ON THE USE OF NS-2 IN SIMULATIONS OF INTERNET-BASED DISTRIBUTED EMBEDDED SYSTEMS

Mitko P. Shopov, Nikolay R. Kakanakov,Grisha V. Spasov,
Department Computer Systems, Technical University of Sofia, Plovdiv branch

Abstract: *The paper discusses the application of Network Simulator (NS-2) in simulation of Internet-based distributed embedded systems. It envisions the pros and cons of NS-2 as a simulation tool for distributed embedded systems and suggests some design techniques to implement simulation scenarios. At the end, an example of time-delay simulation of switched Ethernet as a communication media for distributed embedded systems is presented.*

Keywords: *Network Simulations, Distributed Embedded Systems, Switched Ethernet, NS-2.*

1. INTRODUCTION

Nowadays, control and automation has become more complex due to the task and processes that must be managed. Complex automation processes cannot be controlled and monitored by a single controller. This has lead to new research and development in the field of distributed automation. In such systems, the control loop is closed over some network and the network parameters as delay and connection speed must be studied and tuned. Different standards and communication technologies are proposed to solve this problem. Some of them are fully customized, other are based on popular standards form the business and office networks. As long as Ethernet is the most often used network standard, more and more research efforts are done on its application in distributed automation and control. The unpredictability of the MAC algorithm when collision occurs was the main problem in the earlier standards of Ethernet. With the release of 100 and 1000 Mbps standards and the addition of full-duplex lines and micro-segmentation, this problem is obsolete. The research efforts are shifted towards congestion in the network switches itself, their multiplexing latencies and Ethernet traffic prioritization schemes [2, 5, 9, 12].

The distributed approach in the design of complex systems can bring a lot of benefits compared to the centralized one. This also applies to the design of embedded systems. Besides these benefits, the design of distributed system has many requirements for the building blocks and communication middleware. This makes the distributed systems so complex that are difficult to design correctly without an appropriate model. In [17] three models to help the designers to understand and evaluate both system requirements and implementations are identified. One of these – the simulation model involves the use of executable programs to demonstrate emergent system behavior. Different type of simulation techniques exists: discrete event, cycle-based, coupled analytic, and others. These techniques can use different configuration of the workload inputs like random, abstract, or even traces [17].

Network Simulator (NS-2) [13] is one of the most popular tools in academia for evaluation of network protocols and topologies. It represents a discrete, event-based simulator that has an ability to be easily extendible and modifiable due to its open source nature. NS-2 has a good set of supported queue management policies. Besides its advantages, the NS-2 simulator misses some of the most commonly used network elements and there are extra efforts needed in the simulation preparation phase. In this paper, some techniques used to simulate intelligent network switch will be presented.

2. SIMULATION TECHNIQUES

The node element is a basic primitive in NS-2. It generally consists of two classifiers: address and port. These classifiers are used to distribute incoming packets to the correct agent or outgoing link. The existing implementation of NS-2 node element does not include more complex elements like intelligent layer 2 switch. However, by using the provisions of the link and node primitives in NS-2, it is possible to emulate such a device in the Otcl level with a little more efforts [13].

For the use-case an off-the-shelf network device Cisco Catalyst 2950 (1P3Q1T) will be used. The specification 1P3Q1T stands for one strict priority (SP) queue, three weighted round-robin (WRR 70/25/5 %) queues and one trigger. These queues configuration is used to implement CoS (class of service). Class of Service (CoS) mechanisms reduce flow complexity by mapping multiple flows into a few service levels. Instead of the fine grain control of QoS, CoS applies bandwidth and delay to different classes of network services. Two common CoS mechanisms are: IEEE 802.1p – tagging (Layer 2); and Type of Service (ToS) – prioritization (Layer 3) [5, 18].

Since not all of the queue types used in the CoS implementation on the real device has realization in NS-2, their functions will be mimic with the use of CBQ/WRR [14] queue management mechanism (figure 1). First, the inbound traffic is classified and the SP traffic is forwarded without any hold. This is implemented on the link between node #1 and node #2. The CBQ here is realized as three node tree with one root and two leafs. The bandwidth is set to 100% for the SP traffic and no reservation is made for the rest. This means that the other traffic will be forwarded only when there is extra bandwidth not used by the SP traffic. The mechanism used is formal link share mechanism. Since in NS-2 the delay is added on the link primitive and not on the node, the latency of the switch is emulated by the delay of the link between two nodes (figure 1).

