Live Demonstration of Application Layer Traffic Monitoring at 100 Gbps

Viktor Puš, Lukáš Kekely, Jan Kučera
CESNET, a. l. e.
Zikova 4, 160 00 Prague, Czech Republic
Email: pus,kekely,jan.kucera@cesnet.cz

Denis Matoušek
INVEA-TECH a.s.
U Vodárny 2965/2, 616 00 Brno, Czech Republic
Email: matousek@invea.com

***Abstract*—The increasing speed of network links, together with great complexity of application protocol processing, require a new way of high-speed and precise network monitoring. To tackle this challenge, we have designed a brand new concept of hardware acceleration for flexible flow based traffic monitoring which we call Software Defined Monitoring (SDM). Application layer processing is performed by monitoring tasks implemented in software in conjunction with a configurable hardware accelerator. The hardware accelerator is a high-speed application-specific processor tailored to stateful flow processing. The software monitoring tasks control the level of detail retained by the hardware for packets of each flow.**

The live demonstration will show a real-time analysis of HTTP traffic at 100 Gbps using SDM implementation on our unique COMBO-100G card. COMBO-100G is the world's first PCIe card with 100G Ethernet interface and wire-speed packet capture functionality. The influence of SDM on CPU load and packet drop rate will be presented in real-time graphs together with the results of HTTP analysis. The demo will clearly present evident benefits of SDM usage for application layer processing.

I. Introduction

Modern network engineering and security heavily rely on network traffic monitoring. Requirements imposed on the quality of network security monitoring information often lead to the requirement to process unsampled network traffic. That ability is crucial in order to detect even single-packet attacks. Network monitoring nowadays is usually realized as collecting only the basic statistics about network flows at the Internet and Transport layers and reporting them to a central storage collector using a handover protocol such as NetFlow [1] or IPFIX [2].

The ongoing new trend in the field of network traffic monitoring is towards creation of richer flow records, carrying some extra information in addition to the basic flow size and timing statistics. The added information often include values from the application layer protocol headers, such as HTTP, DNS etc. Therefore, it seems that the ability to analyze application layer in the monitoring process is crucial for improvement of the quality of network threat detection. We argue that the flexibility of software programming (compared to a fixed functionality of hardware) is necessary for practical and extensible application layer processing.

However, with the ongoing process of shifting as much network functionality as possible towards software processing, the hardware acceleration of dataplane's time critical operations is often neglected. This is highly disproportional to recent increases in network speeds, which easily surpass any improvements in the CPU performance. In particular, devices with 100 Gbps Ethernet interfaces have become available in recent years, but practical traffic processing at such speed is often limited to rather simple packet switching and routing. Other important applications, such as network security monitoring, DDoS protection or application-layer processing, are extremely expensive or unavailable at all.

While the original intention of the Software Defined Networking paradigm is to improve network manageability, extensibility and flexibility, we have proposed an elegant way to use its ideas in conjunction with hardware acceleration. Our demonstration provides an example of an application that embraces the Software Defined paradigm, yet utilizes hardware acceleration to achieve very high throughput and is therefore suitable for 100 Gbps networks.

Our Software Defined Monitoring system [3] offloads the processing of bulk traffic that is not (or no longer) interesting for the application layer processing tasks into the hardware accelerator. The offload is controlled on a per flow basis by the monitoring software and adjusted in real time to its current needs. Offload control is done by dynamically specified set of rules. These rules are installed into the accelerator to determine the type of packet offload (preprocessing acceleration) used for individual network flows. The preprocessing method that best aids the performance and does not decrease the required precision of advanced software processing is selected.

II. Demo Description

We have implemented the SDM system using the COMBO-100G hardware accelerator (see Fig. 1) [4], [5]. It uses Xilinx Virtex-7 FPGA for Ethernet and PCIe communication, and is capable of lossless 100 Gbps packet capture. The card is connected to a commodity server PC with an Intel Xeon CPU.

Due to the programmable nature of FPGAs, various traffic processing functions can be implemented according to the application requirements. In the case of SDM, the FPGA firmware receives packets from 100G Ethernet line, parses them, finds the appropriate processing rule, and then performs an action based on that rule. The list of possible actions include passing full or shortened packet to host RAM, passing the parsed packet header in a fixed format to host RAM, or updating an associated flow record directly in the on card memory, without immediate communication with the host PC. To utilize multi-core CPUs, the firmware uploads data into

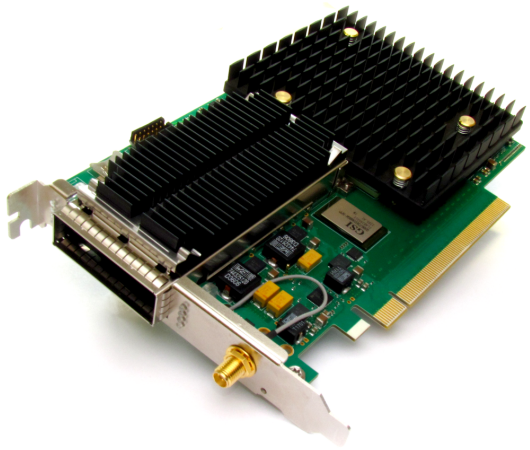

Fig. 1. COMBO-100G hardware accelerator

eight independent RAM buffers, while maintaining consistent mapping of network flows to the buffers.

