1. Alicja STARCZEWSKA[1], 2. Krzysztof DANIEC[1], 3. Jarosław HOMA[1], 4. Aleksander NAWRAT[1]

Silesian University of Technology (1)

doi:10.15199/48.2023.12.08

Development of the system for testing Ethernet network and network applications performance and security, including RFC 2544 test and DDoS attack resistance test

Abstract*. In the age of the Internet a very important issue is network and network application performance and security. Network architects and web application designers need appropriate tools to examine their services. This paper describes the new application for testing mentioned parameters that do not require specialized hardware. Performance tests implemented in presented application are basing on RFC 2544 document. That standard determines a.o. duration of performed tests. Authors tried to shorten times of tests and compared results with full size RFC 2544 compliant tests results.*

Streszczenie. *Bardzo ważnym zagadnieniem w dobie Internetu jest bezpieczeństwo i wydajność sieci internetowej i aplikacji sieciowych. Architekci sieciowi oraz projektancji aplikacji sieciowych potrzebują odpowiednich narzędzi pozwalających na przetestowanie ich produktów. Ten artykuł przedstawia nową aplikacją służącą do testowania wybranych parametrów sieci i aplikacji sieciowych, która nie wymaga wykorzystania specjalistycznego sprzętu. Testy wydajności są zaimplementowane na podstawie dokumentu RFC 2544 definiującego m.in. interwały czasowe potrzebne do przeprowadzenia rzetelnych testów. W ramach badań podjęto próbę skrócenia czasu trwania testów i porównano wyniki z wynikami testów zgodnych z RFC 2544.* ***(System do analizy wydajności i bezpieczeństwa sieci ethernet i aplikacji sieciowych z uwzględnieniem RFC 2544 oraz odporności na atak typu DDoS)***

Keywords: RFC 2544, performance testing, DDoS, Ethernet network, network measurements, network application.
Słowa kluczowe: RFC 2544, testy wydajnościowe, DDoS, sieć Ethernet, pomiary sieciowe, aplikacje sieciowe

Introduction

Nowadays almost all areas of human life are automated. The Internet allows to connect everything with each other, that makes the opportunity to control many things remotely and to communicate easily and quickly with the whole world. That task is often performed using numerous web applications. The main requirements for them are high performance and security. In the 21st century everything should work quickly and smoothly, otherwise it is considered malfunctioning. Simultaneously amount of transmitted information is increasing. At the same time people value the protection of their data very much. That is the reason why IT designers need appropriate tools to examine their systems.

At 1999 in RFC 2544 [1] there was defined benchmarking methodology for network interconnect devices, that is focused on four main parameters characterizing network performance:

- latency – the time interval starting when the end of the first bit of the input frame reaches the input port and ending when the start of the first bit of the output frame is seen on the output port [2],
- throughput - the maximum rate at which none of the offered frames are dropped by the device [2],
- frame loss rate - percentage of frames that should have been forwarded by a network device under steady state load that were not forwarded due to lack of resources [2],
- back-to-back - fixed length frames presented at a rate such that there is the minimum legal separation for a given medium between frames over a short to medium period of time, starting from an idle state [2].

In the case of network application important issues are also parameters specifying possibility of servicing many users simultaneously. Such an application should run smoothly regardless of the number of queries. Besides it should be resistant to various types of attacks. One of the most popular is Distributed Denial of Service (DDoS).

This article demonstrate a specialized test stand used to perform a set of tests including performance tests, e.g. tests based on RFC 2544, and security tests, e.g. DDoS resistance test.

State of the Art

a. Related literature

The most often discussed in research articles parameters, that characterize network performance are latency and throughput.

Waheed et al. [3] described solution that performs stress testing using automatic script generator. That generator is basing on models created using Interaction Flow Modeling Language (IFML) and Unified Modeling Language (UML). Hussain [4] proposed system consist of models written using Java language, witch aim is to generate and simulate users traffic and then to analyze the tested server behavior. An important element of a network performance is latency. Lots of web applications transmit large amounts of data, so it is necessary to ensure low latency of network to smooth running of the application. Park et al. [5] created system based on FPGA called Formullar. Its main goal is to precisely measure latency form ultra-low latency systems. That precision is achieved by measuring time on the FPGA, that eliminates external noise such as delay caused by OS routines.

