# A Benchmarking Methodology for NPU-Based Stateful Firewall

Lee Eng Kean, Sulaiman bin Mohd. Nor
Department of Microelectronics & Computer Engineering,
Faculty of Electrical Engineering,
Universiti Teknologi Malaysia.
81310 UTM Skudai, Johor,
Malaysia.
engklee@yahoo.com, sulaiman@suria.fke.utm.my

*Abstract*—**It is currently a challenge to evaluate and compare network processor-based (a.k.a. NPU-based) applications due to the heterogeneity of the application designs and hardware architectures. Many research works have been done in the area of packet classifications, which were directed towards proposing high-speed firewall implementation algorithms. However, few researches focus on providing standard benchmarking methodology for a specific class of firewall. This paper presents a benchmarking methodology for NPU-based stateful firewall. The aim is to allow 'apple-to-apple' comparison among similar firewalls available in the market, and to provide practical benchmarks that exhibit realistic performance numbers. The key aspects of this work are benchmark specifications, separation of benchmark granularity in a layered manner, and the means of measurement. Finally, a system-level benchmark is applied on a stateful firewall implemented on Intel's IXP1200 network processor as a proof-of-concept of this work.**

***Keywords - stateful firewall; network processor; performance; benchmarking methodology***

## I. INTRODUCTION

Firewall is a software- or hardware-based network security mechanism. It is often implemented in a router to control packet flows. Due to the extra packet-processing overheads, this mechanism may turns out to be the performance bottleneck of a network system. If the system experiences degradation in response times, it may unable to consistently establish connections through the network, thus adversely affecting the security and efficiency of the system. Therefore, performance is a vital concern, especially when a choke point firewall is implemented.

Stateful firewall is one of the firewall systems that incorporate capability to keep track of each connection session while performing IP packet filtering. In most of the current solutions, network address translation (NAT) is included into the device as well. This type of firewall provides higher security level in comparison with traditional stateless firewall. Nonetheless, it is more compute-intensive and resource consuming.

Network processors (NPUs) are software programmable chips specially designed to process packets at wire-speed. They are targeted for applications spanning from layer 3 through 7 of Open System Interconnection (OSI) [1]. Flexibility and programmability place network processors in a better position compared to inflexible ASICs chips. Consequently, there are increasing numbers of firewall vendors implementing their solutions on NPUs.

Benchmarking NPU-based stateful firewalls is complicated by a variety of factors. Firstly, vendors employ widely varying implementation algorithms. Secondly, there is wide range of NPUs available in the market with diverse hardware architectures. NPUs are still emerging and it is lack of standard definitions. Finally, the benchmarks have to cater to different audiences, from NPU programmers to end-users. As of this writing, there is no standard benchmarking methodology for this specific class of firewall.

Motivated by the above discussion, this paper aims at proposing a benchmarking approach for NPU-based stateful firewall. The rest of this paper is organized as follows: First, summary of works on NPU and firewall benchmarking; Second, discussion of benchmark specification principles; Third, definition of benchmark granularity in a layered manner; Forth, proposal of NPU-based stateful firewall benchmarking methodology at system level; Finally, demonstration of system-level benchmarks implemented on an IXP1200-based stateful firewall.

## II. RELATED WORKS AND STANDARD BODIES

There are a number of efforts in NPU benchmarking. NPF Benchmarking Work Group is a standard body that actively works on NPU benchmarks in application domains. This work group defines benchmarks based on "micro-level," "function-level," and "system-level" applications. It is currently working on implementation agreements and benchmarking templates for IPv4, IPv6, MPLS forwarding, and DiffServ QoS. The proposed approach addresses the environment and system-related issues [2]. However, to our knowledge, there is no benchmarking effort on stateful firewall application being worked at present.

On the other hand, IETF Benchmarking Methodology Working Group publishes a series of Request for Comments (RFCs) describing benchmarking terminology and methodology for a wide range of networking devices. This

0-7803-8114-9/03/$17.00 ©2003 IEEE.

working group defines generalized terminology and methodology for firewall performance benchmarking in RFC2647 [3] and RFC3511 [4]. Both of these RFCs were published to provide standard benchmarking terminology and methodology for all classes of firewalls. The suggested approaches are too general to be applied to a particular class of firewall.

## III. Benchmark Specification Principles

A comprehensive benchmark specification requires detailed description of test environment setup, measurement, performance metrics, and granularity.

