Journal of Theoretical and Applied Information Technology
15th July 2022. Vol.100. No 13


ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# ROUTING METHOD IN MULTI DOMAIN SOFTWARE DEFINED NETWORKS

[1]AS'AD MAHMOUD AS'AD ALNASER, [2]HAZEM (MOH'D SAID) HATAMLEH, [3]YURII KULAKOV

[1,2] Department of Applied Science, Ajloun University College, Al-Balqa Applied University, Ajloun, Jordan
[3]National Technical University of Ukraine "Igor Sikorsky, Kyiv Polytechnic Institute"
37 Peremohy ave. 03056 Kyiv, Ukraine

E-mail: asad1-99@ bau.edu.jo

## ABSTRACT

This article examines the issues of routing in multi-domain software-defined networks (SDN) of large size. The advantages of routing with the help of software-defined networks technology are given. The expediency of using multipath routing in computer networks of large size is justified.

A method for dividing the network into domains by the *k*-means method is proposed. The division of the network into domains and the use of intra-domain and inter-domain routing can reduce the time complexity of routing. A centralized routing information method in the SDN controller is proposed.

A modified algorithm for multipath routing along the distance vector is proposed and substantiated. In the routing process, with the formation of a given route, routes from all intermediate sites to the destination node are formed simultaneously. This eliminates the repeated formation of routing information for the previously formed paths.

Algorithms for inter-domain and intra-domain multipath routing are proposed to simplify the re-routing procedure if the network topology changes.

**Keywords:** *Multipath Routing, Software-Defined Networking; SDN controller, Routing domains, Domain controller, Time Complexity, Routing Algorithm, Distance Vector, re-routing*

## 1. INTRODUCTION

Modern computer networks are characterized by large dimensions and a diverse composition of equipment. In this regard, the managing process of this type of networks becomes more complicated particularly, routing and traffic engineering. To solve such problems, software-defined networking (SDN) technology is currently used [1]. In work [2], [3] several methods and approaches were used to solve the problems of optimizing data traffic in networks using SDN. Compared with the traditional networks, routing in SDN has the advantage that the formation of routes is held centrally in the SDN controller, so more efficient routing process is achieved. The routing information for the SDN switches are updated by the SDN controller which also updates the routing tables for these switches to choose the best route based on the specified QoS criteria. Compared to distributed routing methods, the service information in centralized method doesn't need to be exchanged between network switches, result in reducing the number of control packets in the network by 60-70% [4], [5].

One controller can manage a limited number of switches. This creates problems with scalability, reliability, and network availability [6]. A distributed controller architecture was used to solve these problems, the controller can be either a distributed (flat) or hierarchical architecture, where each controller is responsible for its own domain and updates either neighboring controllers or the root controller [7]. Figure 1 shows a hierarchical two-level domain structure of the SDN network [8]. The use of routing between domains and within domains can reduce the time complexity of routing.

Multiple paths are always found between source-destination pairs in large networks. This allows you to organize multipath routing. Multipath routing allows to increase the performance of the network by 10-15% by decreasing the service packets volume [9], [10]. As a result, the use of multipath routing

Journal of Theoretical and Applied Information Technology
15th July 2022. Vol.100. No 13

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

allows you to quickly reroute after failures of links and network switches [11].

