Development and Research of an Adaptive Traffic Routing Algorithm Based on a Neural Network Approach for a Cloud System Oriented on Processing Big Data*

Irina Bolodurina[1[0000-0003-0096-2587]], Denis Parfenov[1[0000-0002-1146-1270]], and Vladimir Shardakov[1[0000-0001-6151-6236]]

Orenburg State University, Orenburg, Russia
{prmat,fdot_info}@mail.osu.ru

Abstract. Today, to create modern systems designed to process Big Data in real time, it is necessary to develop scalable solutions based on the adaptive computing infrastructure, capable of providing prompt and efficient service of incoming requests, as well as storing information obtained from many sources. A significant difference in the paradigm of working with Big data is the lack of a clear data storage structure, as well as the heterogeneity of information, flows coming from different sources with high intensity and in large volumes. The scientific problem, the solution of which is the present study, is to increase the efficiency of routing using modern virtualization technologies and machine learning methods for managing information flows. Solving this problem will allow implementing a data analysis system that can adapt to changes in information flows of high intensity due to self-organization of computing resources, self-learning, algorithmic solutions based on artificial intelligence.

Keywords: Virtual network function · Big Data · Neural network · Software-defined network · Cloud computing

1 Introduction

In recent years, the amount of data that needs to be processed and analyzed grows exponentially. Modern technologies used for data processing require the storage of Big Data. For example, for the application of methods of machine learning or neural networks, it is necessary to ensure the storage of reference samples. Thus, a lot of data that were previously considered useless, not stored

* The research work was funded by Russian Foundation for Basic Research, according to the research projects No. 16-37-60086 mol_a_dk, No. 16-07-01004 and No. 18-47-560016, the President of the Russian Federation within the grant for state support of young Russian scientists (MK–1624.2017.9), DAAD and Ministry of Education and Science of Russian Federation cooperation program "Michail Lomonosov" (project number No. 12770.2018/12.2) and the Ministry of Education of Orenburg region (project no. 22 on 31 July 2018.

for long and not processed in automated order now have a high value. Searching for hidden patterns, trends, emissions, automatic analysis of correlations with data from completely different formats and origin helps solve many problems in more simple and quick ways. But the amount of data supplied for analysis, even by a small set of sensors or a server, can be calculated in terabytes per day. Most companies do not have the resources either for real-time analysis or for their long-term storage. One of the possible solutions to overcome this problem is the use of cloud computing technologies.

As a rule, cloud systems are located on the basis of data centers. Such data centers have significant amounts of computing, network and software resources, allowing to scale cloud solutions for processing Big data [1, 14].

Basically, large providers use ready-made solutions for deploying a software and hardware platform for cloud computing. The most popular are the systems for managing cloud resources OpenNebula, OpenStack, CloudStack, Doku, etc. However, existing solutions for the deployment of cloud systems are universal. This does not allow to take into account a number of features of working with large data. The key disadvantages of cloud systems when working with Big Data are:

- the lack of the possibility of parallelizing the operation of the network (alternative routing and switching of communication channels);

- a large amount of consumed memory and calculations for one analysis cycle.

These shortcomings do not allow efficient processing of Big data by specialized services deployed on the basis of an arbitrarily chosen cloud computing infrastructure.

To solve these problems, providers provide users with ready-made sets of cloud services in BigData-as-a-Service format. As a rule, BigData-as-a-Service includes a Hadoop / Spark cluster with storage on HDFS / Hive / Gluster, as well as NoSQL-services like Redis / MongoDB. However, as the practice of using such services shows, end-users use the service not only to calculate statistics or retrieve data. BigData-as-a-Service is often used to build models aimed at predicting data changes or their automatic classification. Such algorithms are difficult to implement on the Hadoop platform. This is due to the fact that the execution time of the model in Hadoop can be comparable with the usual processing on one computing node. Processing data in the cycles Map and Reduce each time initiates access to the file system. Spark also uses a full mesh network architecture and pre-build RDD tables that require additional samples [4]. Such features do not allow creating effective hybrid structures from private and public components. In most cases, the customers of the BigData-as-a-Service service already have everything they need in the internal storage and does not have the desire or the ability to upload them to the cloud for analysis each time.