### A. Test Environment Setup

The test environment specification describes testbed and traffic configurations. Typically, a minimal testbed setup consists of a tester and device under test (DUT). A tester generates and sends packets to the DUT. Then, the DUT sends its output back to the tester. In this case, the tester serves as both packet source and sink. Alternatively, packet source and sink can be separated into two devices: a packet sender (traffic generator) and a packet receiver (packet capture) [5].

The test data must include a range of packet sizes, from the minimum to maximum allowable packet size. For Ethernet devices, it is recommended to use evenly distributed frame sizes of 64, 128, 256, 512, 1024, 1280, and 1518 bytes [5]. For Internet mix traffic, a distribution pattern is depicted in Table I [6].

TABLE I. Internet Mix Traffic Distribution

| IP (bytes) | Ethernet (bytes) | Proportion (%) |
|---|---|---|
| 40 | 64 | 56 |
| 576 | 594 | 20 |
| 1500 | 1518 | 24 |

### B. Measurement

The measurement specification defines means and procedures of performance measurement. Before measuring the performance, functional correctness must be proven. Functional correctness can be tested using reference implementation such as Click executable description [7].

Generally, overall performance can be measured as the aggregate throughput, where no legal traffic is dropped. This is measured by iteratively adjusting the intended load until the maximum rate is observed without packet loss. Measurement procedures will be further discussed in Section V.

### C. Performance Metrics

The performance metrics specification describes the relevant metrics to exhibit performance numbers. The key units in line-speed performance measurement are packets per second and bits per second. The notion of time is the central measurement of most of the performance metrics.

Performance metrics is chosen based on the measurement purpose. Throughput and latency provides insight into the computational performance. In some literatures, derived metrics such as cost effectiveness (performance per cost) was proposed to determine proportion of performance and cost. Detail of performance metrics is provided in Section V.

### D. Granularity

Granularity is the size of a benchmark compared to the largest application [8]. In general, there are four granularity levels: system-level, functional-level, micro-level, and hardware-level. Granularity of a benchmark can be defined according to performance bottlenecks of different subsets of an application. It pinpoints the performance factors systematically at a layered manner. This specification will be discussed in Section IV.

## IV. Layered Benchmark Granularity

This section briefly discusses each of the benchmark granularity proposed by P. Chandra et al. [9]. Benchmarks at hardware, micro, and function levels have been provided in their work. Thus, only system-level benchmarks will be demonstrated in this paper.

### A. Hardware-level

This level of benchmarks measures the throughputs and latencies for accessing hardware resources. These measurements assist software engineers in choosing appropriate hardware resources and data placement. The key performance latency involved memory accesses. However, this paper does not demonstrate benchmarks on hardware level because it is largely dependent on the target hardware platforms architecture rather than firewall designs.

### B. Micro-level

This benchmark level targets elementary operations that can be combined to make up a function. The operations used must be separable to facilitate independent measurement. Examples of micro-level benchmarks are longest prefix match table lookups, and CRC calculations. This level of benchmark helps NPU developers in designing value-added functionalities.

### C. Function-level

This level of benchmarks measures the performance of a standalone networking function. It is useful for evaluating the performance of an NPU for a single function such as NAT and IP forwarding. It is targeted for NPU customers who will use standalone functions provided by the NPU vendors.

### D. System-level

System-level benchmarks are targeted at performance of a complete networking system. Performance is measured at the entire system. However, it cannot pinpoint the performance of a particular application on a single NPU. This shortcoming can be complemented by function-level benchmarks. Examples of system-level benchmarks are IPv4 router and firewall.

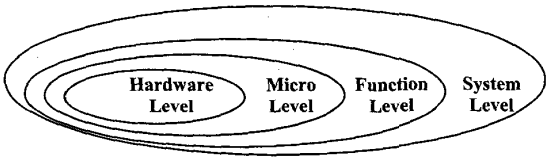


Figure 1. Layered benchmark granularity

## V. SYSTEM-LEVEL BENCHMARKS

This section describes the system-level benchmark for NPU-based stateful firewall. Major aspects that need special attentions are test environment setup, reference architecture, performance metrics definition, test traffic configuration, and reporting format. The following description is essentially based on the IETF RFC1242 [10], RFC2544 [5], RFC2647 [3], RFC3511 [4], and NPF IPv4 Forwarding Application-Level Benchmark Implementation Agreement [11].

### A. Test Environment Setup

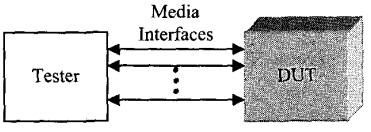


Figure 2. System-level benchmark test setup

Figure 2 illustrates a typical test environment setup of system-level benchmarking for NPU-based stateful firewall. The firewall or DUT is viewed as a black box. The tester acts as traffic source and sink, which sends and receives packet streams with configurable network and data link layer headers. It is also capable to count number of packets received and transmitted.

