EXPERIMENTAL ANALYSIS OF CoS PERFORMANCE IN LOCAL CONTROLLER NETWORKS

Nikolay Kakanakov, Mitko Shopov, Grisha Spasov

Computer Systems and Technologies, Technical University of Sofia, branch Plovdiv, 61 St. Petersburg Blvd., 4000, Plovdiv, Bulgaria, phone: +359 32 659758, e-mail: netlab@tu-plovdiv.bg

The paper shows some test-bed experimental results on the performance of CoS (IEEE 802.1p) implementation in Controller networks. The experiments are taken out for four different embedded devices with different implementation of TCP/IP stack and communication interface. The communication delay of the packets that is measured between embedded systems and a test computer are observed while the switch is flooded with bulk traffic. Two different protocols are being analyzed: network layer – ICMP and application layer – CNDEP.

Keywords: Controller Networks, Real-time Communication, Ethernet, CoS, QoS.

1. INTRODUCTION

Ethernet in its nature is not developed for real-time and control applications. Nevertheless, its low cost of cabling and scalable infrastructure has led to its application in the control field. There are applications of Ethernet in every level of control – device, field, plant. These applications are time sensitive and have different characteristics and requirements from office networks. Controller networks based on Fast and Gigabit Ethernet are possible due to the number of available embedded systems with integrated Ethernet controller and TCP/IP stack. Using Ethernet and Internet technologies in controller networks has led to new perspectives for application development.

1.1 Motivation

The vast application of distributed embedded systems and Internet appliances have led to new research efforts in the field of embedded systems communication. The packet lost rate should be kept to its minimum, as long as communication delay and jitter. The concept of QoS and CoS in audio and video streaming over Internet must be adapted to this new field. Priority management should be used in switches to guarantee low delay and jitter in control applications. This concept has been proven by analytical and simulation analysis in the literature, but there are only few efforts targeted at test-bed estimation of the delay and jitter and the parameters they are influenced by. This paper shows some test-bed experiments for estimation of the delay in local switched Ethernet network of embedded devices. The experimental network is built in "Distributed Systems and Networking, Virtual Laboratory" in Technical University of Sofia, branch Plovdiv [1, 3, 8].

1.2 Background

For more accurate examination of the delays measured they can be separated. The delay on local network without collisions can be separated to: delay form the source communication stack (D_S), delay from the destination communication stack (D_R), delay for packet transmission (number of bits in the packet divided by the speed of the network) (D_{Tx}), delay form signal propagation (D_{prop}) and the delay from queuing in the switch (D_Q) [7]:

Equation (1): $D_E = D_S + D_{Tx} + 2D_{prop} + D_R + D_Q$

Delays form the communication stacks of the source and destination nodes can be calculated by a series of test-bed experiments. They are a specific parameter for every class of controllers and depend on its CPU, memory, NIC, interrupt handling mechanism and so on. Some results on its calculation for two classes called medium and high performance are presented in [4]. The medium class is ARM-7 based device running on 40MHz and the high performance – Intel 586, running on 166MHz. The delays introduced form the devices are the following, depending on data size: high performance – 1.02 ms for 128 bytes, 1.87 ms for 500 bytes, and 2.88 ms for 1000 bytes; medium performance – 1.94 ms for 128 bytes, 3.10 ms for 500 bytes, and 4.65 ms for 100 bytes. Typical switch multiplexing delays are in order of 45 µs for Fast Ethernet and 25 µs for Gigabit Ethernet [7]. Typical values for the transmission delay for the maximum frame size of 1514 bytes is 121 µs Fast Ethernet and 12 µs for Gigabit Ethernet. For the minimum frame size of 64 bytes for Fast Ethernet the transmission delay is about 5.12 µs [2]. Gigabit Ethernet has different minimum frame sizes for different physical standards but the transmission delay is as that of Fast Ethernet.

2. Test-bed Architecture and Setup

Four different embedded systems are chosen for the test-bed experiments. These represent different classes of embedded devices. The first embedded system is based on Microchip PIC18F25J10 8-bit processor with ENC28J60 Ethernet chip and Microchip TCP/IP stack. It has no operating system and API, so the access to the hardware is direct and fast [10]. The second embedded system used in experiments is Beck DK40 development board with SC12 processor. SC12 is system-on-chip with I80186 processor and integrated Ethernet controller and real-time operating system with TCP/IP stack. The access to the hardware and networking is by means of operating system API functions [9]. The third embedded system used is DS TINI. It is based on DS80c400 16-bit processor from Dallas. It has specific operating system and TCP/IP stack and application are mostly written in Java. Access to hardware and network are made through Java native interface API [12]. The last embedded system used is a single board computer with ARM9 TDMI processor at 200 MHz. It has Fast Ethernet controller by Micrel. The operating system is general purpose Debian Linux, kernel 2.6.18 [13].

