electronics

MDPI

Article

A Cost-Efficient Software Based Router and Traffic Generator for Simulation and Testing of IP Network

Su Jun [1], Krzysztof Przystupa [2,*], Mykola Beshley [3], Orest Kochan [1,3], Halyna Beshley [3], Mykhailo Klymash [3], Jinfei Wang [4] and Daniel Pieniak [5]

1 School of Computer Science, Hubei University of Technology, Nanli Road, 28, Hong-shan District, Wuhan 430068, China; sjhosix@gmail.com (S.J.); orestvk@gmail.com (O.K.)
2 Department of Automation, Lublin University of Technology, Nadbystrzycka 36, 20-618 Lublin, Poland
3 Department of Telecommunications, Lviv Polytechnic National University, Bandery 12, 79013 Lviv, Ukraine; mykola.i.beshlei@lpnu.ua (M.B.); halink@ukr.net (H.B.); mklimash@polynet.lviv.ua (M.K.)
4 School of Mechanical Engineering, Northwestern Polytechnical University, 127 Youyi Ave. West, Xi'an 710072, China; wjfwenxn@sina.com
5 Department of Mechanics and Machine Building, University of Economics and Innovations in Lublin, Projektowa 4, 20-209 Lublin, Poland; daniel.pieniak@wsei.lublin.pl
* Correspondence: k.przystupa@pollub.pl

Received: 13 November 2019; Accepted: 24 December 2019; Published: 27 December 2019

Abstract: The development was carried out using the Qt5.2 integrated development environment, which uses the programming language C++. The main advantage of this environment is that the code written in it can be compiled to different platforms (for example, Windows, Linux, Mac OS). A software router based on a modular architecture has been developed. It uses the socket technology, which allows forming a program-oriented packet network with any topology, including full-coupled topology. A network traffic generator to test the developed software router has been designed. We proposed a scheme to measure the packet processing time of a router using a specialized packet-capture network interface cards (NIC 1 and NIC 2) and a novel traffic generator installed on PC. Based on an experimental test bed we confirmed that our software router provides a cost-efficient alternative to the expensive, special hardware router CISCO 2801.

Keywords: software-based router; hardware router; packet delay; network traffic generator; packets processing time

1. Introduction

Local and global computer networks are the basis of the communication infrastructure of modern society. Due to the growth of networks, the problem of choosing the optimal network equipment (routers, switches) is becoming more and more acute [1]. The central element of the information network is the router. The main purpose of which is to unite the subnetwork so that any computer can exchange packets with other computers in the network.

The router can be implemented fully by a software approach (in this case, it is an operating system module installed on a general-purpose computer that operates as a server) or a hardware-software method (which is a specialized computing device, in which some functions are performed by non-standard equipment, and some of the software modules that work under the specialized operating system). The main advantages of software routers over hardware ones are flexibility, intelligence and simplicity of algorithms modification [2]. It is possible to implement the most non-standard network solutions on the basis of a software router. The majority of software routers operate under Linux. That allows providing high productivity and flexibility of a configuration at realization of routing, processing of the network traffic arriving on physical port of the router [3].

At present, specific and expensive hardware platforms offered by suppliers and vendors are used to solve the problems of high-speed routing. However, we propose a software router, which has a much lower cost and comparable computing power to multi-core general purpose platforms. It will allow achieving similar performance characteristics using purely software mechanisms. There will be additional features such as incomparably greater flexibility and almost unlimited possibilities to increase functionality, integrate all new services, as well as to carry out adaptation and fine-tuning (customization) to the specific tasks of each user.

Today, all types of global IP traffic, communication performance, user numbers and the number of connected devices are predicted to grow significantly. These growth rates lead to an increase in the load on the network infrastructure, namely several types of traffic flows traverse the network equipment (switch and router), thus Quality of Service (QoS) testing is required. Network traffic generators help network administrators, developers and researchers to prepare, test and deploy technologies to ensure reliable and quality network infrastructure [4]. Traffic generators are classified into hardware or software based on different performance criteria. Hardware-based traffic generators (e.g., Ixia IxChariot) typically achieve higher performance and accuracy than software-based tools, which depend on many factors (endpoint performance, operating system, etc.). On the other hand, hardware devices are usually commercial products, while software tools are usually open source or cost-effective tools developed by researchers [5]. Despite these arguments, software network traffic generators are widely used in corporate networks due to their flexibility, simplicity and cost effectiveness [6].

For this reason, the paper proposes its own software traffic generator, which, unlike the known ones, can estimate the average delay in servicing individual flows, current delays, loss and jitter of packets transmitted through the network infrastructure. The research will be carried out using this generator to compare the developed router with the Cisco 2801 hardware one.

This paper is organized as follows: Section 2 describes the related research work of the satellite software-based router, network simulator, software traffic generators and the problems of model adequacy assessment of infocommunications system. Then, Section 3 introduces the performance analysis of the proposed software-based router using the designed network traffic generator. Section 4 describes the method to measure and compare packet processing time of the software router with the Cisco 2801prototype hardware router, verifies the proposed solution, including the experimental test bed. Finally, Section 5 concludes this work.

2. Related Work

2.1. Related Research on Software-Based Router

In this part, we illustrate the status of research development of software-based routers and software traffic generators with packet delay monitoring. Today, the latest technological advances provide an opportunity to do something truly effective in the area of open Internet devices, sometimes called open routers (OR) based on software solution. With regard to the current state of the software router, a number of initiatives have been taken over the past few years to develop and research the software-based router and related topics [7–9]. In the field of software, one of the most important initiatives is the Click Modular Router Project [10], which offers an effective solution for building a data plane. Authors [11] consider the inefficiency of kernel-level packet processing inside modern OS-based software routers and explores whether a redesign of kernel network stacks can improve the incompetence. They proved that the proposed Kafe neither adds any new API nor depends on proprietary hardware features, but the Kafe outperforms Linux by seven times and RouteBricks by three times. For the speed-up in the routing table lookup in software routers, [12] introduced a new data structure called sTable that achieves space efficiency while causing low processing overhead without hardware parallelism. In [13] a performance analysis of an OR architecture enhanced with Field Programmable Gate Arrays (FPGA) line cards, which allows direct network interface card to network interface card (NIC-to-NIC) packet forwarding, is introduced. In [14] the virtualization of

a multiservice OR architecture is discussed: the authors propose multiple Click forwarding chains virtualized with Xen. The authors of [15] proposed an in-depth study of the IP lookup mechanism included in the Linux kernel.

2.2. Related Research on Network Simulator and Software Traffic Generators

Performance measurement and simulation are important approaches for identifying bottlenecks of such systems to predict and improve their performance [16]. Today there is a large number of software tools that allow one to conduct modeling of individual network devices and the whole network as a whole. The main means of modeling, which are used by scientists around the world to test their hypotheses and developments, are: Network Simulator (NS) [17], OPNET [18], NetSim [19], OmNET++ [20]. The listed modeling tools make it possible to investigate the functioning parameters of network nodes, systems, protocols and allow introducing own changes in the configuration of the model of some devices, allows conducting research of own algorithms or protocols developed by scientists. However the listed means are based on a principle of modelling of discrete events [21]. Nevertheless, a significant disadvantage of these tools is that they use statistical methods and analytical dependencies to calculate the state of the system at a certain point in time [22]. Thus, an hour of work of a real network can be simulated within tens of seconds that is not effective when modelling is carried out in real time, for example, modelling of algorithms of work with memory of the network device, formation and service of queues of packages in the router.

The authors of [23] offer a solution for traffic modelling in NS-2 network simulator, and whereas [24] is engaged in traffic modelling, the authors of [25] offer a solution for realistic generation of HTTP traffic. Articles [26–28] analyze the production network traffic behaviour.

According to the authors of [26], network traffic can be generated in three ways:

- stochastic generation,
- replication of production network traffic,
- using list of instructions (communication scenario) for applications in the tested network.

The authors of [27] divide network generators according to the layer on which they work:

- Application-level traffic generators: they emulate the behaviour of specific network applications in terms of the traffic they produce.
- Flow-level traffic generators: they are used when the replication of a realistic traffic is requested only at the flow level (e.g., number of packets and bytes transferred, flow duration). For example, Bit-Twist [28] is representative of this group.
- Packet-level traffic generators: with this term we refer to generators based on packet's Inter departure time (IDT) and packet size (PS). The size of each packet sent, as well as the time elapsed between subsequent packets, are chosen by the user, typically by setting a statistical distribution for both variables. Most current packet generators belong to this group.

In a thesis [29], the author examines the software traffic generators Iperf, Mausezahn, Ostinato in a closed loop physical and virtual environment to evaluate the applicability of the tools and find sources of inaccuracy for a given traffic profile. One can easily find comparisons of some popular tools, such as in [30–35].

Despite the fact that many tools for the traffic generation are available, none of the reviewed tools fully suits all the requirements on network traffic generator required for network experiments. None of them has not the ability to control the generated stream without synchronizing the input and output interfaces and determine the current delay of packets in real time.

2.3. Related Research on Problems of Model Adequacy Assessment and Experimental Investigations of Infocommunications System

In the process of building an information and telecommunication system model, some important dependencies may be missed and may not be included in it. In this case, the model will be inadequate,

i.e., the behaviour of the model for these input parameters will not correspond to the behaviour of the real system. It is clear that an inadequate model is practically inexpedient. If a quality indicator can be measured, the adequacy of the model can be checked by comparing the real and model values of the indicator [36].

If the difference between them exceeds the acceptable limit, it indicates the inadequacy of the model. To evaluate the quality of a system, such a simple check of the model adequacy is not correct. One of the ways to solve the adequacy problem is to build more detailed model. However, this can lead to the inclusion in the model a large number of parameters and the relationships between them, so that the overall model will be difficult to inspect. Analysis of such a model is very complex, and simulation requires a lot of machine time. As a result, the model will not provide additional knowledge about the system, as significant links in it will be lost among the secondary ones.

Therefore, as noted above, it is advisable to have several models of different levels of detail for the same computer system, each of which can be used for different purposes. The ideal model should contain exactly as much detail as is necessary for the purpose of its construction.

In order to practically use the system models, information on the real course of the information-computing process is required in many cases. Such information is also necessary in order to evaluate the quality of design solutions used during the creation of computing devices and development of mathematical support with a greater degree of probability, as well as to solve the problems connected with adjustment of an operating system according to concrete conditions of operation [37,38]. The necessary information is collected with the help of special means, providing the measurement of parameters, characterizing the dynamics of the system functioning in the normal operation mode [39–41].

The main problems that arise during measurement and need to be solved can be classified as follows:

- Conducting a meaningful analysis of the system under study and the specific conditions of its operation;
- Building a simulation model of the system functioning process, which should reflect all those events in the information-computing process, which cause the change of measured parameters;
- Development of measurement algorithms for selected parameters of the system functioning process on the basis of simulation model;
- Performance of measurement algorithms in the system under study with the help of appropriate hardware and software measuring instruments.

In some cases, when, for example, the performance of the conveyor processor is estimated, the model can be clearly defined from the substantial analysis of the system under study. In a more complex situation, when it comes to the analysis of the system, the construction of simulation models requires special consideration.

Measurement results should be presented in a form suitable for further analysis using special software and hardware processing tools. The measurement process should be connected both with the hardware-software measuring instruments and with its monitoring. In particular, it concerns the choice of common data formats convenient not only for conducting measurements, but also for processing their results. If the type of processing is fixed in advance and does not require complex calculations, it can be carried out during the measurement process. In general, the measurement phase usually precedes the processing phase and the hardware and software can be used effectively to process large amounts of information. In these cases, there is a very high density of recorded data.