When reporting the results of the benchmarking test, hardware information listed in Table II should be provided.

TABLE II. HARDWARE INFORMATIONION NEEDED

| | |
|---|---|
| **Environment setup** | • Block diagram of test environment setup, and internal data path. |
| **Network Processor** | • Number of network processor(s) used.<br>• Programmable Processing Engine (PPE).<br>• Control processor core. |
| **Media Interface** | • 10/100Mbps Ethernet, Gigabit Ethernet, etc.<br>• Total number of media interface(s) supported.<br>• Can be homogenous or combination of heterogeneous media interfaces. |
| **External Memory** | • External memory interfaces.<br>• Amount of memory on the DUT used for certain function. |
| **Bus Interface** | • Frequency and bandwidth of bus interfaces. |
| **Hardware list** | • List of additional hardware components used. |

### B. Reference Architecture

A typical reference architecture or packet traversal of an NPU-based stateful firewall is shown in Figure 3.

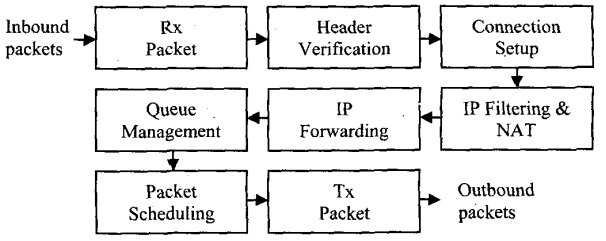


Figure 3. Packet traversal in a typical NPU-based stateful firewall

### C. Performance Measurement

#### 1) IP Throughput

*a) Definition:* The maximum rate at network layer which none of the received packet is dropped by the DUT without activating filtering rules.

*b) Measurement units:* Input packets per second or input bits per second. The bits to be counted are in the IP packets, excluding data link layer headers and trailers.

*c) Procedure:* The tester sends a specific number of unicast frames at a constant rate on each of the media interfaces of the DUT simultaneously, then counts the number of frames received. If the number of frames sent and received by the tester are the same, the rate at which frames are received at the tester is throughput of the DUT. If fewer frames are received than were sent by the tester, the rate of the frames generated is reduced and the test is rerun. This process is iteratively repeated until the maximum rate without packet loss. The duration of each test should be at least 120 seconds [11]. If NAT is implemented, tests should be run with NAT disabled and enabled [3].

*d) Test traffic configuration:* For Ethernet test, evenly distributed frame size of 64, 128, 256, 512, 1024, and 1518 bytes should be used. For Internet mix test, it is recommended to use traffic distributions shown in Table I.

*e) Reporting format*: The results must be reported in the form of graph. The x-axis must be the frame sizes and the y-axis must be the agregate throughput in packet per second or bits per second. Beside measured throughput, the theoretical throughput calculated at each of the input frame sizes must be shown in the graph. It is recommended that separate lines drawn for cases where NAT disabled and enabled.

#### 2) Latency

*a) Definition:* The time interval starting when the last bit of input frame reaches at the input interface of the DUT, and ending when the first bit of the output frame is observed at the output interface of the DUT.

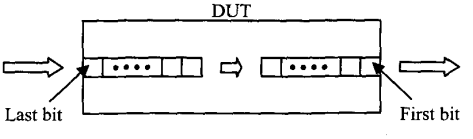


Figure 4. Latency measurement

*b) Measurement units:* Seconds.

*c) Procedure:* The tester sends a stream of frames at a specific size to the DUT at a constant rate. The generated traffic should be at least 120 seconds in duration. Tagged frames are used for timestamping. The time at which the tagged frame is fully transmitted by the tester is recorded. Then, the tester records the time at which the tagged frame is received on the other side after transmitted by the DUT. The latency is the time interval between both recorded times.

*d) Test traffic configuration:* Same as in Section C.1.d.

*e) Reporting format:* The result must be reported in the form of graph. The x-axis must be the frame sizes and the y-axis must be the measured latency in seconds. Separate lines for latencies measured at different throughput rates should be drawn in the same graph. The reported results should be the average latency in an aggregate fashion over all interfaces.

*3) Goodput*

*a) Definition:* The rate at which packets are forwarded to the correct destination interfaces of the DUT, excluding any packets dropped due to the rule set definition.