The other traffic is already separated from the strict priority, but is not yet classified and prioritized. This occurs on the outbound link between node #2 and the network segment. On this step the traffic is classified by the CBQ/WRR, mapped to the CoS scheme of the real device and WRR scheduling is made for the packets in the three queues.

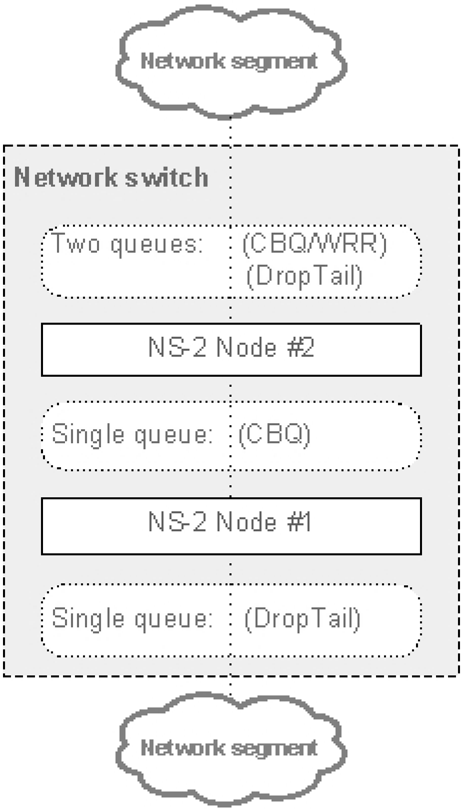

Fig.1: Implementation on the element ‘Network switch’ in NS-2.

The drop tail queues shown on figure 1 represent FIFO buffers of the network device. The shown connections represent the physical links in the topology. The logical links are not shown on the figure. Next sections will show a use-case on how the techniques from above are applied in the simulation of Ethernet for industrial networks.

3. SIMULATION SCENARIOS

In traditional network applications the main parameters that are evaluated are the performance and the throughput. In distributed real-time and embedded systems the most valuable parameters are the delay of the messages, the probability of error messages and the jitter – the deviation of the delay [3, 5]. These parameters are closely related with the parameters of the local network. On the network nodes main parameters are the message size, packets distribution times, delay from the node's communication stack. On the communication subsystem parameters are network topology, bandwidth, bit error rate, capacity of the switches, switch priority and queue management mechanism [6, 10].

3.1. Traffic types and workload distribution

The choice of appropriate traffic types and load distribution is a key factor in evaluation of a communication protocol. In distributed embedded system environment, devices exchange data between each other and with a master controller. Specific characteristics of the exchanged data and the time of exchange are the main differences from the office networks. In a network of controllers there are four main scenarios for data exchange. The first scenario is exchange of request-reply pairs for control or monitoring of devices. The second scenario is sending configuration data to devices. The third scenario is diagnostics – the master request specific parameters and the devices sends their values periodically. The last scenario is sporadic sending of alarms [3, 11].

Based on this exchange scenarios and the probability of their occurrence, the system workload can be defined. Most of the authors define four classes of workloads for controller networks. The first class (WT1 in the paper) is mapped to the exchange of messages for diagnostics, monitoring and control. It is about 75-90% of the overall traffic. The second class (WT2) is mapped to the exchange of event-driven exceptional messages – alarms. It is about 5-15% of the overall traffic and usually has Poisson distribution of the packets inter-arrival times.

The third class (WT3) is mapped to the configuration scenario and includes configuration of the device and the network itself (DHCP, STP, CDP, etc.). Its part is under 5%. The last class (WT4), in contrast to other three classes, exchanges big-sized messages. Its part is under 1% of the overall traffic and occurs mostly in the initialization part of the lifecycle of the systems. It includes file and/or code upload to devices [4, 6]. The main parameters are shown in Table 1.