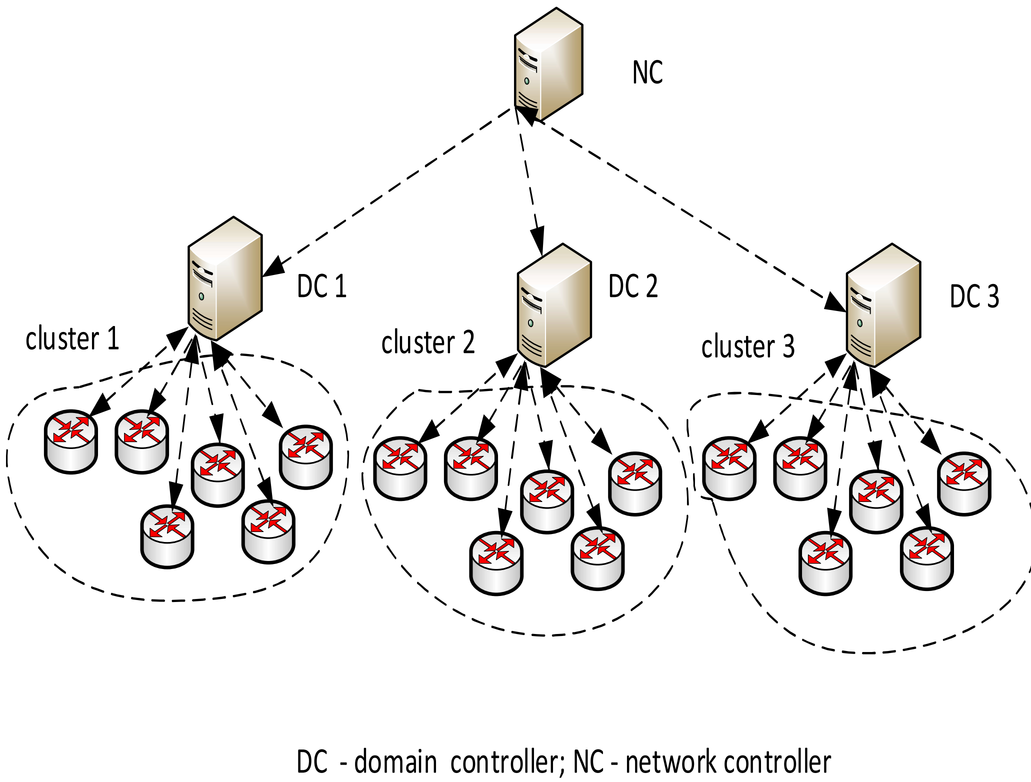


*Figure 1: Two-layer domain structure of the SDN network*

The method of centralized formation of routing information makes it possible to exclude the repeated formation of routes between intermediate sections of the generated path [12], [13].

Work [14], proposed a method for balancing traffic, which, according to the centralized method of generating routing information in the SDN controller and using multipath routing, makes it possible to simplify the traffic reconfiguration procedure and ensure the most uniform network load. Wave routing algorithm is used to generate Routing information. At the same time, paths between all intermediate nodes and the final node are formed simultaneously. This makes it possible to simplify the procedure of forming routes between intermediate nodes of previously formed routes [15].

Thus, the use of centralized algorithms for multipath routing within domains and between domains will improve the efficiency of a large SDN network. To do this, it is necessary to divide the network into the optimal number of domains and develop an efficient centralized interdomain routing algorithm

## 2. ROUTING METHOD IN A TWO-LAYER SDN NETWORK

Large SDN networks are divided into domains [16], [17].

The optimal division of the network into domains affects the efficiency of its function. The division of the network into domains is carried out using the *k*-mean method.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The number of domains is determined based on the average degree of vertices of the network graph and its diameter. The average number of vertices in domains is determined by the ratio of the total number of vertices to the number of domains. As the controller (*Ci*) of the domain *Di*, vertices with the maximum degree and equidistant from each other are selected.

For example, the network (Figure 2) is divided into domains $D_1 = \{ S_1, S_2, S_3, S_4, S_5, S_6, S_{14} \}$, $D_2 = \{ S_7, S_8, S_9, S_{10}, S_{11}, S_{12}, S_{13} \}$, $D_3 = \{ S_{15}, S_{16}, S_{17}, S_{18}, S_{19}, S_{20}, S_{23}, S_{25} \}$, $D_4 = \{ S_{21}, S_{22}, S_{24}, S_{26}, S_{28} \}$. *In this case* $C_1= S_4$; $C_2=S_9$; $C_3=S_{19}$; $C_4=S_{24}$.

The starting vertex ($S_s$) and the final vertex ($S_n$) of the path $P_i$ ($S_s$,$S_n$) can be in the same or in different domains. Therefore, routing in multi-domain networks is carried out separately between domains and within domains. Routing between domains and within domains is based on a modified multipath routing algorithm [18-21]

With inter-domain routing, based on the link table between domains (Table 1), a set of virtual paths is formed between domains in the network controller.

