CESNET Technical Report 6/2012

COMET—COMBO Ethernet Tester

Tomáš Čejka, Pavel Benáček, Štěpán Friedl, Radek Krejčí

Received 18. 12. 2012

Abstract

This technical report describes the implementation of an Ethernet tester called COMET. The COMET is a testing and measuring system based on the COMBOv2 programmable card family. The programmable hardware allows the user to change the functionality of the device. We describe an implementation of the system at speed of 10 Gbps, moreover we discuss necessary changes to migrate to 40/100GE. COMET is able to generate test traffic, analyse incoming traffic and collect some statistical information. COMET can be used via command line tools, simple web graphical interface or via the NETCONF protocol.

Keywords: COMET, Ethernet tester, COMBOv2, traffic analyzer, traffic generator, 10GE, 40GE, 100GE

1 Introduction

In current days, networks are still getting faster. The spread of mobile technologies and voice/video services (e.g. Video-on-demand) causes that users rely on high speed of their connection. Therefore, Internet Service Providers (ISP) and network operators need to test and diagnose their high-speed network connection to ensure quality of services. In case of failure, they need to react flexibly and promptly.

In order to help network administrators and operators to diagnose their network links, we designed and implemented the COMET system. The name COMET stands for COMBO Ethernet Tester. The tester is intended for testing 10GE and it is ready to be used for faster challenging technologies like 40GE and 100GE. These new interfaces are described in IEEE 802.3ba standard [7], which was ratified in June 2010. In November 2010, CESNET prepared a study of 40/100GE card implementation [4].

The idea and realization of a new Ethernet tester came from the department of Tools for Monitoring and Configuration at the CESNET, z.s.p.o. COMET is one of projects using programmable hardware for hardware acceleration of network applications.

Nowadays, commonly used testers are very expansive and they are not flexible enough. The currently used Ethernet testers, for example [5], are single-purpose devices. When an Ethernet tester is required, administrator has to transport it and connect it into the network infrastructure. Usage of COMBOv2 card allows us to avoid physical transportation of the tester and affecting the infrastructure. The tester could be set up from a common PC with COMBOv2 card that is already in the network. For example, we can create the COMET device from a monitoring probe just by switching card firmware.

COMET is focused on testing of physical (L1) and data (L2) layers. Typical use cases for COMET include:

- generator of PRBS pattern for Eye pattern test of the physical link,
- problem analysis of the physical line bit error rate tests (this is very important for 40/100GE networks because of more physical lines),
- multimedia traffic tests – packet generator with precise time based sending,
- stress tests – testing of maximal device throughput.

The structure of the text is organized as follows. After the Introduction, a brief list of COMET features is depicted in Section 2. The system architecture and implementation details are described in Section 3. The details of software and hardware parts of the COMET system can be found in Section 4 and Section 5 The checking of the functionality of the system is described in Section 6. Section 7 discusses future improvements of COMET to support 40/100GE. The technical report is concluded in Section 8.

2 Features of COMET

The following list contains main features of COMET. The listed features are grouped by Ethernet layers (PMD, PCS, PMA) and other COMET functionalities (network traffic generator and analyzer).

- Optical modules (PMD):
  - information about signal strength, temperature and voltage of the module,
  - loopback enable/disable,
  - software TX enable/disable.
- Physical Coding Sublayer (PCS):
  - loopback enable/disable,
  - PCS Fault(s) indicator,
  - BER counter and indicator,
  - PRBS31 Test pattern generator and analyzer,
  - user pattern test,
  - RX block descrambler switch,
  - TX block scrambler switch.
- Physical Media Access (PMA):
  - PMA Fault(s) indicator,
  - loopback enable/disable switch,
  - TX data inversion switch,
  - TX MSB inversion switch.
- Packet generator:
  - packets reconstruction from PCAP file,
  - packets transmission,
  - packets replication,
  - precise time based transmission,
  - packet extension with special protocol header.
- Packet analyzer:
  - basic L2 statistics (number of received packets, min/max packet size, transferred bytes),

- packet order analysis and number of lost packets,
- histograms of packet sizes, inter-packet gaps, one-way delay of the packet with nanosecond accuracy.

3 Architecture

The COMET system enhances and extends the Ethernet testing functionality of the COMBOv2 cards. The COMBOv2 cards contain L1 monitoring and testing capabilities supplied by on-board phyter chips accessible via MDIO and I^2C buses. These capabilities are normally hidden to user and COMET makes them accessible. Moreover, COMET adds traffic generator and analyzer which are implemented in the FPGA and controlled by software tools.