The controller network uses a star topology with a Catalyst 2950 intelligent Ethernet switch in the middle. It has 24 Fast Ethernet ports (100-Base-Tx), 8.8 Gbps switching fabric with 6.6 Mpps wire forwarding rate, 16 MB DRAM, 8 MB queuing memory and 8 MB flash. It supports VLANs (802.1q), SPAN, and Cos (802.1p). At each egress port of the switch four queues with 1P3Q1T management are supported.

A number of general purpose computers are used in the experiments for the tasks of bulk traffic generators and sinks, testing and monitoring. Computers used for the bulk traffic generation are Pentium III at 900 MHz and Fast Ethernet cards, running OS Debian Linux. The test and monitoring computer is Pentium IV at 3.0 GHz and Ethernet card, supporting 10/100/1000 [11].

Instead of making one huge experiment, a series of small experiments are made to separate the components of the delay and to observe the parameters that influence the delay. Different scenarios are proposed to test the interference between VLAN, delays in network and performance of CoS implementation on the switch. The measured delays are taken for two different protocols: one at the network layer – ICMP (Internet Control Message Protocol), and one at the application layer – CNDEP (Controller Network Data Extracting Protocol) [5]. This separation is made because most of the embedded systems have functions of the network layer integrated in the kernel and the application layer protocols introduce additional delay caused by switching between user and kernel space.

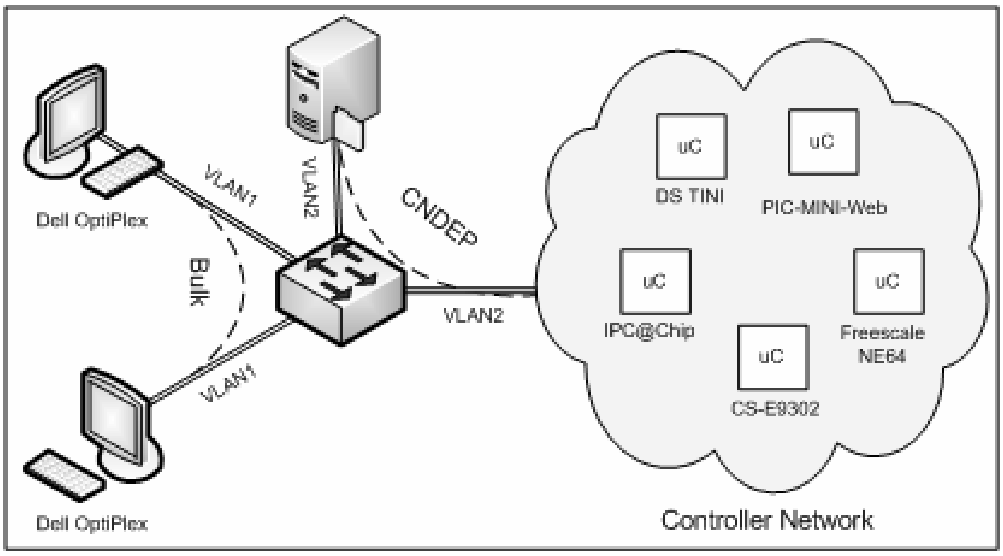

Figure 1

A series of experiments are made separately in several scenarios for the two protocols. In scenario (a) embedded systems are directly connected to the test computer. In the next scenario (b) same experiments are taken out but the embedded systems are connected to the test computer using Catalyst 2950 switch. The round-trip-time is again observed for each controller. In the following scenarios controllers are connected to the switch, together with the test computer and the computers for bulk traffic generation. In scenario (c) the controller's traffic and the bulk traffic travels in separate VLANs. That way an observation of the interference between VLANs is made (figure 1). In scenarios (d) and (e) the bulk traffic and the traffic from controllers to the test computer are directed through the same egress port of the switch. In scenario (e) CoS policies are configured at this egress port – the traffic from the controllers is mapped to the strict priority queue of the port and the bulk traffic to one of the WRR queues (figure 2). This scenario aims to show if the use of CoS will decrease the delay of the controller traffic.