At the end of the experimental studies, the results of the measurements are analyzed, and this allows drawing meaningful conclusions about the system under investigation. An important condition for forming such conclusions is the successful representation of these results. It is reasonable to choose the form of their presentation taking into account the specific task of the study and the quality indicators used. Another aspect of the analysis of measurement results relates to ensuring the reliability of the

formulated conclusions [42–46]. For example, when the parameters are registered periodically, without reference to the events in the computer system that cause their change, there is a problem of analyzing the statistical reliability of the obtained data. A similar problem arises when measurements are made within a short time frame. However, when the method of measuring parameters takes into account all the events in the system that cause changes in these parameters, the observation interval can be quite large and the problem of reliability of measurement results virtually does not arise.

3. Development of Software-Based Router and Traffic Generator

Nowadays the majority of infocommunication networks are constructed on the basis of the proprietary equipment whose functionality is realized by hardware means, which demands specialised knowledge from the system administrator and is closed for modification of functioning of a network directed to the requirements of users. Each addition or change of functions by the system administrator in the network infrastructure, as a rule, leads to difficult deployment tasks, which must be carefully planned in advance. Transition from traditional hardware-based to software-based networks will allow a qualitative leap in terms of flexibility, productivity and production and will become a new solution for future info-communication networks in the telecommunications market [47,48].

For this reason, a software router and software traffic generator have been developed in this chapter. The development was carried out using the Qt5.2 integrated development environment, which uses the programming language C++ (standard C++11, 2011). The main advantage of this environment is that the code written in it can be compiled to different platforms (for example, Windows, Linux, Mac OS).

3.1. Development of Software-Based Router

In this work the developed model of the software router is presented, which allows estimating the traffic created by service messages, time of staying of a packet in the node, number of packets in the buffer, probability of loss of packets, jitter, adequacy of routing tables in nodes, establishing optimal structural and functional parameters of a node and parameters of a routing protocol [49].

The model of the software router has the following features:

- Each component of the model is implemented in the form of a separate application and has its own IP-address and port, which allows you to select and connect components depending on the specific conditions;
- The simulation model of the router can be as close as possible to the specific model of the manufacturer;
- Distribution of components of the data network model is possible both on one and on several computers (servers) for the purpose of maximum approximation to real conditions and modelling networks with an unlimited number of nodes, not limiting the resources of one computer.

The router software is based on socket technology, which is the software object of the operating system and consists of the IP address of the device and the TCP port. Using the API of the operating system, the software router receives the generated socket object and uses it to communicate with other software routers that are installed on other physical machines in the local network.

In the process of developing the router model, the modular structure of the program in the form of a tree is first built [50,51]. Then the program modules are programmed alternately, starting with the modules of the lowest level (leaves of the tree of the modular structure of the program), in such an order that for each program module all modules to which it can address are already programmed. After programming all the modules of the program, the individual modules are tested and debugged in the same (ascending) order as their programming. This order of program development seems quite natural at first glance: each module is expressed through already programmed directly subordinate modules, and during testing it uses already debugged modules.

To describe the structure of such a system the diagram of classes is used (Figure 1). This is one of the main ways of description, because Unified Modeling Language (UML) is first of all an object-oriented language, and classes are the main (if not the only) "building material". Dashed arrows represent dependencies between elements, in particular on the presented diagram those elements that have the arrow pointing at them depend on the elements from which the arrows are outgoing. White diamond arrows show aggregation, elements that have diamond pointed at them, aggregate those elements from which the arrow is outgoing. When an element aggregates other elements it means that the element doesn't much depend on the number of contained elements and will function properly even with a single inner element. Black diamond arrow shows composition. The difference of composition from aggregation is that element strongly depends on the number of contained elements and without inner elements will not be able to operate normally, so inner elements can be considered like composite parts of the overall element.

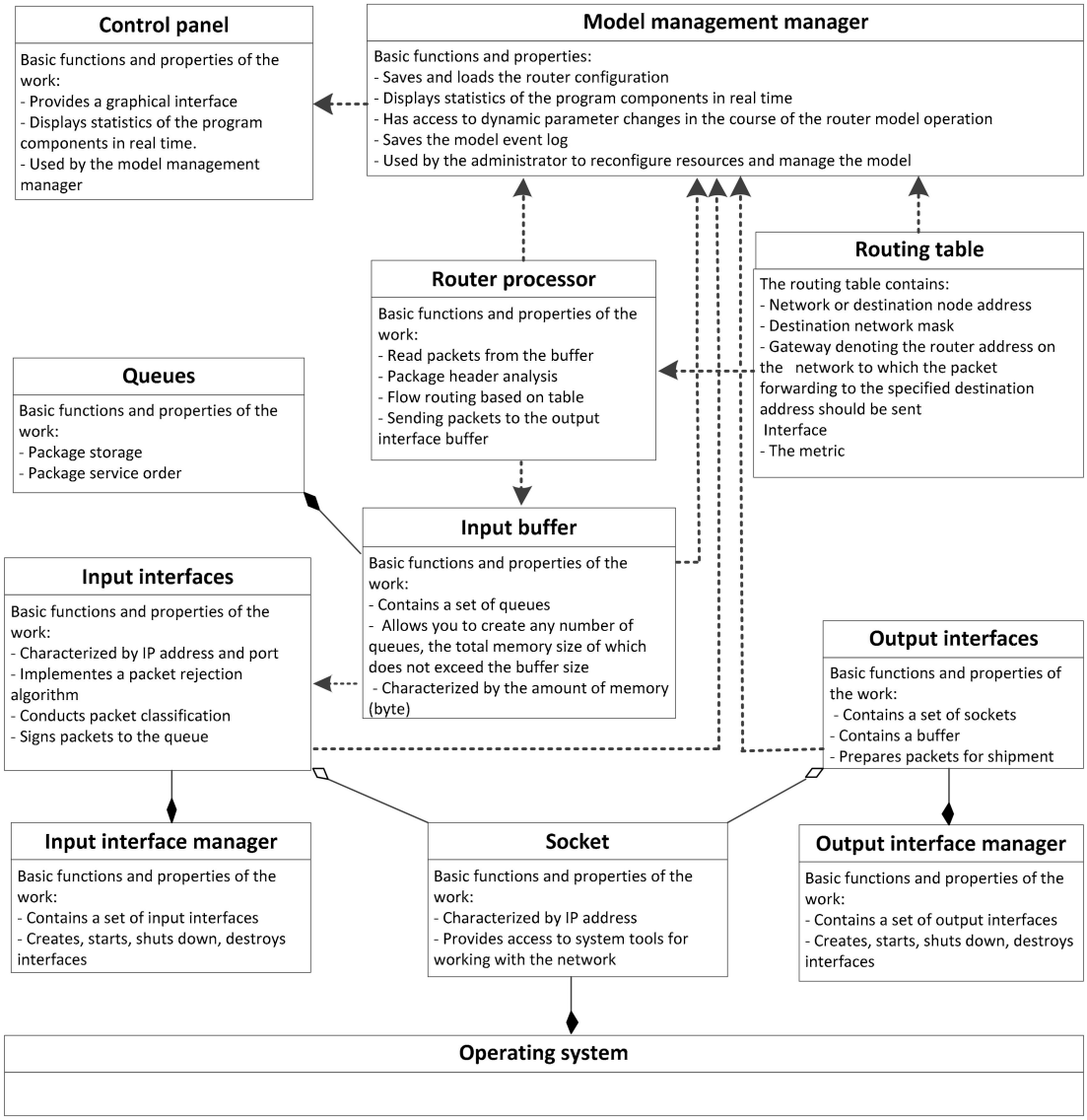

Figure 1. UML diagram of software-based router model.

The router model is represented by a set of queues and service devices (Figure 2) and reflects the process of packet transmission as follows:

(1) Aggregated multiservice traffic arrives at the ports of incoming interfaces.

(2) Packets of streams arrive in the input queue
(3) The processor of the router on the basis of the classifier, taking into account the discipline of queuing, selects the packet and analyzes its header type of service (ToS), differentiated services code point(DSCP);
(4) The processor which realizes the report of routing, defines a direction for message transfer and supports an urgency of tables of routing by an exchange of service packages with other knots (at construction of a network from several program routers)
(5) After processing the packet header and selecting the output interface port, the packet is queued in the output channel waiting list.
(6) The next step is to transfer the packets to another network device defined by the routing protocol.

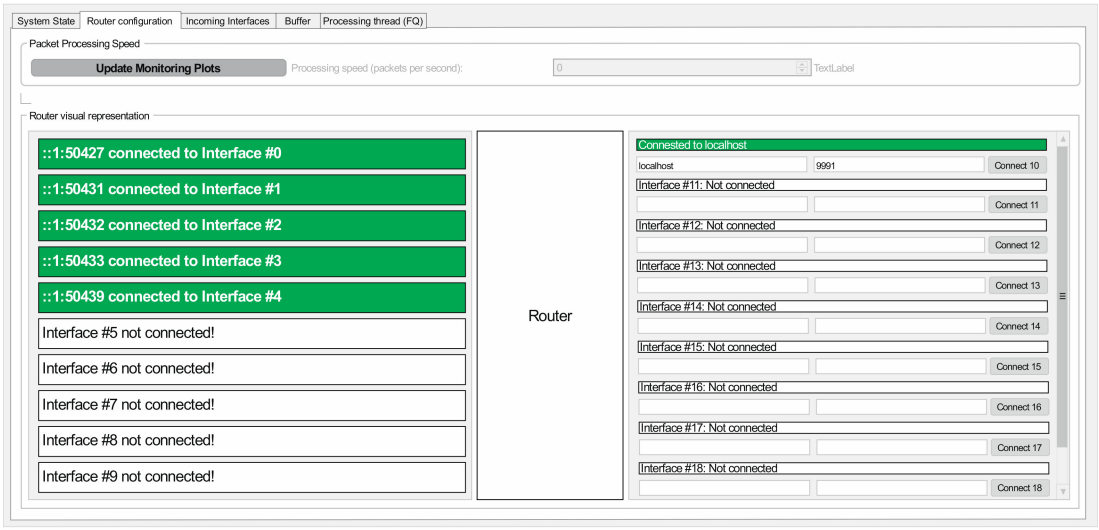

Figure 2. Software-based router model with modular structure.

The designed software router has the following main functions: transfer of information packets between different subnetworks, support of static and dynamic routing protocols, filtering of information packets at the channel, network and transport layers of the open systems interconnection (OSI) model, traffic prioritization according to the requirements.

3.2. Performance Analysis of The Software-Based Router Using the Designed Network Traffic Generator

In order to deploy the test network and obtain adequate simulation results, the developed software components (the router and generator) must first be tested. The maximum performance of these models is tested in accordance with RFC 2544 recommendation [52]. This recommendation was developed by the Network Working Group, created within the framework of the open international community of designers, scientists, network operators and Internet Engineering Task Force (IETF) providers. On the basis of this document, specialists from the Ethernet/IP working group (ODVA) developed a detailed description of testing procedures for network devices operating on the first and second layers of the TCP/IP model, and a somewhat extended set of tests. In the case where the study is conducted not on a model, but on a real network, it is necessary to use software that will act as a generator and receiver of traffic to test performance. Such software can simulate the behavior of an individual user as well as a group of users. In the latter case, as the number of users increases, the traffic characteristics deteriorate, as the program processes only one user at a time and the operating system protocol stack delays the transmission of packets. These factors will affect the time parameters of packet transmission, degrade the quality and accuracy of the obtained results of the study of the quality of service parameters of multiservice flows in general. This method can be used to investigate traditional IP networks, as the

destination host address and service type are important when routing packets. It is this information that is used by routing protocols when routing and most methods of optimizing the process of data transmission and load balancing in the network. The main advantage of this method is the ability to generate large amounts of traffic of different classes (the maximum flow rate depends on the maximum speed of the network card of the server on which the generator is installed). A traffic generator has been designed to test the developed software router. UML-diagram of the developed packet generator is shown in Figure 3.

