ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 7, Issue 9 , September 2020

Algorithm of Distributing Traffic Flows in a Software-Defined Network

Salimjon Olimjonovich Mahmudov

Senior teacher, Department of Data Communication Networks and Systems, Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Tashkent, Uzbekistan

ABSTRACT: The article suggests transporting packets of heterogeneous traffic in a multi-service network on the basis of a Software Defined Networks (SDN). The optimization problem of transmitting flows of heterogeneous traffic at the level of the SDN infrastructure in case of its load is formulated. The criterion used is the total delay time of the threads. Development of algorithms for search and analysis of traffic flows.It is proposed that packets of heterogeneous traffic are transported at the multiservice network transport level to organize on the basis of the SDN, the expediency of organizing the transmission of packets of heterogeneous traffic is not justified on one, but on different virtual channels.

KEY WORDS: Network, Model, Algorithm, Flow distribution, Traffic.

I. INTRODUCTION

The sharp increase in the volume of external traffic dictates the need for a phased introduction of Software Defined Networks (SDN) into traditional networks, the main idea of which is to separate the control level from the data transfer level, which involves transferring a number of basic control functions from routers and traditional switches network to a centralized system. A network operating system installed on a dedicated server (controller), with the help of certain programs and protocols, solves the problem of traffic control and routing. The development and implementation of SDN solutions requires the improvement of routing protocols and flow control processes. This article is devoted to this actual problem, in which the optimization problem of the distribution of heterogeneous traffic flows under network congestion conditions is formalized [1, 4, 5, 7, 8].

II. FORMULATION OF A PROBLEM

The software-defined network architecture has three levels: network infrastructure level — includes a set of network devices (switches, routers) and data transfer channels; management level — monitors and maintains the global network view (the global network view is the network topology and the status of network devices); network application layer — implements various network management functions: network data flow management, security management, traffic monitoring, service quality management, policy management, and so on [9, 10].

The controller interacts with network devices of the infrastructure level using the OpenFlow protocol, which allows you to process network loads dynamically.

The essence of the problem of traffic control and determining the route paths solved at the control level of the SDN is to determine the shortest paths between the switches of the SDN infrastructure level and transfer the calculated information to the corresponding switches via a special secure channel.

The solution to this problem is advisable to implement on the basis of a flow model based on graph theory. With this approach, the network structure is represented as a graph $G\,\{M, Z\}$ (where M is a non-empty set of vertices and Z is a set of edges, such that any $a \in A$ has a finite number of ordered pairs of elements (i, j), $(i, j \in M)$, which determine the beginning and end of the edge), each edge (i, j) corresponds to the probability distribution function of the transmission time of the application on the channel (edge) $F_{i,t}(t)$. Input flows are described by the distribution function of the moments of receipt of requests to the network, i.e. to the node (switch) i for node $j - \psi_{i,j}(t)$. Input and output flows are assigned to nodes associated with consumer terminal devices.

The routing procedure in the flow model is determined by the matrix of route variables:

$$\Phi = \left\|\varphi_{i,j}^{(k)}(t)\right\|, \tag{1}$$

where, $\varphi_{i,j}^{(k)}(t)$ is the probability of sending a message (packet) on node *i* at time *t* addressed to node *k* by edge (*i*, *j*), *k = 1, 2,. . . M*, *(i, j)* ∈ *Z*. The route variable $\varphi_{i,j}^{(k)}(t)$is interpreted as a fraction of the stream at node *i*, addressed to node k and sent in direction *(i, j)*. It's obvious that,

$$\sum\nolimits_{j \in M} \varphi_{i,j}^{(k)}(t)=1,\ \ \varphi_{i,j}^{(k)}(t) \geq 0. \tag{2}$$

The intensity of the resultant flow through node i to node j is determined from the expression:

$$\varepsilon_{i,j} = \gamma_{i,j} + \sum\nolimits_{k \in M} \lambda_{k,i}^{(t)}, \tag{3}$$

where, $\gamma_{i,j}$ is the intensity of the input stream in the *i*-node intended for the j-node; $\lambda_{k,i}^{(t)}$ is the intensity of the flow at the *k* - node (*k = 1, 2, ... M*) intended for the *j* - node and passing through the *i* - node.