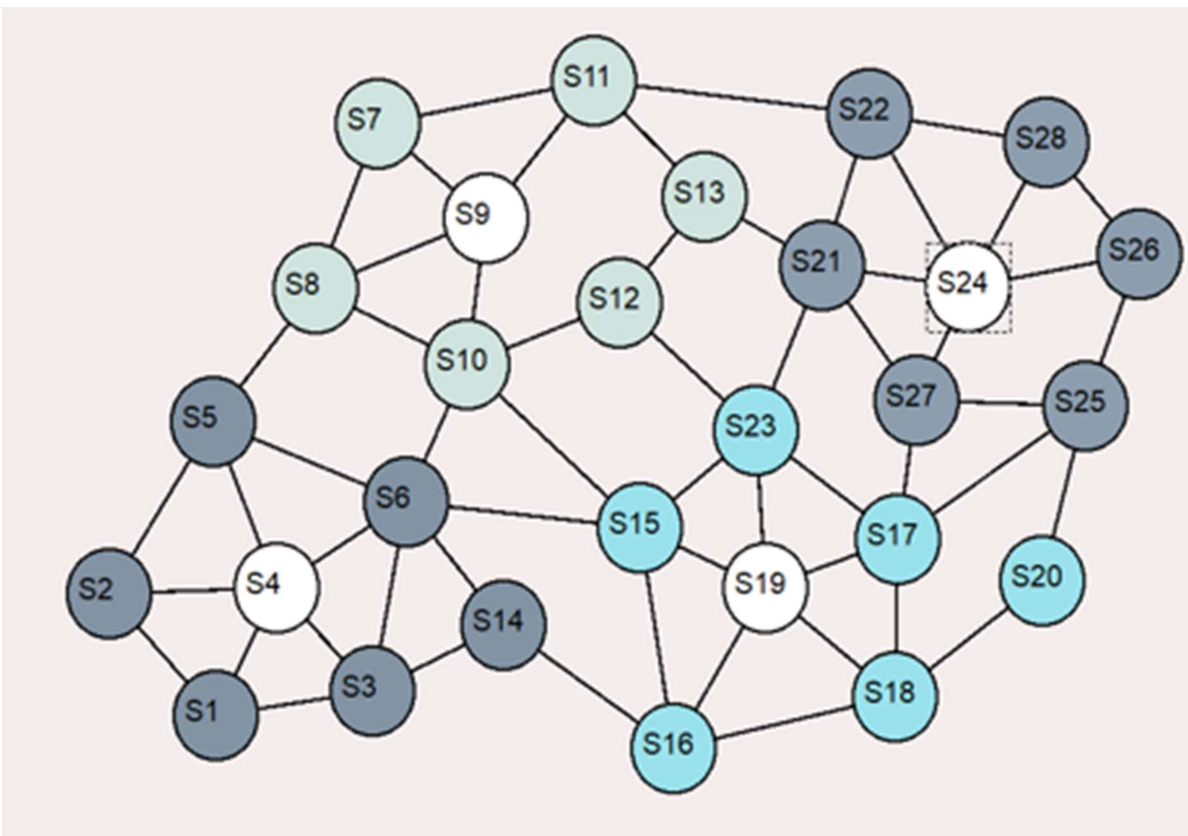


*Figure 2: Structure of a domain computer network*

Journal of Theoretical and Applied Information Technology
15th July 2022. Vol.100. No 13
© 2022 Little Lion Scientific
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

*Table 1: Relationship between adjacent domains*

| | | $D_1$ | | | $D_2$ | | | | | $D_3$ | | | | | $D_4$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $S_5$ | $S_6$ | $S_{14}$ | $S_8$ | $S_{10}$ | $S_{11}$ | $S_{12}$ | $S_{13}$ | $S_{15}$ | $S_{16}$ | $S_{17}$ | $S_{20}$ | $S_{23}$ | $S_{21}$ | $S_{22}$ | $S_{25}$ |
| $D_1$ | $S_5$ | | | | 0.4 | | | | | | | | | | | | |
| | $S_6$ | | | | | 0.1 | | | | 0.1 | | | | | | | |
| | $S_{14}$ | | | | | | | | | | 0.3 | | | | | | |
| $D_2$ | $S_8$ | 0.4 | | | | | | | | | | | | | | | |
| | $S_{10}$ | | 0.1 | | | | | | | 0.1 | | | | | | | |
| | $S_{11}$ | | | | | | | | | | | | | | | 0.1 | |
| | $S_{12}$ | | | | | | | | | | | | | 0.1 | | | |
| | $S_{13}$ | | | | | | | | | | | | | | 0.1 | | |
| $D_3$ | $S_{15}$ | | 0.1 | | | 0.1 | | | | | | | | | | | |
| | $S_{16}$ | | | 0.3 | | | | | | | | | | | | | |
| | $S_{17}$ | | | | | | | | | | | | | | | | 0.1 |
| | $S_{20}$ | | | | | | | | | | | | | | | | 0.1 |
| | $S_{23}$ | | | | | | | 0.1 | | | | | | | 0.5 | | |
| $D_4$ | $S_{21}$ | | | | | | | | 0.1 | | | | | 0.5 | | | |
| | $S_{22}$ | | | | | | 0.1 | | | | | | | | | | |
| | $S_{25}$ | | | | | | | | | | | 0.1 | 0.1 | | | | |