Network performance testing is an important issue concerning different types of networks. Loiacono et al. [6] described ENSIGHT, that is EOSDIS network performance measurement system. It tests network in two ways: passive and active. Passive measurements are based on data stored on network devices, while active measurements check response to generated traffic. Goenka et al. [7] proposed Client-side Active Measurement platform (CLAM) using mechanism called Network Error Logging (NEL) implemented in Chromium-based browsers. That platform enables performance measurements between user and HTTP server. According to authors it is dedicated to Content Delivery Networks and it is able to eg. track network latency, track user-performance changes and capture client-LDNS mapping. Mayer et al. [8] proposed technique measuring throughput using a tool called SRPerf. Mendiola et al. [9] were analyzing network throughput and latency using SmartBit 600B device from SPIRENT. Yang et al. [10] proposed technique measuring network throughput and latency using tool called qperf. Huynh et al. [11] compared latency of different types of networks.

A set of tests characterizing network performance is described in RFC 2544 [1]. One of the way of realization these ideas was proposed by Lifu et al. [12]. The solution uses gSOAP tools, that enable easy development of SOAP/XML Web services and client application in C/C++, by providing a SOAP/XML-to-C/C++ binding. System architecture consist of Windows client working as User Interface and Linux server performing tests.

Another approach to performance web testing basing on RFC 2544 is to use FPGA. Ozcan and Yalcin [13] proposed solution implemented on Xilinx SDK, consist in FPGA board performing test, connected witch PC host acting as User Interface via RS232. Wang et al. [14] described system, where FPGA solely transmits and receives ethernet packets. It is connected via PCI bus with test management PC host that creates tests using libnet library.

b. Software solutions

According to Pradeep and Sharma [15] there are various approach of web application testing, eg. load testing, stress testing, security testing, smoke testing, unit testing, acceptance testing, GUI testing, gorilla testing or performance testing. These approach can be realized in different programming languages, such as Java, Python or Ruby. To the most popular web testing tools are as follows:

- Locust – an open-source tool basing on Python language and having Command Line Interface (CLI). It is able to generate multiple virtual users traffic [16].
- Apache JMeter - an open-source tool with Integrated Development Environment (IDE) used to examine numerous of internet protocols e.g. FTP, TCP, SMTP, POP, IMAP, HTTP or HTTPS [17].
- Hulk - one of web application testing approach consist in perform DDoS attack on HTTP server, for example HTTP Unbearable Load King (HULK). That attack can be realized using Hulk DoS Tool API written in Python or Go language [18].
- The Grinder - an open-source Java Framework enabling easy load testing with flexible scripting in Jython or Clojure [19].
- Capybara - a tool for simulate user behavior written in ruby with intuitive API. It can be used with others tools such as Cucumber, RSpec or Minitest [20].
- Pylot – an open-source Python tool for testing web application by simulating HTTP requests and analyzing server responding [21].
- Tsung – an open-source load testing tool written in Erlang. It supports various protocols such as HTTP, SOAP, MySQL, PostgreSQL, TCP, UDP, TSL/SSL and is able to simulate numerous of virtual user simultaneously [22] .
- SoapUI – an open-source tool for testing REST, SOAP and GraphQL-based web services with easy-to-use graphical interface [23].
- OWAMP – a tool measuring latency between hosts, running as a command line client application [24]. To appropriate results it requires a synchronized with NTP protocol and stable clock.

c. Hardware solutions

There are lots of commercial hardware ethernet network testers, e.g.

- Ethernet Inline Protocol Analyzer MGA2510 from Aukua Systems [25] – solution supported from 100 Mbps up to 10 Gbps data rate. It has intuitive browser-based interface and programmable RESTful API. The device is able to decode Hundreds of protocols with nanosecond precision timestamp. It includes latency monitoring analyzer and realtime streaming capture.
- Ethernet Traffic Generator and Analyzer XGA4250 from Aukua Systems [26] – solution used to Bit Error Rate Testing (BERT), throughput validation, latency measurement, monitoring or negative testing. It support data rates up to 25 Gbps. It has intuitive browser-based GUI and full RESTful API.
- Multi-Functional Ethernet/IP Tester PacketExpert 10G from GL Communications Inc. [27, 28] – solution supporting tests such as BERT, RFC 2544, Y.1564 or RFC 6349 up to 10 Gbps. It can be controlled by GUI or CLI/API enabling automated testing.
- GET-100A Gigabit Ethernet Tester from ShinewayTech – compact device that could be used as network tester, data packet sniffer, traffic generator, cable tester or RFC 2544 compliant tester. It includes 4.3 inch LCD color touchscreen and USB interface.