The intensity of the stream flowing along the edge (i, j) from all the k - nodes is equal to:

$$\lambda_{i,j} = \sum\nolimits_{k \in M} \lambda_{k,i}^{(t)} \tag{4}$$

(note that in the last expressions i and f have a different meaning).

Flow and route variables are related as follows:

$$\left.\begin{array}{r} \lambda_{i,j} = \sum \varphi_{i,j}^{(k)} \varepsilon_{i,j} \text{by } k \in M \\ \varphi_{i,j} = \lambda_{k,i}^{(t)} / \varepsilon_{i,k}\ \ by\ \ \varepsilon_{i,k} > 0 \end{array}\right\} \tag{5}$$

These expressions for the fluxes $\lambda_{i,j}$ is a system of linear equations with known input streams. Its solution will allow determining the mart-smart network table (the route table of node *i* is a set of all routes by which information can be delivered to node *i* with a given set $\left\{\varphi_{i,j}^{(k)}\right\}$.

There are many methods for solving this problem, in which a number of interesting methods and algorithms for constructing route tables according to the chosen criterion are proposed. Without dwelling on them, we note that in the overwhelming majority of cases, the task of finding effective flow distribution options is solved on the basis of satisfying the needs of consumers in all respects, that is, the number and throughput of channels and nodes, the number and performance of powerful servers, etc. - that is, the entire network structure is selected from the "need". In reality, these characteristics are determined from the "possibilities", i.e., the quantitative and qualitative parameters of the components of the computer network structure are ultimately finite (everything is specified less than necessary). Therefore, it is necessary to search for the desired network parameters precisely from these positions, since in this case, apart from the quantity and quality of the network's hardware and technical resources, their every component performs its functions in the best way for this mode. This is directly related to the SDN, in which the bandwidth of the channels is limited and it is necessary to rationally distribute them in order to organize the transfer of heterogeneous traffic flows through them.

III. THEORY

The section formalizes the optimization task of distributing flows of heterogeneous traffic at the level of the SDN infrastructure. So let them be given:

- the topological structure of the SDN infrastructure level, which consists of *N* nodes, *M* communication channels, and *K* sender-receiver (SR) pairs.
- communication channels and nodes are imperfect and have finite noise immunity, the characteristics of the channels are set, i.e. bandwidths of communication channels C_i are known;
- the nodes (switches or routers), including a single-channel service device, receive three incoming queues of requests, that is, flows (traffic) of three categories:
- packet traffic with an arrival rate of γ_1, in which their losses during transmission are allowed;
- packet traffic with an arrival rate of γ_2, which may be delayed and in which some lost packets may be recovered;
- packet traffic with the arrival rate γ_3, which requires end-to-end transmission and does not allow losses;
- in each pair, the sender-receiver (SR) can be in *Z* traffic classes. Between the *k*-th pair of SR there are $u^{(k)}$ possible paths;
- sources of network latency is both communication lines and nodes;
- the packet length in the network is fixed;
- external traffic is stationary, i.e. flow intensities do not have a dynamic over time.
- traffic packets (flows) are independent and cannot move from one queue to another.

The task of distributing flows for a network with *Z* traffic classes will be formulated as follows: for given bandwidth capabilities Ci and flows γ_z ($z \in Z$), find for each SR pair a set of flows and traffic classes and for each link a set of flows so that the total average delay of flows (traffic) in the network of the SDN Ť infrastructure should be minimal with the following restrictions:

1) all incoming flows to the SDN infrastructure level must pass through the appropriate switches, subject to the condition of preserving flows;
2) total values of flows in communication channels should not exceed their carrying capacity;
3) in each switch to which a stream is directed, a single outgoing channel should be chosen, through which it will leave the switch;
4) for counter-incoming flows $\lambda^{(i,j)}$ and $\lambda^{(j,i)}$ matching routes should be provided.

