Load Balancer for IP Traffic Based on NetFPGA Cards

Marek Michalski, IEEE Member
Poznan University of Technology, Faculty of Electronics and Telecommunications, Polanka 3, 60-965 Poznan, Poland
e-mail: marek.michalski@put.poznan.pl

***Abstract*—In this paper, there is presented a project realized on NetFPGA cards for load balancing in IP networks. In many cases, it is hard to serve whole stream of incoming packets in one device with satisfactory deep analysis. This system allows to split high throughput stream of packets into several ones with smaller throughput. Each of them can be served by dedicated device with smaller requirement on performance. The presented project can also filter Internet traffic and extract only packets with interesting features. System has been realized and tested with cards with 1Gbps interfaces, it can be easily ported into cards with the speed of 10Gbps.**

I. Introduction

It is very hard to serve a high volume of Internet traffic in one device. Usually, traffic incoming to our device is a mixture of streams from different clients, from different services and protocols. To distinguish packets from particular stream, detailed analysis has to be performed. In most cases, it is enough to analyze only the header of IP packet or UDP/TCP segment. When analyzed traffic contains a lot of short packets, in such a stream, there is a lot of headers what gives a huge amount of information to analyze. Devices based on processors are not optimal solution for such a task. It is also almost not possible to serve such a traffic with relative low cost hardware. However, it is possible to use hardware load balancer and divide big stream into several smaller ones and serve them independently. In this paper I would like to present such a load balancer which is realized as a project implemented in NetFPGA cards and allows us to serve high throughput stream with devices with nominal throughput smaller then incoming traffic.

The rest of paper is organized as follows: Section II shortly describes NetFPGA cards, in Section III the main idea of project is presented. Section IV presents usage scenarios. Finally, Section V presents results of testing, last Section gives some conclusions and describes plans for future works in this area.

II. NetFPGA Cards

NetFPGA cards have been developed by University of Cambridge and Stanford University networking groups as an "open hardware" project [1]–[3]. NetFPGA card is an extension card for PC, it has four physical network interfaces ($4 \times 1Gbps$ or $4 \times 10Gbps$ electrical or optical ($SFP+$) Ethernet ports) and FPGA - programmable hardware as a main chip. More details about structure, architecture and usage of such a card can be found in the literature. What is very important, there is a public framework which implements basic network functionality (routers, switches and interface cards) and it is relatively easy to add our own functionality. In the literature, there is a lot of information about typical usage and projects prepared by NetFPGA community [4].

The main chip of NetFPGA card receives packets from physical ports as input traffic, analyzes and modifies them, and sends them out to physical ports as output traffic. Functionality of main chip is programmable and can be very flexible. Very important fact is that this functionality is realized by hardware, hence, the performance of such an appliance allows to serve whole traffic with speed of line, in case of these Ethernet ports it can be 1Gbps or 10Gbps.

III. Idea of Project

In most typical cases, there is used framework from one of NetFPGA reference project extended by additional functionality [6], [7]. In project presented here, I modified network interface card ($ReferenceNIC$) and used it to direct particular packets to proper VLANs and physical ports.

The first goal of project presented here, is to divide one stream of incoming packets into several smaller ones which can be served independently. Lets assume that we have to serve 1Gbps stream of packets from our clients. Such a stream can contain packets with source IP address form different IP networks, but all of them are directed to one destination IP address (lets assume $10.0.0.1$).

Each packet, basing on its IP source address, is assigned to one of N groups (N is power of 2). Number of groups can be defined by configuration software. The decision is realized by hashing function which takes into consideration three oldest octets in IP address. With such an approach, it is highly probable that packets from the same IP networks will be assigned to the same group. It can be useful for servers of higher layer protocols. Hashing function can be also changed by configuration, it can use different number of bits from IP address and realize different formulas. With almost uniform distribution of IP addresses the division will be almost equal, i.e. in each of group there will be the same number of packets. This is quite big approximation and detailed distribution depends on distribution of addresses of incoming packets, but such an accuracy is sufficient in this project.

NetFPGA cards have only four physical interfaces, but on each of them VLANs can be used, so, the number of logical

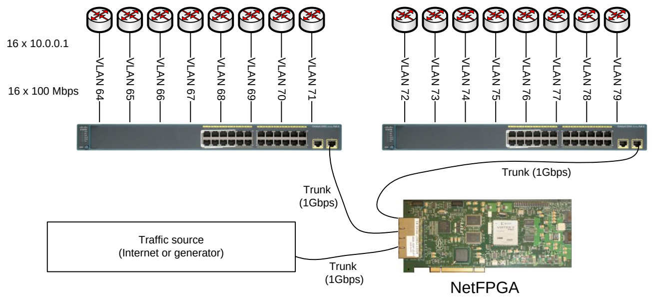

Fig. 1. Example of system configuration with one NetFPGA card and 2 Cisco 2960 switches

interfaces is not a limitation. One physical port is treated as an input port. Due to performance calculation, it is assumed, that at this port there is trunk (802.1q), it means, each Ethernet frame contains 4 additional bytes in its header. These bytes work as VLAN tag and have to be taken into consideration when we analyze throughput of traffic on this port.