Test stand

a. System architecture

The application consists of two parts: a main application and a server application. The main application is responsible for initiating all tests. It also acts as HTTP server for its Graphical User Interface (GUI). The server application is necessary for carrying out network performance tests. Both parts of the application are able to run only on the Linux Operating Systems. The application is able to perform two kinds of tests: network tests and web application tests. System architecture required to perform network tests is depictured at the figure 1. Devices under test (DUT) include whole tested infrastructure and are placed between the client machine and the server machine. The client machine is the device where the main application is installed. The server machine is the device where the server application is installed. System architecture required to perform web application tests is depictured at the figure 2. In that case it consist only of the main application and the machine with tested HTTP server.

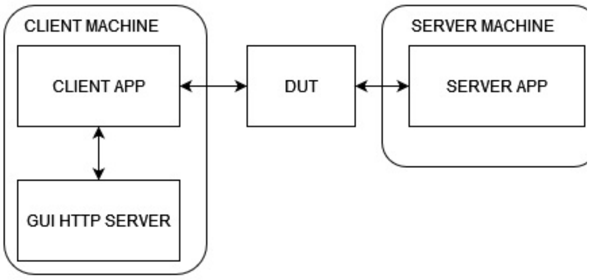

Fig.1. System architecture for network tests

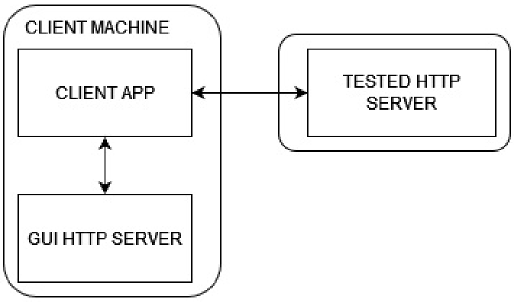

Fig.2. System architecture for network application tests

b. Used technologies

The application is implemented based on the following technologies:

- Kali-Linux – an open-source, Debian-based Linux distribution dedicated mainly to penetration testing, security research, computer forensics and reverse engineering [29].
- Python3 – a high-level object-oriented programming language. It support easy executing programs and scripts implemented in others languages. Python is developed as open-source project. [30]
- Flask – micro web framework written in Python. It does not require particular libraries or tools and by default it does not include a database abstraction layer [31].
- Ostinato – network traffic generator and analyzer. It has user-friendly GUI implemented using Python API. Both, GUI and API, are available to the users. Ostinato supports different protocols of different ISO/OSI layers, it may be used to generate frames of various lengths and contents [32].
- Iperf3 – a tool for measurements of the maximum available bandwidth on IP networks. It is built on a client-server model and supports various protocols (UDP, TCP, SCTP) [33].
- Slowloris.py – a python script performing slowloris attack [34]. Slowloris is a type of DDoS attack that targets HTTP and consists in sending lots of HTTP GET requests into the victim.
- PycURL – a Python interface to libcurl library, allowing to send HTTP requests [35].

c. Application functionality

The application support performing various tests checking application and link performance.

RFC 2544 compliant test

RFC 2544 [1] is a document describing a number of tests used to define the performance characteristics of a network interconnecting devices. According to this document there should be conducted tests measuring the following parameters: throughput, latency, frame loss rate, back-to-back frames, system recovery and reset for different lengths of frames: 64 bytes, 128 bytes, 256 bytes, 512 bytes, 1024 bytes, 1280 bytes and 1518 bytes. Each of the test consist in transmitting frames into the tested device, that should send the received frames back.

Unfortunately, latency measuring systems are mainly based on hardware solutions such as FPGA [9, 13, 14, 37]. Primorac et al. [36] compared FPGA-based solution measuring network latency with software solutions. According to them hardware solutions are much more accurate that the other ones. Authors focused on create an application that would be able to run on simple computer without any additional devices such as FPGA or NTP server, what excludes using also tools such as OWAMP [24] mentioned in section “Software solutions”. RFC 2544 requires measuring using TCP or UDP protocol, what excludes using tools basing on ICMP protocol. Finally SRPerf used in [8] is not an appropriate for proposed application because there is no option to set throughput rate, what is required by RFC 2544 [1]. There was also problem with transmitting big frames. Packets longer than 1516 bytes transmitted by UDP protocol are fragmented.

The described application performs RFC 2544 compliant test using UDP frames. It measure throughput frame loss rate and back-to-back frames for lengths of 64 bytes, 128 bytes, 256 bytes, 512 bytes, 1024 bytes, 1280 bytes and 1516 bytes.