The formal description of this statement was made on the basis of the above described flow model. To do this, we represent the investigated topology of the SDN infrastructure level in the form of an undirected graph $G\,\{N, M\}$ without loops and multiple arcs with limited throughput capabilities of arcs $C\,\{c_m\}$. For each arc *(i, j)* there is a counter arc *(j, i)*. We introduce the following notation:

γ_z - the intensity of the input stream (external average traffic) of the *z-th* class;
f_z^k - the intensity of the total flow (average traffic) of the *z-th* class between the *k-th* pair SR;
λ_{zi} - the total intensity of the *z*-class class flow in the *i-th* communication channel;
μ_i- the average service time of packets in the *i-th* communication channel;
λ_i - the average intensity of flows in the *i - th* communication channel.

For total input flow γ, its corresponding intensities are defined as:

$$\gamma_1 = \sum_{i=1}^{N}\sum_{j=1}^{N}\gamma_{1\,i,j},\ \gamma_2 = \sum_{i=1}^{N}\sum_{j=1}^{N}\gamma_{2\,i,j},\ \gamma_3 = \sum_{i=1}^{N}\sum_{j=1}^{N}\gamma_{3\,i,j}, \qquad (6)$$
$$\gamma_{i,j} = \gamma_{1\,i,j} + \gamma_{2\,i,j} + \gamma_{3\,i,j}$$

According to the flow model above, the routing rules are determined by the matrix of route variables. Then the average intensity of the total flow of all classes through the i-th node addressed to the j-th node will be equal to:

$$V_{i,j} = \gamma_{i,j} + \sum_{q=1}^{N} V_{q,i}^{(j)} P_{q,i}^{(j)} \qquad (7)$$

where, $P_{q,i}^{(j)}$ is a route variable (matrix) that determines the proportion of the total flow in the *q-th* node $V_{q,i}^{(j)}$ sent from it to node *j* on route (*i*, *j*) (through node *i*). Expression (7) in the theory of flows is called the balance equation.

The average intensity of the total channel stream of all classes along the edge (*i*, *r*) for the j node is determined by the expression:

$$\lambda_{i,r} = V_{i,j}P_{i,r}^{(j)} = (\gamma_{i,j} + \textstyle\sum_{q=1}^{N} V_{q,i}^{(j)} P_{q,i}^{(j)}) P_{i,r}^{(j)}, \tag{8}$$

where, $P_{i,r}^{(j)}$ is the route variable on the channel (*i*,*r*) of the route (*i*, *j*).

Based on the general model, the flow rate (traffic) of the *z*-th class in the *i*-th communication channel will be:

$$\lambda_{z,i} = \textstyle\sum_{k=1} \sum_{j=1} \delta_{i,j}^{(k)} h_{z,i}^{(k)}, \tag{9}$$

$$f_z^k = \textstyle\sum_{j=1}^{u^{(k)}} h_{z,i}^{(k)}, \tag{10}$$

where, $\delta_{i,j}^{(k)} = 1$, if the *i*-th channel lies on the *j*-th path between the *k*-th SR, and $\delta_{i,j}^{(k)} = 0$ otherwise;
$h_{z,i}^{(k)}$ is the flow rate (average traffic) of the *z*-th class of the *j* -th path between the *k*-th SR pair;
f_z^k- full average traffic of the *z*-th class of the k-th pair of the SR. Between the *k*-th pair of SR there are $u^{(k)}$ possible paths.