On the other hand, even large servers with more than 60 cores and terabytes of RAM cannot accommodate the entire amount of data required for processing. For these purposes, as a rule, cluster storage technologies are used. However, this approach is not always applicable to existing methods of analysis and processing of large data (for example, for neural network applications). Distributed execution of neural networks involves great difficulties in synchronizing the layers when the layer is divided into sections. Intermediate storage should have a special logic of work, high performance, as well as a small delay in terms of communication over the network.

To solve the identified problems, this study proposes the use of approaches that optimize the network infrastructure for the intensive traffic flows that arise during the analysis and processing of Big data. The solution of this task is based on the principles of self-management and virtualization of the resources of the heterogeneous infrastructure of the data center. The proposed solution is based on the hybrid application of software-configurable components for adaptive routing of data streams, using the functionality of two breakthrough technologies software-defined network (SDN) and virtual network functions (VNF).

Today, SDN is the most popular and effective approach to the organization of the network for the provision of services on the basis of data centers. The use of this technology is due to a number of advantages. First of all, SDN greatly simplifies the design and operation of the network, since it allows centralized intelligent control at the controller level. Secondly, SDN allows network administrators to quickly configure and optimize network resources based on an aggregated set of data collected in a single location. Thirdly, the use of SDN allows providing protection by dynamically analyzing data flows circulating in a virtual data center [2].

Another technology which used to organize a network based on virtual data centers is the technology of network function virtualization (NFV). Technology NFV offers a new way of designing, deploying network services based on the cloud systems. Virtualization of network functions allows separating network functions, such as NAT, Routers, and Switches from the hardware level [3]. In addition, it allows you to consolidate all the network components necessary to support virtualized infrastructure at the software level. In addition, it allows you to consolidate all the network components necessary to support virtualized infrastructure at the software level.

The problem of organizing adaptive routing for Big data flows can be solved by formalizing an optimization task to maximize the satisfaction of QoS requirements. The scientific novelty of the task is that at the moment there are no effective algorithmic solutions for automatically providing adaptive routing within the allocated network resources for processing Big data.

2 Problem Formulation

As part of this study, an approach combining adaptive routing in processing Big data flows and providing QoS (including minimum guaranteed throughput and maximum guaranteed delay) is proposed.

Let $G(t) = (V, E)$, be an oriented multigraph describing the current topology of the virtual network at some time t. The set of its vertices V is the union of the set of nodes representing the set $V = (vNode, NFV, vStg)$, where $vNode$ - virtual computing nodes that process Big data, NFV - virtual network devices (switches, routers, etc.), as well as $vStg$ - virtual data stores in which the data for analysis are directly placed.

Each arc $e \in E$, corresponds to some network connection between the vertices $Start\,(e) \in V$ and $Stop\,(e) \in V$. There can be several parallel arcs between the two vertices, for example, parallel connections between routers. This provides a variety of alternative routes for data transmission and QoS parameters.

On the set of arcs E two functions are given:

1.$L_{speed} : E \to R^{+} \cup \{0\}$ - a mapping characterizing the current capacity of each arc at time t.

2.$L_{delay} : E \to R^{+} \cup \{0\}$ - The maximum guaranteed delay.

Then each service working in the cloud system and dealing with large data processing has its own mechanism for organizing the communication interaction of traffic flows, which can be represented as the next oriented graph

$$G' = (V', L) \tag{1}$$

where V' - a set of virtual nodes on which the components of a service that processes Big data are deployed, L - arcs pointing to data flows between virtual computing nodes, network devices, and a storage system.

To ensure the quality of service in the cloud network, the tasks on the set of arcs L are three functions:

1.$l_{speed}^{min} : L \to R^{+} \cup \{0\}$ - The minimum guaranteed throughput of data flows corresponding to this arc.