Table 1. *The experimental measuring results*

SW		WT1		WT2		WT3		WT4	
R [bps]	$1.0x10^8$	L [bits]	576	b2 [bits]	4096	b3 [bits]	16384	b4 [bits]	48704
S [s]	$4.5x10^{-5}$	T [s]	$1.0x10^{-4}$	r2 [bps]	$4.0x10^4$	r3 [bps]	$1.0x10^5$	r4 [bps]	$1.0x10^6$
				M2 [bits]	2048	M3 [bits]	8192	M4 [bits]	12176
				p2 [bps]	$1.0x10^8$	p3 [bps]	$1.0x10^8$	p4 [bps]	$1.0x10^8$

3.2. Simulation scenario

In the context of the multi-tier model for distributed automation suggested in [15] the master node in the controller network is represented by a transaction server (TS). The slave nodes are presented by a set of controllers, typically around 15-20 controllers per network segment. Since we assume full-duplex micro-segmented connections and only master-slave communications we can reduce the complexity of the simulated topology to the one shown on figure 1. It could be concluded that the master-slave communication will lead to a highly asymmetric workload in the switch node, thus the connection between TS and switch node is chosen to be 1Gbps and the connections with controllers – 100Mbps (figure 2).

The output buffers of the controller devices are not a subject of interest for the current case and a simple FCFS (First-Come-First-Served) queue management for the links between controllers and the switch is used. Of major concern here is the queue management of the output buffer of the switch port connected to the TS. Performance parameters of all four traffic types are directly influenced by this choice.

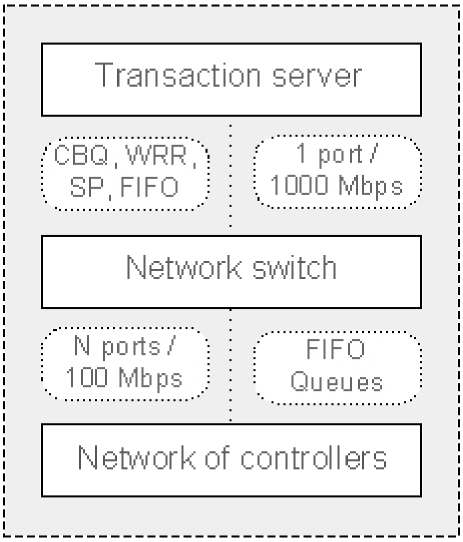

Fig.2:Simulation topology – abstract view.

The queue management to be used is a combination of SP (strict priority) and WRR (weighted round-robin) to retain maximum closeness to the implementation in the off-the-shelf network device (1P3Q1T – Cisco Catalyst 2950 switch). It consists of one SP queue and three queues in WRR (70/25/5 %). Implementation in NS is based on a CBQ/WRR (figure 1). First, the inbound traffic is classified and the SP traffic is forwarded without any hold. The other traffic is forwarded to a CBQ node and further classified in three queues and WRR scheduling is made for the packets in the three queues.

The selection of NS traffic generators should be as much realistic as possible to ensure accurate results. Traffic WT1 is implemented as several UDP/CBR (Constant bit rate). Traffic WT2 is implemented using TCP as transport agent to ensure guaranteed delivery and Exponential application generator to map the sporadic nature of alarm messages. Traffic WT3 is characterized by burst exchange of large packets (during startup and network reconfiguration) and small packets at fixed periods (keep-alive messages). The first one is implemented as Exponential and the second one as a CBR. Traffic WT4 is implemented as TCP/HTTP/FTP (figure 3).

4. SIMULATION RESULTS

Each simulations run for a period of 120 seconds. CBR traffic generators are sequentially started during the first second of the simulation and are stopped at the 100-th second of the simulation. This allows analysis of the influence of the high priority traffic on other traffic types that are running during entire simulation. During the simulation the following information is collected: end-to-end packet delay for each flow, jitter for the periodic flow, bandwidth utilization, backlog, and packets loss of the switch port. The collected traces are processed and the minimum, maximum and average delays are extracted and compared to analytical bounds to examine correctness of the simulation model – Table 2 [16].

Table 2. *The experimental measuring results*

Flow	Simulation results, Delay [ms] min / avg / max	Analytical results, Delay [ms]
WT1	27 / 27 / 35	37
WT2	37 / 38 / 201	204
WT3	28 / 77 / 290	470
WT4	28 / 75 / 248	981

For most of the packets the jitter value is zero, only 0.18% examines jitter and only 0.02% examines the maximum jitter of 8 µs. This is due to the SP scheduling used for the periodic flow. The jitter only occurs when there is a packet from some other flow that has already occupied the switch. The maximum jitter is observed when the packet needs to wait transmission of maximum sized packet from some other flow 11.5sµ. The distribution of packets according to their jitter is shown on figure 4.