The communication between hardware and software is done using Linux kernel drivers and the *libcombo* library. The *libcombo* provides access to MI32, MDIO and I^2C bus controllers implemented in the FPGA. COMET libraries (*libcomet* and *libcomregs*) use and extend the functionality of the *libcombo*. The whole system architecture is shown in Figure 1.

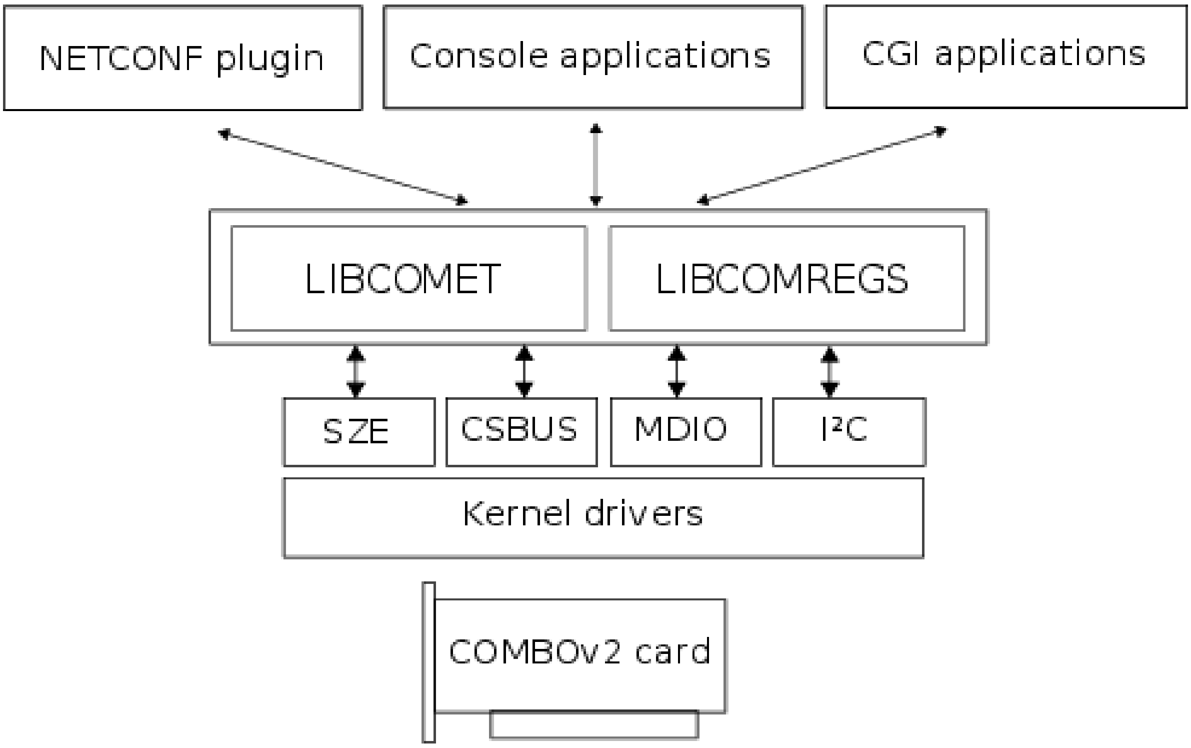


Figure 1. Architecture scheme of the COMET system, including hardware and software layers.

4 Hardware

As the first step we developed 10GE COMET version to get practical implementation experiences. Figure 2 shows block schematics of the COMET as implemented on the COMBO-LXT card with the COMBOI-10G2 interface. The system consists of these main physical blocks: PCIe connector, FPGA chip, phyter chip (PHYTER) and XFP optical transceiver. Two main logical blocks, referred as "firmware", are implemented in the FPGA: *packet generator* and *packet analyzer*. This report will focus on the generator and receiver implementation; even more details can be found in [1].

