

PC BASED REAL TIME DATA EXCHANGE ON 10GbE OPTICAL NETWORK USING RTOS *

R.P. Gupta, Green Systems, Ahmedabad, India
H. Dave, IPR, Gandhinagar, India

Abstract

The traditional embedded systems are expensive to adapt to the new requirements. The Personal Computer based systems offer alternatives for industrial controls. It reduces the capital cost and provides a solution for multiple applications. However, limitations of PC based controls should be resolved. PC operates on a non-real time OS with non-deterministic response to real time events and data. The real-time pre-emptive kernel for Linux uses Xenomai for better solution. A real-time 10GbE data exchange optical network using Xenomai extension for Linux is demonstrated. The hardware based on Intel_82599 10GbE Ethernet PCIe network card supports IEEE1588 standard for synchronization, deterministic response to real-time interrupts and events.

The benchmark testing comprises nodes and data sources, for data exchange among nodes, which would improve the performance of PC control systems. Data sources and consumers include time synchronization, hardware and software events broadcasting. A single network cable is used for exchange of status and control data among nodes. Moreover, the open source Ubuntu Linux RTOS will help the future development.

INTRODUCTION

Industrial control systems are evolving from dedicated systems to generalized and flexible systems. Traditional controls were designed around dedicated embedded system, wherein the design was applicable for a given set of control parameter and process. These systems would need major changes for new set of control parameters, computations and processes. The personal computer based controls offer better computational power, flexible and faster systems, via Ethernet [1]. The complex scientific experiments, [2] require more computational power and better controls [3]. Multiple applications and wider communication support on PC could offer cost-effective solution for real-time network.

Implementation of PC based industrial control system encounters many limitations [1, 2]. PCs were basically designed for applications with non-real time responses. Operating Systems (OS) for PC do not exhibit deterministic response to real time events and data [4]. Most of the common OS are not pre-emptive. However, as per [4], Xenomai kernel for Linux OS is a better pre-emptive kernel, which provides deterministic real time response. Xenomai is an alternative to the proprietary real time operating system, because it extends GNU/Linux with real time performance [5].

The pre-emptive kernel needs a better support from the system hardware, so that hard-real time interrupts are serviced within a given time-frame. PC based controllers require networked nodes for data acquisition, event monitoring and control.

With the advent of 10GbE Ethernet controller cards, it is now feasible to setup a multi-node network with reasonable throughput and latency performances [6]. Ethernet controllers such as 82599 and X540-T2 are now available from Intel, with Linux drivers. Hard-real time network are possible with Ethernet controllers [6], due to faster PCI express interface. The fast PCIe 2.0 interface on a PC ensures faster response from host CPU. The 10GBASE-T converged network adapter from Intel such as X540-T2, offers advantages of low latency, load balancing on multiple CPUs, MSI-X support and flexible I/O virtualization.

The real-time data exchange needs reliable time synchronization. In particular, a hard real-time network needs synchronized time stamping on data packets. Intel's controller support IEEE1588 Precision Time Protocol (PTP). Time synchronization is achieved with accurate time reference from GPS connectivity.

According to the above facts, network adapters for optical cable (X-520-SR2), as well as for Cat6A copper cable (X540-T2) were used to create 10GbE network, running on Linux with Xenomai real time framework.

An experimental real-time network is demonstrated for benchmark testing of 10 GbE network using Linux RTOS. Standard network performance test utilities such as Netperf, Iperf and Jperf were used for the benchmarking. Tests were performed both for TCP and UDP data exchange.

REAL TIME DATA EXCHANGE

The real time data exchange require pre-emptive kernel with deterministic response time. There are two open source real time extensions for Linux namely RTAI and Xenomai. However, Xenomai is being updated by the open source community. The comparative study [2] of VxWorks, RTAI and Xenomai indicates that performance of Xenomai is acceptable for most applications. It is necessary to apply ADEOS patch for Xenomai. We have used standard open source patches available for Xenomai kernel. For the present testing, the interrupts priority was ensured with minimum external devices connected to the CPU. The main objective is the basic setup of real time network using 10GbE network card, wherein the PC is used exclusively for the benchmark testing.

* Work is supported by research grant No. BRFST/NFP/2012/Feb/N/05; under the National Fusion Programme of the BRFST at IPR, Ahmedabad

Network Setup

The 10GbE network was setup around a PC with Intel Ethernet network adapters. For hard-real time network a pre-emptive kernel is required with reasonably fast Ethernet controller adapters. Intel provides new range of 10GbE Ethernet controllers with virtualization features as well. Linux drivers are available for X520 and X540 series of network controllers; hence we selected these Ethernet cards with dual adapters.