As can be seen from figure 5, most of the WT1 packets examine 27 µs delay which means that they are forwarded without queuing. For WT3 flow packet delays are mostly distributed around two values. The smaller one occurs when they arrive at the switch in the idle period of the WT1 flow. For WT4 flow packet delays are distributed like those of WT3 but are bigger because of its relatively low weight in WRR scheduling. The delay of WT2 flow has little influence from WT3 and WT4 because it has relatively high weight in the WRR scheduling.

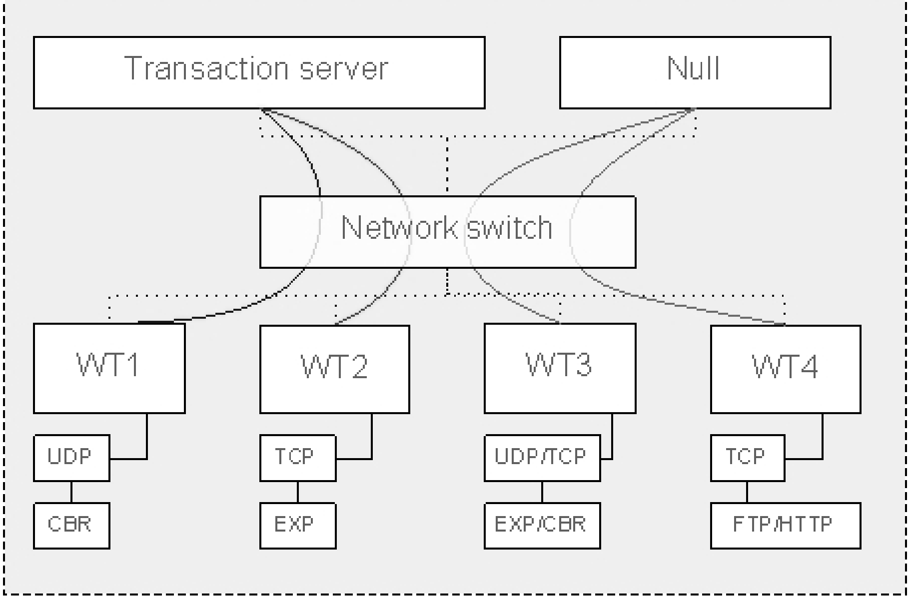

Fig.3: Simulation topology – traffics view.

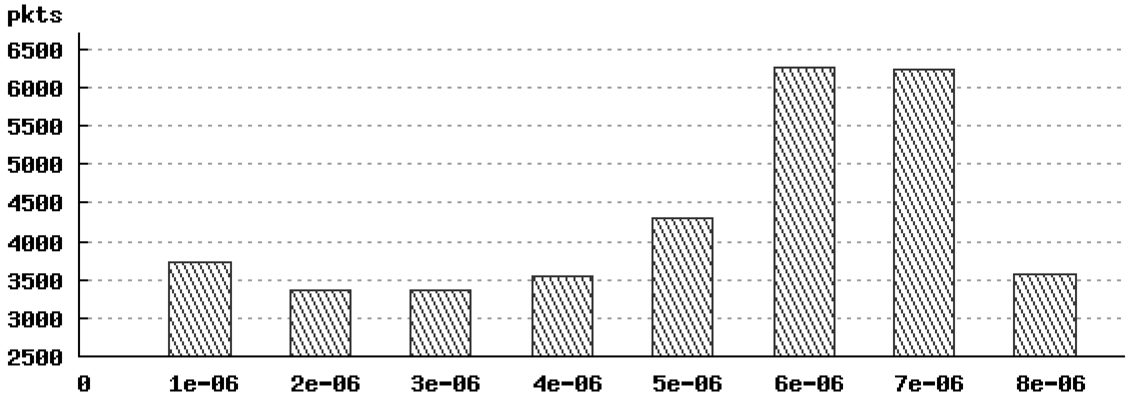

Fig.4: Jitter density

The average bandwidth utilization of the link switch-TS is about 80 Mbps during first 100 seconds and 10 Mbps afterwards. After stopping of the WT1 flow in the 100-th second of the simulation the delay of other flows reduces significantly and is becomes closer to its average value.