2.$l_{delay}^{max} : L \to R^{+} \cup \{0\}$ - The maximum guaranteed delay.

3.$l_{delay}^{avg} : L \to R^{+} \cup \{0\}$ - estimate the average delay that occurs when processing data flow packets on ports of network devices.

Then the QoS routing and provisioning algorithm must construct a function $\Omega : L \to Path\,(G\,(t))$ such that each data flow $l \in L$, is associated with its transmission path r leading from the vertex $Start\,(l)$ to $Stop\,(l)$. Here, as $Path\,(G\,(t))$, the set of routes between any pairs of vertices in the graph of topology $G\,(t)$, which is actual at time t, is indicated. The function Ω can be described as a vector $R = \left(r_1, \ldots, r_{|L|}\right)$, where $r_i = \Omega\,(l_i)$ is the route for the data flows.

Also, we introduce an additional notation, let $\mu : E \to 2^{l}$ be a function that maps each network connection $e \in E$ to a set of data flows $l \in L$ for which the corresponding routes pass through e, that is, $\mu\,(e) = \{l \in L | r = \Omega \& e \in r\}$.

Obviously, the vector R must contain routes that satisfy the following QoS constraints:

1. The bandwidth of each route r_i, taking into account the effect of other data flows, should not be less than the guaranteed throughput for the flows

$$l_i : \forall r_i \min_{e \in r_i} \sum_{e \in r_i} \left\{ L_{speed} - \sum_{l \in \frac{\mu(e)}{\{l_i\}}} l_{speed}^{min}\,(l) \right\} \geq l_{speed}^{min}\,(l_i)$$

2. The total delay of each route r_i, taking into account the influence of other data flows, should not be greater than the guaranteed delay for the flow

$$l_i : \forall r_i \sum_{e \in r_i} \left\{ l_{delay}^{avg}(e) + \sum_{l \in \frac{\mu(e)}{\{l_i\}}} l_{delay}^{avg}(l) \right\} \leq l_{delay}^{max}(l_i)$$

It should be noted that these restrictions are flexible and to some extent may be violated for some data streams. For example, in the case when the network is overloaded.

There are also strict restrictions on routes $r_i = (e_{i,1}, \ldots, e_{i,n_i})$.

1. r_i really is a route:

$$\forall r_i \forall j = \overline{1, n_i - 1}, Start(e_{i,j}) = Stop(e_{i,j+1}).$$

2. r_i must start at the vertex corresponding to the beginning of arc l_i and end in the vertex corresponding to the end of l_i, i.e.:

$$\forall r_i, Start(e_{i,1}) = Start(l_i) \& Stop(e_{i,n_i}) = Stop(l_i).$$

3. r_i must not pass several times through the same vertex, i.e.:

$$\forall r_i \forall j, k = \overline{1, n_j}, j \neq k \Rightarrow Start(e_{i,j}) \neq Stop(e_{i,k}).$$

Thus, for effective organization of adaptive traffic routing when processing Big data flows, it is necessary to solve two tasks:

- automate the construction of rules for routing traffic, based on data based on data on the current state of the network and the services that are running and the processes running in them;

- optimize the rules for routing traffic to ensure the quality of service in the virtual network.

To solve the first task, it is necessary to perform classifications of network traffic passing through the corporate network. To solve the second problem, we use the iterative method, in which we distinguish two main stages. At the first stage, we will perform the clustering of the rules obtained as a result of solving the first task, and the subsequent deducing of the rules from the clusters. At the second stage, to solve the second problem, we apply the algorithm of conflict-free optimization of the list of routing rules.

3 Model for Constructing Rules for Adaptive Traffic Routing

To implement the presented plan, it is necessary first of all to determine the model of routing rules used to provide the functioning of a virtual network of a cloud system built for processing Big data.