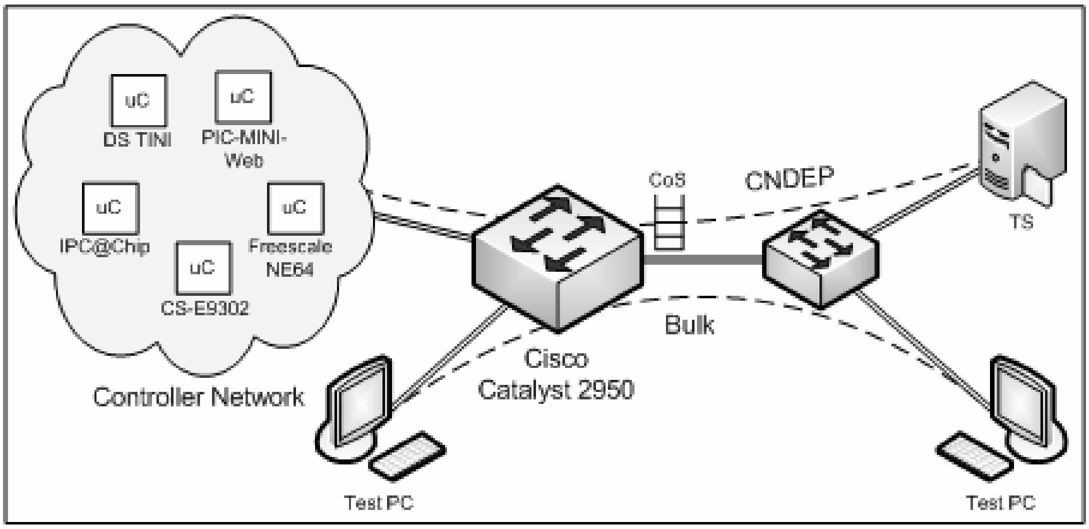

Figure 2

All these scenarios are executed for both protocols. For the ICMP protocol the experiments are executed 100000 times with packets of 64 Bytes size. Experiments for CNDEP protocol are executed 101 times.

3. EXPERIMENTAL ANALYSIS AND RESULTS

In the experiments with ICMP the ping command is used with following parameters: *“ping –c <count> –f –a”*. The option *“-c”* gives the number of sent packets, *“-a”* defines that the intervals between the packets sent should be as small as the round-trip-time of the network, and *“-f”* skips the output info for every packet. The result of the outputs number of sent and received packets, percentage of lost packets, minimum, maximum and average round-trip-time and its deviation. In the scenarios with CNDEP again the round trip time is observed for every embedded system. A test program is developed for the experiments. It is an implementation of CNDEP client with addition for time measure. The time is measured with QueryPerformanceCounter, which has a precision of about 1 microsecond, depending on the processor frequency. The program outputs the measured times, together with command type and queried controller in comma separated file for further analysis.

Table 1: Results for Round Trip Time for ICMP

board	EP9302					PIC				
scenario	*a*	*b*	*c*	*d*	*e*	*a*	*b*	*C*	*d*	*e*
min	0.232	0.243	0.256	0.303	0.268	1.871	1.883	1.889	1.876	1.902
avg	0.289	0.311	0.311	0.395	0.377	1.952	1.964	1.978	2.043	2.044
max	4.019	4.036	4.029	4.773	4.13	6.14	5.724	5.686	6.203	6.213
mdev	0.084	0.088	0.086	0.11	0.103	0.167	0.194	0.172	0.186	0.191
board	IPC					TINI				
scenario	*a*	*b*	*c*	*d*	*e*	*a*	*b*	*C*	*d*	*e*
min	2.673	2.69	2.709	2.726	2.731	1.361	1.374	1.38	1.368	1.396
avg	3.074	3.06	3.081	3.367	3.371	1.603	1.661	1.482	1.813	1.809
max	11.57	8.546	7.647	10.25	8.259	5.671	5.478	5.453	6.149	6.14
mdev	0.375	0.345	0.348	0.508	0.509	0.273	0.241	0.216	0.423	0.423

The results for the scenarios with ICMP are shown in table 1. The average values shown in the table are close to minimal and the deviation is small. The relatively high maximum values can be explained with spontaneous latencies from the communication tasks of the embedded systems and the test computer. Due to the relatively low utilization of the network of the switch and the absence of lost packets, it could be concluded that the major component of the delay is from the processing in the communication stack of the embedded system. This explains the bigger differences in the delay values for different embedded systems. As expected, the best results are obtained when the controllers and the bulk traffics are in separate VLANs. The scenario with CoS gives worse results, perhaps, because of the time needed for packet classification, policing and marking at the switch.

Table 2: Results for Round Trip Time for CNDEP

Board	EP9302					PIC				
Scenario	*a*	*b*	*c*	*d*	*e*	*a*	*b*	*C*	*d*	*e*
Min	1.71	1.70	1.73	1.79	1.79	3.43	3.28	3.32	3.32	3.34
Avg	2.13	2.09	2.14	2.32	1.96	3.96	3.53	3.63	3.97	3.97
Max	30.25	20.10	23.79	29.00	9.60	27.40	12.80	25.60	28.90	49.54
Mdev	0.45	0.24	0.44	0.38	0.08	0.50	0.12	0.28	0.38	2.43
Median	1.85	1.85	1.85	1.95	1.95	3.48	3.45	3.48	3.55	3.61
Board	IPC					TINI				
Scenario	*a*	*b*	*c*	*d*	*e*	*a*	*b*	*c*	*d*	*e*
Min	6.91	6.87	6.92	6.48	6.91	15.39	15.86	15.95	16.11	15.88
Avg	8.15	7.81	7.65	10.01	7.57	16.97	17.17	17.12	17.24	16.85
Max	20.84	29.68	27.03	99.00	13.20	99.82	20.74	47.56	51.22	18.92
Mdev	0.23	0.23	0.30	1.55	0.07	1.05	0.08	0.48	0.35	0.05
Median	7.77	7.68	7.51	7.72	7.57	16.96	17.05	16.99	17.10	16.82

