Combining FPGAs and processors for high-throughput forensics

Franck Cornevaux-Juignet, Matthieu Arzel, Pierre-Henri Horrein, Tristan Groleat, Christian Person

Combining FPGAs and processors for high-throughput forensics

Franck Cornevaux-Juignet*, Matthieu Arzel*, Pierre-Henri Horrein*, Tristan Groléat† and Christian Person*

*IMT Atlantique Bretagne-Pays de la Loire campus de Brest
Technople Brest-Iroise CS 83818 29238 Brest Cedex 3 FRANCE
Email: firstname.lastname@imt-atlantique.fr

†OVH
230 rue Roland Garros, 29490 Guipavas FRANCE
Email: tristan.groleat@corp.ovh.com

***Abstract*—Data centers availability is mandatory and is conditioned by a quick response to failures and attacks thanks to efficient live forensics. However, this task is lately impossible to complete with classic systems because of encountered data rates and service diversity. Moreover, Software-Defined Networking (SDN) devices agility requirements prevent the use of Application Specific Integrated Circuits (ASIC) solutions due to long development time.**

New solutions of smart Network Interface Cards (NIC) with embedded Field Programmable Gate Arrays (FPGA) are considered, as in Microsoft Azure solution. FPGAs ensure high throughput processings without packet loss to offload CPU processing, but their configurations support only sparse firmware upgrades and shut down processings.

This paper proposes an hybrid architecture to realize agile high performance traffic forensics. This work combines hardware performance, high throughput, and software high flexibility to achieve data rates beyond 40 Gb/s while being configurable at runtime through parameters. A software API allows a user-friendly configuration without stopping processings. The implementation of a flexible packet parser, first block of the packet processing chain, demonstrates the viability of the concept.

I. Introduction

Network throughput is always increasing as well as the number of services making network monitoring more complex and difficult. In September 2016, an attack of 1 Tb/s through connected object on the web service provider OVH [1], among others, has shown the necessity of high performance forensic systems sustaining 40 Gb/s links and over, as found in data centers. In addition to cope with high data rates, probes have now to be agile in order to be compliant with the SDN paradigm. It dissociates the data plane and the control plane to set programmability on the architecture on the condition to have flexible probes.

Despite their compliance, current full software solutions are unable to process such traffic data rates [2], as the opposite of ASIC solutions. But ASICs suffer from long development time, so that agility requirements can not be met. Current solutions are more likely smart NICs using an FPGA to offload CPU processings, as seen in Microsoft Azure solution [3]. Thanks to high parallelization possibilities, FPGA devices allow to process data at high throughput without loss. The reconfiguration possibility brings flexibility to the system. Few solutions exist to facilitate access of network engineers to FPGA systems. In [4], a packet processing architecture is able to forward results to the host at 100 Gb/s. Once configured, only the rule set can be configured. Xilinx SDNet design flow [5] provides a configurable solution at 100 Gb/s. Partial dynamic reconfiguration allows to reconfigure parts of the processing chain. However, FPGA reconfiguration means new design generation with proprietary tools and shutting down the probe. Therefore, these solutions can not support the runtime flexibility needed for live forensics and lack portability.

This research is supported by Brittany region, Finistère regional council, Brest Métropole funds.

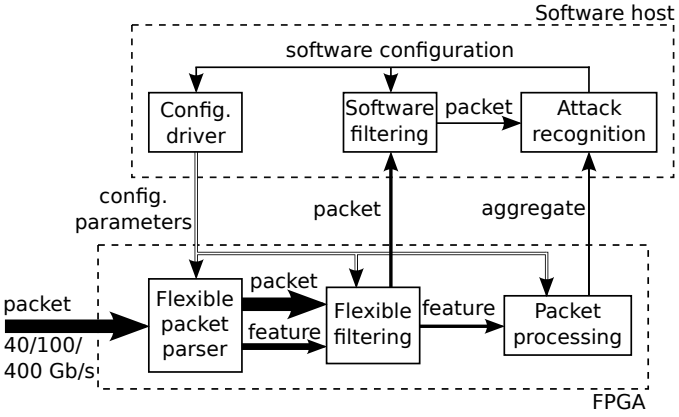

Fig. 1. Flexible probe architecture

This paper presents a network probe architecture built on top of a smart NIC to enable adaptative monitoring. This hybrid design is based on a fully configurable packet processing chain, answering agility and high performance requirements. A simple software integration allows its adoption by end users, *i.e.* network engineers.

II. Agile traffic monitoring

A. Hardware and software co-design

The main interest in using an hybrid design is to make a context-aware probe capable of processing packet traffic at line rate. This allows to run precise and agile forensic applications. A static flexible FPGA design ensures sustaining high data rates without packet loss. Configuring this design with parameters avoids the classic reconfiguration of the FPGA and allows runtime configuration. Indeed, the complementary software is based on an API setting the parameters of the FPGA design. In this way, this solution is user-friendly for network engineers, as no Hardware Description Language (HDL) knowledge is needed. Moreover, development is cost-efficient and does not depend on external proprietary tools.

B. Agile probe architecture

The figure 1 describes the proposed architecture. Front end traffic is handled by the FPGA processings in order to reduce the data bandwidth sent to the software. The CPU works then

only on data of interest extracted from the main traffic as required by the software.

1) Packet parsing: Packet parsing is a mandatory step for every packet processing device. Packets are decapsulated to extract data of interest from different headers, like specific fields or selected bytes. As every packet must be processed by this block, an hardware implementation is necessary in order to support data rates. With configuration parameters, parsed protocols are set at runtime. Thus, this packet parser can be adapted to the situation enabling agile processings. It is then easily possible to go further than the utilization of the classical network 5-tuple: IP addresses, TCP/UDP port and protocol. Any information in any header can be extracted for the benefits of forensic algorithms.