A routing rule is usually a string consisting of certain characteristics of a network connection. Within the framework of the study, the following characteristics were chosen from a variety of characteristics describing network connections: the IP address of the destination network, the network mask, the IP address of the next router, the output interface, and the metric characterizing the priority of using the route. When selecting these characteristics, the routing rule will generally look like:

$$< dst\_ip, net\_mask, gateway, interface, metric > \tag{2}$$

where: *dst_ip* - IP address of the destination network; *net_mask* - network mask; *gateway* - IP address of the next router; *interface* - output interface; *metric* - the metric that characterizes the priority of using a route in the routing table.

To analyze the demand for communication channels and improve routing efficiency, services that collect statistical information are deployed on virtual network nodes. The list of records about the packets passing through virtual network nodes is represented by the form of the set of the following form:

$$R = \{r_k\}, k = \overline{1, n} \tag{3}$$

where n - is the length of the list of records stored on the virtual network node that provides the operation of the cloud platform.

Let's define areas of admissible values for elements of the given vector, namely characteristics of traffic on which collection of statistics of use of communication channels in a network of the software-defined infrastructure will be carried out. They are represented as a list of lines of the form (2). Each such line is represented as a vector of the form:

$$r_k = \{r_{k1}, r_{k2}, r_{k3}, r_{k4}, r_{k5}\}, \tag{4}$$

where: k - number in the list of traffic rules; r_{k1} -– IP address of the destination network;r_{k2} -– network mask; r_{k3} -– IP address of the next router; r_{k4} - output interface; r_{k5} -– the metric that characterizes the priority of using a route in the routing table.

3.1 Algorithm for Deriving Routing Adaptive Rules from Clusters

After reducing the input data to the required form, we formulate the statement of the problem. Given a lot of records about the traffic passing through the virtual network in the form of a set of the type (4), it is required to build a set of non-conflicting rules $R = \{r_i\}, i = \overline{1, m}, m \to min.$

This work assumes the solution of the problem by the iterative method in two stages. The first step is the construction of the initial list of rules R_1 by classifying the set X into two classes. The second step consists of making of the set R_{opt} by clustering the set R_1 with the subsequent deduction of the rules from the clusters, that is, the construction of the set R_2 with the subsequent optimization of this set by the algorithm of conflict-free optimization, that is, the construction of the set R_{opt}.

There are many different methods for solving this problem. In the framework of this study, we will use the classification based on the neural network. This is due to the fact that classification is a classic task for neural network methods. The most applicable for solving such a problem is a multi-layered perceptron type architecture. The number of neurons in the input layer is calculated depending on the input data. In this case, the size of the input layer of the neural network will be 99 neurons. It takes 32 bits to write an IP address, and 16 bits to write ports, the number of protocols considered is 7, therefore, 3 bits are required to write them. To train the selected neural network model, we will use the algorithm for backpropagation of the error.

3.2 Classification of Adaptive Routing Rules

One of the problems with processing Big data is the constant access to a specific resource for obtaining data for analysis. As a result of such activity, there is a rapid growth of traffic and as a result of the load on the processor and the increase in the amount of memory consumed on the virtual network nodes involved in organizing the route. This leads to a drop in performance of the cloud network as a whole. In order to reduce computing, it is necessary to solve the task of compiling a list of rules without losing the characteristics responsible for the performance of the network and the cloud system itself. For these purposes, it makes sense to break routing rules into clusters in order to derive new, generalized rules from them. Let's formulate the mathematical formulation of the rules of firewall clustering tasks.

Let there be given a set of rules $R = \{r_i\}, i = \overline{1, m}$. It is required to compose a sample partition into disjoint subsets called clusters in such a way that each subset consists of objects that are close in some metric. It is necessary to compose a clustering function $f(r) : R \to Y$, that assigns to each element of the set R an element of the set $Y = Y_1, Y_2, \ldots Y_m$ - the set of cluster numbers.