The results for the scenarios with CNDEP are shown in table 2. As it can be seen from the results, the difference between the delays in directly connected and connected through switch (scenarios *a* and *b*) when the switch is not loaded with bulk traffic is extremely small. The expected latency from the switch in this case is around 25-45µsec [6], witch is lower than the error from the statistical processing of the data and the mean value is slightly bigger. Applying CoS on the egress port of the switch reduces the delay much more than the scenario with ICMP. This could be explained by the mechanisms of the CoS that do not allow marking and classification of ICMP packets.

4. Conclusions and Future work

The presented results show that applying CoS policies on the network switch in Controller networks can significantly reduce the message delay to values comparable to the values from the scenarios b and c. However, protocols like ICMP cannot benefit from CoS and the measured delay values are closer to the worst case scenario – scenario d.

The observed delays for all four embedded systems are very diverse. That could be explained with the different access to the network hardware – drivers, APIs, direct or JNI; different operating system (OS) – no OS, real-time OS, general purpose OS; different processor frequency; different Ethernet implementation and thus, different transfer speed limits.

Some future work includes analysis of the delay parameters when multiple controller networks are interconnected through backbone switches and routers (or Internet) and a quality of service (QoS) is applied between entry points of these networks. Further, some protocol with periodic nature (typical case for automation systems) must be included in the experiments to evaluate how parameters critical for real-time process like jitter will be influenced by CoS.

5. ACKNOWLEDGMENTS

The work in this paper is supported by National Science Fund of Bulgaria project – **"VU-966"/2005**, entitled "Web Services and Data Integration in Distributed Automation and Information Systems in Internet Environment".

6. REFERENCES

[1] Alves, M., E. Tovar, and F. Vasques, *Ethernet goes real-time: a survey on research and technological developments*, Tech. Report, The Polytechnic Institute of Porto, Portugal, January 2000.

[2] Costa, P.; J. Netto; C. Pereira, *Analysis of Traffic Differentiation on Switched Ethernet*, In Proc. RTN 2004, 3rd int. workshop on real-time Networks, Italy, 2004.

[3] Georges, J.P., E. Rondeau and T. Divoux, *Classification of service for networked control systems*, Proc. 1st NeCST Workshop, Ajaccio, France, 2005.

[4] Jasperneite, E.; Neumann, P., *Switched Ethernet for factory communication*, Proc. of IEEE International Conference on Emerging Technologies and Factory Automation, pp.205-212 vol.1, 2001.

[5] Kakanakov, N., I. Stankov, M. Shopov, G. Spasov, *Controller Network Data Extracting Protocol – design and implementation*, Proc. of CompSysTech'06, V. Tarnovo, Bulgaria, 2006, pp.IIIA-14-1 – IIIA-14-6.

[6] Kakanakov, N., M. Shopov, and G. Spasov, *Time-Delay Simulation Analysis of Local Controller Networks*, Proc. of CompSysTech'08, Gabrovo, Bulgaria, p. IIIA.12, 2008.

[7] Loeser, J.; Haertig, H., *Low-latency hard real-time communication over switched Ethernet*, Real-Time Systems, 2004. ECRTS 2004. Proceedings. 16th Euromicro Conference on, pp. 13-22, 2004.

[8] Pedreiras, P.; R. Leite, and L. Almeida, *Characterizing the Real-Time Behavior of Prioritized Switched-Ethernet*, In Proc. 2nd RTLIA, Porto, Portugal, June 2003.

[9] Beck IPC GmbH, IPC@CHIP, website: http://www.beck-ipc.com/en/products/sc1x/, [Accessed: May 2008].

[10] Bjr-labs, RFID board with Ethernet LAN, website: http://rfid.bjr-labs.com/, [Accessed: May 2008]

[11] Cisco Systems, Cisco Catalyst 2950 documentation, website: http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/, [Accessed July 2007].

[12] Dallas Semiconductor, TINI - Tiny InterNet Interfaces, website: http://www.maxim-ic.com/products/microcontrollers/tini/, [Accessed: May 2008].

[13] Olimex, EP-9302 development board, http://www.olimex.com/dev/cs-e930x.html, [Accessed: May 2008]