2) Packet filtering: Packet filtering is used to select packets according to defined rules. Rule processors are dynamically programmed *via* parameters by the software at runtime. Only packets of interest are extracted from the main traffic. This traffic reduction lowers the communication bandwidth between the FPGA and the CPU in a controlled way. Thus, the processor can apply complex rules without sub-sampling or being overloaded.

3) Packet processing: Hardware packet processings supply the software with precise data aggregates on the incoming traffic impossible to get otherwise to offload the core processing on the CPU. Finally, software applications have the possibility to use a maximum CPU workload to realize complex attack detection adapted to the incoming traffic. Configurable aggregates and selected packets are available to refine the detection if the software algorithm needs them.

III. Test and measurements

A. Test platform

The NetFPGA project proposes an affordable platform for smart NIC prototyping. The test board, the NetFPGA SUME, is composed of a Virtex 7 XC7VX690T FPGA, four 10 Gb/s network interfaces and a PCIe port. A reference design is given containing a packet datapath and a register configuration datapath from the host machine. This board allows an implementation of the proposed agile probe.

This test platform is used to implement the proposed probe, available as open-source [6]. Extracted features from headers are sent to configurable filters feeding counters. This design tests the validity of the proposed architecture. A simple volumetric attack detection program is built on top of extracted counter values and determines which protocol is targeted.

B. Measurements

In order to test high performance and runtime agility, the test traffic contains 4 successive volumetric attacks of 10 Gb/s integrated inside a base traffic of 30 Gb/s. This synthetic traffic is composed of worst case 64-byte packets. The figure 2 shows the evolution of the software detection application response to the incoming traffic. The number of received packets is seen by the application thanks to counter values transferred from the FPGA. It is worth noting that this number reaches 59,523 Mpps, corresponding to a full 40 Gb/s link with 64-byte packets. This application succeeds in getting information on the traffic with no packet loss. Once an anomaly in the evolution of the number of packets is detected, the program sends new parameters to the FPGA to refine the detection on the different protocols. A refinement loop sets coarse-gained to fine-grained configurations of the parser and of the filters

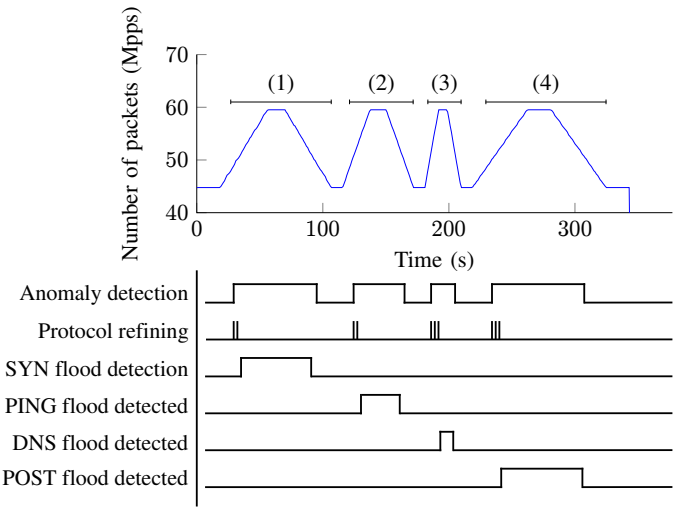

Fig. 2. Probe reaction chronograms to received traffic

in order to determine the exact protocol targeted by the attack. This is only possible because of the runtime configuration ability of the proposed probe.

IV. Conclusion and future works

The proposed architecture succeeds in using of a smart NIC architecture in order to propose an agile probe for live forensics. Configuring a flexible FPGA design through parameters allows CPU applications to benefit a feedback loop and to be content-aware of the traffic while sustaining the line rate without shutting down the probe. The implementation of the packet parser demonstrates the viability of this design. No packet loss is detected at 40 Gb/s and any information can be extracted from any protocol.

The utilization of boards with more recent FPGAs will allow to achieve data rates over 100 Gb/s while increasing the processing capability. Hardware filtering and offload processings will complete this part to produce a fully runtime-configurable forensic probe.

References

[1] OVH, "OVH Mirai attack," accessed: 2017-02-27. [Online]. Available: https://www.ovh.com/fr/a2367.goutte-ddos-n-a-pas-fait-deborder-le-vac

[2] V. Moreno, J. Ramos, P. M. S. del Río, J. L. García-Dorado, F. J. Gomez-Arribas, and J. Aracil, "Commodity packet capture engines: Tutorial, cookbook and applicability," *IEEE Communications Surveys Tutorials*, vol. 17, no. 3, pp. 1364–1390, thirdquarter 2015.

[3] D. Firestone, "Smartnic: Accelerating azures network with fpgas on ocs servers," in *OCP U.S. SUMMIT 2016*, San Jose, CA, March 9-10 2016. [Online]. Available: http://files.opencompute.org/oc/public.php?service=files&t=5803e581b55e90e51669410559b91169&download&path=//SmartNIC%20OCP%202016.pdf

[4] L. Kekely, J. Kuera, V. Pu, J. Koenek, and A. V. Vasilakos, "Software defined monitoring of application protocols," *IEEE Transactions on Computers*, vol. 65, no. 2, pp. 615–626, Feb 2016.

[5] Xilinx, "SDNet," accessed: 2017-02-23. [Online]. Available: https://www.xilinx.com/products/design-tools/software-zone/sdnet.html

[6] *Removed for review.*