An important aspect in solving the clustering problem is the choice of the distance function, or metric. The metric is a measure of proximity, which is algorithms. As part of the study, the Euclidean distance was taken as the metric by the following characteristics: the source and destination IP addresses, the corresponding ports, the protocol by which the connection is made and the decision on the admissibility of the connection. Thus, the formula for the distance function has the following form:

$$D(r_1, r_2) = \sqrt{a\left(r_{1,2} + r_{2,2}\right)^2 + b\left(r_{1,3} + r_{2,3}\right)^2}, \tag{5}$$

This function was used with empirically selected parameters $a = 0,55$, $b = 0,55$.

3.3 Clustering of Adaptive Routing Rules

The next step is to remove the rules from the clustered list of rules. Input data will be a list of rules, broken into clusters R_{kl}. The output of the algorithm is expected to list the generalized rules R_{opt}. In the framework of this study, an algorithm was developed. The list of clusters is denoted by u. Then for all clusters from u, we have:

Input: Clustered rule list $R_{k,l}$, list of clusters u.
Output: Generalized list of rules R_{opt}.
$r_{i,1} = min\left(r_{k,1}\right)/mask = 32 - log_2\left(max\left(r_{k,1}\right) - min\left(r_{k,1}\right)\right)$;
$r_{i,2} = min\left(r_{k,2}\right)/mask = 32 - log_2\left(max\left(r_{k,2}\right) - min\left(r_{k,2}\right)\right)$;
$r_{i,3} = \{r_{1,3}, \ldots, r_{k,3}\}$;
$r_{i,4} = \{r_{1,4}, \ldots, r_{k,4}\}$;
$r_{i,5} = \{r_{1,5}, \ldots, r_{k,5}\}$.

Algorithm 1: CLUSTERING of adaptive routing rules

3.4 Algorithm for Optimizing the List of Rules by Ranking Sorting

An important parameter for the list of routing rules, in addition to the breadth of resource coverage and the size of the list, is also the arrangement of rules. To ensure the quality of service on the network, it is necessary to apply a sorting of the list of rules in order to place the least loaded routes in the top of the list. To solve this problem, the following algorithm was developed. Let's describe it.

Input: - A set R of traffic headers, passing through a virtual network device;
- Non-conflicting list of rules $\mathbf{R} = \{R_i\}, i = 1, n$;
Step 1: To assign each rule and each element of the set R, the weight by formula $w_i = \frac{k_i}{k}$, where k_i - the number of traffic passes by the rule i; k - total number of packets passing through the network.
Step 2:
if $w_n > w^*$ **then go to Step 3**;
else go to Step 5;
Step 3: Create a rule R_{n-1}, according to the current flow.
Step 4: Sort the set $\mathbf{R}$ by its value, place the most popular rule on the last place, assigning the corresponding metric **go to Step 2**.
Step 5: End.

Algorithm 2: ALGORITHM for optimizing the list of rules by ranking sorting

This algorithm does not reduce the size of the routing list, but allows you to make a more optimal list of rules.

4 Experimental Results

Based on the proposed solution, the module implemented adaptive routing for a software-configurable network. The software is implemented as a virtual network function based on the Open Platform for NFV (OPNFV), assembled as a Docker container.

Comparative analysis was carried out by comparing the results of traditional routing methods for a software-defined network and a developed software module. Before the experiment, the developed software module was trained, and a comparable set of rules for traditional routing methods was developed, which allows for a correct comparison of the compared means. Within the framework of the study, work was evaluated under different loads for two key indicators: response time in the network; load the central processor on the NFV Router.

For carrying out of the load, test of scenarios. It includes 4 OpenFlow switches (2 HP 3500yl, 2 Netgear GSM7200), 8 computing nodes (32Gb RAM, 4 cores), 1 server (32Gb RAM, 8 cores) with OpenFlow controller and 1 server (32Gb RAM, 4 cores) for monitoring function. As a selected fat tree topology with three levels. Routers are connected with the speed of 1000 Mbit / s, and the computers are connected to the third level. In this infrastructure was prepared 100 virtual machines. Also we include one node that controls the data flows. We select random five virtual machines (service host). Load is created using a specially designed tool - hping3. It allows you to generate packets of a certain type in certain directions, which makes possible on different values of the number of packets, and as a consequence. The experiment consists in the sequential measurement of the indicators with a consecutive increase in the intensity of traffic flows in single-load frames in the range from 20 to 350 Mbit / s.

