unformatted bytes. The utilities were configured to send one packet of a given number of bytes equivalent to the payload size of the SCA Packet. The test results are shown in the figures below.

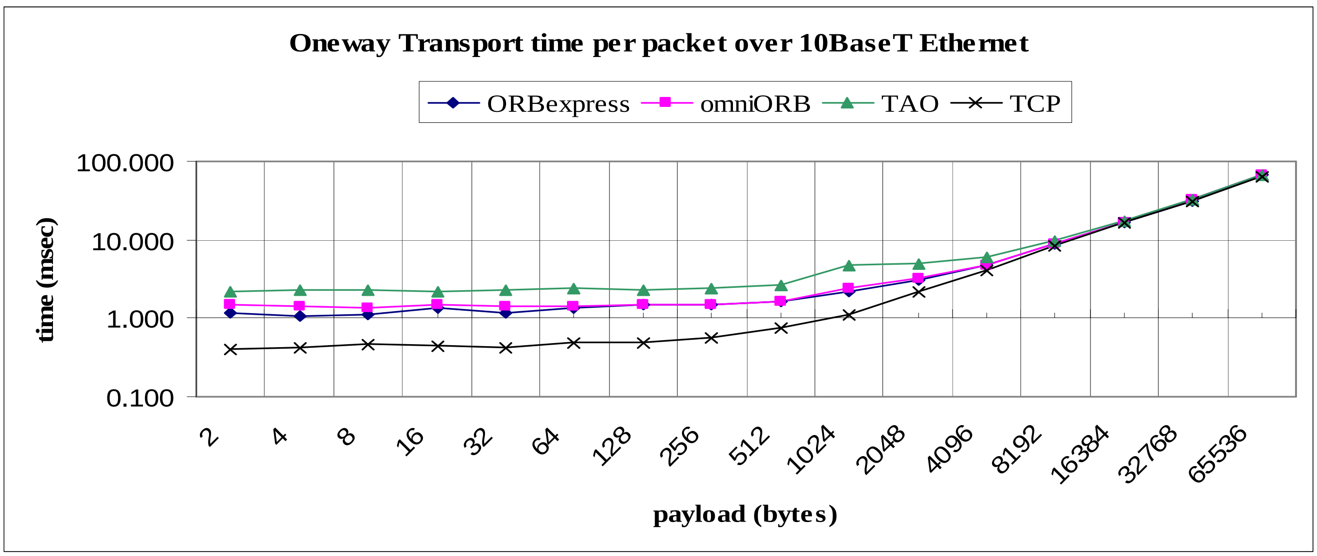


**Figure 1 Time (in msec) to send a packet payload of xx bytes between two processing nodes interconnected through a <u>10BaseT</u> Ethernet switch.**

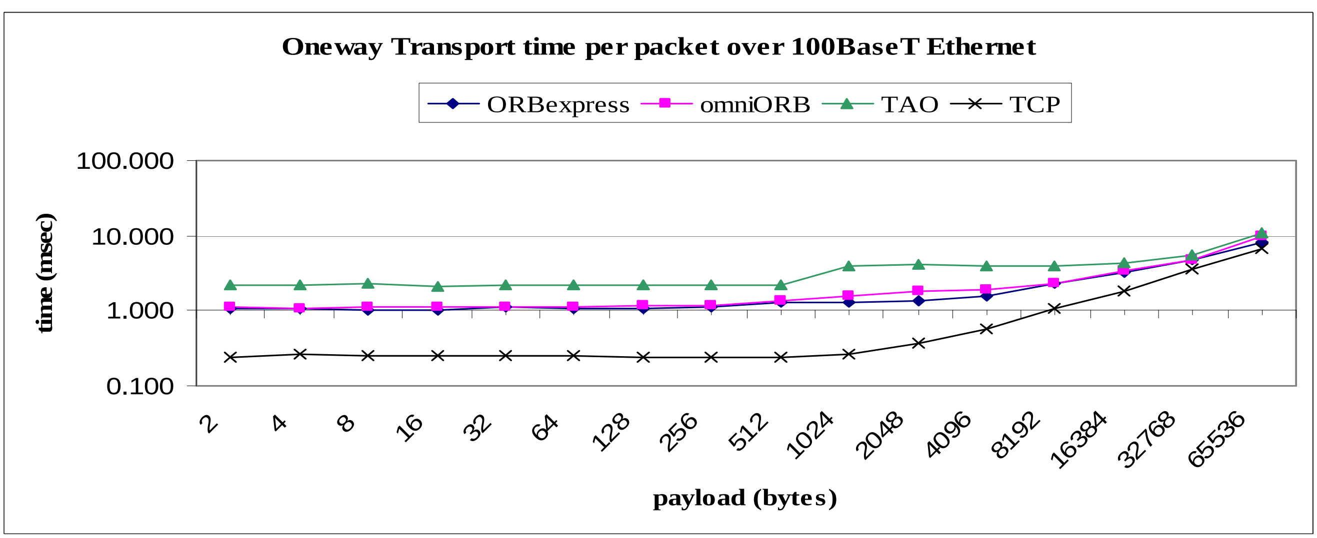


**Figure 2 Time (in msec) to send a packet payload of xx bytes between two processing nodes interconnected through a <u>100BaseT</u> Ethernet switch.**