The PC based real time data network comprises of the following setup:

- A test node PC (named as NewI5) with Intel Core i5-CPU @3.10GHzx4, Memory 3.6GB, with PCIe 2.0 x8 host interface. Installed with Ethernet controller card: Intel X540-T2 10GbE dual adapter with RJ45x2 for Cat6A sealed cable. The Cat6A cable provides better performance for 10Gbps network.
- A test node PC with Intel® Core i5- CPU @ 3.10 GHz x 4; Memory: 3.8 GiB with PCIe 2.0 x8 host interface. Installed with Ethernet controller card: Intel X-520-SR2; Intel 82599 based 10GbE dual port optical network adapter; Optical cable is LC/LC multimode fibre.

The 10GbE network was setup around standard PC running on Ubuntu Linux OS. Dual adapters were loop backed with switchless network, for testing the maximum possible performance. Following OS and kernels were used to create Linux kernel with Xenomai extension.

- OS: Ubuntu 12.04, Vanilla Linux kernel version 3.2.21;
- Adeos patch 3.2.21 -x86-1.patch was used with Xenomai 2.6.1.

It created an Ubuntu bootable version with Xenomai extension. The stable version 2.6.1 of Xenomai has been released recently on July 10, 2012, with many bug fixes. The Xenomai related tests such as Xeno-test etc., were successfully confirmed on both the nodes.

Network Performance Benchmark Tools

Network performance is measured by utility software that measures TCP/IP and UDP performance at the application or protocol level. The popular and proven benchmark tools are mostly used for reliable comparison of performance results. The benchmarking tool should test throughput and RTT (Round trip time) latency of the network.

Standard open source tools such as Netspec, Netperf, Iperf, Jperf, Sockperf etc., were evaluated. We finally decided on Netperf, Iperf and Jperf, having popular support, for all measurements. Netperf measures throughput and also RTT latency. Iperf with Jperf provides graphical results with jitter information for UDP protocol. All the selected tools work under Server/client communication system with command line interface. The Netperf tools also provide CPU utilization for useful indicator of the network performance. The results are presented in graphical format by Jperf. We have selected TCP and UDP protocols for comprehensive evaluation of the network.

BENCHMARK TEST RESULTS

For performance measurement, the proposed network was tested with Ethernet as the only active port. USB port and all other ports were inactive, so that CPU is mainly utilized for Ethernet activity. The aim of the test was to determine the upper limit performance of the network under ideal test conditions...

Netperf Test

The measurement command and result for Netperf tool is shown in Table 1. The server / client setup provides result for UDP protocol on client node. The Trans rate measures the latency time, without lost datagram for UDP.

Table 1: Netperf results for UDP protocol

```
NEWI5:~# netserver -L 192.168.10.1 -p 5679
Starting netserver with host '192.168.10.1' port '5679' and
family AF_UNSPEC
NEWI5:~$ netperf -t UDP_RR -L 192.168.10.2 -H
192.168.10.1 -p 5679
MIGRATED UDP REQUEST/RESPONSE TEST from
192.168.10.2 () port 0 AF_INET to 192.168.10.1 () port 0
AF_INET : first burst 0
```

Netperf results for UDP protocol	163840
Netperf results for UDP protocol	163840
Request Size bytes	1
Resp. Size bytes	1
Elapsed Time(s)	10.01
Trans. Rate/ s	102245.9

The measurement result for TCP protocol is shown in Table 2, wherein it indicates that transfer rate is 82986 per sec, about 19% lower than UDP. The ping result is shown

Table 2: Command for TCP protocol

```
NEWI5:~# netserver -L 192.168.10.1 -p 5679
Starting netserver with host '192.168.10.1' port '5679'
and family AF_UNSPEC
NEWI5:~$ netperf -t TCP_RR -L 192.168.10.2 -H
192.168.10.1 -p 5679
MIGRATED TCP REQUEST/RESPONSE TEST
from 192.168.10.2 () port 0 AF_INET to
192.168.10.1 () port 0 AF_INET : first burst 0
```

Netperf results for UDP protocol	16384
Netperf results for UDP protocol	87380
Request Size bytes	1
Resp. Size bytes	1
Elapsed Time(s)	10.01
Trans. Rate/ s	82986.52

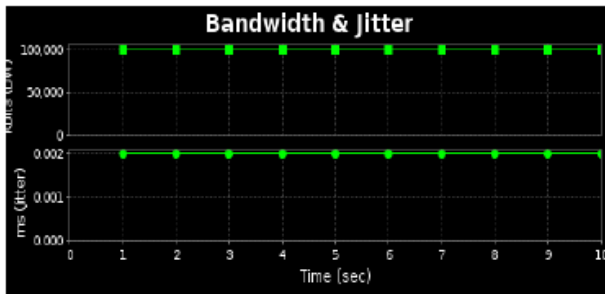
in Table 3, which shows average RTT of 10 us and mdev is 6 us. This fast response is useful for real time applications.