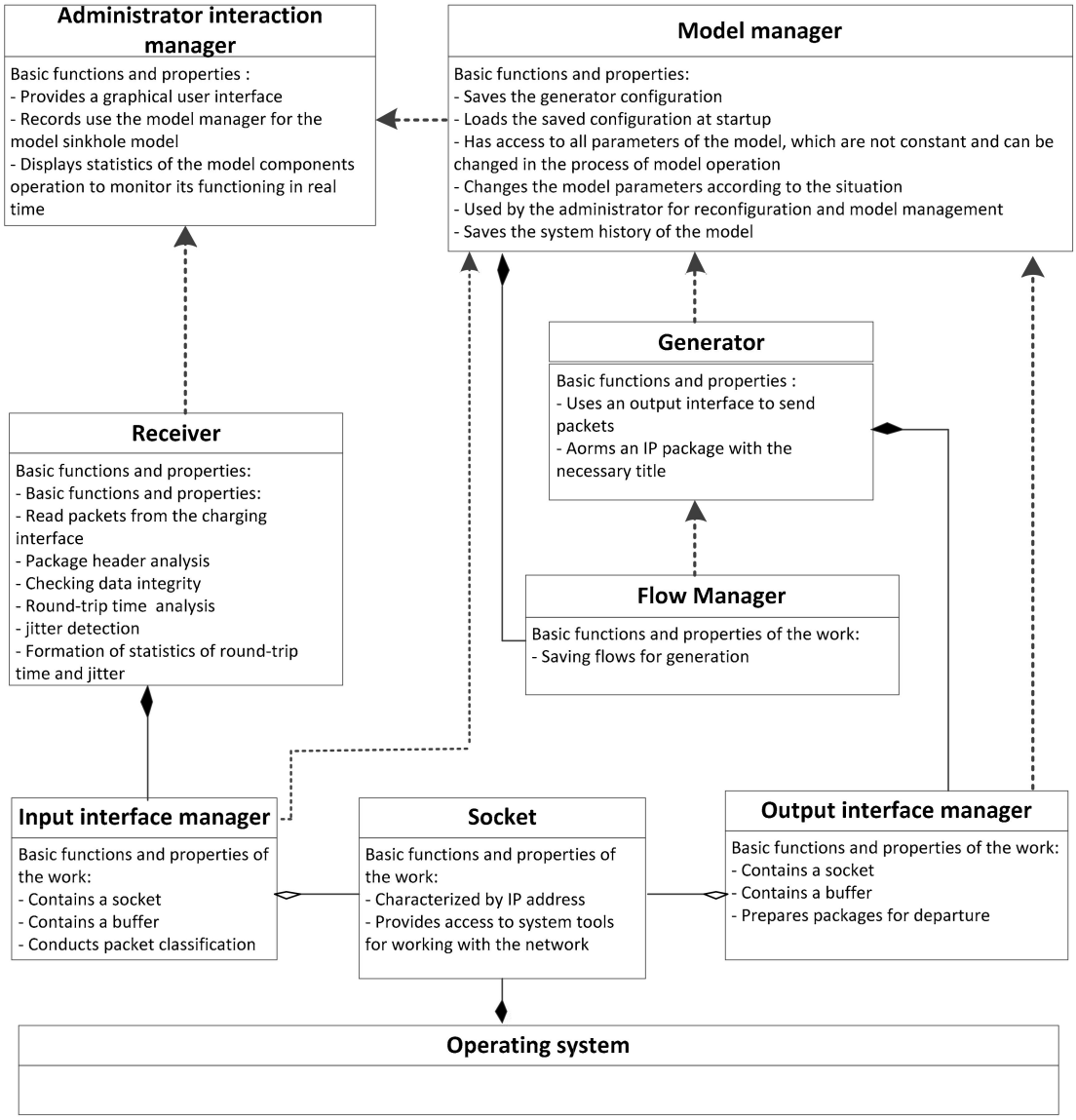

Figure 3. UML diagram of the traffic generator.

With this generator it is possible to create a data stream with different packet sizes. These packets can be transmitted both by UDP protocol and TCP. Using the Wireshark program, the characteristics of real VoIP, video on demand, IPTV and internet data service streams (packets size and their transmission intensity per second) are determined and researched. Using the developed generator it is possible to reproduce any type of service. It is also possible to generate different flows simultaneously, which will allow to generate multiservice traffic (Figure 4).

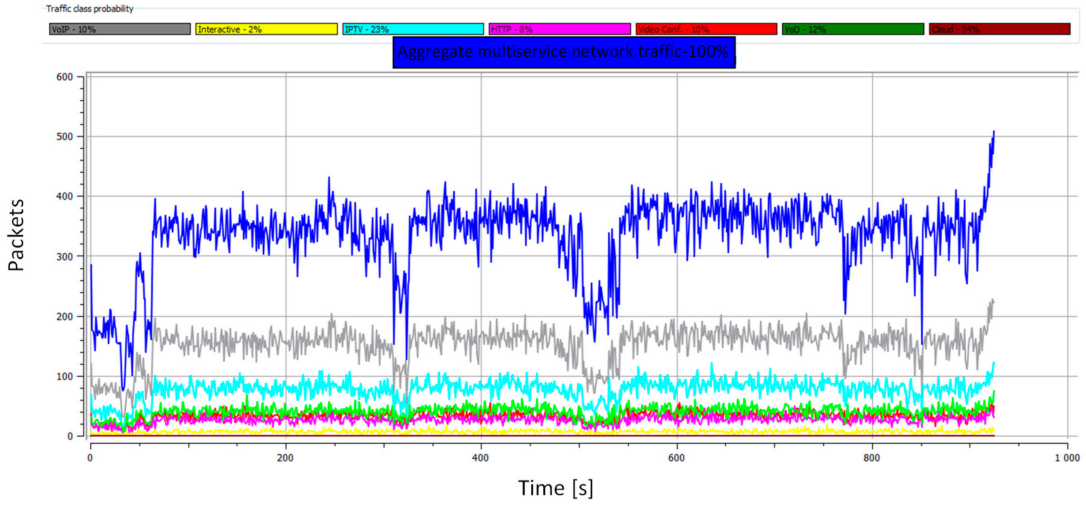

Figure 4. Multiservice traffic generation.

To get adequate results with the software router models it is necessary to check the quality of the traffic generator. This check is to determine the measurement error of the inter-packet interval when generating packets. The smaller the standard deviation of the inter-packet interval values, the smaller the error and greater adequacy of the simulation results. Investigation of the generator quality is carried out using a series of tests. Each series differs in the size of the inter-packet interval and the size of the package itself.

Figure 5 shows a graph of inter-packet interval values for 100,000 packets. The inter-packet interval is 10 ms, the size of the packet is 1500 bytes.

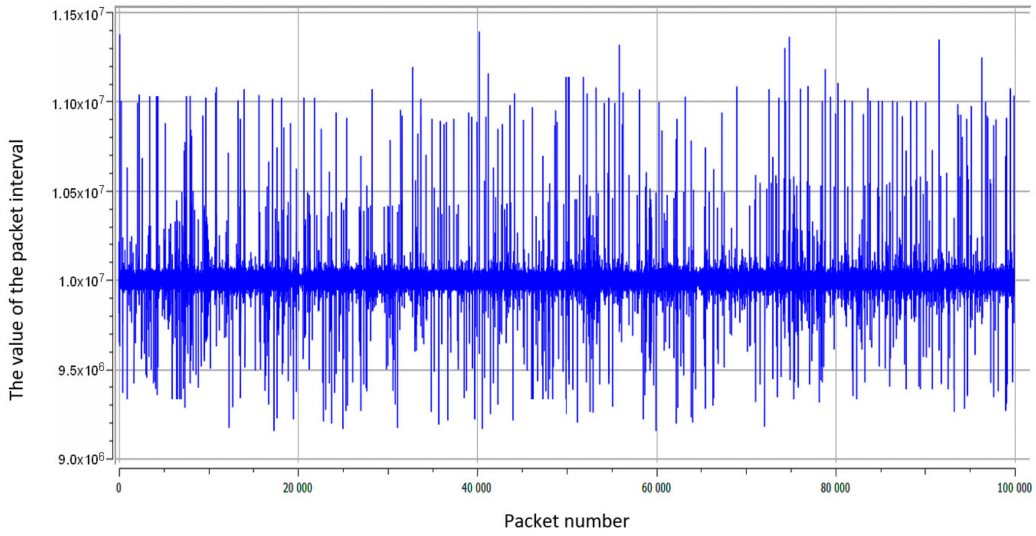

Figure 5. The stability characteristic of the interval between two sequentially generated packets.

As a result of modelling, the following statistical characteristics of a number of values of the inter-packet interval were obtained: mean value -1.00845×10^7 ns; variance -1.73237×10^{12} ns^2; standard deviation -1.3162×10^6 ns; average error -4163.33 ns.

The advantage of the developed traffic generator is that, unlike the known ones, it allows you to control the generated flow without the need for synchronize the input and output interfaces and to determine the packets delay of flow. It allows you to evaluate the parameters of service of hardware and software telecommunications facilities of any purpose [53–59].

Furthermore, in a real packet network, a few delay times can occur due to queuing, processing, transmission, and propagation, as depicted in Figure 6.

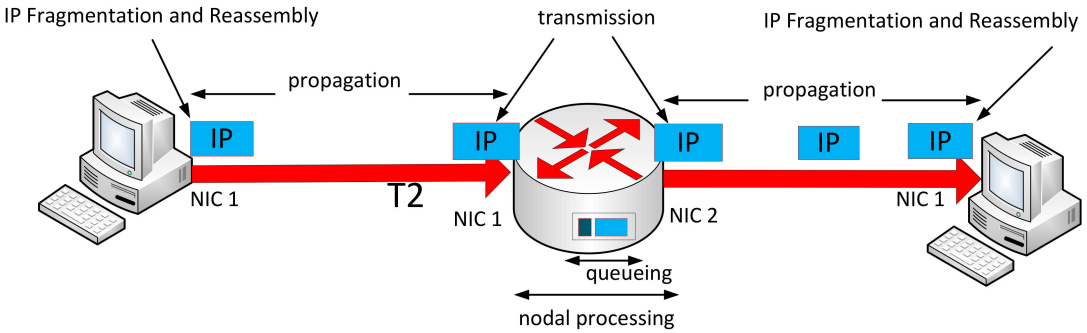

Figure 6. The delay through router.

If we let $d_{frag.}$, $d_{proc.}$, d_{queue}, $d_{trans.}$, and $d_{prop.}$ denote the IP fragmentation, processing, queuing, transmission and propagation delays, then the total nodal delay is given by:

$$d_{nodal.} = d_{frag.} + d_{proc.} + d_{queue} + d_{trans.} + d_{prop.}, \quad (1)$$

where $d_{frag.}$ is the time it takes to split a packet into smaller pieces (fragments), so that the resulting pieces can pass through a link with a smaller maximum transmission unit (MTU) than the original packet size. The fragments are reassembled by the receiving host. $d_{prop.}$ is the delay due to the propagation speed of the link. The propagation speed depends on the physical link (e.g., twisted-pair copper wire or multi-mode fiber) and is in the range of 2×10^8 or 3×10^8 m/s, thus it almost equal to the speed of the light [60]. Hence, if d is the distance between two routers and s is the propagation speed, it follows that the propagation delay is d = s. In general, the propagation delays are on the order of ms in wide-area networks. $d_{trans.}$ is the transmission delay, also called the "store-and-forward" delay. It represents the amount of time required to transmit all the packet bits along the link. For instance, if L is the length of the packet (in bits), R is the transmission rate of the link (typically in Mbps), it follows that the transmission delay is L = R. Transmission delays are typically on the order of ms or less in practice. $d_{proc.}$ the time required to examine the packet's header and determine where to direct the packet is part of the processing delay. In high-speed routers, is typically on the order of μs (microseconds) or less. d_{queue} is the time that a packet has to wait for the transmission along the link. The queuing delay of a specific packet depends on the number of other packets in the queue, thus, the delay of a given packet can vary significantly from packet to packet. On the one hand, if the queue is empty and no other packet is currently being transmitted, the packet queuing delay is approximately zero. On the other hand, if the traffic is heavy and many other packets are also waiting to be transmitted, the queuing delay will be long. Queuing delays can be on the order of μs to ms in practice.

