# Hardware Acceleration for Measurements in 100 Gb/s Networks

Viktor Puš

CESNET, association of legal entities,
Zikova 4, 160 00 Prague, Czech Republic
pus@cesnet.cz

**Abstract.** The paper presents the NetCOPE experimental platform for evaluation of 100 Gb/s networks. The platform consists of acceleration boards with FPGAs, FPGA firmware and control software. While the NetCOPE for 10 Gb/s networks is a mature platform, multiplying the throughput ten times brings major challenges in the platform design.

**Keywords:** FPGA, 100 Gb/s, hardware acceleration, Ethernet, PCI Express

## 1 Introduction

With the growing amount of Internet traffic, it is inevitable that current 10 Gb/s lines, often used in data centers and backbones, will be eventually replaced by faster technologies. Standard for 100 Gb/s Ethernet was proposed as IEEE standard 802.3ba in 2008 and ratified in June 2010. While it is still waiting for wider deployment, preparing and evaluating the tools and algorithms for this technology is a major research topic.

While theoretical works in this area certainly have significant impact, we argue that experimental research will also bring many valuable results. We aim to create and evaluate general measurement platform for 100 Gb/s networks, which can not be examined by means of formal description and analysis. Our work falls into the category of experimental computer science [4]. We expect the first application to employ a limited set of operations, but the platform allows to create wide selection of complex applications. The NetCOPE platform is a team work of several researchers, author's contribution is the overall design of the platform firmware and the design of several firmware modules, such as the packet header parser HFE-M and the packet filter with TCAM.

The following chapters introduce the hardware, firmware and software design of the experimental platform together with several selected challenges in the 100 Gb/s application design.

## 2 Platform Design

Our experimental platform employs Virtex-6 FPGA [2] to receive, process and transmit the 100 Gb/s traffic. FPGA is placed on the COMBO family acceleration boards [1]. Daughter card communicates with the mother card via several high-speed channels and is equipped with the 100 Gb/s CXP connector. The mother card is connected to the host PC via the PCI-Express Gen2 x8 slot. The Linux operating system running on the host PC is supplemented by the device drivers to support configuration accesses as well as high-speed DMA transfers. Structure of the platform is shown in Fig. 1.

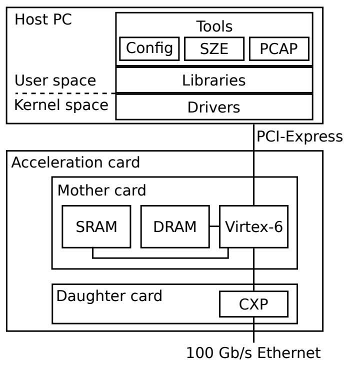

**Fig. 1.** Structure of the experimental platform.

This arrangement requires smart partitioning of the platform functions between firmware and software. Fig. 2 shows the basic arrangement of firmware modules in the FPGA. Mother card is responsible for packet reception, basic processing (compute frame checksum, check MTU, etc.) and eventually packet transmission. Valid packets are passed to the following modules, where additional processing, such as packet parsing and filtering is done. Filtered packets are then sent to the host RAM by the DMA controller via the PCI Express interface. Additional modules can be connected as plug-ins into the system. The software part of the platform can perform various functions such as routing, traffic storage, monitoring, etc.

## 3 Challenges

While the platform design is straightforward and logical, there are several design considerations that must be engineered carefully. This chapter presents some of the most interesting research topics.

### 3.1 On-chip Communication

The 100 Gb/s Ethernet is capable of packet transmission at the rate of 150 million packets per second (6.6 ns/packet) for the shortest 64 B packets. Due to the minimal 20 B inter-frame gap, more data is effectively transmitted with long packets. The example of data bus capable of transmitting the full 100 Gb/s data stream is 512 bits wide and runs at 196 MHz.

