Versatile FPGA-based Hardware Platform for Gigabit Ethernet Applications

Matei Ciobotaru*, Mihail Ivanovici*, Razvan Beuran*, Stefan Stancu†,
* CERN,
1211 Geneva 23, Switzerland
and
"POLITEHNICA" University
Bucharest, Romania
Email: matei.ciobotaru@cern.ch, mihail.ivanovici@cern.ch, razvan.beuran@cern.ch
† University of California, Irvine
Irvine, CA 92697-4575
Email: stefan.stancu@cern.ch

***Abstract*— Hardware platforms are most suitable to Gigabit Ethernet applications, which require high packet-processing rate capabilities. FPGA-based implementations offer the possibility of changing the functionality of the platform to perform several tasks. Using an FPGA-based custom-design PCI platform we developed two highly-configurable applications: a Gigabit Ethernet Tester and a Network Emulator.**

The Gigabit Ethernet Tester, a flexible traffic generator, makes it possible to evaluate network equipment before deployment in a complex network infrastructure, as well as to assess the performance of in-place network devices. We describe the use of this system for the evaluation of Gigabit Ethernet switches, as part of the design process of the ATLAS data acquisition network at CERN.

The Network Emulator project implements a "network-in-a-box" that permits reproducible experiments in realistic scenarios. By introducing controlled degradation to the network traffic, we study real-application behaviour under a wide range of specific network conditions. We investigate the performance of commonly-used network applications, such as file transfer, Internet telephony (VoIP) and video streaming.

I. Introduction

Despite the steady increase of CPU power, conventional CPU based systems are still overwhelmed when faced with high-frequency I/O operations. Gigabit Ethernet applications which require line-rate processing for all frame sizes call for hardware implementation. FPGA based-platforms fulfill the performance requirements and provide extra flexibility in comparison to ASIC implementations. Using an FPGA-based custom-design PCI platform we developed tools that make it possible to perform two different tasks:

1) characterize Gigabit Ethernet devices so that they can be selected based on pre-defined requirements;
2) study application behaviour under given network conditions and determine the dependency between application performance and experienced network quality degradation.

The paper first describes the Gigabit Ethernet tester. Assessing the performance of the existing Gigabit Ethernet switches is a prerequisite for the design of any high-performance specialized network, such as the one to be used in the ATLAS[1] data acquisition system. The quantitative evaluation of the degradation likely to be introduced by the switches allows to predict the behaviour of network applications under given network conditions. Our second application is the Network Emulator. Emulation is a technique that allows determining the dependency of application performance on network quality degradation.

Combining the results from the active measurements on real switches with the ones from application performance assessment using a network emulator offers the possibility of having a global view on how the whole system will behave, before actually building and deploying it. It also allows for architectural choices in the process of designing the network, based on the application-level requirements.

A. State of the art

Platforms that are using hardware acceleration for network-based applications have been built before. In our group we developed the Enet32 FastEthernet Tester based on FPGAs [1] and a Gigabit Ethernet Tester based on the Alteon programmable network interface cards [2]. A system similar to the one described in this article is the GNET-1 [3] that provides functions for network emulation and traffic generation. Commercial applications are also available. Celoxica, the company that makes the Handel-C language compiler, has the RC Series of development platforms, which provide a complete environment that can be used among other things for networking applications [4].

Grace to the flexible design of both the hardware and the low level firmware, the system we describe can be used to implement various Gigabit Ethernet applications. This feature is achieved through the use of programmable hardware and of a high-level programming language, Handel-C from Celoxica [5], enabling a rapid implementation of algorithm into hardware. Moreover, the platform provides high port count testing

[1] One of the experiments being built at CERN, Geneva, Switzerland.

equipment, at a fraction of the cost of equivalent commercial systems.

The methodology for testing switching devices is currently specified in various RFCs (Request For Comments from IETF) [6], [7], [8], [9]. These RFCs define the types of traffic to be used for benchmarking switches, the parameters to be measured, pre-defined states to be tested etc.

Many commercial solutions are available for network testing—such examples are the Ixia 1600T [10] and the Spirent Smartbits systems [11]. However these are mainly oriented toward the market of device manufacturers and lack the flexibility required for specialized research and development. On the other hand, using customizable platforms allows performing various special tests. For example we developed a methodology of measuring the size of internal switch buffers, which requires special traffic patterns to be sent to the switch under test. This can not be achieved with the commercial testers that were built to perform mainly the tests described in benchmarking RFCs.