In order to determine the packet delay, a series of experiments were conducted by the developed software-based router. In the case of a software router, its performance significantly depends on the performance of the hardware server. The same software router installed on different servers with different hardware characteristics can serve packets with different delays. To test the maximum performance, the proposed traffic generator is installed on a hardware server with the following parameters: the central processor—Intel Core i5-2410M 2.30 GHz, RAM—DDR3 6Gb, network card—Realtek PCIe FE Family Controller 1 Gbit/s. Windows 7 Ultimate Service Pack 1 (2009) is installed on the server.

In each experiment, the packet size changed, the number of packets was 100,000, and the inter-packet interval was 10 ms. These measurements were carried out in the conditions of the software router installation on different servers with different hardware characteristics (Celeron J1800|2 cores|2.41 GHz| RAM 4 GB|; Intel Core i3-3230M (2.6 GHz)/RAM 4 GB; Intel Core i5-2410M (2.3 GHz)/RAM 6 GB; Core i7-7700|4 cores|3.6 GHz| RAM 8 GB). We proposed a scheme to measure the packet processing time of a router using a specialized packet-capture network interface card in a traffic generator (Ethernet

port 1 and Ethernet port 2). This generator can send and receive traffic at the same time, and thus calculate the time difference between sending time—T1 (Ethernet port 1) and receiving time—T2 (Ethernet port 2) each packet, as a T2–T1. This allows the generator to calculate the time of packets transit over the network and processing on the router. Experimental scheme for determining the total nodal delay by the software router installed on the servers of different performance is depicted in Figure 7.

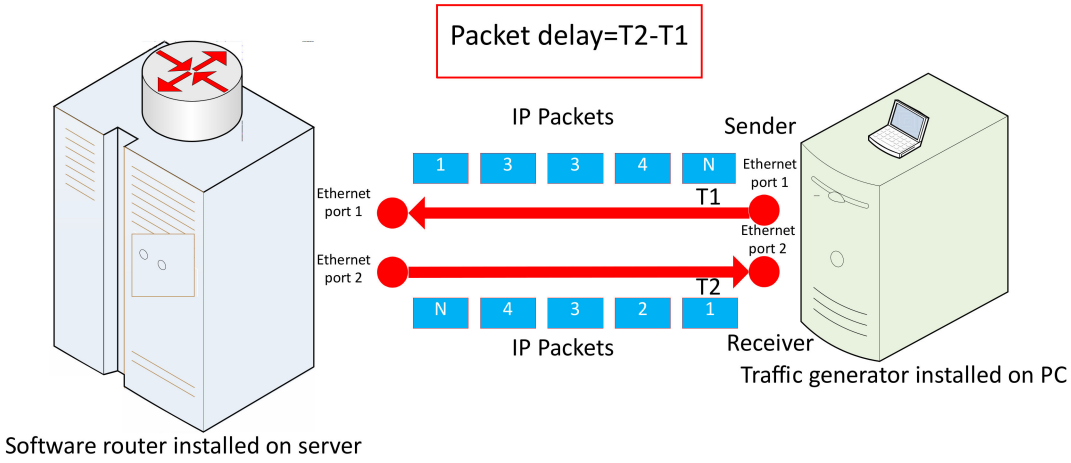

Figure 7. Experimental scheme for determining the total nodal delay by the software router installed on the servers of different performance.

The scheme does not require synchronization between the router network interface card and the packet-capture network interface card. Also the scheme does not require synchronization between the NICs of traffic generator (Ethernet port 1 and Ethernet port 2).

The proposed basic model of the traffic generation system can be used to create test environments to study the real network using several separate physical computers to simulate clients. In studying the characteristics of the real network data transmission, the main and signaling traffic required for the test environment should be separated. At best, the signaling traffic should be transmitted over a separate physical network. Otherwise, it is necessary to create an additional virtual local area network at the test network. Such a scheme will allow to conduct experiments remotely, using a single device to control all other devices, significantly expands the capabilities of the researcher, simplifies the conduct of new experiments and saves time on their formulation. In the process of research to test the efficiency of the software router and traffic generator, an experiment was conducted. The investigation of the total nodal delay by the software router on the servers of different performance is shown in the Table 1.

So, from the experimental results we can see that the same software router installed on different servers with different hardware characteristics can serve packets with different delays. Thus, it is established that the more productive server on which the software router is installed, the less total nodal delay will be.

Table 1. Investigation of the average time of the total nodal delay by the software router on the servers of different performance.

Celeron J1800|2 Cores|2.41 GHz|RAM 4 GB		
Series	**Packet Size [byte]**	**Average delay [ms]**
1	64	1.24
2	128	1.79
3	512	1.98
4	1024	2.12
5	1500	2.2
Intel Core i3-3230M|2.6 GHz| RAM 4 GB		
Series	Packet size [byte]	Average delay [ms]
1	64	0.42
2	128	0.79
3	512	0.98
4	1024	1.01
5	1500	1.21
Intel Core i5-2410M|2.3 GHz|RAM 6 GB		
Series	Packet size [byte]	Average delay [ms]
1	64	0.33
2	128	0.48
3	512	0.52
4	1024	0.62
5	1500	0.68
Core i7-7700|4 cores|3.6 GHz|RAM 8 GB		
Series	Packet size [byte]	Average delay [ms]
1	64	0.01
2	128	0.04
3	512	0.09
4	1024	0.12
5	1500	0.2

4. Results and Discussions

4.1. Method to Measure and Compare Packet Processing Time of Software Router with Prototype Hardware Router CISCO 2801

The most complicated and interesting component of nodal delay is the queuing delay, d_{queue}. In fact, queuing delay is important and interesting in computer networking Another interesting scientific task is to develop a method of estimating the packets processing delay by only router ($d_{proc.} + d_{queue}$) without taking into account IP Fragmentation, transmission and propagation delays. However, at present there are no practical methods of measuring the delay of packets introduced only by the router ($d_{proc.} + d_{queue}$). Existing router delay testing methods are based on formula 1 and are defined as the sum of all types of delays ($d_{frag.} + d_{proc.} + d_{queue} + d_{trans.} + d_{prop.}$). These methods do not make it possible to accurately assess the router's performance in terms of average packet processing time. Moreover, this method will be a good practical tool for comparing the performance of software and hardware routers.

In the following we will enter the designations Tav.router as a packet processing time by the router:

$$T_{av.router} = d_{proc.} + d_{queue}. \quad (2)$$

Packet processing time $T_{av.router}$ of a host (i.e., router) is the time elapsed between the arrival of a packet in the router (input queue of the NIC, i.e., the data-link layer of the TCP/IP protocol stack) and the time the packet is processed at the application layer ($d_{proc.} + d_{queue}$).

The idea of the experiment is to create a software router as a prototype Cisco 2801 hardware router. To confirm or simplify the hypothesis that the developed software router provides a cost-efficient alternative to expensive, special hardware routers. To do this, it is necessary to select a server machine for the software router, the average time of processing packets which should be identical to the average time of processing packets by hardware routers and then to compare their prices. For the experiment the computer on which the traffic generator is developed is chosen. This generator can send and receive traffic at the same time, and thus calculate the difference between sending and receiving times each packet. This allows the generator to calculate the time of packets transit over the network and processing on the router.

4.1.1. Experiment with 1 Hardware and 1 Software Router

In the work the comparative experiment of packet processing time with use of a hardware router (Cisco 2801 series) and the developed software router is carried out. The experiment was conducted in 215 laboratories of the Cisco XI training building of the Lviv Polytechnic National University. The routers were configured through the console using the FIFO queue service algorithm, which is set by default. The traffic generator (50,000 packets) was used to monitor the intensity of the incoming traffic created on the interface of the Cisco 2801 series router. The scheme of the experiments is shown in Figure 8.

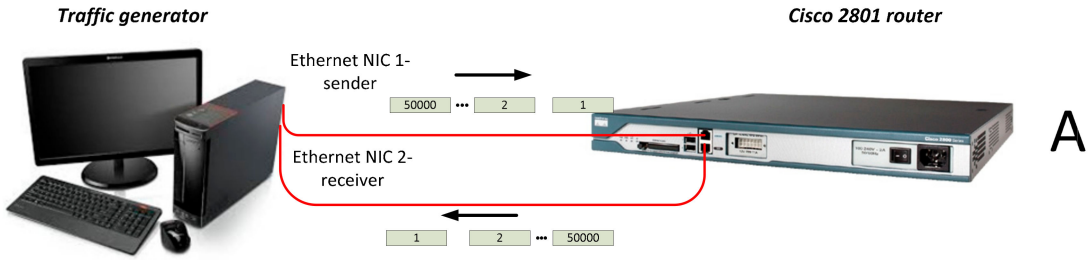

Figure 8. Experimental scheme for determining the packets delay by a Cisco 2801 hardware router.

In real time mode, graphs were built that showed a packets delay and probability density function of these values by quantity. At the same time, the Wireshark network analyzer detected fragmentation, grouping of packets of the transmitted information flow and confirmation of their delivery. The time delay value for packet fragmentation and packet grouping was detected. The use of Wireshark allowed to monitor the process of data transmission through the router at the network level with the detection of alarm data packets and fixation of network anomalies, which affect the time parameters of quality of service of real-time traffic. Such packets delays are single and range from 10 to 50 ms. Measurement results of packet delay via the Cisco 2801 router are shown in Figure 9.

Having determined the average packets delay by the hardware router, the server machine performance is selected for the software router, which provides similar delays. Correspondingly, for comparison we chose the Intel Core i5-2410M (2.3 GHz)/RAM 6 GB server. The compared total nodal delay by the software router installed on the servers of different performance with the Cisco 2801 is depicted in Figure 10.

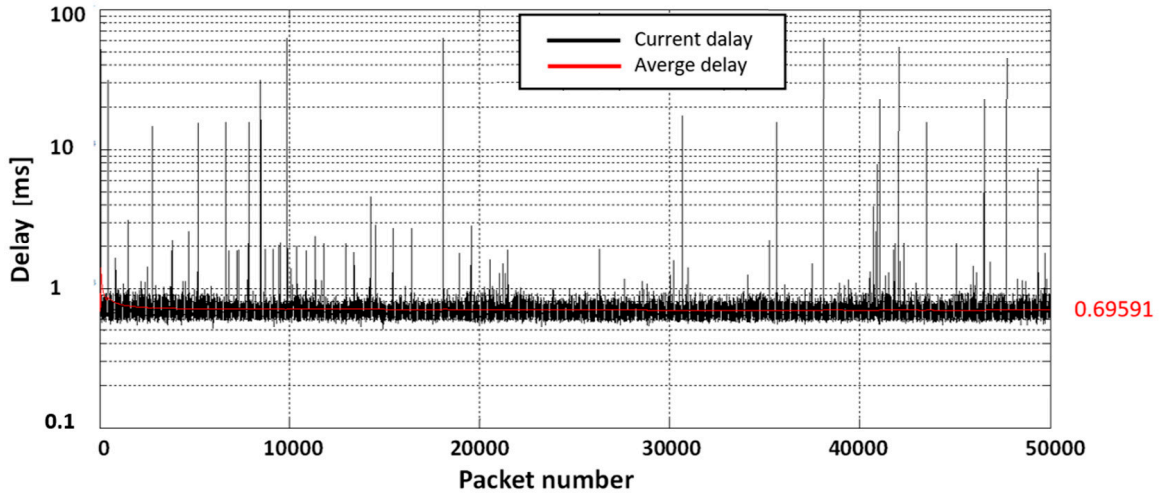

Figure 9. Packets delay caused by the Cisco 2801 router.