RFC 2544 compliant throughput test

The test consists of a numbers of trials. Each of the trails consists in sending packets at the initial speed for 60 seconds and counting sent and received frames. If counts of sent and received frames are equal, the speed is increased, otherwise it is decreased and the next trial is performed. The algorithm is repeated until found maximum speed without frames loss. The result of the test is the measured speed. The test is performed using iperf3.

RFC 2544 compliant frame loss rate test

The test consists of a numbers of trials. The first trial is executed at a maximum medium speed. Each of the next trials is executed at a speed 10% lower than the previous one. Every trial takes 60 seconds. During each of the trials application counts sent and received frames and after the trial it counts frame loss rate according to the equation 1.

$$\frac{(input\_count - output\_count)}{input\_count} \cdot 100\% \quad (1)$$

The test is executed until two consecutive trials result in frame loss rate equals to 0. The result of the test is a set of measured frame loss rates. The test is performed using iperf3.

RFC 2544 compliant back-to-back frames test

The test consists of a numbers of trials. Each of the trails consist in sending the burst of packets and counting sent and received frames. If counts of sent and received frames are equal, the burst length is increased, otherwise it is decreased and the next trial is performed. The algorithm is repeated until found maximum burst length without frames loss. The algorithm is repeated 50 times and the result of the test is the average value of measured burst lengths. The test is performed using simple client-server application implemented in C language.

Test generated by the user

The presented application has GUI allowing users to generate their own test. Configurable parameters are as follows: protocol of transport layer (TCP or UDP), frame length, payload content, source and destination IP address, count of transmitted frames, transmitting frequency and mode (continuous or burst). The test is implemented using ostinato python API.

Distributed Denial of Service test

One of the tests performing by application consists in performing HTTP DDoS attack called Slowloris. The time of attack is set by the users before the start. The result of the test is the time of unavailability of the examined application compared to the time of attack. The test is performed using the python script with the same name [34].

Test results

As a part of the tests of created application authors performed network performance tests of the following configurations:

a) one switch 1Gbps,
b) two switches 1Gbps connected in series,
c) three switches 1Gbps connected in series.

It was noted that RFC 2544 compliant test takes a long time, so authors decided to try if the same tests with shorter duration give correct results. These test were performed at the same configurations. Each scenario was performed 10 times.

Figure 3. shows the dependence of the throughput on the frame length for different configurations. As expected for all tested configurations throughput increases with the frame length. However for RFC 2544 compliant tests results are similar regardless of the number of switches, while in case of shorter tests differences are significant. That fact may suggest inaccuracy of shortened method. Figures 4 - 6 depict the summary of the results of individual tests and their mean value for different frame lengths. In case of RFC 2544 compliant test dispersion of results is slight, while in the others tests gross errors occur. Frequent occurrence of clearly overestimated results might be caused by too short sampling times.

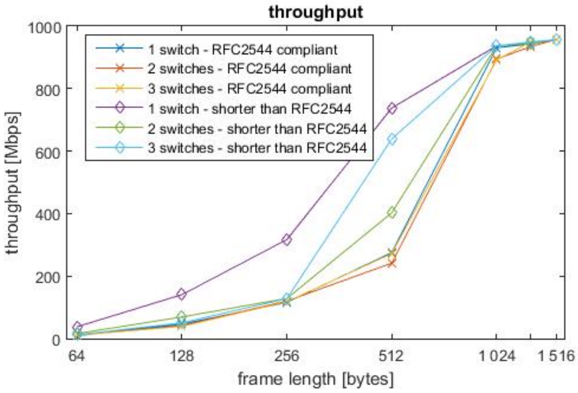

Fig.3. Throughput tests results – average values

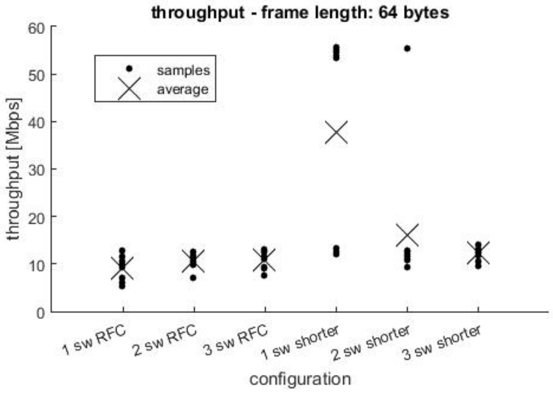

Fig.4. Throughput tests results – comparison of samples and average values for frames of 64 bytes