There are several network emulators available at the moment, many of them as software implementations. They can be installed and run on ordinary personal computers, which makes them very attractive for building cheap experimental setups, such as NIST Net [12] or ONE [13]. In the case of all software implementations, the accuracy of the degradation they introduce, e.g. delay, cannot be guaranteed. Hardware implementations exist as well, and the number of commercial emulators is ever increasing. Products such as Shunra Virtual Enterprise [14], LANForge-ICE [15] are readily available on the market. The main problem with all the existing implementations of network emulators is that the loss and delay they introduce is not correlated, therefore the degradation that a certain traffic flow will experience is not realistic. Since each packet is treated independently, the delay variation may lead to packet reordering which in real network equipment cannot occur when there is only one way from the input port to the output port.

The delay variation should be naturally induced and the original packet sequence should be preserved inside a certain stream. All these requirements are met by our system through the use of an approach that emulates the interdependence between packet loss and delay experienced by traffic flows.

After assessing the degradation introduced by the switching mechanism itself, we quantitatively measure the degradation introduced by the service differentiation algorithms implemented in the existing Gigabit Ethernet switches, especially by the scheduling mechanisms.

The tests are conducted according to the methodology described in [16]. The results of our tests are compared against the user expectations based on ideal switch models.

II. The Hardware Platform

The Gigabit Ethernet platform is based on a PCI card that was designed and built at CERN. The main component of the card is an Altera Stratix FPGA. The floor plan of the card is shown in Figure 1.

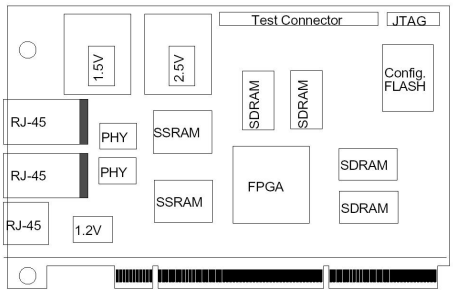


Fig. 1. Floor plan of the PCI card.

Apart from the central FPGA, the card has two Gigabit Ethernet ports, one port for GPS clock synchronization and memory used for packet processing (SDRAM and SRAM). The part of the FPGA firmware that provides the high-level functionality (user application) is written in the Handel-C language [5]. In addition to the user code, the Ethernet MAC, PCI and SDRAM controllers are also integrated into the FPGA. Using this approach we obtained a simpler board layout. More information about the platform is available in [17].

The board fits into a standard PCI connector. Using PCs that provide several PCI slots we built a testbed comprising of 64 cards distributed in 15 industrial PCs. A photo of the system connected to a device that came for evaluation is shown in Figure 2.

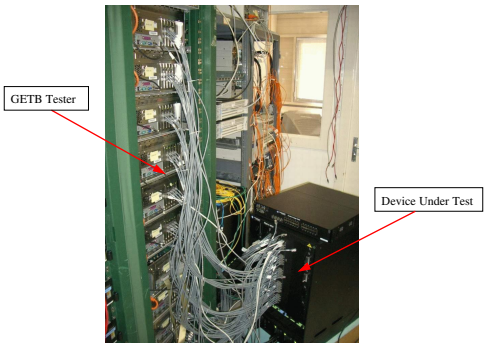


Fig. 2. Gigabit Ethernet testbed.

For the control infrastructure we use the Python scripting language [18]. From a central workstation we can configure and monitor all the cards available. The configuration is done via scripts or from the command line, while the monitoring of the statistics can be done also using a graphical interface. Python was chosen because it allows us to write scripts that drive the system automatically without user intervention. The software currently runs on the Linux OS, but in principle it supports any platform that can run a Python interpreter.

III. The Network Tester

The network tester is the primary application of the FPGA-based Gigabit Ethernet system. Its design was driven by the

need to evaluate network equipment for the ATLAS experiment at CERN [19]. The ATLAS data acquisition system relies on a large Gigabit Ethernet local area network. The real-time nature of the application imposes strict requirements for the network in terms of packet loss, throughput and delay. The tester is used to verify that all switching devices deployed in the network meet these specifications.