The intensity of the total flow in the i - th communication channel is found as the sum of the flows of all types flowing through this channel:

$$\lambda_i = \textstyle\sum_{i \in Z} \lambda_{z,i}. \tag{11}$$

The total average delay of flows in the network in its most general form will be equal to:

$$\bar{T} = 1/\gamma \textstyle\sum_{i=1}^{M} \sum_{z=1}^{Z} \lambda_{z,i} T_{z,i}, \tag{12}$$

$$\gamma = \textstyle\sum_{k=1}^{K} \sum_{z=1}^{Z} f_z^k, \tag{13}$$

where, $T_{z,i}$ is the average delay of the flow (traffic) of the *z*-th class in the *i*-th channel; γ - total intensity of flows *Z* classes of all *K* pairs sender - recipient.

If we assume that the structure of the infrastructure level is a single system, the communication channels and switches included in it are absolutely reliable and robust, with stationary flows with a total intensity γ with exponential distribution of message arrival times (packets) at its input, then the average delay time can be represented by a function:

$$\bar{T} = 1/\gamma \textstyle\sum_{i=1}^{M} {}^{\lambda_i}/_{(\mu_i - \lambda_i)} \tag{14}$$

and, the problem of distribution of different types of flows can be reduced to the well-known problem of convex programming, which is formulated as follows: minimize the convex function (14) with the following restrictions:

$$\begin{cases} \lambda_{z,i} \geq 0, \lambda_i \leq \mu_i \\ \sum_{j=1}^{N} \sum_{z=1}^{Z} \lambda_{z,i\,j}^{(k)} + \sum_{z=1}^{Z} \gamma_{z,i} + \sum_{j=1}^{N} \sum_{z=1}^{Z} \lambda_{z,j\,i}^{(k)} \\ \gamma_{ij} = \gamma_{1\,ij} + \gamma_{2\,ij} + \gamma_{3\,ij}, \gamma_{1\,ij}, \gamma_{2\,ij}, \gamma_{3\,ij} \geq 0 \end{cases} \tag{15}$$

Based on the solution of the problem, it is possible to determine which traffic classes in which proportion can be simultaneously transmitted on each communication channel and on each route. In other words, the result of solving an optimization problem is to determine for each *k-th* pair of SRs and *z-th* traffic class a uniquely ordered set of paths with positive flows, ensuring a minimum of the total average delay. The physical meaning of this solution is that the flows will follow paths for which the end-to-end transmission delay is minimal and the transmission of flows along other paths leads to an increase in the total delay.

When implementing the algorithm, the key element is the task of determining the shortest paths between the switches of the level of the SDN infrastructure with a given topological structure.

In graph theory, a number of effective methods and algorithms for finding the set of shortest paths are known, but this task does not lose its relevance due to its multivariance and today. Below is a description of the developed algorithm for finding the shortest paths.

Let given:

topological network structure consisting of n - nodes and m - communication channels; bandwidth communication channels C_i, (where i = 1, n). They can be specified as a distribution and $\{Ci\}_m$, then the edge weights are the mathematical expectations of a given distribution.

It is required to determine the K-shortest paths between network nodes.

We represent the given topological structure as a graph G = {A, B}, where A = [$\alpha_1, \alpha_2, ..., \alpha_m$] is the set of vertices, m is the number of vertices; B = [$\beta_1, \beta_2, ..., \beta_m$] is the set of edges connecting the vertices, n is the number of edges. The structure of this graph is presented in the form of the following rows:

$\alpha_1, \alpha_2, ..., \alpha_m$ - numbers of outgoing vertices;
$\beta_1, \beta_2, ..., \beta_m$ - numbers of incoming vertices;
$d_1, d_2, ..., d_m$ are the lengths between outgoing $\{\alpha_m\}$ and incoming $\{\beta_m\}$ vertices of the series;
$\gamma_1, \gamma_2, ..., \gamma_m$ is a switch that indicates whether the j-arc is scanned when determining the next shortest path.