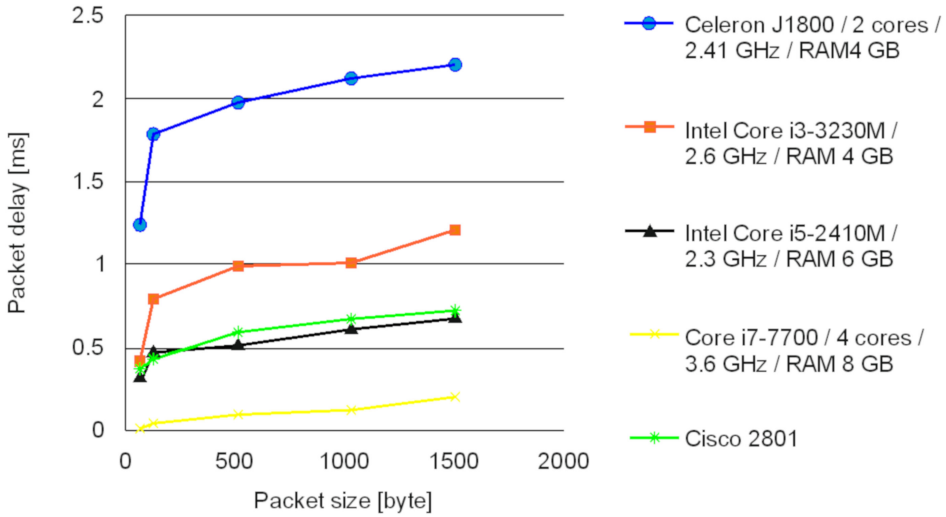

Figure 10. Comparison of the total nodal delay by the software router installed on the servers of different performance with the Cisco 2801 server.

A similar experiment was conducted with the developed software router (Figure 11). Measured result of packet delay via the software—based router is shown in Figure 12.

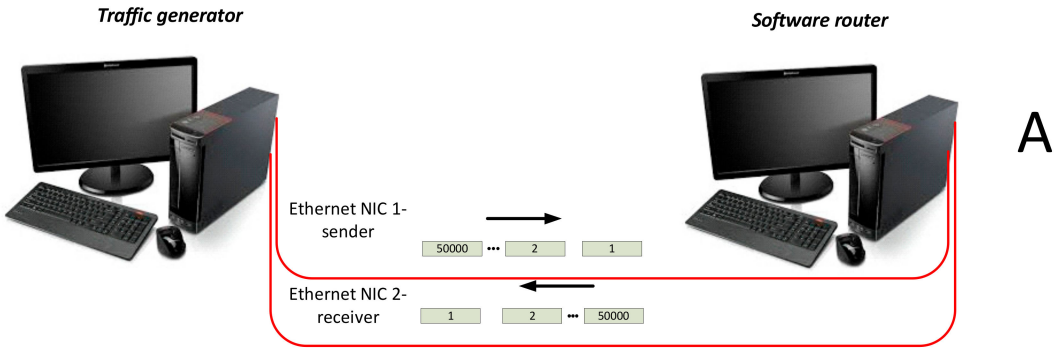

Figure 11. Experimental scheme for determining the packets delay by the software—based router.

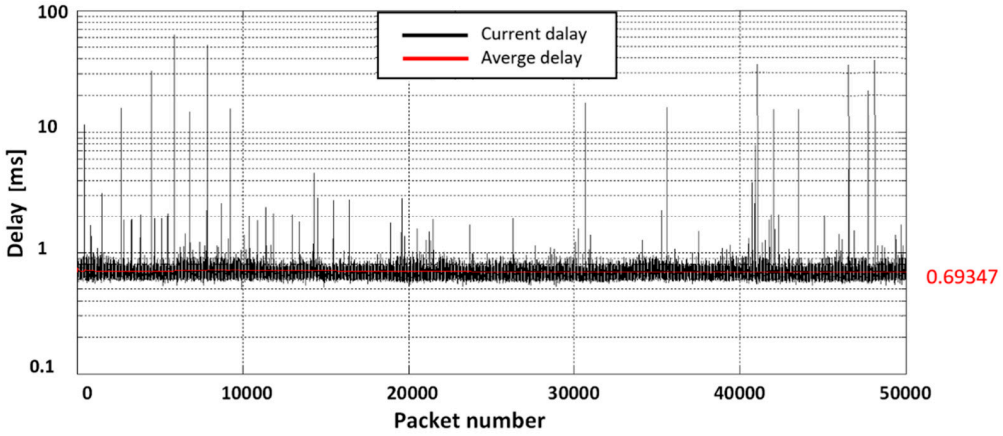

Figure 12. Packets delay caused by software—based router.

Comparison of the average packets delays caused by the software router and the Cisco 2801 hardware router is shown in Figure 13. Comparison of the probability density functions of packets delays caused by the hardware and software router is shown in Figure 14.

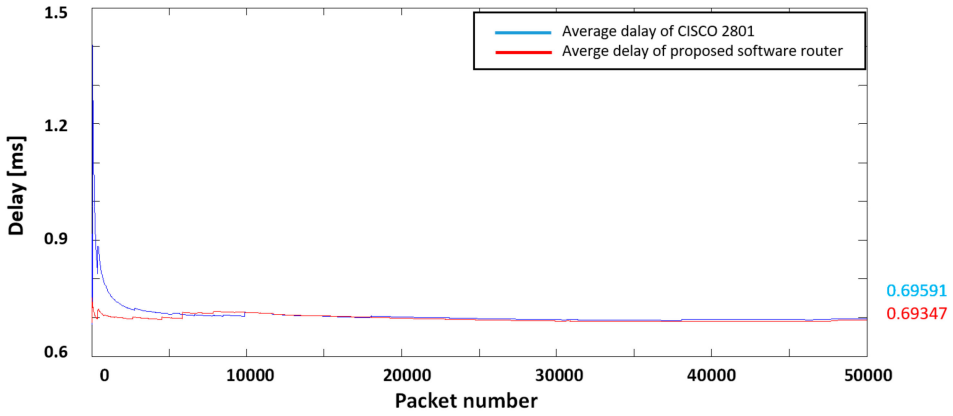

Figure 13. Comparison of the average packets delays caused by the Cisco 2801 hardware router and the software router.

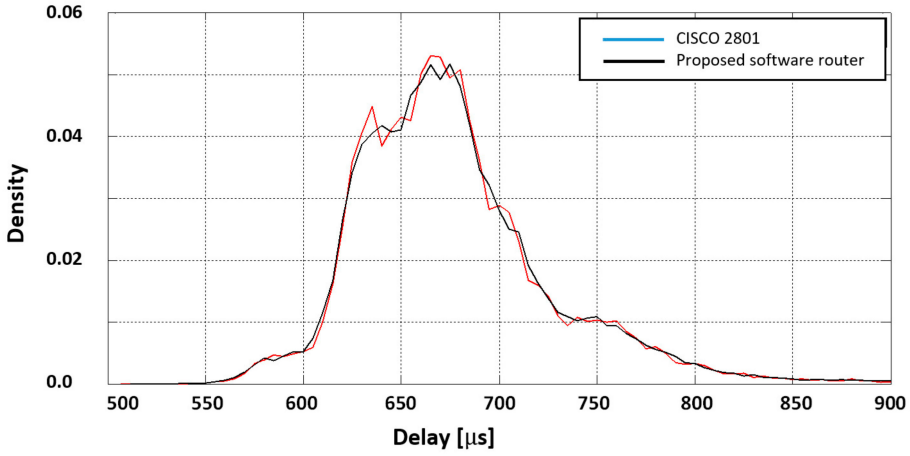

Figure 14. Comparison probability density functions of packets delay caused by hardware and software router.

The conducted studies confirmed the adequacy of the results obtained in the course of experimental simulation of the software router with Cisco 2801 series hardware. The average packets delay caused

by the hardware router (experiment A)—$D_{HR(A)_1}$ is 695.91 μs, and through the software—$D_{HR(A)_1}$ 693.47 μs when processing the same information flow. As we can see, the measured difference between the software and hardware routers is 2.44 μs. And also the probability density functions of packets processing time caused by hardware router is same in comparison with the packets processing time probability density functions obtained in work [60].

Also, for more exact reception of results of a time delay of packages created by the Cisco 2801 router. The experiment when the information flow passed through routers A, A-B and A-B-C is carried out.

4.1.2. Experiment with 2 Hardware and 2 Software Routers

To research the packets processing time by A-B routers in the experiment a network of two Cisco 2801 routers was configured and they were connected to a traffic generator installed on a personal computer with two network interface cards to calculate the difference between the moments of sending and receiving each package (Figure 15). A similar experiment was conducted with the two developed software routers (Figure 16).

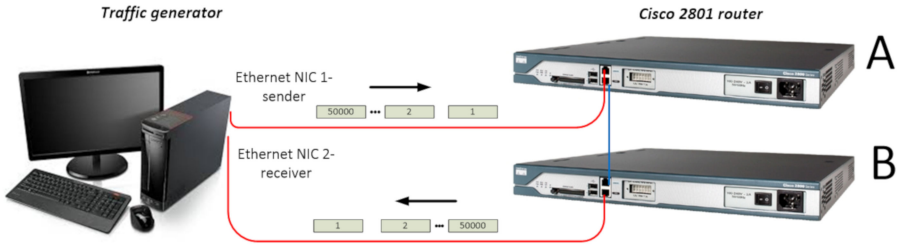

Figure 15. Experimental scheme for determining the packets processing time by two Cisco 2801 hardware routers.

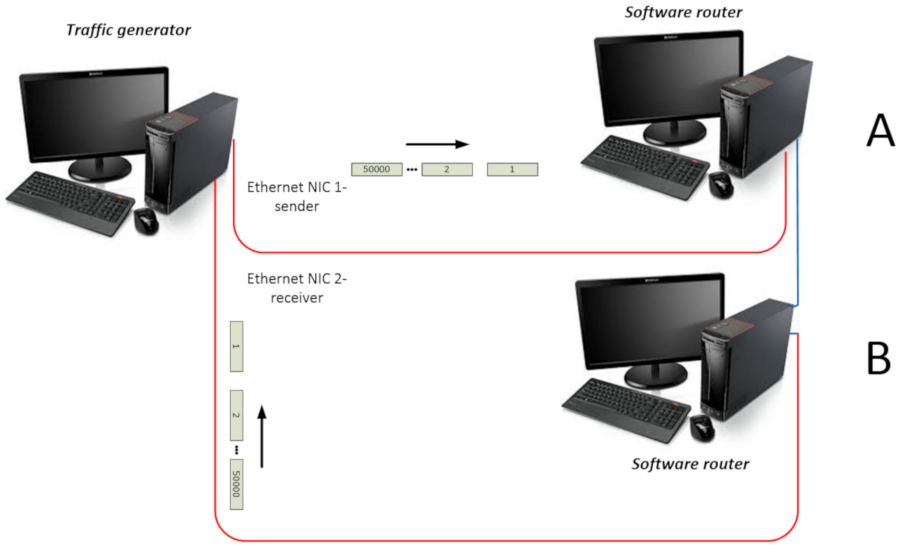

Figure 16. Experimental scheme for determining the packets processing time by two software—based routers.