A detailed description of the tester project can be found in [17]. Here, we only give an overview of its main features.

The strong point of the tester is the ability to generate traffic at Gigabit line-speed and to compute averages and histograms of the network parameters (throughput, delay and packet loss) in real time. Other statistics, such as min, max, n^{th} percentile can be made available.

The stream of packets is generated according to a set of descriptors that are loaded into the memory of the card. A packet descriptor contains information about source, destination, packet size and inter-packet gap. The generated packets can be both Layer 2 and Layer 3 (IPv4 and IPv6). The number of descriptors is only limited by the amount of available memory (currently 2 million).

Depending on the values of the various descriptor fields, which are computed offline, different traffic patterns can be generated. For example, using the value of the inter-packet gap field one can produce any arrival-pattern distribution, such as Constant Bit Rate, Poisson or Erlang.

In addition to the descriptor-based mode, a special mode exists in which the tester emulates a request-reply system. This produces traffic similar to the one between the applications of the ATLAS data acquisition system.

A. Sample results for the Gigabit Ethernet Tester

We further present some results obtained using the network tester: a fully-meshed traffic and a QoS test.

1) Fully-meshed throughput test: During this test every ports sends Constant Bit Rate traffic to all the other, with a uniform random choice of destination. The frame size during a test trial is constant. The following frame sizes have been used: 64, 135, 512, 1027 and 1518 bytes. The 64 byte frames correspond to the worst case in number of events received by the switch and can be used to characterize the per-packet processing overhead. The 1518-byte packets correspond to the most data-intensive case, hence it shows the maximum amount of data that can be forwarded by the device. Three middle range sizes are also used: 135, 512 and 1027. Two "odd" values (135 and 1027) have been chosen in order to reveal the device behaviour for frame sizes which are not common in benchmarking tests ([20] recommends the use of 64, 128, 256, 512, 1024, 1280 and 1518-byte frames for testing switching devices).

Figure 3 illustrates the loss rate versus offered load. The loss rate is zero (or insignificant) for loads inferior to approximately 95%, and becomes non-negligible (yet smaller than 1.5%) for higher loads.

2) QoS test: We configured 8 ports to send traffic with 8 different priorities to the 9^{th} port. Each transmitter is sending

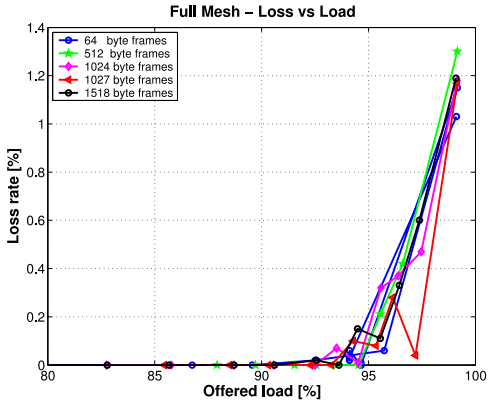


Fig. 3. Fully-meshed throughput results.

at the same rate to the destination, but with a different priority level. In this test we use 802.1q VLAN tagged frames. The priority is part of the VLAN tag and up to 8 priorities can be used. Additional information concerning the evaluation of the delivery QoS characteristics of Gigabit Ethernet switches can be found in [16].

For a given input transmitted rate (load) the receiving port records the packet loss, latency and throughput for each traffic priority. By varying the input load we obtain curves that show the received throughput observed for each flow.

We ran this test on two different devices, denoted here by A and B. These devices implement a Strict Priority scheduling mechanism—device A on 8 queues, device B on 4 queues. For device A we expect that each priority will go to separate queue and for device B we expect that priorities 0 and 1 will be assigned to Queue 1, priorities 2 and 3 will go to Queue 2 and so on. Figure 4(a) illustrates the behaviour of Device A: the QoS scheduling mechanism is properly implemented.

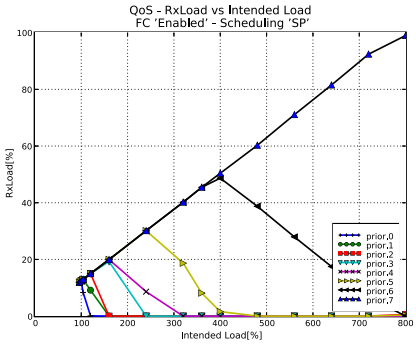
The same test was performed on Device B and the result is shown in Figure 4(b). From the figure we can easily observe that the scheduling is not done properly, since all flows are given approximately the same bandwidth, regardless of their priority.