*b) Measurement units:* Output packets per second, or output bits per second. The bits to be counted are in the IP packets, excluding data link layer headers and trailers.

*c) Procedure:* The tester sends a specific number of unicast frames at a constant rate to each of the media interfaces of the DUT simultaneously, then counts the number of frames received on the other side of the DUT. The duration of each test should be at least 120 seconds. If NAT is implemented, separate tests should be run with NAT disabled and enabled.

*d) Test traffic configuration:* Same as Section C.1.d.

*e) Reporting format:* The results must be reported in the form of graph. The x-axis must be the frame sizes and the y-axis must be the aggregate rate in packets per second or bits per second. Separate lines for 25%, 50%, and 75% legal traffic should be drawn in the same graph. Besides, the IP throughput in Section C.1 should be shown in the graph.

## VI. RESULTS

This section demonstrates system-level benchmarks applied on an IXP1200-based stateful firewall.

### A. Test Environment Setup

TABLE III. HARDWARE INFORMATION

| | |
|---|---|
| **Environment setup** | • Refer to Figure 2, and 3. |
| **Network Processor** | • 1 x IXP1200 Network Processor.<br>• 6 x 232MHz Microengines.<br>• 6 x 4 hardware threads.<br>• 1 x 232MHz SA-1 core. |
| **Media Interface** | • 8 x 10/100Mbps IXF440 Ethernet ports. |
| **External Memory** | • 256MB SDRAM, 8MB SRAM.<br>• 4096 x 64-Bytes rules supported in policy table (2MB SDRAM).<br>• 1024 x 64-Bytes concurrent connection supported in state table (512KB SRAM). |
| **Bus Interfaces** | • 116MHz 64-bit SDRAM interface.<br>• 116MHz 32-bit SRAM interface.<br>• 66MHz 32-bit PCI interface.<br>• 104MHz 64-bit IX Bus interface. |
| **Hardware list** | • 64-bit and 48-bit hash key generation engine. |

### B. Performance Measurement

*1) IP Throughput*

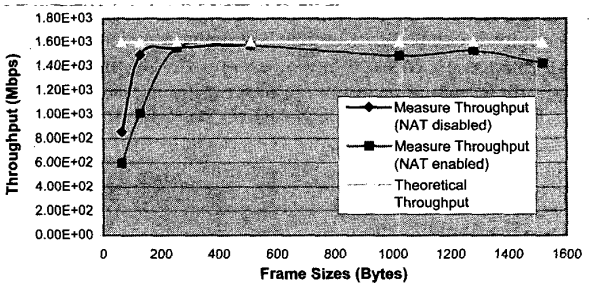


Figure 5. IP throughput in 'bits per second'

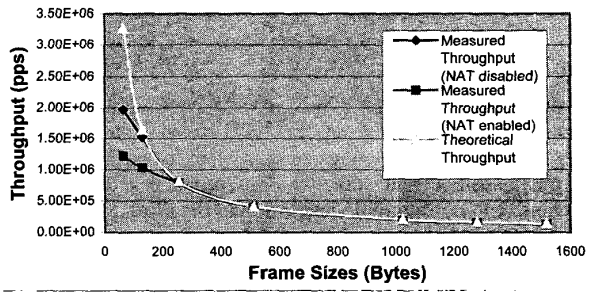


Figure 6. IP throughput in 'packets per second'