Response time is one of the key parameters when analyzing network performance and network resources. It shows how much time passes from the moment the request is sent by the user until the service responds to the request. In accordance with the experiment rules, the response time from the network was removed.

The list of routing rules formed by the built module under load, including avalanche, gives a response time difference of up to 20%, which indicates the effectiveness of the approach developed in the work.

The load on the CPU of the virtual network device is an important parameter in ensuring the quality of service in the network supporting large data processing. This is an indication of whether the device is loaded, or inefficiently used. Based on the obtained results, we have that an optimized set of rules built using the approach considered in the study, has a reduced load on the firewall processor.

5 Discussion

The study of the problems of route optimization and maintenance of quality of service in data transmission networks involved many scientists from different countries.

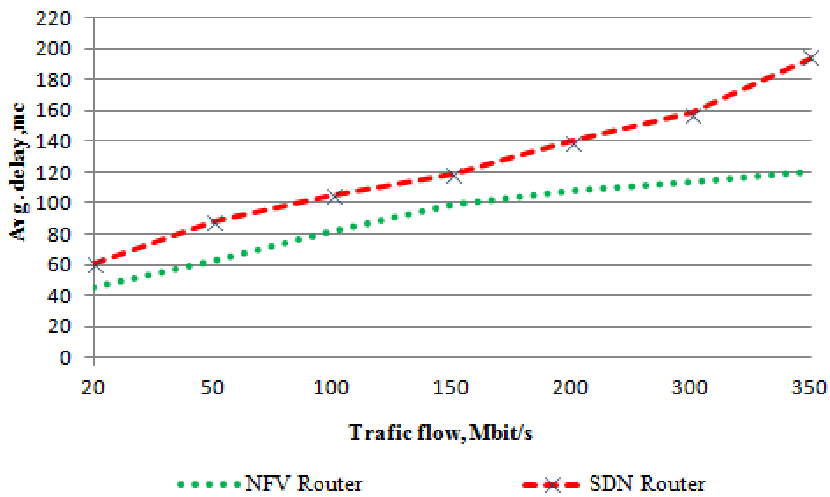

Fig. 1. The delay in the network for different speed of flows

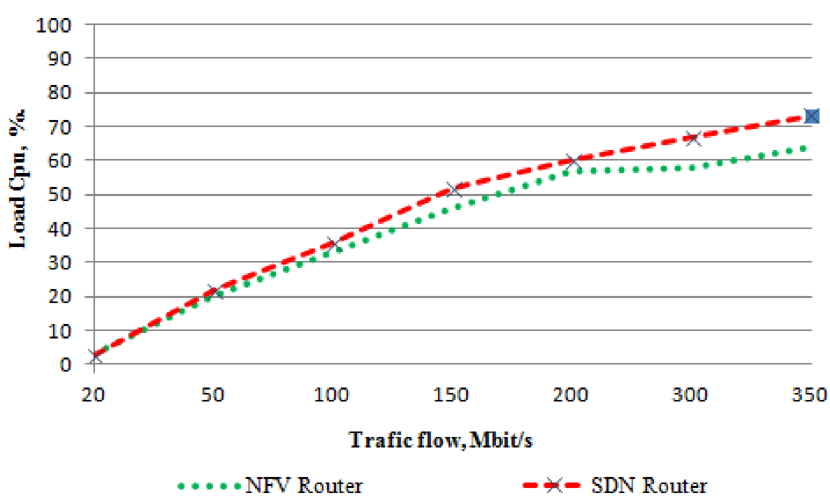

Fig. 2. The load CPU for different speed of flows

For example, authors of the publication [5] proposed a solution for reducing the control traffic generated, as a rule, in the absence of a suitable routing rule. Researchers use the buffer_id function of the OpenFlow protocol, designed specifically to identify individual buffered packets inside the switch. Thus, the OpenFlow controller can only send the first packet from the stream to the controller and buffered the remaining packets until the controller responds or a timeout occurs. The results of the conducted studies demonstrate the reduction of the control traffic.