Three physical ports can be used as direct output ports and they can be connected with servers for divided streams. They can be also connected to a switch with trunk and the switch can split VLANs from trunk to separate physical ports. Due to logical separation on L2, each server connected to switch can have the same destination IP address. Each IP packet goes to its destination, but it can be realized by different hardware. With dedicated software we can configure NetFPGA to work in different configuration. We can set number of groups N, and direct them to certain output ports. Example of basic configurations is presented in the Fig 1. If we assume semi uniform distribution of addresses, the load of two trunks from NetFPGA card to switches will be about 50%. Analogically, the load of each link from switch to router will be about 8% of primary stream (1Gbps), which is 80% of 100 Mbps.

The second goal of this project is to pass almost whole input stream to output port and extract stream of packets which will contain only packets with clearly defined features. Instead of hashing function and IP addresses we can use some parameters which denote requested packets. It is easy to extract for example SSH packets, SIP packets or another protocol. Very interesting set of functionality can be realized when we focus on packets with TCP segments with SYN flag set.

IV. Example of Usage

Many times developers are asked about scalability of their products and solutions. It is very hard question because almost each solution, device or another product is designed to work in certain conditions. General rules can be applied in general cases, in particular situation parameters of working systems are limited by available resources. One typical problem comes from Internet of Things, where huge amount of tiny devices are a client of one server. It is very easy to add more devices to our system. But the scalability of the server can be a problem. We can use more efficient hardware. But this way for efficiency enlargement has physical limitations which are very easy to be achieved. With this load balancer it is possible to multiply servers and use them as one big server, where the administrator can dynamically and flexible divide a traffic to be served by particular server.

V. Analysis of Performance and Throughput

The working project was tested in many configurations [5] with professional and certified network traffic - Spirent Test Center [8]. Several tests were realized in configuration like in Fig. 1: the stream of ICMP requests from generator has been split by one NetFPGA card and two switches into 16 independent streams of packets. These packets have as a source address random IP, but all of them are directed to the same destination IP address. In this test case scenario, this address is served by 16 different routers placed in different VLANs. Even with the highest possible throughput in physical layer (100% of 1Gbps) sent from traffic generator, each ICMP request had its own ICMP-replay.

VI. Conclusions and Future Work

In this paper there is presented NetFPGA project which works as a load balancer. With such an implementation it is possible to serve traffic with high throughput with several devices with nominal performance much smaller than throughput of primary stream. It can be run on NetFPGA cards with interfaces with speed of 1Gbps and 10Gbps. Its functionality will be extended in future projects.

Acknowledgements

The work described in this paper was financed from the funds of the Ministry of Science and Higher Education for the year 2017 under Grant 08/82/DSPB/8221.

References

[1] Website of community and NetFPGA project: *http://www.netfpga.org*.

[2] G. Gibb, J.W. Lockwood, J. Naous, P. Hartke, and N. McKeown. “Netfpga: An Open Platform for Teaching How to Build Gigabit-rate Network Switches and Routers”, In *IEEE Transactions on Education*, Volume: 51, Issue: 3 pp. 364-369, August 2008.

[3] N. Zilberman, Y. Audzevich, G.A. Covington, A.W. Moore: *NetFPGA SUME: Toward 100 Gbps as Research Commodity*, IEEE Micro, vol.34, no. 5, pp. 32-41, Sept.-Oct. 2014, doi:10.1109/MM.2014.61

[4] NetFPGA reference projects: *http://www.netfpga.org/project_table.html*.

[5] M. Michalski, The Configurations for Experimental Study of the Network Performance, *8th IEEE, IET International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP 2012)* 18-20 July 2012.

[6] M. Michalski: The System for Delay Measurement in Ethernet Networks on NetFPGA Cards, IEEE International Conference on High Performance Switching and Routing 2014, Vancouver, Canada 1-4 July 2014.

[7] Michalski, T. Sielach: "The Analysis of Time Reaction in OpenFlow Switches in NetFPGA Cards and ROFL", 2015 International Symposium on Networks, Computers and Communications (ISNCC 2015), 13-15 May 2015

[8] ’Spirent Test Center’: *www.spirent.com/~media/Datasheets/Broadband/PAB/SpirentTestCenter/STC_Hardware_and_BasePackage_Matrix.pdf*.

[9] Oficial website of AM Technologies Company: *http://www.amt.pl*.

[10] JDSU - MTS5800 traffic generator and analyzer: *http://www.amt.pl/produkty/telekomunikacja/sieci-dostepowe/296.html*.

[11] Xilinx University Program: *http://xilinx.com/support/university.html*.