The result of frame loss rate test is the table with columns for each tested frame length and rows for each percentage value of maximum medium speed. In the cells of the table are numbers of lost packets expressed as a percentage. In the case of properly performed test maximum value should be in the upper left corner of the table and the others values should diminish as it approach to the lower right corner. If there are two zeros next to each other in the given column, the next cells should remain empty. Empty cells should form a triangle in the lower-right corner of the table. Tables 1 - 6 show the result of frame loss rate tests for different scenarios. Cells containing the last iterations of a given test are marked in yellow. Green indicates cells containing values greater than 1% considered to be relatively large. Tables 1 - 3 show the results of RFC 2544 compliant tests. In the each column the end of the test occurs earlier or in the same iteration as in previous column. What interesting often just before the occurrence of no loss occurs there is relatively large value. Tables 4 - 6 show the results of shortened tests. In these cases results are not as expected, what indicates incorrect tests execution.

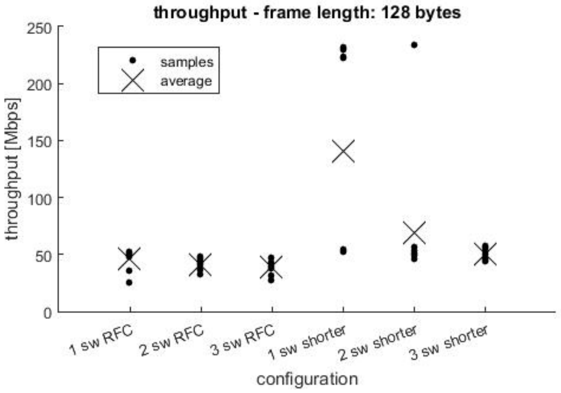

Fig.5. Throughput tests results – comparison of samples and average values for frames of 128 bytes

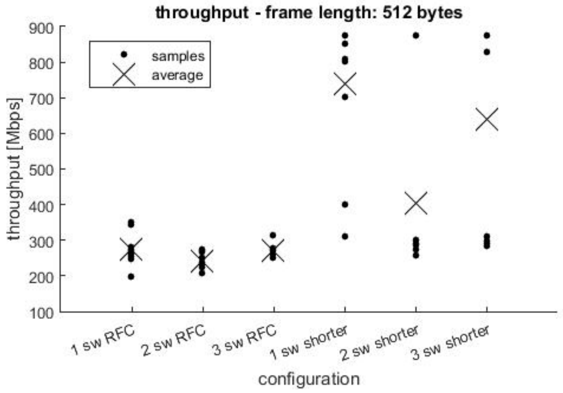

Fig.6. Throughput tests results – comparison of samples and average values for frames of 512 bytes

Table 1. Frame loss rate tests results for 1 switch – duration RFC 2544 compliant

	64	128	256	512	1024	1280	1516
100%	0.7	0,41	0,43	0,014	0	0	0
90%	0.67	0,084	0,37	4,4 e-3	0	0	0
80%	1,1	0,077	0,9	0,012	-	-	-
70%	0,76	0,51	0,37	0,017	-	-	-
60%	1,3	0,42	0,74	0,012	-	-	-
50%	0,6	0,03	0,36	0,59	-	-	-
40%	0,36	0,055	0,76	1,6	-	-	-
30%	0,44	0,085	11	0	-	-	-
20%	0,4	0,013	1,7	0	-	-	-
10%	0,42	0,44	0	-	-	-	-

Table 2. Frame loss rate tests results for 2 switches – duration RFC 2544 compliant

	64	128	256	512	1024	1280	1516
100%	0,17	0,3	0,27	0,025	4,2 e-5	0	0
90%	0,41	0,44	0,44	0,018	0	0	0
80%	0,3	0,42	0,44	0,023	0	-	-
70%	0,37	0,025	0,47	5,5 e-3	-	-	-
60%	0,24	0,037	0,096	0,034	-	-	-
50%	0,049	0,85	0,041	0,49	-	-	-
40%	2,6 e-3	0,051	0,28	1,5	-	-	-
30%	0,039	0,45	11	1,1 e-3	-	-	-
20%	0,064	0,41	4,8	0	-	-	-
10%	0,071	5,3	0	0	-	-	-

Table 3. Frame loss rate tests results for 3 switches – duration RFC 2544 compliant