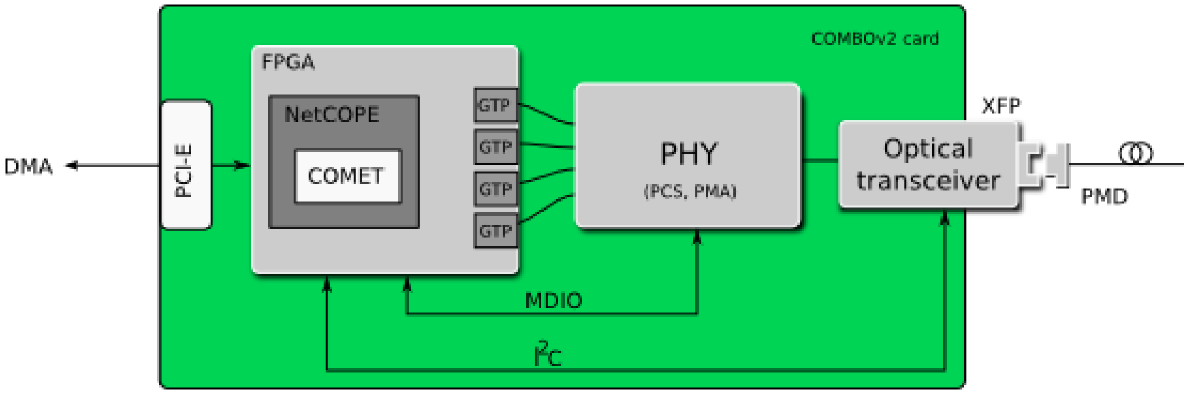


Figure 2. Block schematic of the COMET implemented on the COMBO-LXT card with the 10GE interface

4.1 Packet generator

The generator can fully saturate 10GE network link by sending data stored in a PCAP file. The data are sent to the interface of the network card via a DMA channel. However, the throughput of the DMA is not sufficient to fully saturate the 10GE link. Therefore, maximal transfer rate is achieved by a frame repeating. The information about repetition count for each frame is specified by a control software.

For each packet, the transmission time can be specified by a precise timestamp. Two delay modes are supported – *absolute* and *relative*. In absolute mode, the provided timestamp is compared with timestamp from the hardware Time Stamp Unit (TSU). This unit holds real time clock which can be synchronized to GPS [8]. A packet is transmitted when the timestamp is less or equal to the current timestamp received from the TSU. Relative delay mode is useful for creating a constant gap between packets. The delay is counted from the end of previous data packet.

For purposes of link delay and packet loss testing, special protocol header can be inserted into each outgoing packet. The header is used for later analysis in the receiver. Header contains following fields:

TS

contains dispatching timestamp,

ID

is packet identification number for input order analysis,

HN
: constant identifier of the protocol, default value is 0xC0,

HASH
: carries CRC32 checksum of previous fields. It is used for header validation at receiver.

4.2 Packet analyzer

The receiver deals with the analysis of incoming network traffic. It collects statistical data such as number of received packets, discarded and erroneous packets, packet transport delays and packet ordering status. According to the collected data, COMET provides several statistical summaries including histograms of packet sizes, packet delays and inter-packet gaps. Behaviour of these statistical blocks is controlled by the software.

The STAT_UNIT module implements a snapshot function useful for atomic reading of base statistic counters like number of received bytes, minimal/maximal packet size, minimal/maximal inter-packet gap and others. The snapshot of all these counters is created during the first read operation and released after the last read operation. Furthermore, counters and registers could be cleared by the read operation.

4.3 Firmware implementation

The COMET firmware is composed of blocks that supply functions described in Section 4.1 and Section 4.2. These blocks fit into modified NetCOPE framework for COMBOv2 cards [6]. Due to the block architecture and the standard NetCOPE communication protocols being used, the COMET system is extensible and easily configurable. The architecture is shown in Figure 3.

COMET utilizes two communication protocols for data exchange between firmware blocks: FrameLink (FL) and MI32. The FrameLink protocol is used for transferring packet data by the DMA engines. The MI32 protocol is used for components configuration and statistics collection.

Packet data, supplemented with the configuration for packet repeating and delaying units, is transferred by the DMA engine from the host memory into the internal FIFO front. After that, the packet is loaded into block named REPEATER, which performs packet cloning. Then the packet crosses the DMA and network clock domains via the asynchronous FIFO (ASFIFO). The TIME_SENDER_CORE module is responsible for packet delaying, while the TRIMMER module removes the configuration data. Finally, the packet is expanded with the special protocol header in the PACKET_CHECKER_SEND module and forwarded to the output interface (NETWORK_MODULE).

On the receiving side, the network statistics are created in the NETWORK MODULE and transferred to the STAT_UNIT for later processing. The special protocol header is analyzed in the PACKET_CHECKER_ANALYZER. The last module in chain is FRAME_DISCARD, which is useful for packet dropping in situation, when it is not necessary to forward the incoming packets to the host memory. In such case, the analysis are not affected by the receiving DMA channel and the whole 10 Gbps traffic can be processed by the STAT_UNIT and PACKET_CHECKER_ANALYZER without packet loss.