The essence of the algorithm is as follows:

1. For each vertex, the number of outgoing rays (edges) K_j , $j \epsilon A$ is determined.
2. The next vertices N_V and K_V (the beginning and the end of the vertex) are specified, between which the shortest path must be calculated, the auxiliary variables of the RP(i) are cleared.
3. i = i + 1; RP (i) = N_V; PD = 0.
4. Of the fourth row is determined not included switch corresponding to RP (i). However, situations are possible:

a) all Kj switches are in the “on” state and RP (i) points to the initial vertex - transition to operator 8 is performed;

b) when RP (i) switches are not on the initial vertex — the entire set of switches is switched to the off state, the computational path length is reduced by the value of the corresponding element of the third row and after i = i is assigned, control is transferred to the beginning of operator 4;

c) in the fourth row there is an off element - it is transferred to the “on” state and the following operations are performed:

- if on the array RP (i) there is no element from the second row with a number equal to the number of the switched on switch, then i = i + 1, RP (i) is assigned a checked element from the second row and PD value increases by the length of the arc from the third row, the number of which corresponds to the number of the switched on switch and control is transferred to the operator 5;
- in RP (i) there is an element from the second row - control is transferred to the beginning of operator 4;

5. If RP (i) = K_V - control is transferred 6, otherwise 4.
6. If PD OD, then OD = PD and array OP are assigned to the elements of the RP array (where OD is the optimal length, PD is the intermediate length, RP is the intermediate path, OD is the optimal path).
7. j = j - 1, reduces the path length by the length of the last added arc and transfers control to 4.
8. Checks whether all the shortest paths are found between vertices of the graph and, if not, transfers control 2, otherwise, prints the results and stops the algorithm.

IV. EXPERIMENTAL RESULTS

The algorithm was tested on the topological structure of an experimental network consisting of twenty nodes and thirty-eight channels (Fig. 4.1). The bandwidths of the communication channels (edges) were specified in the form of distribution of message delivery time with an average value of {ti} n.

Table1 summarizes the results of the experiment on the distribution of the shortest paths between nodes 1 and 20. 5000 implementations were carried out. In each case, the value of the message delivery time was determined according to the specified distribution, that is, the "whole edge" was determined and the program for searching the shortest paths was launched. The table summarizes the numbers of nodes along which the shortest path passes between 1 and 20, the number of nodes in the path, and the probability of choosing this path [2, 3].

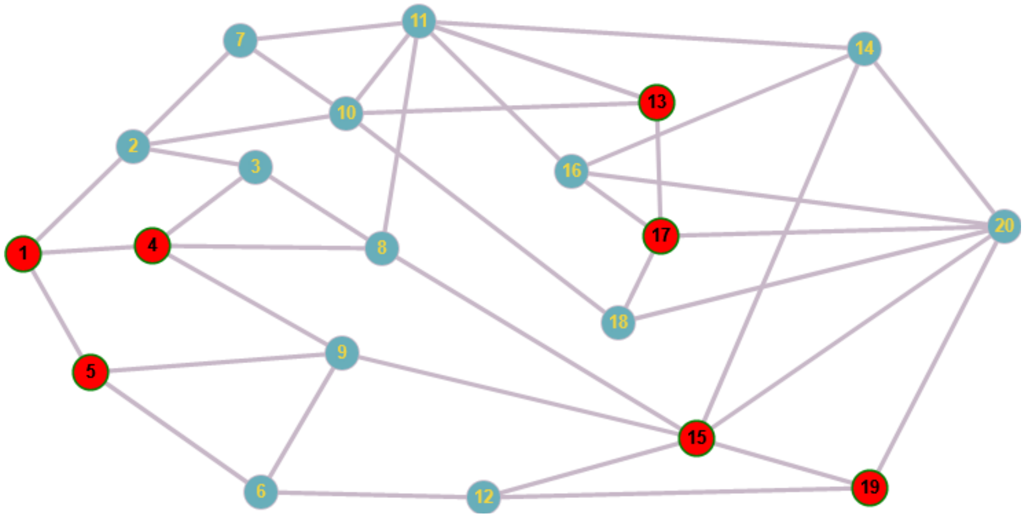

Fig. 4.1 Experimental network topology.