	64	128	256	512	1024	1280	1516
100%	0,42	0,99	0,072	0,028	0	0	0
90%	0,086	0,58	0,43	8,2 e-3	1,3 e-4	0	0
80%	0,93	0,42	0,43	5,6 e-3	1,3 e-4	-	-
70%	1,3	0,069	0,3	0,019	0	-	-
60%	0,59	0,034	0,053	0,029	0	-	-
50%	0,07	0,2	0,66	0,3	-	-	-
40%	0,05	0,2	0,04	0,006	-	-	-
30%	0,8	0,17	19	0	-	-	-
20%	0,045	0,4	4,5	0	-	-	-
10%	0,13	4,4	0	-	-	-	-

Table 4. Frame loss rate tests results for 1 switch – duration shorter than RFC 2544

	64	128	256	512	1024	1280	1516
100%	0,43	0	0	0,055	0	0	0
90%	0	0	3,8 e-4	0	0	0	0
80%	2,4	-	2,5	0	-	-	-
70%	0	-	2,3	-	-	-	-
60%	0	-	2,3	-	-	-	-
50%	-	-	0,035	-	-	-	-
40%	-	-	9,8 e-3	-	-	-	-
30%	-	-	7,6	-	-	-	-
20%	-	-	1,1	-	-	-	-
10%	-	-	0	-	-	-	-

Table 5. Frame loss rate tests results for 2 switches – duration shorter than RFC 2544

	64	128	256	512	1024	1280	1516
100%	9,7 e-3	2,2 e-3	5,8 e-5	0,026	0	0	0
90%	0	5,8 e-4	0,13	0	0	0	0
80%	2,3 e-4	0	0,052	0	-	-	-
70%	1,6 e-3	1,4 e-4	0,15	-	-	-	-
60%	9,1 e-4	0,061	2,4	-	-	-	-
50%	8,7 e-5	1,5 e-3	0,051	-	-	-	-
40%	3,8 e-4	2,3 e-4	0,034	-	-	-	-
30%	0	0	13	-	-	-	-
20%	0	0	1,4	-	-	-	-
10%	-	-	0	-	-	-	-

Table 6. Frame loss rate tests results for 2 switches – duration shorter than RFC 2544

	64	128	256	512	1024	1280	1516
100%	0,18	0,036	0,4	0,16	0	0	0
90%	2,6	0,043	4,8 e-3	0	0	0	0
80%	0	5,5 e-4	2,3	0,013	-	-	-
70%	0	2,2	7,9 e-3	1,7 e-3	-	-	-
60%	-	0,29	2,6	0,015	-	-	-
50%	-	1,6 e-3	0,17	0,024	-	-	-
40%	-	1,7	0,014	0,027	-	-	-
30%	-	0,73	13	0,04	-	-	-
20%	-	2,3	2,3	0	-	-	-
10%	-	6,1	0	0	-	-	-

Figure 7. shows the average results of the back-to-back tests for different configurations. All charts have the same shape, however results of RFC 2544 compliant tests are larger than in the others cases. Figures 8 - 10 depict the summary of the results of individual tests and their mean value for different frame lengths. Similar to throughput RFC 2544 compliant test dispersion of results is slight, while in the others tests gross errors occur. Besides shorter tests returned smaller results.

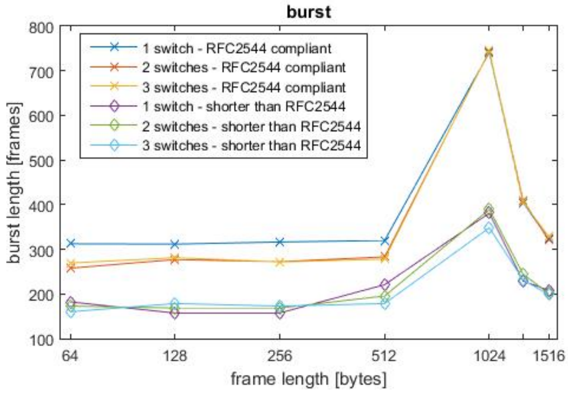

Fig.7. Back-to-back tests results – average values

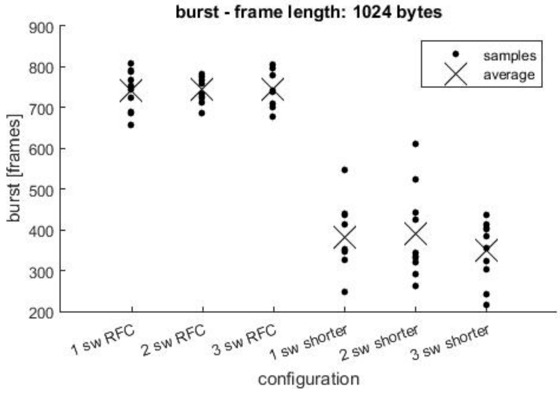

Fig.8. Back-to-back tests results – comparison of samples and average values for frames of 1024 bytes