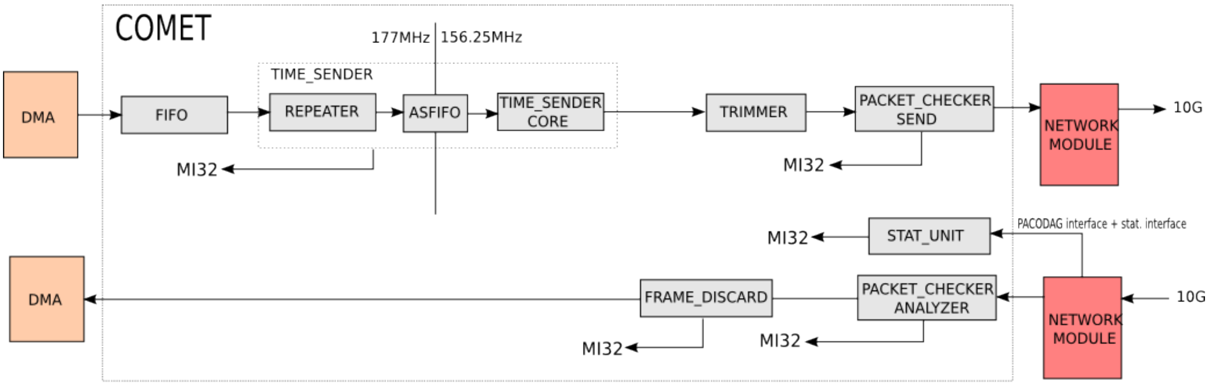


Figure 3. 10GE packet generator/analyzer architecture (only one interface is shown)

5 Software

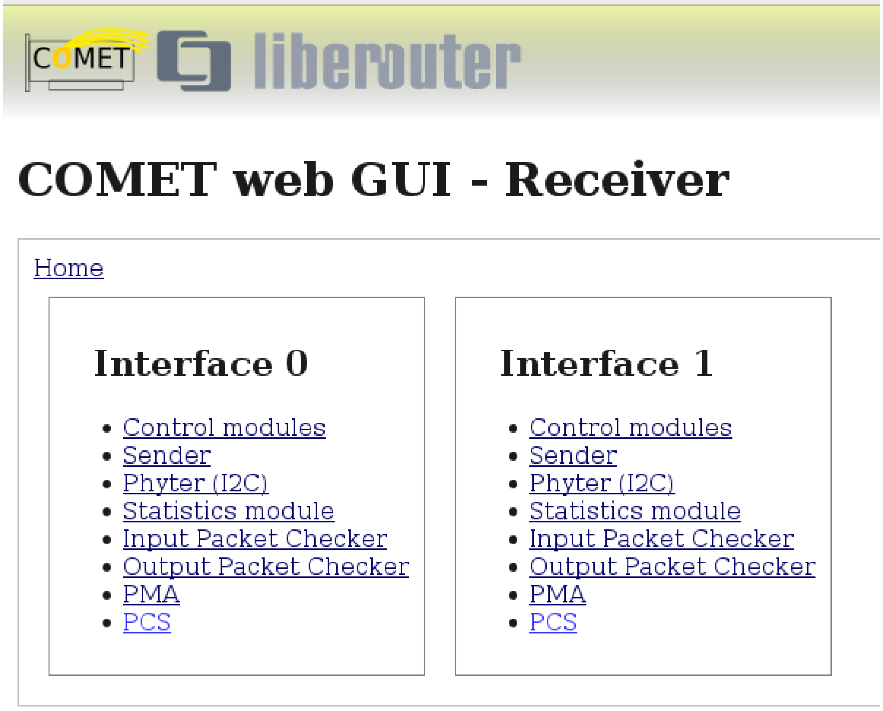
The COMET software contains two main utilities – *pcap2sze(1)* and *comstat(1)*. These console applications use *libcomet* and *libcomregs* libraries. The software part of COMET is described in more details in [2].

The main feature of the libcomet library is a control of PCAP file retransmission. The retransmission follows precise timestamps read from the PCAP file. However, the library is able to modify timestamps according to specified interval to set the average speed of transmission. To saturate the network link, the library instructs COMET to clone single Ethernet frame. *pcap2sze(1)* provides libcomet's functionality via command line interface to the user.