Table1. Summarizes the results of the experiment

№	Network node numbers											Num. of node in transit	The probability of choosing this path
1	1	4	8	11	16	17	20	0	0	0	0	7	0.3018
2	1	4	8	15	20	0	0	0	0	0	0	5	0.296
3	1	2	10	18	20	0	0	0	0	0	0	5	0.056
4	1	4	8	11	16	20	0	0	0	0	0	6	0.054
5	1	4	3	8	15	20	0	0	0	0	0	6	0.05
6	1	2	3	8	11	16	17	0	0	0	0	7	0.046
7	1	2	3	8	15	20	0	0	0	0	0	6	0.042
8	1	4	3	8	11	16	17	20	0	0	0	8	0.038
9	1	4	8	11	13	17	20	0	0	0	0	7	0.022
10	1	4	3	8	11	16	20	0	0	0	0	7	0.01
11	1	2	7	10	18	20	0	0	0	0	0	6	0.008
12	1	5	9	4	8	15	20	0	0	0	0	7	0.008
13	1	4	8	11	14	20	0	0	0	0	0	6	0.006
14	1	2	7	11	16	20	0	0	0	0	0	6	0.006
15	1	4	9	15	20	0	0	0	0	0	0	5	0.006
16	1	4	8	11	10	18	20	0	0	0	0	7	0.006

The shortest paths with a probability of less than 0.005 are excluded from consideration and are not listed in the table. Thus, if you want to define 5 different shortest paths between nodes 1 and 20, then they will be the first five options. Variants of the shortest paths between all other nodes of the network are determined in the same way.

The time spent to calculate the shortest paths based on the developed algorithm is 3.2 percent less than the search time based on the Dijkstra algorithm, To determine the shortest paths between the nodes of the K network, the above approach was used, although other methods can be used.

In this paper, the results obtained on the basis of the proposed algorithm were compared with the well-known Kleinrock algorithm for flow deviation.

Given the higher priority of voice traffic, the task of distributing voice traffic was primarily addressed. Further, the algorithm of distribution of data streams was performed at different load of channels with transmission of voice information. Some results of the computational experiment are shown in Fig. 4.2 (a, b, c) and 3. The parameter α characterizes the degree of network load by voice traffic.
As can be seen from the figures, the growth of voice traffic significantly increases the delay of data packets (from 0.15 to 0.55 when the load of voice traffic changes from α = 0 to α = 0.65 (Fig. 4.3).