Comparison of the average packets delay caused by the two Cisco 2801 hardware routers and two software routers is shown in Figure 17. Comparison of the probability density functions of packets processing time caused by the two hardware and two software routers is shown in Figure 18.

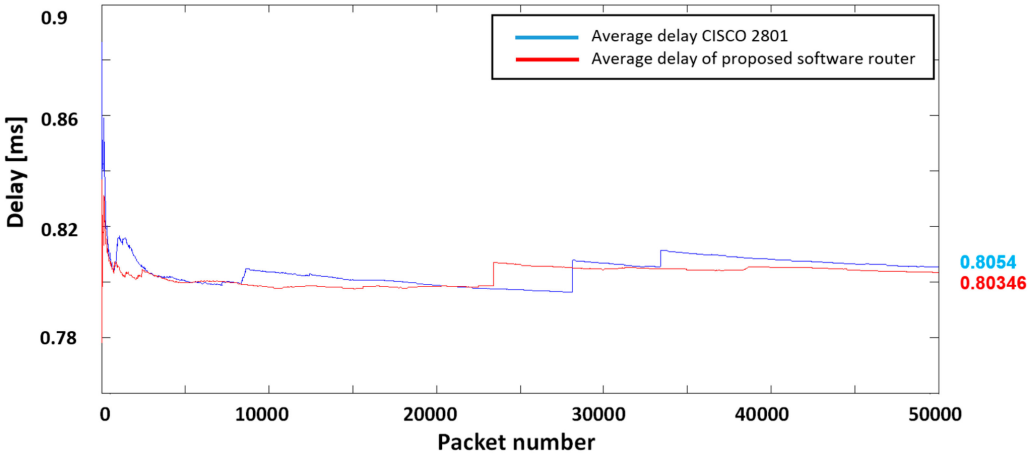

Figure 17. Comparison of the average packets delays caused by two Cisco 2801 hardware routers and two software routers.

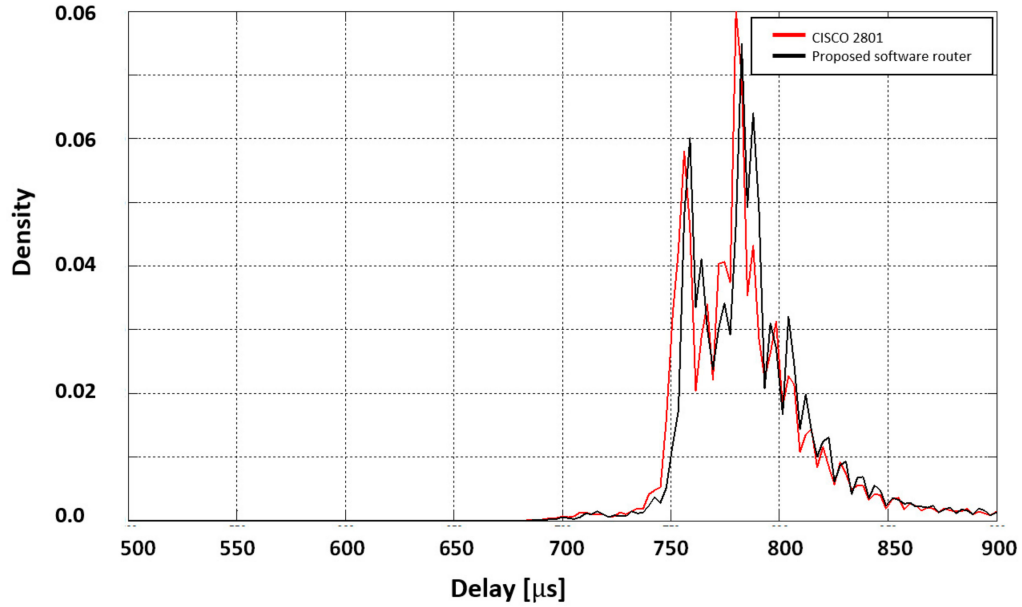

Figure 18. Comparison probability density functions of packets delays caused by two hardware and two software routers.

In the experimental research it is found that transmission of the same traffic over two hardware routers has an average packet delay $D_{HR(A-B)_2}$ of 805.4 μs, while the average packet delays over two software routers $D_{SR(A-B)_2}$ is 803.46 μs. As we can see in the comparison of packets delays the measured difference between the hardware and software routers is approximately 1.94 μs, the same as in the first experiment. However, it is necessary to focus on the average packets delay at the first and second experiments, because the value of the delay caused by hardware routers during two experiments differs by 805.4 − 695.91 = 109.49 μs. As a result, it was found that the packet processing time caused by only one router of the CISCO 2801 series is 109.49 μs ($d_{proc.} + d_{queue}$), and the packet delay of 586.42 μs is caused to the computer's operating system when generating traffic and depends on its performance ($d_{frag.} + d_{trans.} + d_{prop.}$). From Figure 20 we can see that the probability density function of packets processing time of the software routers approaches that of the hardware routers of Cisco 2801 series.

4.1.3. Experiment with Three Hardware and Three Software Routers

For a more accurate assessment of packet processing time by a real router and to determine the packet delay from end to end an experiment using three routers via which the same information flow created by the generator passed and in previous experiments was carried out. The experimental scheme for determining the packets delay by three hardware Cisco 2801 routers is depicted in Figure 19. The experimental scheme for determining the packets delay caused by three software routers is depicted in Figure 20.

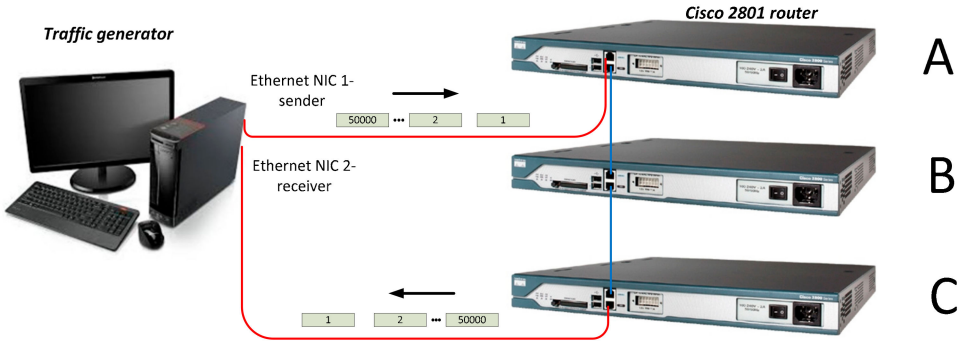

Figure 19. Experimental scheme for determining the packets delay caused by three hardware Cisco 2801 routers.

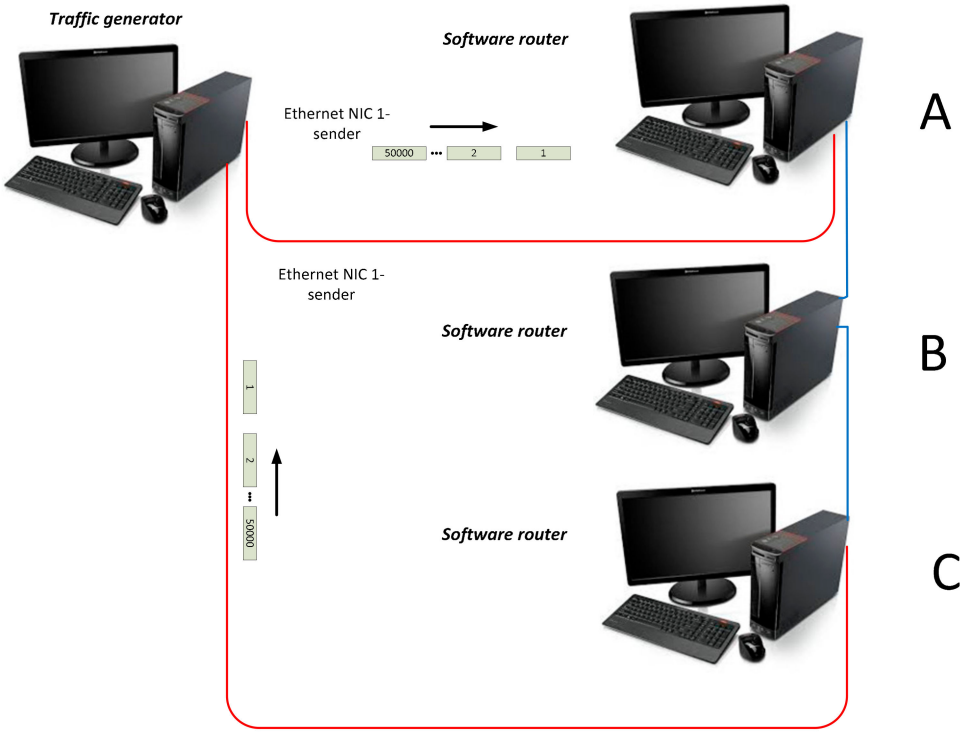

Figure 20. Experimental scheme for determining the packet delays by three software-based routers.

Comparison of the average packets delay caused by the three Cisco 2801 hardware routers and three software routers are shown in Figure 21. Comparison of probability density functions of packets delay caused by three hardware and three software routers are shown in Figure 22.

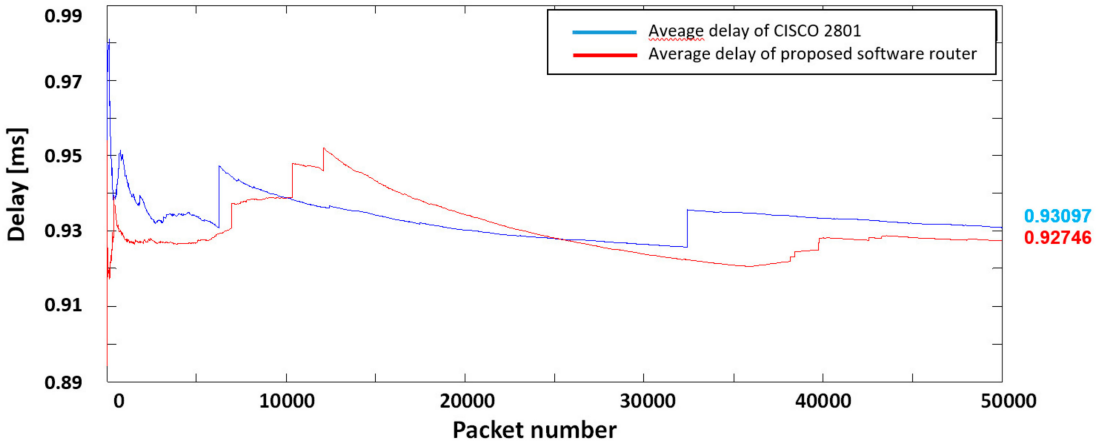

Figure 21. Comparison of the average packet delays caused by three Cisco 2801 hardware routers and three software routers.