IV. The Network Emulator

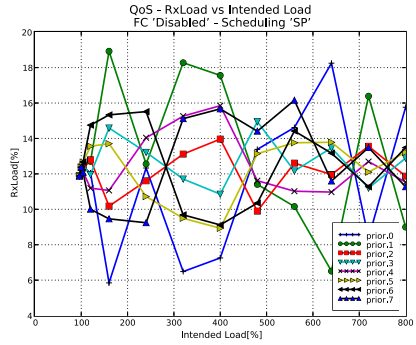
The network emulator is a "packet processor", with a general architecture composed of a "packet path" and a "control path" (Figure 5). The Packet Path (Figure 5(a)) consists of a storage block (the Packet Data Storage) and receiving/forwarding modules (i.e. the Packet Data Receiver and the Packet Data Forwarder). The Control Path (Figure 5(b)) processes the packet references (structures that allow the identification of packet inside the system) by applying different service degradation that acts on traffic flows through packet loss, delay and throughput limitation. It also interacts with the packet path and has the knowledge about the place (memory address) where packets are stored.

A. Application Performance Assessment

Network emulation is a technique that allows the assessment of real network application performance in a laboratory setup.

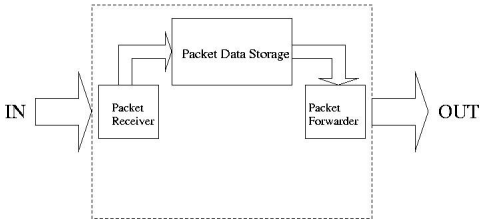


(a) proper implementation

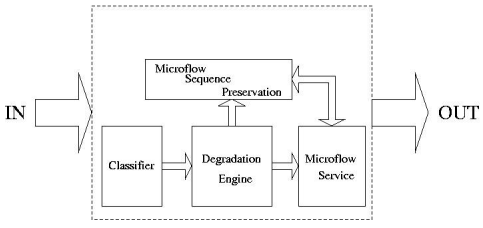


(b) erroneous implementation

Fig. 4. QoS tests for Strict Priority scheduling.



(a) Packet Path



(b) Control Path

Fig. 5. Network Emulator architecture.

By controlling the quality degradation introduced by a network emulator, one can study the application behaviour in a wide range of network conditions.

To assess the application performance we use the setup depicted in Figure 6. A detailed description of this system can be found in [21], [22]. By UPQ we denote the User-Perceived Quality quantified in a specific way for each class of application under test. Note that application experiments were performed so far only with a freely-available network emulator, NIST Net [12], which treated independently the loss and delay. The Network Emulator that we currently develop uses a novel approach to emulation that ensures more realistic reproduction of network conditions by jointly dealing with delay and packet-loss in order to emulate very closely reality. Initial tests have allowed us to study the behaviour of short-lived TCP transfers by HTTP.

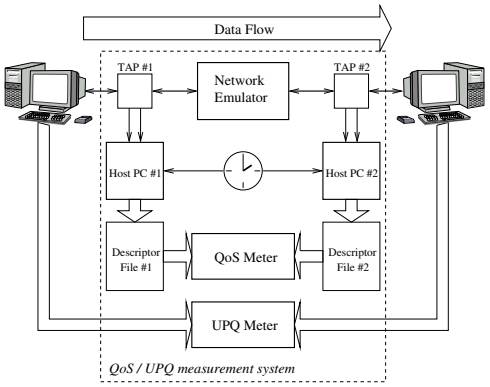


Fig. 6. Application performance assessment setup.

V. Conclusions

We have shown the versatility of FPGA-based hardware platforms. Two Gigabit Ethernet applications were presented, a network tester and a network emulator. The network tester is currently used to evaluate network equipment intended to be used in the ATLAS TDAQ network. The network emulator is still in an intermediate stage of development. The basic functionality is implemented and tests with applications will be carried on in the near future.

The versatility of our platform derives from the fact that different Gigabit Ethernet applications can be implemented on the same board. The firmware provides a low-level library that allows a modular design of the applications implemented on this platform; the functions it provides are general enough to allow building on them various applications. For each application, custom functions can be easily added, an important requirement in a research environment. Moreover, the use of this platform provides a high-port density at a relatively low cost.