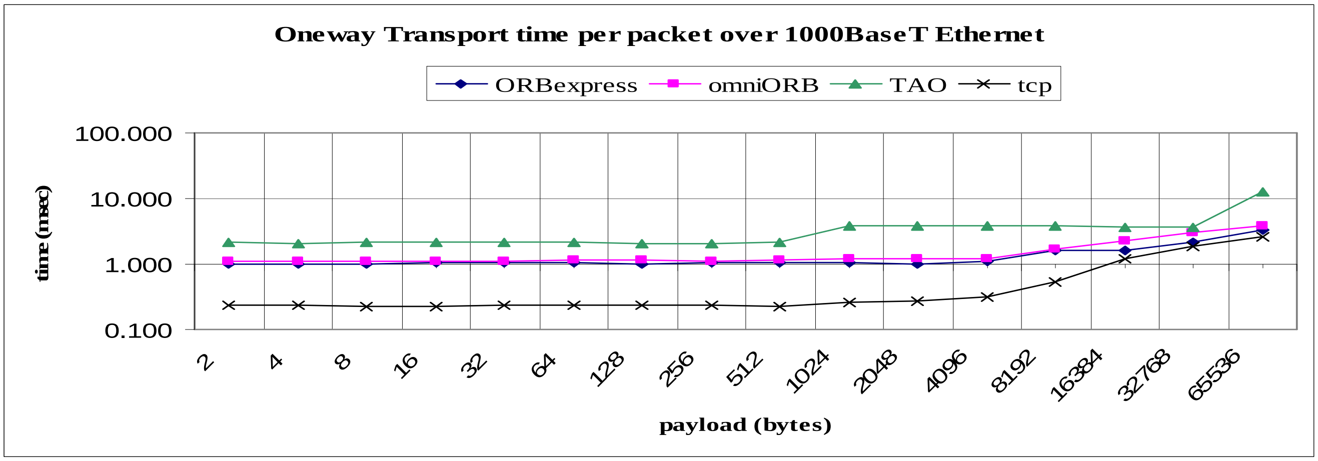


**Figure 3 Time (in msec) to send a packet payload of xx bytes between two processing nodes interconnected through a 1000BaseT Ethernet switch.**

## 3. ANALYSIS OF RESULTS

As can be seen in the figures above, ORBexpress is the fastest of the three ORBs tested. The performance of omniORB was a close second, and TAO was a distant third. The overhead associated with SCA and CORBA as compared with TCP is noticeable for packet payload sizes of 512 bytes or less. The average transit time per packet is relatively constant for packet sizes of 2 to 512 bytes. The measured results are shown in Table 1.

**Table 1 Average time (msecs) per packet size of 2 to 512 bytes**

| | raw TCP | ORBexpress | omniORB | TAO |
|---|---|---|---|---|
| 10Mb | 0.494 | 1.318 | 1.476 | 2.340 |
| 100Mb | 0.248 | 1.093 | 1.150 | 2.190 |
| 1000Mb | 0.234 | 1.043 | 1.115 | 2.135 |

As shown in these results, the raw TCP performance for packet payload sizes of 512 bytes or less is 3-4 times faster than the fastest ORB implementation. However, it must be noted that the raw TCP performance was obtained from tests with ttcp and nuttcp which are optimized to obtain the fastest TCP performance. The SCA applications have not been optimized for performance. For example, ttcp sends unformatted bytes, whereas in the SCA Packet, the bytes are sent as an array of characters.

## 4. CONCLUSIONS

These tests showed that ORBexpress is the fastest of three ORBs tested, with omniORB second, and TAO third. The TCP performance as measured with an optimized TCP test utility, ttcp, is 3-4 times faster than the fastest ORB implementation for Packet payload sizes of 512 bytes or less. The difference in performance between ttcp and SCA applications diminishes for larger payload sizes. Further work is needed to compare the performance of optimized versions of the SCA applications with performance results of ttcp to determine if the performance gap is smaller than these tests indicate.

## 10. REFERENCES


[1] JTRS Software Communications Architecture Specification, version 2.2.2, 15 May 2006, http://sca.jpeojtrs.mil/

[2] Joint Tactical Radio System (JTRS) Standard Device Packet Application Program Interface (API), version 1.1.1, 29 March 2007, http://sca.jpeojtrs.mil/

[3] Common Object Request Broker Architecture: Core Specification, formal/04-03-12, Object Management Group, http://www.omg.org/docs/formal/04-03-12.pdf

[4] Open Radio Communications Architecture Core Framework (OrcaCF), http://www.orcacf.com/

[5] ORBexpress, http://www.ois.com/products/prod-1.asp

[6] omniORB, http://omniorb.sourceforge.net/

[7] TAO, http://www.cs.wustl.edu/%7Eschmidt/TAO.html

[8] ttcp, http://sd.wareonearth.com/~phil/net/ttcp/

[9] nuttcp, ftp://ftp.lcp.nrl.navy.mil/pub/nuttcp/