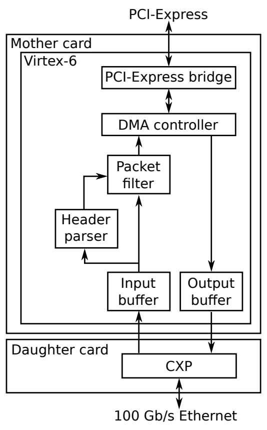

**Fig. 2.** Firmware structure.

It is important to use the bus bandwidth effectively. For example, if the first byte of the packet is always aligned to the lowest byte of the data bus, then 65 B packet occupies two clock cycles on the bus, limiting its throughput to 50.78 %. However, if the packet is allowed to start at eight different positions of the data word (every eight bytes), then the second data word of 65 B packet can already carry 56 bytes of the *next* packet and only seven bytes are unused. Protocol effectivity is 90.27 % in the worst case.

### 3.2 Frame Checksum

The Ethernet requires each frame to be followed by the 32-bit CRC frame checksum (FCS) to detect transmission errors. FCS is checked when frame is received and appended when frame is sent. FCS computing is complicated by the fact that the packet may start at different positions of the data word and can also end at any position in the data word. Adding leading or trailing zeros is not acceptable, as it would affect the FCS value.

We resolve the unaligned frame starts by using carefully selected CRC initial vector. Our solution for the unaligned frame ends is similar to [5]: unaligned last word of the frame is computed in the tree hierarchy of smaller CRC modules.

### 3.3 Header Parsing

High-speed header parsing is another interesting research topic. Key issue is that the protocol fields appear at various offsets in the packet. This is due to shim protocols (VLAN, MPLS) and variable header lengths (IPv4/IPv6 option headers).

Our Modular Header Field Extractor (HFE-M) employs pipelined architecture to determine protocol headers offset. Each protocol parsing module has the protocol start offset as the input and the next protocol type and start offset as the output. These messages are passed among protocol parsing modules which are connected in the expected order of protocol headers. Once the types and offsets of present protocols are known, extracting the fields of interest from the packet is relatively simply done by multiplexers.

### 3.4 Packet Filtering

The PCI-Express bus supported by the card has the theoretical throughput 32 Gb/s. Therefore most applications require hardware preprocessing of incoming packets. One suitable operation may be packet shortening, which is useful for applications processing only packet headers. Other possibility is to filter out uninteresting traffic and pass only the selected packets into the host RAM.

Our first version of packet filtering engine uses the TCAM to classify packets. Our implementation of TCAM in FPGA is inspired by [3]. It uses Look-Up Tables to match the input to the stored value. 6-input LUT matches 6 bits in 1 cycle or 10 bits in two different TCAM rows in two cycles if one bit is used to select the row. For example, TCAM storing 128 IPv6 addresses occupies 2816 LUTs if new matching must be done in each clock cycle, or 1664 LUTs if the matching may take two clock cycles.

## 4 Conclusion

Our experimental work is targeted at post–10 Gb/s network traffic processing. We aim to create universal extensible platform for experiments with high-speed network algorithms and technologies.

The platform is currently being intensively developed, using the experience from our previous 1 Gb/s and 10 Gb/s experimental platforms.

**Acknowledgments** This work is supported by the "CESNET Large Infrastructure" project no. LM2010005 funded by the Ministry of Education, Youth and Sports of the Czech Republic.

## References

1. COMBO-LXT. CESNET, http://www.liberouter.org/card_combolxt.php.
2. Xilinx Virtex–6 FPGA Family. Xilinx, Inc., http://www.xilinx.com/products/silicon-devices/fpga/virtex-6/index.htm.
3. Jean-Louis Brelet. An overview of multiple cam designs in virtex family devices. Xilinx Applixation Note 201, Xilinx Inc., 1999.
4. National Research Council Committee on Academic Careers for Experimental Computer Scientists. *Academic Careers for Experimental Computer Scientists and Engineers.* The National Academies Press, 1994.
5. M. Walma. Pipelined cyclic redundancy check (crc) calculation. In *Computer Communications and Networks, 2007. ICCCN 2007. Proceedings of 16th International Conference on*, pages 365 –370, aug. 2007.