NOVEL ARCHITECTURE OF NETFLOW ADAPTER

Martin ŽÁDNÍK, Bachelor Degree Programme (3)
Dept. of Computer Systems, FIT, VUT
E-mail: xzadni00@stud.fit.vutbr.cz

Supervised by: Ing. Jan Kořenek and Ing. Tomáš Pečenka

ABSTRACT

With growing speed of communication over Internet there is need for reliable devices which are able to perform different operation upon running trough data (i.e. routing, filtering, monitoring). This paper propose to implement network flow monitoring (NetFlow[3]) in specialized hardware. With usage of field programmable gate arrays (FPGAs) it is possible to monitor flows in high-speed environment. That can improve situation today, which is discussed in first part of the article. After that the model of proposed architecture is described. Because the architecture is highly configurable, main variable attributes are analyzed in next part of this article. Expected results and target technology are discussed in conclusion.

1 INTRODUCTION

Todays active network devices are usually busy doing their own job (routing, switching, filtering, etc.). Some of them are able to monitor network but they are limited by processor, bus or memory. Therefore they sample the traffic and during attacks are unable to monitor at all. That significantly decreases value of statistical information they export.

Nowadays FPGA offers high performance in means of computational power (millions of gates operate in parallel), flexibility (reprogrammable), maximal external memory bus utilization and ratio of cost/performance. All these favorable features can be used to optimize architecture for specific application. In this paper a novel architecture of NetFlow monitoring probe is proposed.

The architecture introduces unconventional features for hardware solution but still suitable for implementation in FPGA. Problem is solved with usage of hashes (used for fast and additional lookups), pointers in multiple memories (usage of more memories), unique communication protocol (allows fully parallel operating units), and bidirectional bounded list (implementing LRU algorithm). For abstract see Fig. 2. These techniques should allow to monitor 2,5 Gbps link with simultaneous processing of up to $1,000,000$ flows. It also enables to support various tactics of monitoring (such as [1]) and support of todays most used protocols (IPv4 and IPv6).

2 ARCHITECTURE OVERVIEW

Simplified design of the architecture is on Fig. 1. It is composed of six main units whose function is explained below.

All packets with the same destination and source IP addresses, ports and protocol belongs to one flow. Packet is assigned timestamp in Input Buffer after arrival. Then it is parsed with Header Field Extractor (HFE). Output is information about packet (length, timestamp, identification fields). Identification is hashed because of long and variable fields (IPv4, IPv6) in Hash unit. Hash Search (HSEARCH) then lookups hash values in its memory and provide pointer to following memories. Item in Manager unit addressed by HSEARCH pointer is rebinded to the beginning of bidirectional bounded list. This way the oldest items gather at the end of the list and can be effectively controlled and disposed. HSEARCH pointer also designates record in Storage memory where statistical data are held. Proposed protocol enables independent work of all units on the others (Fig. 3).

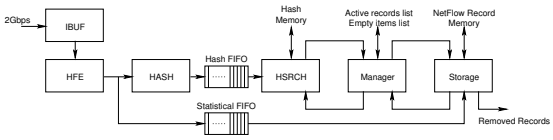

Figure 1: Block structure

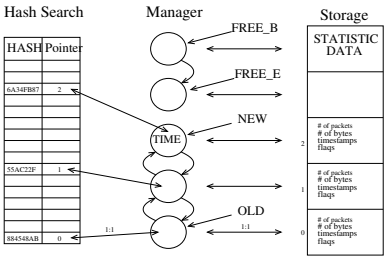

Figure 2: Abstract

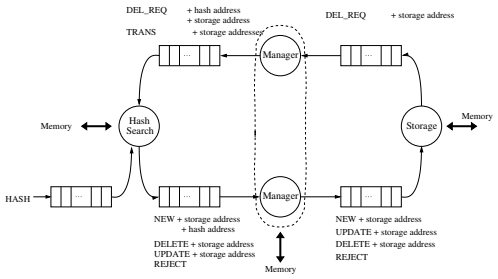

Figure 3: Protocol

Considering reliability and accuracy, the most dangerous points are hash functions and protocol. Therefore it is good to know parameters that fulfill limits of errors that administrator wants to reach.

Let us suppose that the Hash Search is able to distinguish whether flow belongs to certain entry or not. V is the ratio of used entries to number of all ones. Then probability that the Hash Search finds no empty entry for new flow is $P(failure) = V$, is too high. That is why additional lookups have to be done. Then it holds that after n lookups, probability of no available item is:

$$P(failure) = V * \frac{1-V^n}{1-V}$$

Please focus on dependency between number of lookups and usage of memory. Desired probability should be around $1 * 10^{-4}$ (Fig. 4).

As supposed above, Hash Search is able to distinguish whether flow belongs to certain entry or not thanks to another hash number which is stored in every entry point. That is why it is not crucial when first hash map two different flows to one entry.

So let's suppose that we have a set of F possible flows. Every packet is assigned to one of B buckets (by hash distribution), where $B < F$. However, packets of the same flow are always assigned to the same bucket. Collision is a situation when at least one of the buckets contains packets from at least two different flows. We are interested in the probability of the undetectable collision per second (Fig. 5):

$$P(collision) = \frac{k * \frac{F}{B}}{F} = \frac{k}{B}, P(collision/second) = \frac{k}{B} * x$$

where S is number of flows belonging to one bucket, k is number of used items and x number of new flows per second.

The $M/M/1$ Kendall mass service system model can be used for the analytic system model (Fig. 3). Formal verification of protocol is then straightforward.

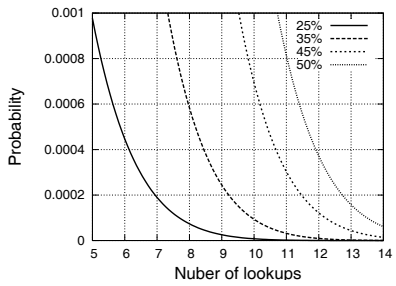

Figure 4: Probability of failed lookups

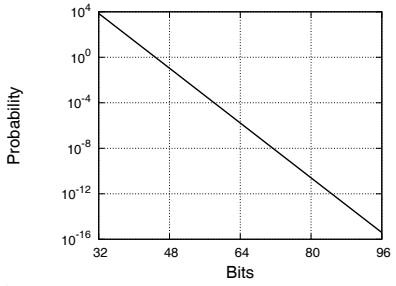

Figure 5: Probability of undetected collision (logarithmic scale)

3 CONCLUSION

This paper introduced model of new architecture for flow monitoring system and proved that well chosen hash and number of lookups, together with safe protocol create reliable as well as flexible solution. The architecture will be implemented in VirtexIIPro (Xilinx) utilizing the COMBO6 platform [3]. The circuit will be capable of monitoring 1 million flows at OC-48 (2.5 Gbps) data rates. Our future work concentrates on implementing simulation model in SIMLIB/C++ as well as on implementing the architecture in VHDL. In addition design can be enhanced by different types of heuristics (sampling, sideway filters [2]).

REFERENCES

[1] Estan, C., Varghese, G. New Directions In Traffic Measurement: Focusing on the Elephants, Ignoring the Mice, San Diego, University of California. http://portal.acm.org/citation.cfm?doid=859716.859719, 2003

[2] Košnár, T.: Notes to Flow-Based Traffic Analysis System Design, CESNET, Prague. http://www.cesnet.cz/doc/techzpravy/2004/ftas-arch/, 2004

[3] Liberouter: Liberouter Project WWW Page. http://www.liberouter.org, 2005