In work [6] of Turkish scientists, a joint mechanism for selecting a server and a route for SDN networks is proposed, which takes into account the current and previous statistics of the use of network resources for each incoming stream. The presented mechanism periodically automatically updates the network state metric of the routing protocol in order to achieve load balancing in the network. The proven concept is implemented using the Mininet emulator.

To define rules for routing network traffic and ensuring quality of service, we need to determine what types of traffic are present in a particular network. To this goal, a number of researchers proposed approaches based on the classification of traffic [7, 9]. Classification of network traffic plays a significant role in the field of network management. This is due to the fact that on the basis of classifiers, packet filters are used that perform application identification. Also on the basis of classification determine the relevance of network resources. The information obtained is used not only to control the routing of traffic, but also in the algorithms of automated intrusion detection systems.

In a joint study of scientists Park J., Tyan H.-R., and Kuo C.-C., a new scheme for classifying network traffic is proposed. The proposed approach does not take into account the features of asymmetric routing and errors in modern measuring instruments. To address these issues, the authors propose a two-stage, scalable classification of traffic. At the first stage, the functions for classification are selected. At the second stage, the authors propose to use the method of reducing the dimension of the classified sample. This will reduce resource intensity and increase productivity [9].

Using multipart modification of the TCP (multipath TCP, MPTCP) protocol in the data centers provides an increase in application performance. The publication of Turkish scientists is devoted to the description of the architecture using MPTCP based on PCS with the aim of guaranteeing customer service in the issues of streaming video [10]. The number of MPTCP subflows and their routes based on the mixed integer linear programming model. The conducted experiments show an increase in throughput and a decrease in the duration of failures in the transmission of the video stream.

Amin Vahdat and its colleagues present design and experience with Google Cloud Platform's network virtualization stack. Authors solutions take into account a variety of factors including performance isolation among customer virtual networks, scalability, rapid provisioning of large numbers of virtual hosts, bandwidth, and latency. But they do not take into account the loud on the network devices and QoS requirements in the network [11].

The researchers Lemeshko et al. in its works prepare the tensor model of QoS-routing, where an accounting of the network state. They reduce the QoS-routing problem to the solution of the optimization problem associated with differentiated maximization the probability of timely delivery of packet flows in the network having a different service priority [12]. But these approaches do not take into account the dynamic characteristics of traffic between network devices.

In the other work by Shuangyin Ren et al. prepared solution for End-to-End QoS guarantee algorithm for traffic routing based on deterministic network calculus. They realized algorithm in Mininet with Open vSwitch as forwarding switch and Ryu as the remote controller. The main idea of the solution is to run Hierarchical Token Bucket queuing discipline to manage bandwidth resource. The main problem with this solution is the lack of consideration of the load on the communication channels at various times [13].

The analysis of scientific sources on the topic of the study has shown that:

a) so far, there are no effective algorithmic solutions for planning traffic routing, application-oriented accounting topology of the computer system, and QoS requirements in communication tasks schemes;

b) the existing methods of data flow routing can be enhanced by taking into account the QoS requirements, but the need for fast solutions to analyze the current state of links in the network.

This demonstrates the novelty of the solutions offered by the project. Thus, the development of new methods and algorithms to improve the efficiency of adaptive traffic routing in the cloud system infrastructure for Big data analysis is a crucial task.

6 Conclusion

The survey reviewed existing solutions in the field of network routing. The architecture of the neural network was selected to achieve this goal, namely, a hybrid artificial neural network consisting of two networks, namely a classifier based on a multilayer perceptron and a clustered based on the Kohonen network, the existing approaches, and algorithms for solving the problem of providing adaptive routing in the virtual network. In the framework of the study, a model for describing the rules for adaptive routing was developed, and on the basis of these model algorithms for automatically constructing and optimizing the list of routing rules were developed. In the projected architecture, an algorithm of conflict-free optimization was implemented, which produces the final optimization by ranking and deducing the most frequently used routes, which allows improving the quality of service in the network. The algorithms and methods considered were implemented as a virtual network function for a software-defined network.