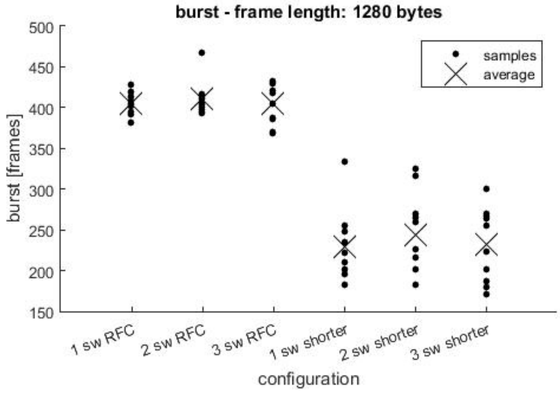

Fig.9. Back-to-back tests results – comparison of samples and average values for frames of 1280 bytes

Conclusion

The problem of testing network and network applications performance and security is very important issue nowadays. Many tools have been created and are still being developed to test these parameters. Many of them are expensive and require specialized equipment. This paper was intended to describe the new application connecting the ability of testing performance and security without large hardware requirements. Besides it was examined legitimacy of long tests duration imposed by RFC 2544 [1]. Performing the same tests described by this document but with shorter durations did not returned repeatable and compatible with full size RFC 2544 tests. In the future work that application will be developed. Especially it is planned to implement a method for inexpensive and accurate latency measurement in accordance with RFC 2544.

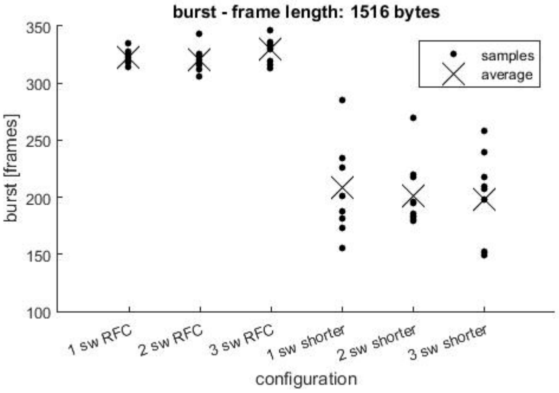

Fig.10. Back-to-back tests results – comparison of samples and average values for frames of 1516 bytes

Authors

M. Sc. Alicja Starczewska, Ph.D. Krzysztof Daniec, Ph.D. Jarosław Homa, Prof. Aleksander Nawrat, Institute of Automatic Control and Robotics, Faculty of Automatic Control, Computer Science and Electronics, Silesian University of Technology, ul. Akademicka 16, 44-100 Gliwice, Poland email: RAU3-KAiR@polsl.pl

REFERENCES

[1] Bradner S., McQuaid J., Benchmarking methodology for network interconnect devices. No. rfc2544. 1999.

[2] Bradner S., Benchmarking terminology for network interconnection devices. No. rfc1242. 1991.

[3] Waheed F., Azam F., Anwar M. W., Rasheed Y., Model Driven Approach for Automatic Script Generation in Stress Testing of Web Applications. Proceedings of the 2020 6th International Conference on Computer and Technology Applications. 2020.

[4] Hussain T., An approach to evaluate the performance of web application systems. Proceedings of International Conference on Information Integration andWeb-Based Applications & Services. 2013.

[5] Park T., Shin S., Shin I., Lee K., Formullar: An FPGA-based network testing tool for flexible and precise measurement of ultra-low latency networking systems. Computer Networks 185 (2021): 107689.

[6] Loiacono J., Germain A., Smith J., Network performance measurements for NASA's Earth Observation System. Computer networks 46.3 (2004): 299-320.

[7] Goenka P., Zarifis K., Gupta A., Calder M., Towards client-side active measurements without application control. ACM SIGCOMM Computer Communication Review 52.1 (2022): 20-27.

[8] Mayer A., Loreti P., Bracciale L., Lungaroni P., Salsano S., Filsfils C., Performance monitoring with h^2: Hybrid kernel/ebpf data plane for srv6 based hybrid sdn. Computer Networks 185 (2021): 107705.

[9] Mendiola A., Fuentes V., Matias J., Astorga J., Toledo N., Jacob E., Huarte M., An architecture for dynamic QoS management at Layer 2 for DOCSIS access networks using Open- Flow. Computer Networks 94 (2016): 112-128.