For example, between domains $D_1$ and $D_4$, virtual paths go through domain $D_2$ or $D_3$. Virtual paths pass through domain $D_2$ $P_1(1,4)= S_5\rightarrow S_8\rightarrow S_{11}\rightarrow S_{22}$; $P_2(1,4)=S_5\rightarrow S_8\rightarrow S_{13}\rightarrow S_{21}$; $P_3(1,4)=S_6\rightarrow S_{10}\rightarrow S_{11}\rightarrow S_{22}$; $P_4(1,4)= S_6\rightarrow S_{10}\rightarrow S_{13}\rightarrow S_{21}$. Virtual paths pass through the $D_3$ domain $P_5(1,4)= S_6\rightarrow S_{15}\rightarrow S_{23}\rightarrow S_{21}$; $P_2(1,4)=S_5\rightarrow S_8\rightarrow S_{13}\rightarrow S_{21}$; $P_3(1,4)=S_6\rightarrow S_{10}\rightarrow S_{11}\rightarrow S_{22}$; $P_4(1,4)= S_6\rightarrow S_{10}\rightarrow S_{13}\rightarrow S_{21}$.

The network controller contains information about the set of vertices in each domain. Based on this information, the network controller determines the domain in which the final vertex $S_n$ of the generated path $P_i$ $(S_s,S_n)$ is located. Then, based on the link table between adjacent domains (Table 1), the network controller selects the optimal path between domains with the initial and final vertices of the path $P_i$ $(S_s, S_n)$. Given the importance of ensuring the reliability and quality of data transmission in real time, this paper considers a routing method taking into account the delay metric in communication channels.

Then in each domain controller $D_j \in P_i( S_s,S_n )$ optimal non-intersecting paths are formed inside the domains between the boundary vertices on the path $P_i$ $( S_s,S_n )$. Paths are formed using a wave algorithm based on the distance vector.

In this case, the paths between the initial and all intermediate switches are simultaneously formed, eliminating the repeated formation of paths from the initial switch to the intermediate and final switches.

Sequential formation of paths between adjacent sets of vertices $W_i$ and $W_{i+1}$. The adjacent set of vertices means the sets of vertices $W_i =\{ S_i \mid i =1, 2 , \dots , n \}$ and $W_{i+1} =\{W_j \mid j = 1, 2, \dots , n \}$ , with a common set of edges $Es = \{ e$ , where: $s_j \in S_{i+1}$ and $sk \in S_i\}$. The formation of paths starts from the graph vertex $S_s$ corresponding to the destination $S_d$ at $i = 1$. Here, the set $S_1 = \{ s_d \}$, and the set $S_2 =\{s_j\}$ is the set of vertices that are adjacent to the vertex $s_d$ . Then for each commutator $S_j$ corresponding to the vertex $s_j \in S_2$ , adjacent to $s_d$ , is entered in its route table. The formation of routes from the vertices $S_j \in W_i$ to the vertices of $Sk \in W_i$ takes place in the second routing wave. Resulting in the formation of route tables of switches $S_j$ for vertices $s_j \in S_{i+1}$, and so on until all paths between the vertices $S_s$ and $S_d$ are formed.

Journal of Theoretical and Applied Information Technology
15th July 2022. Vol.100. No 13

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

For each formed path $P_i\ (S_s,S_n)$ the distance vector $R_i(S_s,S_n)=\{S_a,\ M_i\}$ is set. $S_a$ is the vertex adjacent to the vertex $S_a$ on the path $P_i(S_s,S_n)$, $M_i$ is the metric of the path $P_i(S_s,S_n)$. The value of the distance vector is entered into the table of distance vectors (Table 2).