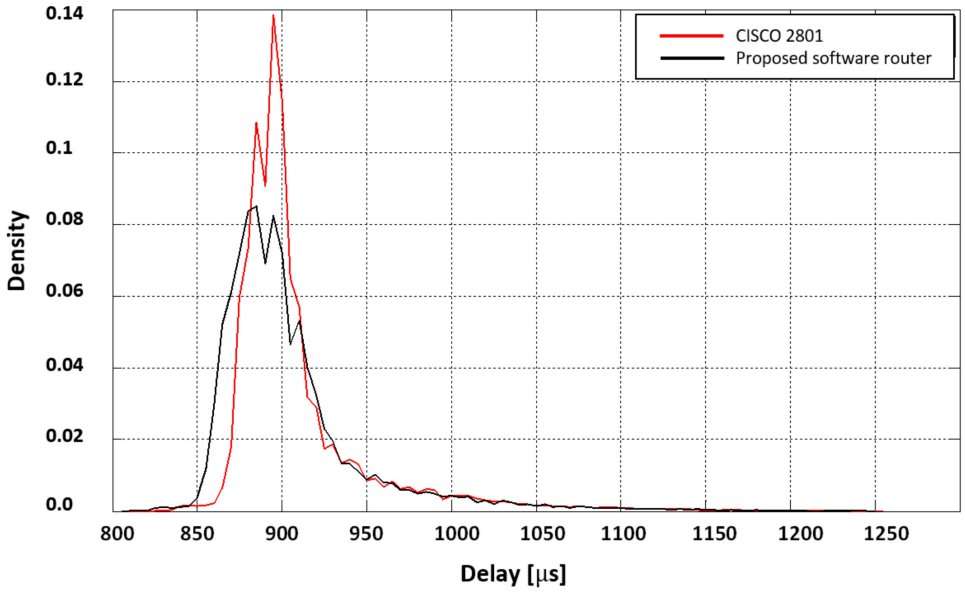

Figure 22. Comparison of probability density functions of the packet delays caused by three hardware and three software routers.

The average packets delay caused by the three hardware router $D_{HR(A-B-C)_3}$ is 930.97 µs, and through the software $D_{SR(A-B-C)_3}$ it is 927.46 µs when processing the same information flow. As we can see by the comparison of average packets delays, the measured difference between the hardware and software routers is 3.57 µs. The probability density function of packets delay of the three software routers approaches that of the three Cisco 2801 series hardware routers.

4.1.4. Experimental Test Bed

Based on the experimental test bed depicted in Figure 23 we calculate the average packets processing time by only one hardware and one software router using Equation (3).

The average packets processing time by a hardware and software router during three experiments is defined as follows:

$$T_{av.hardware|router} = \frac{(D_{HR(A-B)_2} - D_{HR(A)_1}) + (D_{HR(A-B-C)_3} - D_{HR(A-B)_2})}{2} \quad (3)$$

$$T_{av.software|router} = \frac{(D_{SR(A-B)_2} - D_{SR(A)_1}) + (D_{SR(A-B-C)_3} - D_{SR(A-B)_2})}{2} \quad (4)$$

By substituting the obtained values into Formulas (3) and (4), we obtain:

$$T_{av.hardware|router} = ((0.8054 - 0.69591) + (0.93097 - 0.8054))/2 = 0.11753 \text{ ms} \quad (5)$$

and:

$$T_{av.software|router} = ((0.803446 - 0.69347) + (0.9274 - 0.803446))/2 = 0.116965 \text{ ms} \quad (6)$$

Thus, we can see that the performance testing by the delay criterion of the developed software router is similar to the hardware router. Accordingly, the price of the server machine for which the software router is installed is 400 dollars, and the price of a Cisco 2801 is 800 dollars, which is cost-efficient.

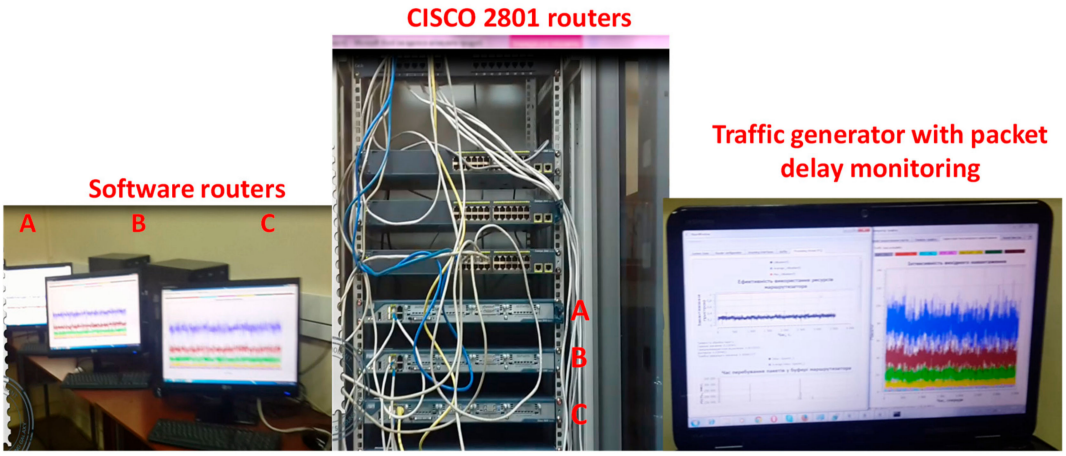

Figure 23. Experimental test bed.

5. Conclusions

A model of a software-based router that supports differentiated service of multiservice traffic has been developed in this paper. The software router is based on the technology of sockets, which are software objects of the operating system and consist of IP address of the device and TCP port. Using the API of the operating system, the software router receives the generated socket object and uses it to communicate with other software routers that are installed on other physical machines in the local network. Also, for generation of multiservice traffic and performance testing of routers, the software generator is developed, using sockets. This generator allows forming traffic with any parameters, on the basis of dynamic mixing of flows with various statistical characteristics and determining the packets delay. The adequacy of the software router model and the software generator was assessed. From the results of the experiments it was established that the models of both the generator and the router are adequate. In particular, the adequacy of the software router was evaluated in relation to the real Cisco 2801hardware router, and the adequacy of the generator was evaluated on the basis of comparing the statistical characteristics of the multiservice traffic of the real transport network and statistical characteristics of the traffic generated by the software generator. In this paper we investigated the packets delay by the software router installed on servers of different performance. It is established that the more productive the server is, the less the packet delay caused by the software router will be. The main advantages of proposed software routers over hardware ones are flexibility, intelligence and simplicity of algorithms modification.

The method to measure and compare packet processing time of software router with a Cisco 2801 prototype hardware router has been proposed. For this the experimental test bed based on developed software router, traffic generator and Cisco 2801 is created. After a number of experimental

studies it was determined that the average packet processing time by hardware router is 0.11753 ms, and through the software-router is 0.116965 ms when processing the same information flow. The measured difference of processing time between hardware and proposed software router is 0.5 µs. The experimental results show that our testing scheme can consistently measure the packet processing time of the router, and without clock synchronization. A proposed novel software-based router can also be a cheaper long-term option for businesses of all types. If one has already spent money on a server chassis for a data center, one can forgo the capital expenditure of purchasing a special hardware router appliance and instead opt for a lower-cost of software router option.

Author Contributions: All authors contributed to the study conception and design; methodology: J.W., K.P., M.B., O.K., H.B., M.K., S.J., D.P.; formal analysis and investigation: J.W., K.P., M.B., O.K., H.B., M.K., S.J.; writing—original draft preparation: J.W., K.P., M.B., O.K., H.B., M.K.; writing—review and editing: J.W., K.P., M.B., O.K., H.B., M.K., S.J.; funding acquisition: K.P., D.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the project No. 0117U004449, "Methods of heterogeneous information and communication systems engineering for the deployment of software-defined networks of 5G for dual purpose".

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Parane, K.; Prabhu Prasad, B.M.; Talawar, B. Design of an Adaptive and Reliable Network on Chip Router Architecture Using FPGA. In Proceedings of the 2019 International Symposium on VLSI Design, Automation and Test (VLSI-DAT), Hsinchu, Taiwan, 22–25 April 2019; pp. 1–4. [CrossRef]
2. Meyer, T.; Raumer, D.; Wohlfart, F.; Wolfinger, B.E.; Carle, G. Low Latency Packet Processing in Software Routers. In Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2014), Monterey, CA, USA, 6–10 July 2014; pp. 556–563. [CrossRef]
3. Zhu, T.; Lan, J. A Novel Router Software Architecture Supporting Reconfiguration. In Proceedings of the 2010 2nd International Workshop on Intelligent Systems and Applications, Wuhan, China, 22–23 May 2010; pp. 1–4. [CrossRef]
4. Cárdenas-Benítez, N.; Aquino-Santos, R.; Magaña-Espinoza, P.; Aguilar-Velazco, J.; Edwards-Block, A.; Medina Cass, A. Traffic Congestion Detection System through Connected Vehicles and Big Data. *Sensors* **2016**, *16*, 599. [CrossRef] [PubMed]
5. Botta, A.; Dainotti, A.; Pescapé, A. Do You Trust Your Software-Based Traffic Generator? *IEEE Commun. Mag.* **2010**, *48*, 158–165. [CrossRef]
6. Emmerich, P.; Gallenmüller, S.; Antichi, G.; Moore, A.W.; Carle, G. Mind the Gap—A Comparison of Software Packet Generators. In Proceedings of the 2017 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS), Beijing, China, 18–19 May 2017; pp. 191–203. [CrossRef]
7. Gupta, V.; Vajpeyee, M.; Kar, S.; Kumar, T.R.N. Performance Analysis and Redundancy Implementation of Open Source Embedded Router. In Proceedings of the 2012 National Conference on Communications (NCC), Kharagpur, India, 3–5 February 2012; pp. 1–5. [CrossRef]
8. Hua, H.; Qin, Y.; Xu, H.; Hao, C.; Cao, J. Robust Control Method for DC Microgrids and Energy Routers to Improve Voltage Stability in Energy Internet. *Energies* **2019**, *12*, 1622. [CrossRef]
9. Molina Zarca, A.; Garcia-Carrillo, D.; Bernal Bernabe, J.; Ortiz, J.; Marin-Perez, R.; Skarmeta, A. Enabling Virtual AAA Management in SDN-Based IoT Networks. *Sensors* **2019**, *19*, 295. [CrossRef] [PubMed]
10. Kohler, E.; Morris, R.; Chen, B.; Jannotti, J.; Kaashoek, M.F. The Click Modular Router. *ACM Trans. Comput. Syst.* **2000**, *18*, 263–297. [CrossRef]
11. Hong, C.-H.; Lee, K.; Hwang, J.; Park, H.; Yoo, C. Kafe: Can OS Kernels Forward Packets Fast Enough for Software Routers? *IEEE/ACM Trans. Netw.* **2018**, *26*, 2734–2747. [CrossRef]
12. Lee, H.; Nakao, A. Improving Routing Table Lookup in Software Routers. *IEEE Commun. Lett.* **2015**, *19*, 957–960. [CrossRef]
13. Bianco, A.G.; Birke, R.; Botto, G.; Chiaberge, M.; Finochietto, J.M.; Galante, G.; Mellia, M.; Neri, F.; Petracca, M. Boosting the Performance of PC-Based Software Routers with FPGA-Enhanced Network Interface Cards. In Proceedings of the 2006 Workshop on High Performance Switching and Routing 2006, 6-NaN, Poznan, Poland, 7–9 June 2006. [CrossRef]