The libcomregs library is an abstract layer for reading/writing data from/into the COMBOv2 card. The library supports MI32, MDIO and I^2C communication interfaces. The COMET project uses libcomregs to retrieve statistical data from the COMBOv2 card. The library reads the list of requested *items* from the configuration file in XML format. The configuration file contains information about the way how the *item* can be read. For *item* that is writeble, we can also change its value. The writeable items are used to control hardware. Therefore, this library can be used in other projects that need to read a set of data from the COMBOv2 card. The *comstat(1)* console application can be used to perform the functionalities of libcomregs.

The COMET software includes a simple graphical user interface (GUI) in the form of simple web pages. It is based on CGI application for communication with hardware. GUI is made of PHP scripts providing user interface. Therefore, COMET can be easily controlled using a standard web browser. GUI covers full set of operations provided by command line tools.

Figure 4 shows *Receiver* part of the COMET GUI. Receiver depicts measured data and state information. The Receiver section can be also used for device control via writable *items*. The page is divided according to network interfaces of the COMBOv2 card. Network interface consists of sections (Control modules, Sender, ...) shown in the figure. Sections are automaticaly generated from the configuration file.



COMET web GUI - Receiver

Interface 0

- Control modules
- Sender
- Phyter (I2C)
- Statistics module
- Input Packet Checker
- Output Packet Checker
- PMA
- PCS

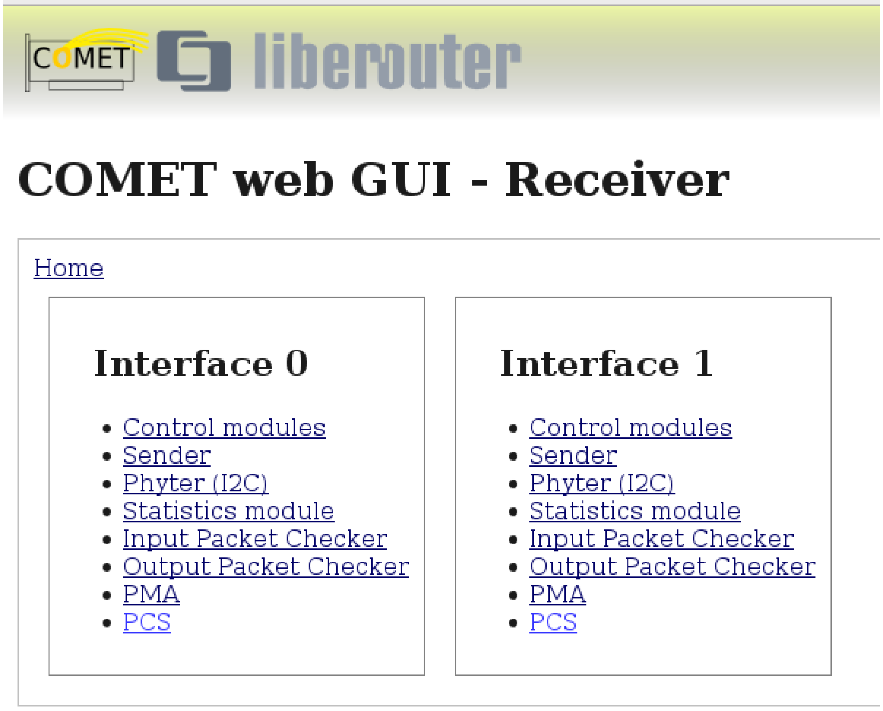
Interface 1

- Control modules
- Sender
- Phyter (I2C)
- Statistics module
- Input Packet Checker
- Output Packet Checker
- PMA
- PCS

Figure 4. Receiver – This part of GUI allows user to read state information and statistical data. It is also possible to control the system by modification of writable values.

Figure 5 shows the user interface for *Sender*. It simplifies the usage of *pcap2sze(1)* allowing user to retransmit data from stored PCAP file. There are three possible modes of network traffic transmission: full speed, user defined speed and interval based. According to the selected transmission mode, only specific settings for each mode are shown. Finally, the transmission can be started using the *Start* button.

The COMET system supports the NETCONF protocol [9]. It is done by *netcomet* – the implementation of ”Netopeer device config module” (Netopeer plug-in). Netcomet is represented by dynamically loadable shared object file *netopeer-cfgcomet.so* that is loaded by the Netopeer server [3]. Netcomet allows users to retrieve data of the COMET device and to manage its configuration via the NETCONF protocol.