*Table 2: Distance vectors of the commutator $S_j$*

| Destination | Adjacent vertex | Metrics Way |
|---|---|---|
| $s_d$ | $s_j$ | $M_1$ |
| $s_d$ | $s_r$ | $M_2$ |

Consider the formation of distance vectors for the path $P_i(S_1,S_5)$. In Table 3, the values of the transmission delay between adjacent nodes of the cluster are given as a metric.

*Table 3: Metrics of communication channels*

| | $S_1$ | $S_2$ | $S_3$ | $S_4$ | $S_5$ | $S_6$ | $S_{14}$ |
|---|---|---|---|---|---|---|---|
| $S_1$ | - | 0.1 | 0.2 | 0.2 | | | |
| $S_2$ | 0.1 | - | | 0.4 | 0.3 | | |
| $S_3$ | 0.2 | | - | 0.1 | | 0.4 | |
| $S_4$ | 0.2 | 0.4 | 0.1 | - | 0.3 | 0.2 | |
| $S_5$ | | 0.3 | | 0.3 | - | 0.2 | |
| $S_6$ | | | 0.4 | 0.2 | 0.2 | - | 0.8 |
| $S_{14}$ | | | | | | 0.8 | |

For $i = 1$ $W_1 = \{ S_1 \}$ , the set of adjacent vertices $W_2= \{ S_2, S_3, S_4 \}$.

Path distance vectors $P_i(S_1,S_2)$, $P_i(S_1,S_3)$, $P_i(S_1,S_3)$ are formed between the set of vertices $W_1=\{S_1\}$ and $W_2=\{S_2, S_3, S_4\}$ (Table 4)

*Table 4: Distance vectors of switch $S_1$*

| Destination | Adjacent vertex | Metrics way |
|---|---|---|
| $S_2$ | $S_2$ | 0.1 |
| $S_3$ | $S_3$ | 0.2 |

For $I = 2$ $W_2 = \{ S_2, S_3, S_4 \}$, the set of adjacent vertices $W_3 = \{ S_5, S_6, S_{14} \}$.

Between the set $W_2 = \{S_2, S_3, S_4\}$ and $W_3 = \{S_5, S_6, S_{14}\}$, distance vectors of paths $P_i(S_2,S_5)$, $P_i(S_3,S_6)$, $P_i(S_3,S_{14})$, $P_i(S_4, S_5)$, $P_i(S_4, S_6)$.

Table 5 shows the distance vector of switch $S_2$

*Table 5: Distance vectors of switch $S_2$*

| Destination | Adjacent vertex | Metrics way |
|---|---|---|
| $S_5$ | $S_5$ | 0.3 |

Table 6 shows the distance vector of switch $S_3$

*Table 6: Distance vectors of $S_3$ switch*

| Destination | Adjacent vertex | Metrics way |
|---|---|---|
| $S_6$ | $S_6$ | 0.4 |
| $S_{14}$ | $S_{14}$ | 0.8 |

Table 7 shows the distance vector of switch $S_4$

*Table 7: Distance vectors of $S_4$ switch*

| Destination | Adjacent vertex | Metrics way |
|---|---|---|
| $S_5$ | $S_5$ | 0.3 |
| $S_6$ | $S_6$ | 0.2 |

As a result, three minimal non-intersecting paths $P_1(1,5)= S_1$ are formed from the vertex $S_1$ to the vertex $S_5\rightarrow S_2\rightarrow S_5$; $P_2(1,5)= S_1\rightarrow S_4\rightarrow S_5$ ; $P_3(1,5)=S_1\rightarrow S_3\rightarrow S_6\rightarrow S_5$.

Path $P_4(1,5)=S_1\rightarrow S_3\rightarrow S_{14}\rightarrow S_6\rightarrow S_5$ intersects at the vertex $S_6$ with the path $P_3(1,5)= S_1\rightarrow S_3\rightarrow S_6\rightarrow S_5$ with a smaller metric.