14. Pryslupskyi, A.; Panchenko, O.; Beshley, M.; Seliuchenko, M. Improvement of Multiprotocol Label Switching Network Performance Using Software-Defined Controller. In Proceedings of the 2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM), Polyana, Ukraine, 26 February–2 March 2019; pp. 106–109. [CrossRef]
15. Bolla, R.; Bruschi, R. The IP Lookup Mechanism in a Linux Software Router: Performance Evaluation and Optimizations. In Proceedings of the 2007 Workshop on High Performance Switching and Routing, Brooklyn, NY, USA, 30 May–1 June 2007; pp. 1–6. [CrossRef]
16. Meyer, T.; Wohlfart, F.; Raumer, D.; Wolfinger, B.E.; Carle, G. Validated Model-Based Performance Prediction of Multi-Core Software Routers. *Prax. Inf. Kommun.* **2014**, *37*, 93–107. [CrossRef]
17. Agrawal, N.; Mishra, N. RTT Based Wormhole Detection Using NS-3. In Proceedings of the 2014 International Conference on Computational Intelligence and Communication Networks, Bhopal, India, 14–16 November 2014; pp. 861–866. [CrossRef]
18. Cao, C.; Zuo, Y.; Zhang, F. Research on Comprehensive Performance Simulation of Communication IP Network Based on OPNET. In Proceedings of the 2018 International Conference on Intelligent Transportation, Big Data Smart City (ICITBS), Xiamen, China, 25–26 January 2018; pp. 195–197. [CrossRef]
19. Rozi, N.N.H.; Ghafar, A.S.A.; Nordin, N.M.; Saparudin, F.A. Performance Evaluation of Multihop Device to Device (D2D) Communication Using Network Simulator and Emulator (NetSim). In Proceedings of the 2019 IEEE 10th Control and System Graduate Research Colloquium (ICSGRC), Shah Alam, Malaysia, 2–3 August 2019; pp. 46–51. [CrossRef]
20. Oujezsky, V.; Horvath, T. Case Study and Comparison of SimPy 3 and OMNeT++ Simulation. In Proceedings of the 2016 39th International Conference on Telecommunications and Signal Processing (TSP), Vienna, Austria, 27–29 June 2016; pp. 15–19. [CrossRef]
21. Dronyuk, I.; Fedevych, O.; Lipinski, P. Ateb-Prediction Simulation of Traffic Using OMNeT++ Modeling Tools. In Proceedings of the 2016 XIth International Scientific and Technical Conference Computer Sciences and Information Technologies (CSIT), Lviv, Ukraine, 6–10 September 2016; pp. 96–98.
22. Bu, G.; Xie, D.; Chen, H.; Gao, M. Impacts on High Frequency Communications Based on OPNET. In Proceedings of the 2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), Hunan, China, 10–11 August 2018; pp. 524–527. [CrossRef]
23. Weigle, M.; Adurthi, P.; Hernández-Campos, F.; Jeffay, K.; Smith, F. Tmix: A Tool for Generating Realistic TCP Application Workloads in Ns-2. *Comput. Commun. Rev.* **2006**, *36*, 65–76. [CrossRef]
24. Barford, P.; Crovella, M. Generating Representative Web Workloads for Network and Server Performance Evaluation. In Proceedings of the 1998 ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS'98/PERFORMANCE'98, Madison, WI, USA, 22–26 June 1998; pp. 151–160. [CrossRef]
25. Hashemian, R.; Krishnamurthy, D.; Arlitt, M. Web Workload Generation Challenges—An Empirical Investigation. *Softw. Pract. Exp.* **2012**, *42*, 629–647. [CrossRef]
26. Choi, Y.; Silvester, J.; Kim, H. Analyzing and Modeling Workload Characteristics in a Multiservice IP Network. *IEEE Internet Comput.* **2011**, *15*, 35–42. [CrossRef]
27. Abhari, A.; Soraya, M. Workload Generation for YouTube. *Multimed Tools Appl.* **2009**, *46*, 91. [CrossRef]
28. Veloso, E.; Almeida, V.; Meira, W.; Bestavros, A.; Jin, S. A Hierarchical Characterization of a Live Streaming Media Workload. *IEEE/ACM Trans. Netw.* **2006**, *14*, 133–146. [CrossRef]
29. Best Network Traffic Generator Software & Tools for WAN & LAN Testing! Available online: https://www.ittsystems.com/network-traffic-generator/ (accessed on 9 November 2019).
30. Patil, A.G.; Surve, A.R.; Gupta, A.K.; Sharma, A.; Anmulwar, S. Survey of Synthetic Traffic Generators. In Proceedings of the 2016 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 26–27 August 2016; Volume 1, pp. 1–3. [CrossRef]
31. Srivastava, S.; Anmulwar, S.; Sapkal, A.M.; Batra, T.; Gupta, A.K.; Kumar, V. Comparative Study of Various Traffic Generator Tools. In Proceedings of the 2014 Recent Advances in Engineering and Computational Sciences (RAECS), Chandigarh, India, 6–8 March 2014; pp. 1–6. [CrossRef]
32. Kolahi, S.S.; Narayan, S.; Nguyen, D.D.T.; Sunarto, Y. Performance Monitoring of Various Network Traffic Generators. In Proceedings of the 2011 UkSim 13th International Conference on Computer Modelling and Simulation, Cambridge, UK, 30 March–1 April 2011; pp. 501–506. [CrossRef]
33. Ostinato Packet Generator. Available online: https://ostinato.org/ (accessed on 9 November 2019).

34. iPerf-The TCP, UDP and SCTP Network Bandwidth Measurement Tool. Available online: https://iperf.fr/ (accessed on 9 November 2019).
35. Packeth. Available online: http://packeth.sourceforge.net/packeth/Home.html (accessed on 9 November 2019).
36. Goldstein, M.; Seheult, A.; Vernon, I. Assessing Model Adequacy. In *Environmental Modelling*; John Wiley & Sons, Ltd.: London, UK, 2013; pp. 435–449. [CrossRef]
37. Przystupa, K. Reliability Assessment Method of Device Under Incomplete Observation of Failure. In Proceedings of the 2018 18th International Conference on Mechatronics-Mechatronika (ME), Brno, Czechia, 5–7 December 2018; pp. 1–6.
38. Przystupa, K.; Kozieł, J. Analysis of the Quality of Uninterruptible Power Supply Using a UPS. In Proceedings of the 2018 Applications of Electromagnetics in Modern Techniques and Medicine (PTZE), Raclawice, Poland, 9–12 September 2018; pp. 191–194. [CrossRef]
39. Wojciechowski, S.; Wiackiewicz, M.; Krolczyk, G.M. Study on Metrological Relations Between Instant Tool Displacements and Surface Roughness During Precise Ball End Milling. *Measurement* **2018**, *129*, 686–694. [CrossRef]
40. Glowacz, A. Acoustic-Based Fault Diagnosis of Commutator Motor. *Electronics* **2018**, *7*, 299. [CrossRef]
41. Shu, C.; Kochan, O. Method of Thermocouples Self Verification on Operation Place. *Sens. Transducers* **2013**, *160*, 55–61.
42. Vasylkiv, N.; Kochan, O.; Kochan, R.; Chyrka, M. The Control System of the Profile of Temperature Field. In Proceedings of the 2009 IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Rende, Italy, 21–23 September 2009; pp. 201–206. [CrossRef]
43. Murakami, M.; Kominami, D.; Leibnitz, K.; Murata, M. Reliable Design for a Network of Networks with Inspiration from Brain Functional Networks. *Appl. Sci.* **2019**, *9*, 3809. [CrossRef]
44. Przystupa, K. Selected methods for improving power reliability. *Przegląd Elektrotechniczny* **2018**, *94*, 270–273. [CrossRef]
45. Sargent, R. Verification and Validation of Simulation Models. In Proceedings of the 2010 Winter Simulation Conference, Baltimore, MD, USA, 5–8 December 2010; Volume 37, pp. 166–183. [CrossRef]
46. Beisbart, C.; Saam, N.J. *Computer Simulation Validation-Fundamental Concepts, Methodological Frameworks, and Philosophical Perspectives*; Springer: Berlin, Germany, 2019.
47. Beshley, M.; Pryslupskyi, A.; Panchenko, O.; Beshley, H. SDN/Cloud Solutions for Intent-Based Networking. In Proceedings of the 2019 3rd International Conference on Advanced Information and Communications Technologies (AICT), Lviv, Ukraine, 2–6 July 2019; pp. 22–25. [CrossRef]
48. Klymash, M.; Romanchuk, V.; Beshley, M.; Arthur, P. Investigation and Simulation of System for Data Flow Processing in Multiservice Nodes Using Virtualization Mechanisms. In Proceedings of the 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), Kiev, Ukraine, 29 May–2 June 2017; pp. 989–992. [CrossRef]
49. Romanchuk, V.; Beshley, M.; Polishuk, A.; Seliuchenko, M. Method for Processing Multiservice Traffic in Network Node Based on Adaptive Management of Buffer Resource. In Proceedings of the 2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET), Slavske, Ukraine, 20–24 February 2018; pp. 1118–1122. [CrossRef]
50. Aweya, J. Software Requirements for Switch/Routers. In *Switch/Router Architectures: Shared-Bus and Shared-Memory Based Systems*; Aweya, J., Ed.; John Wiley & Sons: Hoboken, NJ, USA, 2018; pp. 61–86. [CrossRef]
51. Romanchuk, V.; Beshley, M.; Panchenko, O.; Arthur, P. Design of Software Router with a Modular Structure and Automatic Deployment at Virtual Nodes. In Proceedings of the 2017 2nd International Conference on Advanced Information and Communication Technologies (AICT), Lviv, Ukraine, 4–7 July 2017; pp. 295–298. [CrossRef]
52. Lifu, F.; Dongming, Y.; Bihua, T.; Yuanan, L.; Hefei, H. Technique for Network Performance Measurement Based on RFC 2544. In Proceedings of the 2012 Fourth International Conference on Computational Intelligence and Communication Networks, Phuket, Thailand, 24–26 July 2012; pp. 200–204. [CrossRef]

53. Seliuchenko, M.; Beshley, M.; Panchenko, O.; Klymash, M. Development of Monitoring System for End-to-End Packet Delay Measurement in Software-Defined Networks. In Proceedings of the 2016 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET), Lviv, Ukraine, 23–26 February 2016; pp. 667–670. [CrossRef]
54. Klymash, M.; Beshley, H.; Seliuchenko, M.; Beshley, M. Algorithm for Clusterization, Aggregation and Prioritization of M2M Devices in Heterogeneous 4G/5G Network. In Proceedings of the 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S T), Kharkov, Ukraine, 10–13 October 2017; pp. 182–186. [CrossRef]
55. Beshley, M.; Seliuchenko, M.; Panchenko, O.; Polishuk, A. Adaptive Flow Routing Model in SDN. In Proceedings of the 2017 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM), Lviv, Ukraine, 21–25 February 2017; pp. 298–302. [CrossRef]
56. Renteria-Cedano, J.; Rivera, J.; Sandoval-Ibarra, F.; Ortega-Cisneros, S.; Loo-Yau, R. SoC Design Based on a FPGA for a Configurable Neural Network Trained by Means of an EKF. *Electronics* **2019**, *8*, 761. [CrossRef]
57. Frances-Villora, J.V.; Rosado-Muñoz, A.; Bataller-Mompean, M.; Barrios-Aviles, J.; Guerrero-Martinez, J.F. Moving Learning Machine towards Fast Real-Time Applications: A High-Speed FPGA-Based Implementation of the OS-ELM Training Algorithm. *Electronics* **2018**, *7*, 308. [CrossRef]
58. Michałowska, J.; Tofil, A.; Józwik, J.; Pytka, J.; Budzyński, P.; Korzeniewska, E. Measurement of high-frequency electromagnetic fields in CNC machine tools area. In Proceedings of the 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS) IEEE 2018, Lviv, Ukraine, 20–21 September 2018; pp. 162–165.
59. Mazurek, P.; Michałowska, J.; Kozieł, J.; Gad, R.; Wdowiak, A. The intensity of electromagnetic fields in the range of GSM 188 DECT, UMTS, WLAN inbuilt-up areas. *Przegląd Elektrotechniczny* **2018**, *94*, 202–205.
60. Carlsson, P.; Constantinescu, D.; Popescu, A.; Fiedler, M.; Nilsson, A.A. Delay performance in IP routers. In Proceedings of the 2nd International Working Conference (HET-NETs' 04), West Yorkshire, UK, 26–28 July 2004; pp. 15/1–15/10.