To demonstrate SDM acceleration capabilities, we chose to present standard NetFlow statistics measurement enriched by HTTP protocol analysis at input data rate close to full 100 Gbps. Histograms of accessed URI domains, used HTTP methods and returned status codes computed from the network traffic are shown live as real-time graphs in GUI to demonstrate the functionality of the application layer monitoring (Fig. 2). Packets from a real backbone line of Cesnet2 network are used as input data for the demo.

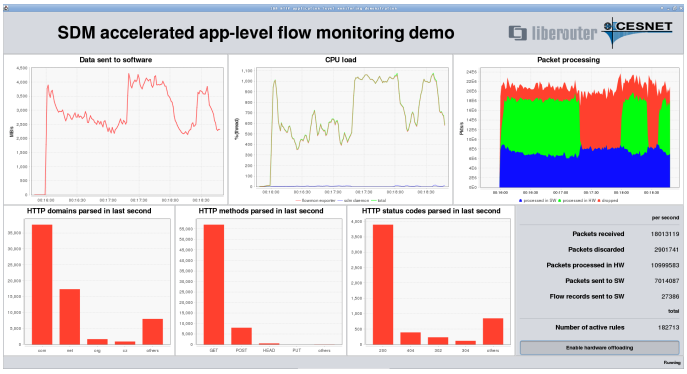

Fig. 2. Interactive GUI with live graphs presented in the demo

Software performing HTTP analysis for the purpose of the demo is realized using the Invea-Tech's Flowmon exporter software [6]. This exporter offers highly-optimized flow based traffic processing with effective multicore support. Furthermore, it performs computation of basic NetFlow statistics and supports easy modification of its functionality by user plugins to the extent required for the demonstration. For this demo, we created a plugin which extends Flowmon exporter functionality by detection of HTTP protocol in packets, extraction of interesting information from HTTP headers and also SDM offload control. This way, one can easily create an effective tool for monitoring of any application layer protocol that is usable in high-speed networks.

Application layer (especially HTTP) monitoring is very hard for general purpose CPU. Therefore, we show that the CPU is fully utilized at 100 Gbps traffic rate and more than half of all the packets are dropped due to buffer overflow, if the hardware behaves as rather ordinary NIC, only receiving packets and uploading them to eight RAM buffers.

However, when we enable the SDM functionality from the interactive GUI, the software starts to offload Internet and Transport layer flow statistics measurement of large, non-HTTP flows to the hardware accelerator. Due to the heavy-tailed distribution of flow sizes in network traffic (small amount of the largest flows convey huge amount of data), great portion of packets never reach the CPU. Only aggregate statistics (flow records) are periodically sent to the software instead of these packets, so the quality of flow measurement is not deteriorated by the SDM offload usage.

Finally, the advantage of SDM is that the offload of L3 and L4 flow measurement reduces the CPU load, leaving more time for HTTP processing. This way, packet drop rate is minimized, therefore the quality of monitoring rises. The benefit in a complete monitoring system is that even single-packet attacks are unlikely to be left unnoticed by a prospective subsequent detection algorithms or manual checking.

III. Conclusion

We demonstrate live HTTP enriched flow monitoring of network traffic at 100 Gbps using our unique SDM system. We show that by using fine-grained offload (hardware acceleration), it is possible to achieve sufficient throughput using a commodity PC and the hardware accelerator. The demonstrated system is designed with flexibility as a primary goal, so that extending it for support of another application layer protocol is only a matter of writing a C plugin following a simple API.

IV. Logistics

The demo will present real-time measurement online by remote access to the actual server with hardware accelerator. Stable Internet connection, power plug and a table are therefore minimal requirements for successful presentation. Additional requirements include a large LCD screen to improve GUI readability and a poster stand for additional material.

Acknowledgment

This research has been supported by the "CESNET Large Infrastructure" project no. LM2010005 funded by the Ministry of Education, Youth and Sports of the Czech Republic.

References

[1] B. Claise, "Cisco Systems NetFlow Services Export Version 9," RFC 3954, Internet Engineering Task Force, October 2004.

[2] B. Claise, B. Trammell, and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information," RFC 7011, Internet Engineering Task Force, Sept. 2013.

[3] L. Kekely, J. Kucera, V. Pus, J. Korenek, and A. Vasilakos, "Software defined monitoring of application protocols," *Computers, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[4] Liberouter / Cesnet TMC group, "COMBO-100G," 2014. [Online]. Available: https://www.liberouter.org/combo-100g/

[5] INVEA-TECH a.s., "COMBO-100G FPGA Card," 2014. [Online]. Available: https://www.invea.com/en/products-and-services/fpga-cards/combo-100g

[6] INVEA-TECH a.s., "FlowMon," 2014. [Online]. Available: https://www.invea.com/en/products/flowmon