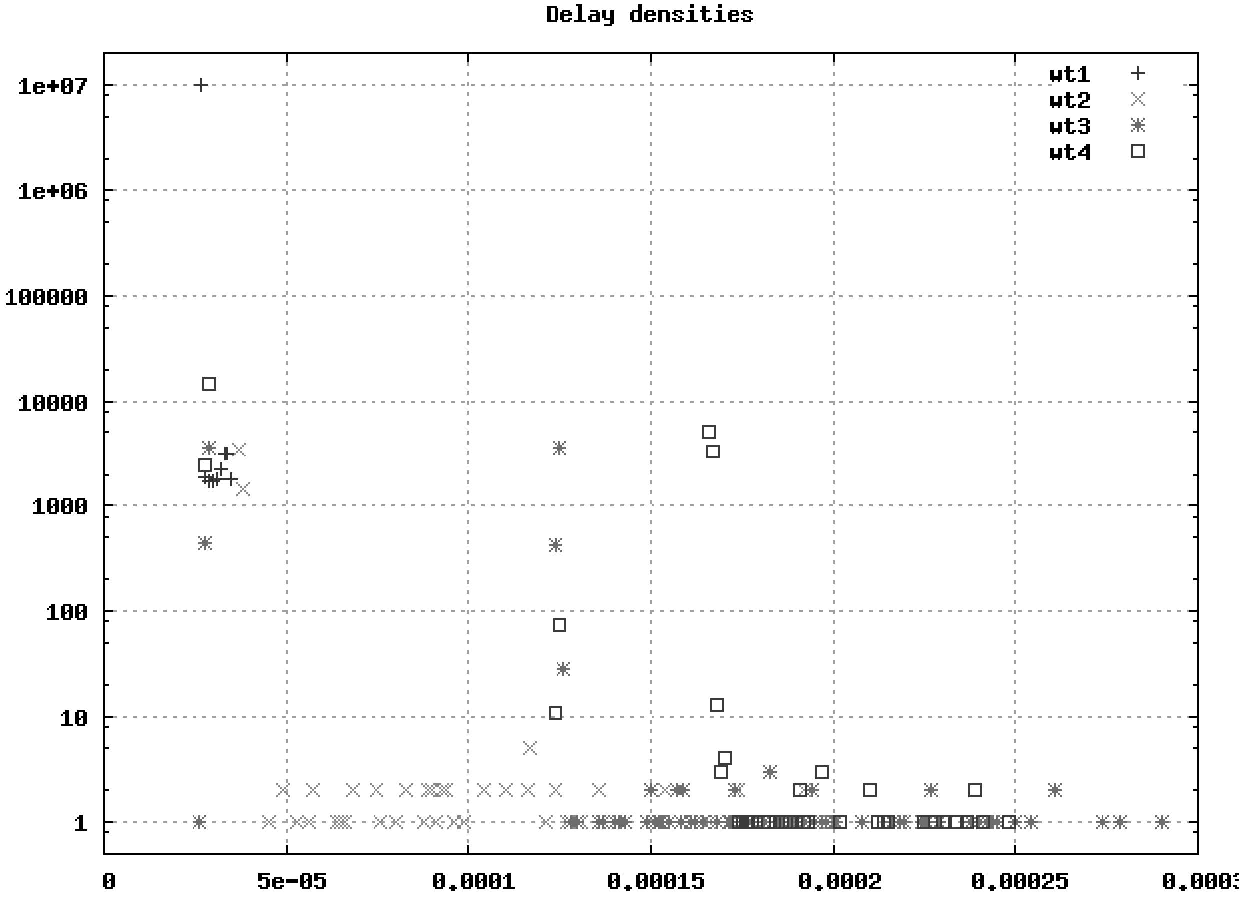

Fig. 5: Delay densities

3. CONCLUSION

This paper introduces the application of NS-2 in simulation of the performance of Ethernet for building distributed embedded systems. Although, it lacks some of the means of other commercially available simulators, it has its strengths in its open-source nature, freely distribution and its embodied dual language support. Many new elements and functionality can be added at Otcl level. These features make it a competitive solution in distributed embedded systems simulations.