*2) Latency*

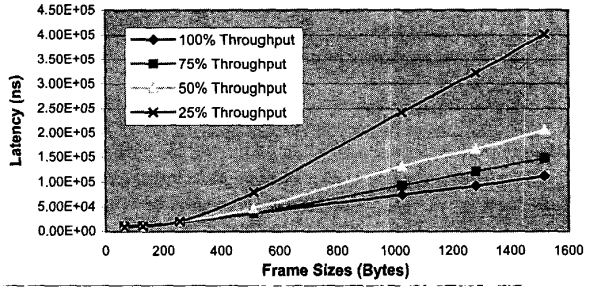


Figure 7. Latency with NAT disabled

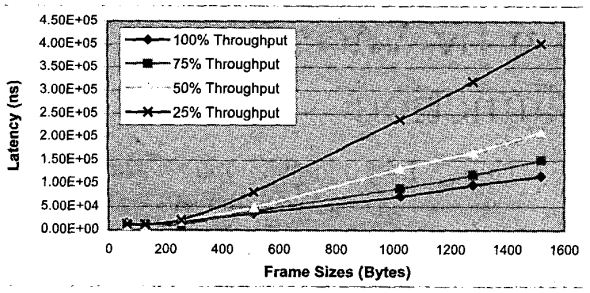


Figure 8. Latency with NAT enabled

*3) Goodput*

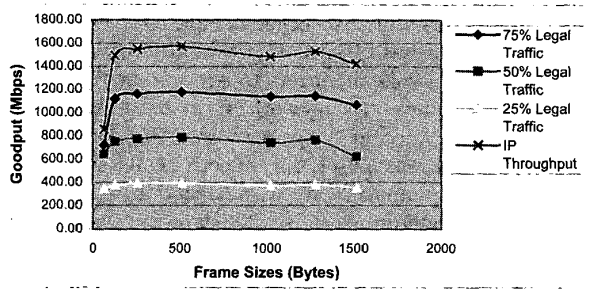


Figure 9. Goodput in 'bits per second'

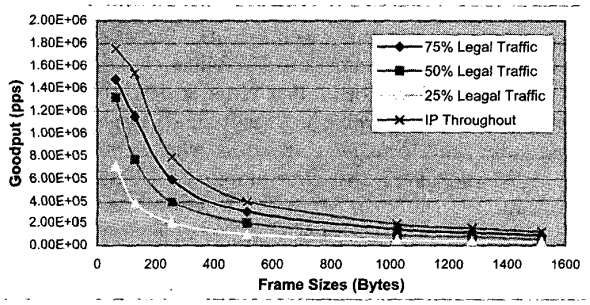


Figure 10. Goodput in 'packets per second'

## VII. SUMMARY

Evaluating and comparing NPU-based applications pose challenges due to the heterogeneity of design architectures and hardware platforms. A practical benchmarking methodology is required to facilitate comparison among similar systems. Key aspects of a benchmarking approach are precise definitions of benchmark specifications and separation of benchmark granularities. Test environment setup, reference architecture, performance metrics, and traffic configurations are subsets of benchmark specification that need to be identified and defined accurately. There are four levels of granularity: system-level, function-level, micro-level, and hardware-level. This paper describes benchmarking methodology for NPU-based stateful firewall only at system level. Finally, the proposed benchmarking approach is applied on an IXP1200-based stateful firewall as a proof-of-concept of this work.

## VIII. FUTURE WORKS

The approach proposed in this paper involves only data plane test. We are working to provide control plane performance measurement for stateful firewall, including TCP connection establishment and tear down times. Besides, benchmarks can also be developed in other platforms, such as IBM's PowerNP, and Motorola's C-Port network processors.

## ACKNOWLEDGMENT

Funding of this research work is currently provided by UTM-PTP. Besides, the author would like to thank Intel Corporate for the IXP1200 Ethernet Evaluation System sponsorship.

## REFERENCES

[1] S. Lakshmanamurthy, K.Y. Liu, Y. Pun, L. Huston, U. Naik, "Network Processor Performance Analysis Methodology", Intel Technology Journal, vol. 6, August 2002.

[2] P. Chandra, and S.Y. Lim, "Framework for Benchmarking Network Processors", Rev. 1.0, NPF Benchmarking Workgroup, August 2002.

[3] D. Newman, "Benchmarking Terminology for Firewall Performance", Request for Comments 2647, Network Working Group, August 1999.

[4] B. Hickman, D. Newman, S. Tadjudin, T. Martin, "Benchmarking Methodology for Firewall Performance", Request for Comments 3511, Benchmarking Working Group, October 2002.

[5] S. Bradner, and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices", Request for Comments 2544, Network Working Group, March 1999.

[6] Mitsuhiro Miyazaki, "Workload Characterization and Performance for a Network Processor", Princeton Architecture Laboratory for Multimedia and Security, May 2002.

[7] N. Shah, W. Plishker, K. Keutzer, "NP-Click: A Programming Model for the Intel IXP1200", University of California, Berkeley, Feb. 2003.

[8] M. Tsai, C. Kulkarni, C. Sauer, N. Shah, and K. Keutzer, "A Benchmarking Methodology for Network Processors", University of California, Berkeley, 2002.

[9] P. Chandra, F. Hady, R. Yavatkar, T. Bock, M. Cabot, and P. Mathew, "Benchmarking Network Processors", Intel Corporation.

[10] S. Bradner, "Benchmarking Terminology for Network Interconnection Devices", Request for Comments 1242, Network Working Group, July 1991.

[11] Prashant R. Chandra, "IPv4 Forwarding Application-Level Benchmark Implementation Agreement", Rev. 1.0, NPF Benchmarking Workgroup, 2002.