COMET web GUI - Receiver

Home

Interface 0

- Control modules
- Sender
- Phyter (I2C)
- Statistics module
- Input Packet Checker
- Output Packet Checker
- PMA
- PCS

Interface 1

- Control modules
- Sender
- Phyter (I2C)
- Statistics module
- Input Packet Checker
- Output Packet Checker
- PMA
- PCS

Figure 5. Sender – This part of GUI simplifies the settings of the pcap2sze(1) utility. It allows user to select one of the transmission modes.

6 Evaluation

The COMET firmware is implemented in VHDL as a NetCOPE plugin. Synthesis was done using Xilinx ISE 13.1. Required FPGA resources of 10GE COMET are shown in Table 1. The COMBO-LXT card is equipped with Virtex-5 LX155T with speed grade -2. Virtex-6 is planned to be used on the next COMBO card generation, which will support 40/100GE interface. Frequency f_{gen} is the estimated value for the generator and the f_{rec} is the estimated frequency for the analyzer of the implemented design.

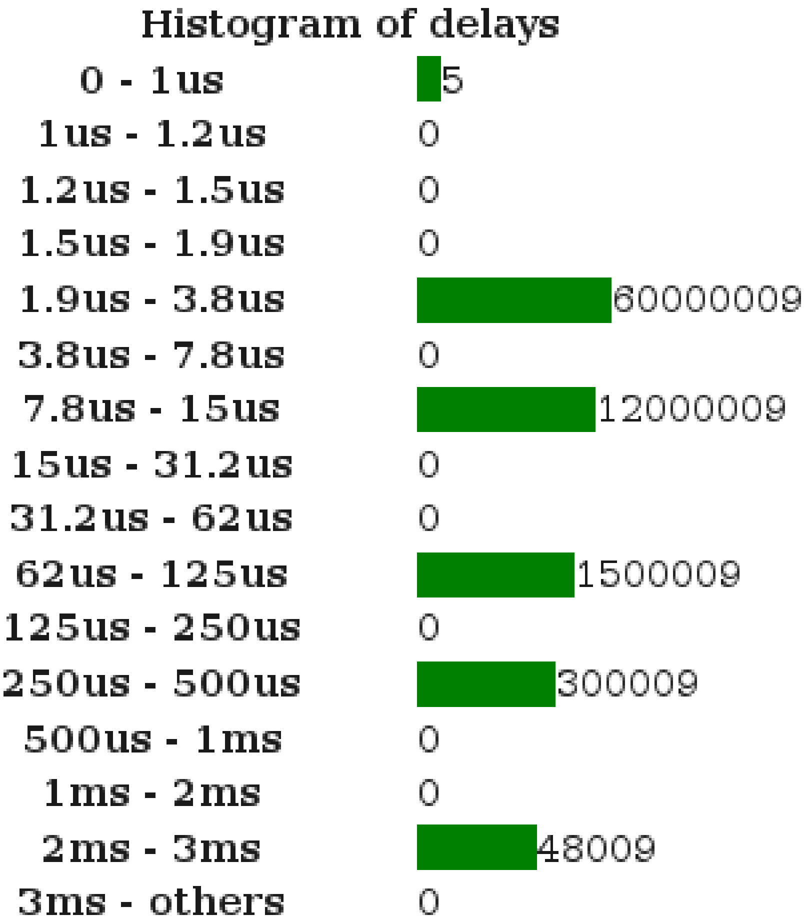
Table 1. The amount of resources required for COMET (tool XST 13.1;chip speed grade -2; f_{gen} is the estimated frequency of generator; f_{rec} is the estimated frequency of analyzer)

Chip	**Virtex-5 LX155T**	**Virtex-6 VHX565T**
LUT	18517	17512
Registers	13286	13257
BlockRAM	40	42
f_{gen}	240.409 MHz	263.145 MHz
f_{rec}	201.401 MHz	231.024 MHz

The tests of the 10GE COMET system can be divided into two groups. For the tests from the first group we used the Spirent AX/4000 Ethernet tester. These tests were based on transmission of generated traffic from/into the Spirent tester. In the second group of tests two COMET testers were involved. The first tester was used to generate the traffic according to the precise timestamps. On the second tester the link delay was analyzed. During the tests of the second group, we were also able to test the incoming packet order and packet loss.