During the tests, it was revealed that the approach proposed by this study gives an increase in two key parameters, namely, response time, on average by 20%, with an increase in load and load of the central process of the shielding device, by an average of 4.5% load growth. Thus, the developed approach is effective for solving practical problems.

References

1. Bolodurina, I., Parfenov D.: The development and study of the methods and algorithms for the classification of data flows of cloud applications in the network of the virtual data center. International Journal of Computer Networks and Communications **2**10(2), 15–22 (2018)
2. Pashkov V., Shalimov A., Smeliansky R.: Controller failover for SDN enterprise networks. In: Proceedings on 2014 International Science and Technology Conference (Modern Networking Technologies) (MoNeTeC), pp.1–2. IEEE Press, Moscow, Russia (2014)
3. Qu L., Assi C., Shaban K., Khabbaz M.: Reliability-aware service provisioning in NFV-enabled enterprise datacenter networks. In: Proceedings on 12th International Conference on Network and Service Management (CNSM), pp.153–159. IEEE Press, Montreal, QC, Canada (2016)
4. Ahmed Oussous et al.: Big Data technologies: A survey. Journal of King Saud University - Computer and Information Sciences, 1–18 (2017)
5. Atli A.-V. et al.: Protecting SDN controller with per-flow buffering inside OpenFlow switches. In: Proceedings on 2017 Black Sea Conference on Communications and Networking (BlackSeaCom), pp.1–5. IEEE Press, Istanbul, Turkey (2017)
6. Akyıldız H.-A. et al.: Joint server and route selection in SDN networks. In: Proceedings on 2017 Black Sea Conference on Communications and Networking (BlackSeaCom), pp.1–5. IEEE Press, Istanbul, Turkey (2017)
7. Dainotti A., Pescape A., Kimberly C.: Issues and future directions in traffic classification. Network IEEE **2**26(1), 35–40 (2012)
8. Callado A., Kamienski C., Fernandes S.-N., Sadok D.: A survey on internet traffic identification and classification. IEEE Comm. Surveys and Tutorials **2**11(3), 37–52 (2009)
9. Park J., Tyan H.-R., and Kuo C.-C.: Internet traffic classification for scalable QoS provision. In: Proceedings on IEEE International Conference on Multimedia and Expo, pp.1221–1224. IEEE Press, Toronto, Ont., Canada (2006)
10. Herguner K., Kalan R.-S., Cetinkaya C., Sayit M.: Towards QoS-aware routing for DASH utilizing MPTCP over SDN. In: Proceedings on 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), pp.1–6. IEEE Press, Berlin, Germany (2017)
11. Dalton M. et al.: Andromeda: performance, isolation, and velocity at scale in cloud network virtualization. In: Proceedings on 15th USENIX Symposium on Networked Systems Design and Implementation, pp.373–387., USENIX Association, Renton, WA (2018)
12. Lemeshko O., Yeremenko O., Hailan A.-M.: Design of QoS-routing scheme under the timely delivery constraint. In: Proceedings on 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM), pp.97–99. IEEE Press, Lviv, Ukraine (2017)
13. Shuangyin Ren, Wenhua Dou, Yu Wang: A deterministic network calculus enabled QoS routing on software defined network. In: Proceedings on IEEE 9th International Conference on Communication Software and Networks (ICCSN), pp.1–5. IEEE Press, Guangzhou, China (2017)

14. Parfenov D., Bolodurina I.: Methods and algorithms optimization of adaptive traffic control in the virtual data centerIn: Proceedings on 2017 International Siberian Conference on Control and Communications (SIBCON 2017), pp.1–6. IEEE Press, Astana, Kazakhstan (2017)