FLOOD ATTACKS GENERATION

David Hudec

Bachelor Programme (3), FEEC BUT

E-mail: xhudec27@stud.feec.vutbr.cz

Supervised by: David Smékal

E-mail: xsmeka11@stud.feec.vutbr.cz

Abstract: Proposal of a high speed packet flooding device is presented in this paper. The product is based on the FPGA (Field-programmable Gate Array) platform, developed in VHDL (Very high speed integrated circuit Hardware Description Language) and set into the NetCOPE environment. It uses an existing solution of packet generator. Once done, it should be implemented on a COMBO-80G board to serve as a network stressing tool.

Keywords: FPGA, VHDL, NetCOPE, Denial of Service, DoS, network attack, network tester, packet generator

1. INTRODUCTION

For the evaluation of current high speed packet switching networks' performance, high speed testing equipment is needed. Existing solutions are not suitable enough, as software traffic generators lack on throughput and appropriate hardware products are very expensive. FPGA platform software-hardware combination, as described in this paper, is capable of delivering good performance for a fraction of the costs spent on commercial hardware-based solutions, nevertheless providing great amount of configurability in the pack.

2. BACKGROUND

Concerns described in this section are essential to comprehend for understanding the project.

2.1. DENIAL OF SERVICE

Amongst many kinds of network intrusion, Denial of Service (DoS) attacks are one of the well-known time-proven ways of attacking a network-enabled device. Its simplicity lies in flooding with traffic so heavy the target cannot withstand, being forced to focus on excessive requests from the attacker and ceases being of any use for other, legitimate users. If the target is a server, its services will not be available for customers (for the time of the attack) and a financial loss is on the way. Great amount of data is therefore necessary. Enhancing this approach is possible by populating the attacker's base and distributing the disruptive force, thus creating a Distributed DoS.

2.2. FIELD-PROGRAMMABLE GATE ARRAY

When a computable task is not as big as to cause a standalone integrated circuit being designed for it, neither it is simple enough for conventional CPU systems to handle, FPGA boards are an option. It represents a technology of integrated circuitry designed to be configured by the customer using a hardware description language, hence programmable. The architecture contains a matrix of logic blocks, and a net of reconfigurable interconnects, allowing the blocks to be knitted together in many different ways. Thus, a unique junction of hardware-oriented solution and great configurability might be achieved, keeping it low-cost, yet effective.

Phases of *simulation*, *synthesis*, *bit stream generation* and *implementation* are what needs to be

completed before a VHDL application can run on an FPGA board.

2.3. NETCOPE

The NetCOPE framework [3] has been created to ease the development of network applications based on the FPGA platform. The firmware provided takes place between the user program core and input/output interfaces of the board used itself, making network communication programming much simpler and more transparent. As it is a living project, though, new versions of NetCOPE are being made, not necessarily keeping the backward compatibility.

2.4. HARDWARE

One COMBO-80G FPGA card is available to use, representing the powerful hardware needed. It contains eight identical RJ45 network interfaces supporting 10 Gb Ethernet technology each. The card can be mounted on a hosting computer via PCI-e bus, enabling sustainable high speed transfers between these two. Virtex-7 processing unit and 8 GB of RAM are installed.

2.5. FLEXIBLE, EXTENSIBLE, OPEN-SOURCE AND AFFORDABLE FPGA-BASED TRAFFIC GENERATOR

An existing open source packet generator called Flexible, extensible, open-source and affordable FPGA-based traffic generator (hereinafter referred to as the FEOSA generator) [1, 2] is used in this project. The solution implemented aims at great configurability, granted ability to fulfil a 10 Gb link even with the smallest packets and a high degree of modularity, hypothetically allowing for external usage, adaptation and extension. Although this is a highly functioning traffic generator, its development was conducted under conditions (2011 – 2013) different to ours (2016). This means that an older version of NetCOPE firmware was used on an earlier than the last generation of COMBO cards, resulting in various compatibility problems. Thus, implementing the FEOSA generator on the card currently available will most likely require a considerable amount of time and effort.

3. IMPLEMENTATION

This project's main goals are:

- To alter the FEOSA generator, so it can be compiled, implemented and run on modern FPGA boards. This includes remapping all ports used in the top entity file (project part responsible for mapping software-described pins on their hardware counterparts, physically present on the board), adjusting signals used in it and make it runnable on COMBO-80G card. This task is plausible.
- To transfer the environment used in FEOSA generator onto current version of NetCOPE framework. Knowledge of both versions of the software might be necessary to compare and find the discrepancies, which is in no matter guaranteed possible. Because of that, this step may prove difficult.
- To configure the generator to create DoS-like flood packets, once the previous steps are done. Due to its open-source nature and the creators having modularity in mind while developing it, this step is supposedly simple.
- To synthetize, compute and implement the final solution on the board provided. This does not stand any problem, if previous steps were completed successfully.
- To use it as a test tool, capable of stressing the device under test as much as necessary.

3.1. PROGRESS

Currently, first two phases of the plan as described above are being executed. Obstacles regarding the board differences (step 1) are time consuming to solve, yet appear that can be coped with. Concerns with the NetCOPE framework evolution are hard to grasp, providing no results or certainties whatsoever. Hard work and concentration on the matter are ahead. So far, a functioning standalone copy of the current NetCOPE environment has been successfully loaded into Vivado Design Suite [4] and examined using the test benches provided, as well as those modified by me exactly for the purpose. Same procedure was applied to the original FEOSA generator (not yet connected to NetCOPE), resulting in further understanding of its structural parts and hierarchy, along with confirming the functionalities.

The current task comprises of combining the genuine generator top file (*traffic_generator.vhd*) and the top file of the NetCOPE package (*application_core.vhd*) into a new, merged architecture. The purpose of the first one is to receive the configuration from hosting computer, set its parts accordingly, control the process of packet generation and send the outcome to the board's output interfaces. The latter one's task is to connect all the available interfaces and provide an example application. This is the code, which this project intends to replace with the contents of the traffic_generator file mentioned. The result of this combination would be able of both controlling the older entities in the generator, and communicating with the necessary newer parts of the environment. Many compatibility errors arise during this task, however all of those found so far have been managed and solved one by one. In theory, this should lead to a fully working project made from both of the parts running almost separately, despite both being controlled by one file.

4. CONCLUSION

Design and ideas presented in this paper are yet subject to development, hence it is not possible to present their results and achievements. As it seems, all the steps should be doable, meaning that a modern network high speed stress testing device / flood attack tool should be created, using an open-source software running on a widely available hardware, keeping the solution reasonably expensive.

ACKNOWLEDGEMENT

The project is being conducted at The Faculty of Electrical Engineering and Communication, Brno University of Technology, under the supervision of Ing. David Smékal. I must express my gratitude to him.

REFERENCES

[1] Groléat, Tristan. GitHub. *Hardware-traffic-generator*. [Online] 11. 3 2016. https://github.com/twisterss/hardware-traffic-generator/tree/master/hw/traffic_generator.

[2] HAL. *Flexible, extensible, open-source and affordable FPGA-based traffic generator*. [Online] 11. 3 2016. https://hal.archives-ouvertes.fr/hal-00859291/document.

[3] NetCOPE Technologies [Online] 26. 3. 2016. http://www.netcope.com.

[4] Xilinx, All programmable. *Vivado Design Suite*. 23. 6. 2016 http://www.xilinx.com/products/design-tools/vivado.html.