We successfully verified all of implemented functions, however we will describe only two following testing scenarios as an example: the packet delays measurement and output transmission speed.

The packet delays measurement was performed by generating traffic with given precise inter-packet delays. Figure 6 shows the measured distribution of packet delays. The COMET analyzer did not loose any packet (the total number of received packets is equal to the number of sent packets). We can see that 5 packets were received with delay less than 1 microsecond. This is caused by immediate transmission of the first packet. We measured independently 5 different delays.



Histogram of delays

0 - 1us	5
1us - 1.2us	0
1.2us - 1.5us	0
1.5us - 1.9us	0
1.9us - 3.8us	60000009
3.8us - 7.8us	0
7.8us - 15us	12000009
15us - 31.2us	0
31.2us - 62us	0
62us - 125us	1500009
125us - 250us	0
250us - 500us	300009
500us - 1ms	0
1ms - 2ms	0
2ms - 3ms	48009
3ms - others	0

Figure 6. The example of histogram from COMET GUI. The figure shows results of the test of Packet delay measurement using COMET.

As another test, we checked the transmission speed of the COMET system. Figure 7 shows generated number of packets with respect to packet size. The number of generated packets was high enough to saturate the network link at wire-speed. The transmission was done using the *pcap2sze(1)* utility with prepared PCAP files. Each PCAP file contains one packet of the specific size that was repeated by COMET. We used testing packets of the following sizes: 64, 128, 256, 512, 1024, 1280 and 1514 bytes. All packets were successfully sent and received without errors.

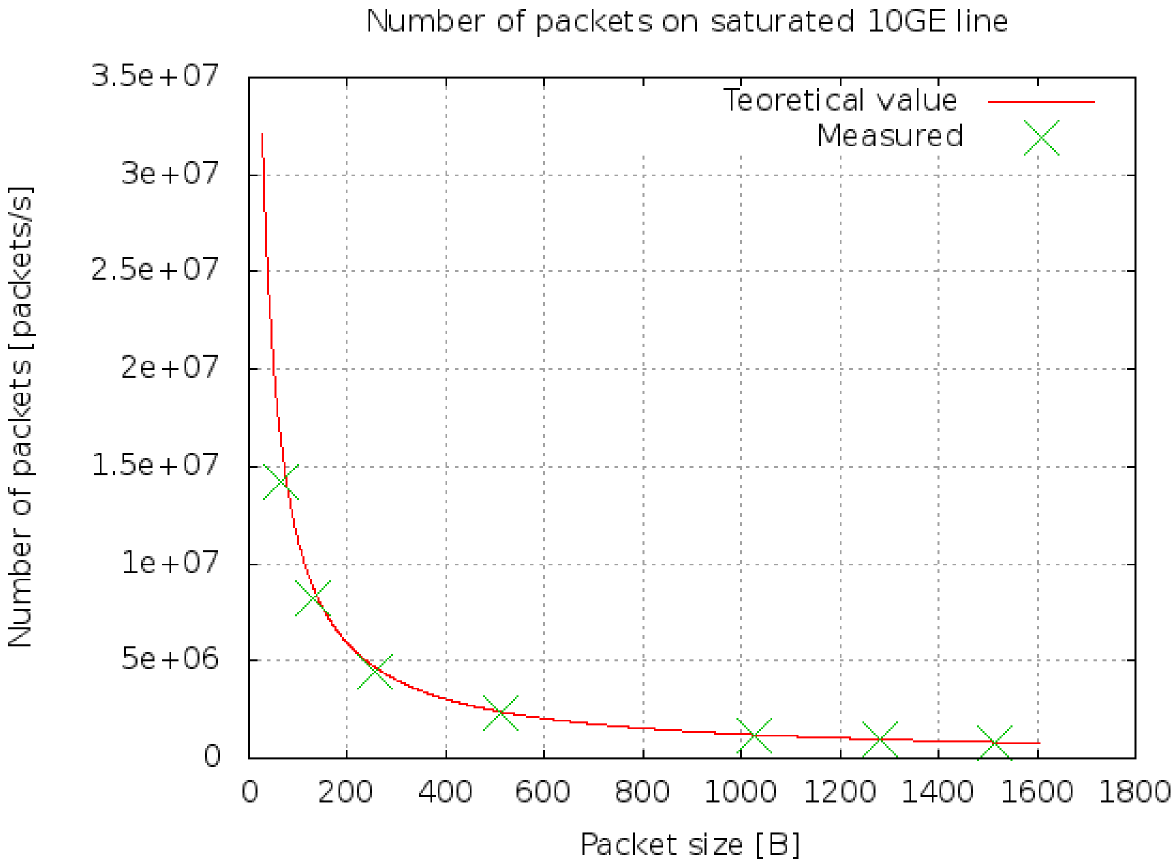


Figure 7. This graph shows theoretical and measured output packet rate; Packets were sent using COMET and received on the AX/4000 Spirent tester.