Another application, which is used for traffic emulation in the ATLAS data acquisition system, has already been implemented [17]. Additional usages can be envisaged, such as a monitoring system, similar to the one described in [21], running at 1 Gb/s.

REFERENCES

[1] F. R. M. Barnes, R. Beuran, R. W. Dobinson, M. J. LeVine, B. Martin, J. Lokier, and C. Meirosu, "Testing ethernet networks for the atlas data collection system," *IEEE Trans. Nucl. Sci.*, vol. 49, pp. 516–520, Apr. 2002.

[2] R. Dobinson, S. Haas, K. Korcyl, M. J. LeVine, J. Lokier, B. Martin, C. Meirosu, F. Saka, and K. Vella, "Testing and modeling ethernet switches and networks for use in atlas high-level triggers," *IEEE Trans. Nucl. Sci.*, vol. 48, no. 3, pp. 607–612, 2001.

[3] Y. Kodama, T. Kudoh, R. Takano, H. Sato, O. Tatebe, and S. Sekiguchi, "Gnet-1: Gigabit ethernet network testbed," in *Proc. IEEE International Conference on Cluster Computing (Cluster2004)*, 2004, pp. 185–192.

[4] Celoxica. Celoxica RC250 Platform. [Online]. Available: http://www.celoxica.com/products/rc250/default.asp

[5] ——. The Handel-C Programming Language. [Online]. Available: http://www.celoxica.com/technology/c_design/handel-c.asp

[6] S. Bradner, "Benchmarking terminology for network interconnection devices," RFC 1242, July 1991.

[7] S. Bradner and J. McQuaid, "Benchmarking methodology for network interconnect devices," RFC 1944, Mar. 1999.

[8] R. Mandeville, "Benchmarking terminology for lan switching devices," RFC 2285, Feb. 1998.

[9] R. Mandeville and J. Perser, "Benchmarking methodology for lan switching devices," RFC 2889, Aug. 2000.

[10] Ixia. Ixia Performance Testing. [Online]. Available: http://www.ixiacom.com/products/performance_applications/

[11] Spirent. Smartbits. [Online]. Available: http://www.spirentcom.com/

[12] National Institute of Standards and Technology. NIST Net. [Online]. Available: http://snad.ncsl.nist.gov/itg/nistnet/

[13] M. Allman, A. Caldwell, and S. Ostermann, "One: The ohio network emulator," Ohio University Computer Science, USA, Tech. Rep. TR-19972, Aug. 1997.

[14] Shunra. Shunra Virtual Enterprise. [Online]. Available: http://www.shunra.com/products/VirtualEnterprise.php

[15] Candela Technologies. LANForge-ICE. [Online]. Available: http://www.candelatech.com/lanforge_v3/lf_marketing.html

[16] R. Beuran, M. Ivanovici, N. Davies, and B. Dobinson, "Evaluation of the delivery qos characteristics of gigabit ethernet switches," CERN, Switzerland, Tech. Rep. CERN-OPEN-2005-002, Dec. 2004.

[17] M. Ciobotaru, S. Stancu, M. J. LeVine, and B. Martin, "GETB, a Gigabit Ethernet Application Platform: its Use in the ATLAS TDAQ Network," in *Proc. IEEE Real Time 2005 Conference*, Stockholm, Sweden, June 2005, p. (to appear).

[18] The Python Programming Language. [Online]. Available: http://www.python.org/

[19] CERN. ATLAS—A Toroidal LHC ApparatuS. [Online]. Available: http://www.atlas.ch/

[20] S. Bradner and J. McQuaid, "Benchmarking terminology for network interconnect devices," RFC 2544, May 1999.

[21] R. Beuran, M. Ivanovici, B. Dobinson, N. Davies, and P. Thompson, "Network quality of service measurement system for application requirements evaluation," in *Proc. International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, Montreal, Canada, July 2004, pp. 380–387.

[22] R. Beuran, M. Ivanovici, and V. Buzuloiu, "File transfer performance evaluation," *Scientific Bulletin of University "POLITEHNICA" Bucureşti*, vol. 66, no. 2-4, pp. 3–14, 2004.