Table 3: RTT, as given by ping command

NEW15:~# ping -I 192.168.10.1 192.168.10.2
64 bytes 192.168.10.2: icmp_req=4 ttl=64 time=0.006 ms
6 packets Tx, 6 received, 0% packet loss, time 5000ms
rtt min/avg/max/mdev = 0.006/0.010/0.022/0.006 ms

Jperf/Iperf Test

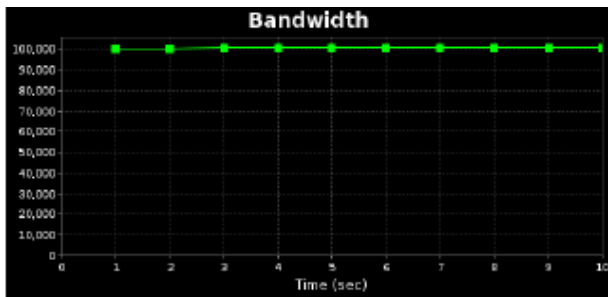
Measurement results with Iperf and Jperf are shown in the Fig. 1 and Fig. 2. For basic UDP test, the BW is restricted to 100 Mbps, hence data loss is 0%.



UDP Server with jitter of 2uSec, BW= 100Mbps			
Transfer	Bandwidth	Jitter	Lost/Total Datagrams
122649 KBytes	100454 Kbps	0.002 ms	0/85436 (0%)

Figure 1: UDP Server output.

The Table 4 for UDP output with bandwidth of 1Gpbs indicates data loss of 0.78%. With UDP, data loss increases with increase in the required bandwidth.



Transfer	Bandwidth
122649 KBytes	100453 Kbits/sec

Figure 2: UDP Client output.

CONCLUSION

The benchmark testing of Ethernet network using Linux PCs, with Xenomai framework extension has been performed with 10GbE Ethernet controller cards. The results, suggest the following:

Table 4: UDP server output with BW = 1Gbps

iperf -s -u -P 0 -i 1 -p 5001 -C -f g			
Interval	Bandwidth	Jitter	Lost/Total Datagrams
(s)	(Gbps)	(ms)	(%)
0.0-10.0	0.68	0.002	0.78
(4566/582135)			
0.0-10.0	1 datagrams received out-of-order		

The RTT of the order of 6 to 10 uSec is possible. Real time pre-emptive Linux kernel with Xenomai extension using Intel's 10GbE Ethernet controllers can be used for complex applications. TCP protocol offers reasonable performance with higher reliability. UDP protocol may be used for lower bandwidth application, however at higher bandwidth the data loss increases. For multiple node networks, TCP protocol is recommended. A real time network with the order of 10usec round trip response time is possible with pre-emptive Linux system.

Table 5: Netperf command for UDP request response with CPU utilization

NEW15:~# netserver -L 192.168.10.1 -p 9999	
NEW15:~\$ netperf -t UDP_RR -L 192.168.10.2 -H 192.168.10.1 -p 9999 -c -C	
Socket Send bytes	163840
Size Recv bytes	163840
Request Size bytes	1
Resp. Size bytes	1
Elapsed Time(s)	10.01
Trans. Rate/ s	98200.4
CPU local (% S)	22.72
CPU remote (% S)	22.72
S.dem local (us/Tr)	9.255
S.dem remote(us/Tr)	9.255

The Table 5 below shows the UDP response request with CPU utilization, wherein CPU utilization is 23% and transaction rate is higher than TCP protocol.

REFERENCES

- [1] Kun Ji and Won-jong Kim, "Real-Time Control of Networked Control Systems via Ethernet" International Journal of Control, Automation and Systems, vol. 3, no. 4, pp. 591-600, Dec.2005
- [2] F. Leroux et al., "New Developments on Tore Supra Data Acquisition Units", Proceedings of ICALEPCS, Grenoble, France, pp. 922-925, 2011.
- [3] Ondrej Cevan, "Installation and Implementation of Xenomai on an Industrial ... IO-Card to the Xenomai Infrastructure", Vienna University of Technology TRAINEESHIP PROJECT, pp.1-87, July-August 2006

- [4] A. Barbalace, et al., "Performance Comparison of VxWorks, Linux, RTAI and Xenomai in a Hard Real-Time Application", IEEE TRANS.ON NUCLEAR SCIENCE, VOL.55, NO. 1, FEB.2008
- [5] Byoung Wook Choi,et al., "Real-time control architecture using Xenomai for intelligent service robots in USN environments", Intel Serv Robotics(2009) 2:139-151 DOI 10.1007/s 11370-009-0040-0
- [6] Amarpreet Ugal "Hard Real Time Linux *using Xenomai*" on Intel Processors", Intel Xenomai on i7 core,Oct.2009.