7 Modifications for 40/100GE

A design implementation for 40/100GE network links is very challenging and it is still in progress. First of all, the design must adapt to the new network blocks IBUF and OBUF. IBUF stands for input network buffer, it is responsible for packet receiving from the network. OBUF stands for output network buffer and it is used for packet transmission out of the NetCOPE to the network.

To ensure the precise timing of packet output, we developed a special version of the 40GE OBUF, which supports timestamp-driven dispatch and packet extension with the special protocol header, described in the previous chapter. We also wanted to make the dispatching timestamp stored in the header more accurate, therefore we moved the header insertion from the position before OBUF to the position after

OBUF. However, the detailed description of the OBUF is out of the scope of this technical report.

For data transfers inside the system, the FrameLinkUnaligned (FLU) is used instead of FrameLink(FL), which was used in 10GE version of the COMET. The main difference between FL and FLU is that FLU supports unaligned start of the next packet. Therefore, one FLU word can contain the end of the previous packet and the start of the next packet. This situation sometimes leads to very complex design implementation. For example, if we want to insert data into the FLU flow, sometimes we have to reorder words in the packet flow. This operation consumes too much FPGA resources and it also degrades the working frequency.

On the other hand, the statistical units in the 10GE COMET were implemented with respect to 40/100GE. Therefore, we do not have to reimplement them and they should serve well also for this version.

8 Conclusion

This technical report described the implementation of COMET. COMET is the system that consists of the software part and the hardware design. The project is built on the COMBOv2 card using the modified NetCOPE platform.

COMET makes accessible L1 testing and allows a wire-speed output generation and analysis of incoming traffic on 10GE link. The traffic generation is based on packet retransmission in defined periods of time. Both functions (traffic generation and analysis) are hardware accelerated in FPGA. The system can be used from the command line via console applications or using simple web graphical user interface. COMET also supports configuration via NETCONF protocol. Therefore, we can manage the system using any NETCONF client.

Since existing network testers are not flexible enough, we proposed a new possible way which is based on flexibility of COMBOv2 cards. This approach allows us to load the functionality of COMET on every working server with the COMBOv2 card. .

The implemented COMET software will work at higher speeds than 10GE without changes. We have also implemented some hardware blocks for wire-speed on 40GE and 100GE.

References

[1] Benáček, P. *Ethernetový tester pro vysokorychlostní sítě* Diplomová práce. ČVUT v Praze, Fakulta informacních technologií, Praha, Czech Republic, 2012.

[2] Čejka, T. *Control Software for the COMbo Ethernet Tester and its integration into the Netopeer configuration system: Master's thesis* Czech Technical University in Prague, Faculty of Information Technology, Prague, Czech Republic, 2012.

[3] Cesnet TMC group. *Netopeer | Liberouter* 2012. Available at WWW: https://www.liberouter.org/?page_id=827

[4] Štěpán F., Novotný J., Lhotka L. *Study of 40/100GE card implementation* CESNET technical report 22/2010

[5] Spirent Communications *Solutions Directory* 2012. Available at WWW: http://www.spirent.com/Solutions-Directory

[6] Liberouter group *Programovatelný hardware* 2012. Available at WWW: http://www.cesnet.cz/doc/2007/zprava/proghw.html

[7] The Institute of Electrical and Electronics Engineers, Inc. 3 Park Avenue,New York, NY 10016-5997, USA *Part 3:Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications* 2002, IEEE Std 802.3ae™ -2010 , ISBN 0-7381-3288-8.

[8] Martínek T., Žádník M. *Precise Timestamp Generation Module and its Applications in Flow Monitoring* CESNET technical report 13/2009. Available online[1]

[9] Enns R., Bjorklund M., Schoenwalder J., Bierman A. *Network Configuration Protocol (NETCONF)* RFC 6241 (Proposed Standard), Internet Engineering Task Force, 2011. Available at WWW: http://tools.ietf.org/html/rfc6241

[1] http://www.cesnet.cz/doc/techzpravy/2009/timestamp-module-flowmon/