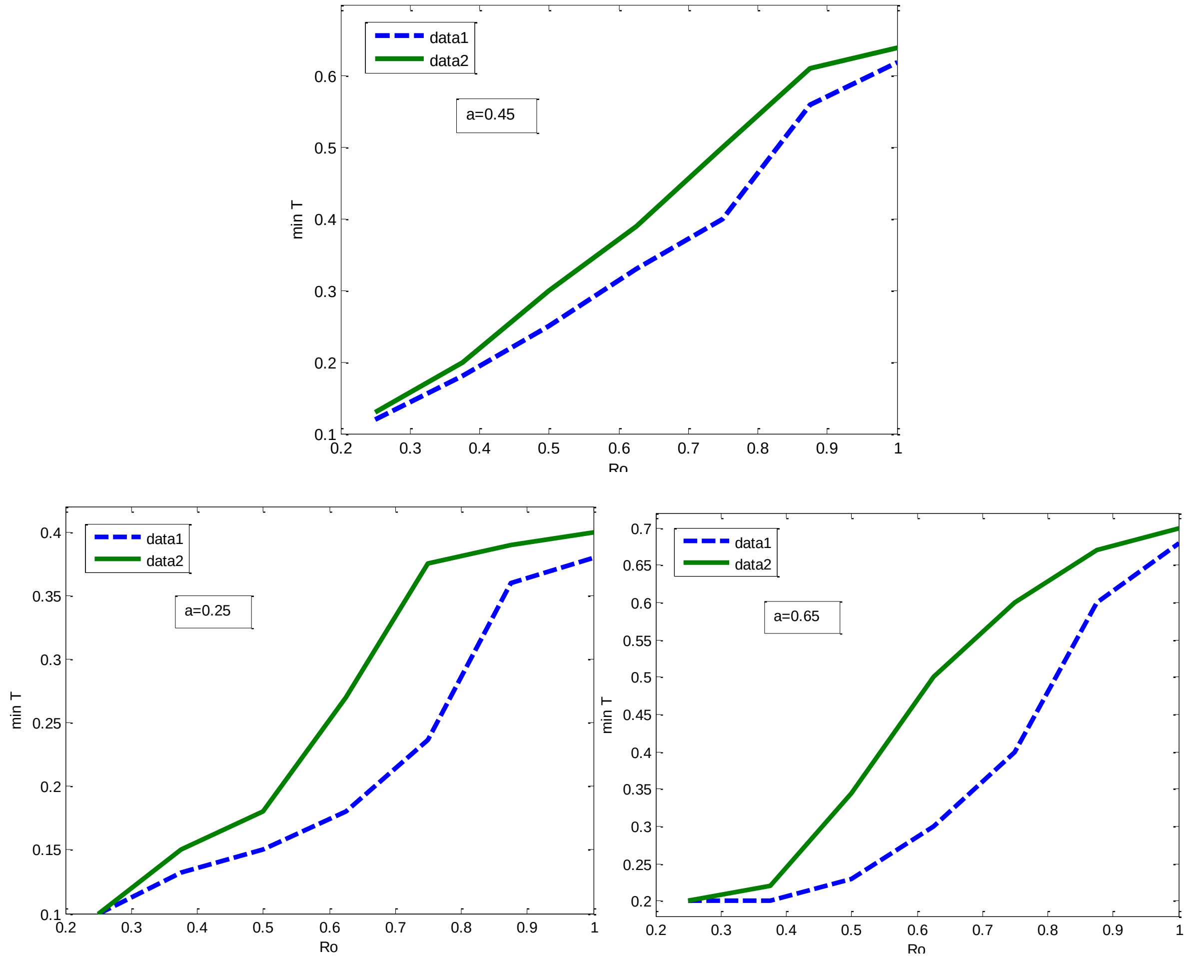

Fig. 4.2 Dependence of the minimum of the total average data latency on the performance of the channels (ρ) at different workloads.
1 - the proposed method; 2 - method of flow deviation.

It should also be noted that in all the cases studied, the algorithm for the distribution of flows in a “paired” way in the region of medium and heavy channel load provides a shorter delay time (curve 1 in Fig. 4.2 a, b, c) compared to the traditional algorithm using the deviation method flows.

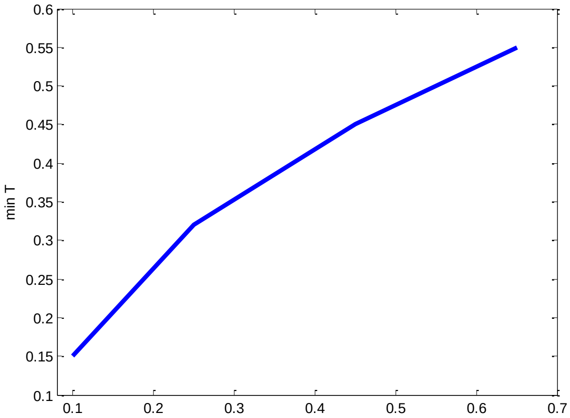

Fig. 4.3 The nature of the change in the average data latency with increasing voice traffic load.