The presented simulation results must be further checked against corresponding test-bed experiments, using market-available switches and controllers [17]. Further analysis must be made for other components of the end-to-end delays to obtain the complete view. The results can be used for analysis of the data flows and QoS policies in the context of the multi-tier model for Distributed Automation [15] and especially to model the behaviour of its lowest tier – the Data producing tier.

4. ACKNOWLEDGEMENTS

The presented work is supported by National Science Fund of Bulgaria project – “ВУ-966/2005”, entitled “Web Services and Data Integration in Distributed Automation and Information Systems in Internet Environment”, under the contract “ВУ-МИ-108/2005”.

5. REFERENCES

[1] Cruz, R.L., "Quality of service guarantees in virtual circuit switched networks ," Selected Areas in Communications, IEEE Journal on, vol.13, no.6, pp.1048-1056, 1995.

[2] Culler, D. E., J. Hill, P. Buonadonna, R. Szewczyk, and A. Woo, “A network-centric approach to embedded software for tiny devices,” in Proc. of the First International Workshop on Embedded Software (EMSOFT), Oct. 2001

[3] Fonseca, José Alberto, Paulo Pedreiras ,"Fieldbuses in real-time distributed embedded systems," 2nd International Workshop on Embedded Systems, Internet Programming and Industrial IT, Kiel, Germany, 20-22 August, 2003

[4] Georges, J.P., E. Rondeau and T. Divoux, "Classification of service for networked control systems," In Proc. 1st NeCST Workshop on Networked Control System Tolerant to faults. Ajaccio, France, 2005.

[5] Hendrey, G.R., "Standard Ethernet as an Embedded Communication Network", MSc Thesis, Dept. Electrical and Computer Engineering, Carnegie-Mellon University, 1999.

[6] Jasperneite, E.; Neumann, P., "Switched Ethernet for factory communication," Emerging Technologies and Factory Automation, 2001. Proceedings. 2001 8th IEEE International Conference, vol., no., pp.205-212 vol.1, 2001.

[7] Le Boudec, J.-Y.; Thiran, P., "A short tutorial on network calculus. I. Fundamental bounds in communication networks," Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on , vol.4, no., pp.93-96 vol.4, 2000.

[8] Loeser, J.; Haertig, H., "Low-latency hard real-time communication over switched Ethernet," Real-Time Systems, 2004. ECRTS 2004. Proceedings. 16th Euromicro Conference on , vol., no., pp. 13-22, 30 June-2 July 2004.

[9] Otanez, P. G., J. T. Parrott, J. R. Moyne, and D. M. Tilbury, "The implications of Ethernet as a control network," In Proceedings of the Global Powertrain Congress, Ann Arbor, September 2002.

[10] Ruping, S., E. Vonnhame, J. Jasperneite, "Analysis of Switched Ethernet Networks with Different Topologies Used in Automation Systems," in Proceedings of Fieldbus Conference (FeT’99), Magdeburg, Germany, pp. 351-358, Springer-Verlag, September 1999.

[11] Topp, U. and P. Mueller, Web based service for embedded devices, Lecture Notes in Computer Science, Vol.2593, pp. 141-153, 2003.

[12] Yiming, Alimujiang; Eisaka, T., "Development of an Ethernet protocol for industrial real-time communications," Communications and Information Technology, ISCIT 2004. IEEE International Symposium on, vol.2, no., pp. 1122-1125, 2004.

[13] Official NS-2 website: http://www.isi.edu/nsnam/ns/.

[14] Floyd, S. and V. Jacobson, “Link-sharing and Resource Management Models for Packet Networks,” IEEE/ACM Transactions on Networking, Vol. 3, No. 4, 1995.

[15] N. Kakanakov, M. Shopov, and G. Spasov, "A new web based multi-tier model for distributed automation systems," Journal Information Technology and Control, vol. 2 , pp.44-48, 2006.

[16] N. Kakanakov, M. Shopov, and G. Spasov, “Time-Delay Simulation Analysis of Local Controller Networks,” International Conference on Computer Systems and Technologies - CompSysTech’08, June 2008, p. IIIA.12.

[17] Koopman Ph., System Architecture & Design Methodologies, (accessed: 2008-09-20), http://www.ece.cmu.edu/~koopman/system.html

[18] Intel Corporation, “Differentiated Services: Moving towards Quality of Service on the Ethernet”, Intel Networking White Papers, (accessed: 2007-04-25).