[10] Yang Y., Jiang H., Zhang G., Wang X., Lv Y., Li X., Fdida S., Xie G., S2H: Hypervisor as a setter within Virtualized Network I/O for VM isolation on cloud platform. Computer Networks 201 (2021): 108577.

[11] Huynh M., Goose S., Mohapatra P., Resilience technologies in Ethernet. Computer Networks 54.1 (2010): 57-78.

[12] Lifu F., Dongming Y., Bihua T., Yuanan L., Hefei H., Technique for network performance measurement based on RFC 2544. 2012 Fourth International Conference on Computational Intelligence and Communication Networks. IEEE, 2012.

[13] Özcan A., Yalçın M. E., RFC 2544 Ethernet Performance Measurements Using FPGA Based Dual-Core ARM. 2021 13th International Conference on Electrical and Electronics Engineering (ELECO). IEEE, 2021.

[14] Wang Y., Liu Y., Tao X., He Q., An FPGA-based high-speed network performance measurement for RFC 2544. EURASIP Journal on Wireless Communications and Networking 2015 (2015): 1-10.

[15] Pradeep S., Sharma Y. K., A pragmatic evaluation of stress and performance testing technologies for web based applications. 2019 Amity International Conference on Artificial Intelligence (AICAI). IEEE, 2019.

[16] Locust. An open source load testing tool [web page] https://locust.io/. [Accessed on 16 Feb. 2023.].

[17] Apache JMeter [web page] https://jmeter.apache.org/. [Accessed on 16 Feb. 2023.].

[18] Hulk DoS tool [web page] https://github.com/grafov/hulk/. [Accessed on 16 Feb. 2023.].

[19] The grinder [web page] http://grinder.sourceforge.net/. [Accessed on 16 Feb. 2023.].

[20] Capybara [web page] https://github.com/teamcapybara/capybara. [Accessed on 16 Feb. 2023.].

[21] TestMatick quality is never too mach. Pylot [web page] https://testmatick.com/testing-tools/pylot/. [Accessed on 16 Feb. 2023.].

[22] A distributed Load Testing Tsung [web page] http://tsung.erlang-projects.org/1/01/about/. [Accessed on 16 Feb. 2023.].

[23] SoapUI Supported by SMARTBEAR [web page] https://www.soapui.org/. [Accessed on 16 Feb. 2023.].

[24] One-Way Ping (OWAMP) [web page] https://software.internet2.edu/owamp/. [Accessed on 16 Feb. 2023.].

[25] Aukua Systems. Ethernet Inline Protocol Analyzer. MGA2510 Product Brief [web page] https://www.aukua.com/docs/AukuaInlineAnalyzerPB_08021BW.[Accessed on 16 Feb. 2023.].

[26] Aukua Systems. Ethernet Traffic Generator and Analyzer. XGA4250 Product Brief [web page] https://www.aukua.com/docs/AukuaXGA4250_GeneratorPB_05021BW.[Accessed on 16 Feb. 2023.].

[27] GL Communications Inc. Multi-Functional Ethernet/ IP Tester – PacketExpertTM 10GX [web page] https://www.gl.com/Brochures/Brochures/PacketExpert-10GX-[Accessed on 16 Feb. 2023.].

[28] GL Communications Inc. PacketExpertTM – RFC 2544 Testing [web page] https://www.gl.com/Brochures/Brochures/PacketExpert-10GX-[Accessed on 16 Feb. 2023.].

[29] Kali. The most advanced Penetration Testing Distribution [web page] https://www.kali.org/. [Accessed on 16 Feb. 2023.].

[30] Python [web page] https://www.python.org. [Accessed on 16 Feb. 2023.].

[31] Flask, web development, one drop at a time [web page] https://flask.palletsprojects.com/. [Accessed on 16 Feb. 2023.].

[32] Ostinato. Traffic Generator for Network Engineers [web page] https://ostinato.org/. [Accessed on 16 Feb. 2023.].

[33] iPerf – The ultimate speed test tool for TCP, UDP and SCTP [web page] https://iperf.fr/. [Accessed on 16 Feb. 2023.].

[34] slowloris.py – Simple slowloris in Python [web page] https://github.com/gkbrk/slowloris/. [Accessed on 16 Feb. 2023.].

[35] Pycurl [web page] http://pycurl.io/. [Accessed on 16 Feb. 2023.].

[36] Primorac M., Bugnion E., Argyraki K., How to measure the killer microsecond. ACM SIGCOMM Computer Communication Review 47.5 (2017): 61-66.