This algorithm, rationally loading the network channels while being loaded with voice traffic, does not greatly increase the value of the average delay with increasing data stream intensity. For example, Tcp = 0.38 for α = 0.25, Tcp = 0.6 for α = 0.65, ΔTcp = 0.6 - 0.38 = 0.22 when using the method of flow deviation, the same result with the proposed algorithm is ΔT = 0.4 - 0.23 = 0.17.

Thus, the proposed algorithm for the distribution of flows allows for each thread of the flows to determine such shortest paths at which the value of the total average information delay in the network is smaller in comparison with the flow deviation algorithm.

V. CONCLUSION

All of the above allows us to conclude that the proposed streaming model and the shortest-path search algorithm make it possible to efficiently manage the distribution of heterogeneous streams in terms of using the ideology of a software-defined network in the transport part of a multi-service network. In this case, the decision on the distribution of flows is made at the controller, and the results are transmitted to the appropriate switches via special secure channels. These operations should be carried out in a very short time if the network is busy and using the developed algorithms will allow you to calculate the desired solution in a very short time.

REFERENCES

[1]. W. Xia, Y. Wen, C. Foh, D. Niyato, H. Xie, "A Survey on Software-Defined Networking", in: IEEE Commun. Surv. Tutor. 2015, 17, pg no: 27–51.

[2]. T. Nishonboyev, S. Mahmudov, "Formalization of the problem of distribution of heterogeneous traffic flows at the infrastructure level of a software-defined network", in: Infocommunications: networks, technologies, solutions, 4 (48) 2018, pg no: 5-11.

[3]. T. Nishonboyev, S. Mahmudov, "Algorithm for the distribution of heterogeneous traffic at the level of infrastructure of a software-defined network", in: Infocommunications: networks, technologies, solutions, 4 (52) 2019, pg no: 25-29.

[4]. Y.L. Lan, K. Wang, Y.H. Hsu, "Dynamic load-balanced path optimization in SDN-based data center networks", in Proceedings of the 10th International Symposium on Communication Systems, Networks and Digital Signal Processing, Prague, Czech Republic, 20–22 July 2016; pg no: 1–6.

[5]. Ghaffarinejad, A.; Syrotiuk, V.R. "Load Balancing in a Campus Network Using Software Defined Networking", in: Proceedings of the Third GENI Research and Educational Experiment Workshop, Atlanta, GA, USA, 19–20 March 2014; pg no: 75–76.

[6]. T. Nishonboev, M. Abdullaev, S. Maxmudov, "A model for the formation of the cloud structure of a data center based on SDN", in: International Conference on Information Science and Communications Technologies (ICISCT) Applications, Trends and Opportunities, November 2019, Tashkent, Uzbekistan.

[7]. T. Nishonboev, S. Maxmudov, M. Abdullaev, "Distribution model of heterogeneous flow in a software-defined multiservice network ", in: International Conference on Information Science and Communications Technologies (ICISCT) Applications, Trends and Opportunities, November 2019, Tashkent, Uzbekistan.

[8]. S. Mahmudov, "Software Defined Networking: Management of network resources and data flow" in: International Conference on Information Science and Communications Technologies ICISCT-2016, November 2016, Tashkent, Uzbekistan.

[9]. T. Nishonboev, S. Mahmudov, M. Abdullaev, "Formalization of the Task of Building a Cloud-data Center based on a Software-Defined Network", in: XIV International scientific- technical conference "Actual Problems of Electronic Instrument Engineering", APEIE-2018.

[10]. T. Nishonboev, S. Maxmudov, M. Abdullaev, "Distribution of external traffic by Virtual Servers of Cloud-data Center with distributed structure" in: Tenth World Conference WCIS-2018 "Intelligent Systems for Industrial Automation".