Table 8 shows the path metrics $P_i(1,5)$ of switch $S_1$

Table 8: Distance vectors of switch $S_1$

| Destination | Adjacent vertex | Metrics way |
|---|---|---|
| $S_5$ | $S_2$ | *0.4* |
| $S_5$ | $S_3$ | *0.8* |
| $S_5$ | $S_4$ | *0.5* |

## 3. ALGORITHM FOR THE FORMATION OF DISTANCE VECTORS

**Notations:**

$S_d$ : top of destination
$S_s$: top of the path
$D_s$: top domain $S_s$
$D_d$: top domain $S_d$
$W_{i+1}=\{ s_j \}$: vertex set adjacent to vertex set $S_i = \{ s_j \}$
$R_j (s_d, s_i)$: path vector from vertex $s_j$ to vertex $s_d$ in the direction of vertex $s_i$
$p_j (s_d, s_i)$: path from vertex $s_j$ to vertex $s_d$
$l_{i,j}$: path link between vertices $s_i$ and $s_j$
$M_i$ : Path metric $P_j$
$m_{i,j}$: Path link metric
$C_i$ : criterion for uniform loading of the path $P_i$.
$TS_j$ : table of path vectors of the vertex (switch) $S_j$
$TD_j$ : domain controller path vector table $D_J$
$TN_j$ : network controller path vector table $S_j$
$BS_i$ : set of adjacent vertices domain $D_i$
$WD$: many domains
$WS_i$ : set of domain vertices $D_i$
$k$ : number of path domains $P_j (s_d, s_i)$

---

1. Begin/* formation of paths $P_j ( S_d, S_i )$ from vertex $s_j$ to vertex $s_d$ */;
2. if $R_j (s_d, s_i) \in TS_j$ then go to 35 /* $TS_j$ table contains path */;
3. if $S_d \in WS_j$ and $S_s \in WS_j$ then go to 23 /* $S_d$ and $S_s$ are in the same domain */;
4. if $R_j (s_d, s_i) \in TN_j$ then go to 35; /* $TS$ table contains the path to $S_d$ */;
5. begin /* Domain definition $D_i$ top $S_d$ */;
6. $i$ =1;
7. if $S_d \in D_i$ then $d:=i$; go to 10;
8. $i = i + 1$;
9. if $i < k$ go to 7;
10. end;
11. Begin /*formation of a vector of paths between domains $D_s$ and $D_d$
12. $i$ = 1;
13. $S_i = \{ s_d \}$;
14. form $S_{i+1} = \{ s_j \mid i=1,..,k \}$;
15. For all $s_j \in S_{i+1}$ determine $R_j (s_d, s_i)$;
16. $R_j (s_d, s_i) \rightarrow TS_j$ /* put $R_j (s_d, s_i)$ into $TS_j$ */;
17. $P_j (s_d, s_i) = P_i (s_d, s_m) + l_{i,j}$;
18. Determine $M_i$;
19. $M_i \rightarrow TS_j$ /* put $M_i$ in $TS_j$ */; ;
20. Determine $D_i$ ;
21. $D_i \rightarrow TS_j$ /* bring $D_i$ in $TS_j$ */;
22. if $S_s \in s_{i+1}$ then go to 15;
23. $S_i := S_{i+1}$ ;
24. go to 4
25. end
26. for $j: = 1$ step 1 to $k$ do
27. Begin /* forming a vector of paths within the domain $D_J$
28. $i$ = one;
29. $S_i =\{s_s\}$;
30. form $S_{i+1} = \{ s_j \mid i =1,.., k \}$;
31. For all $s_j \in S_{i+1}$ determine $Rj (s_d, s_i )$;
32. $R_j (s_d, s_i) \rightarrow TS_j$ /* put $R_j (s_d, s_i)$ into $TS_j$ */ ;
33. $p_j (s_d, s_i) = P_i (s_d, s_m) + l_{i,j}$ ;
34. Determine $M_i$ ;
35. $M_i \rightarrow TS_j$ /* put $M_i$ in $TS_j$ */ ;
36. if $s_s \in S_{i+1}$ then go to 21;
37. $S_i := S_{i+1}$ ;
38. go to 4
39. end
40. $R_j (S_s, S_d) \rightarrow TS_s$ /* the path vector value is entered into the path vector table of the domain switch $D_s$ */;
41. **end**

---

Figure 3 shows the result of the formation of paths within the domains included in the path $P(S_1,S_{28})$. The optimal path between nodes $S_s$ and $S_n$ is chosen based on the minimum packet transmission delay.

Journal of Theoretical and Applied Information Technology
15th July 2022. Vol.100. No 13

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

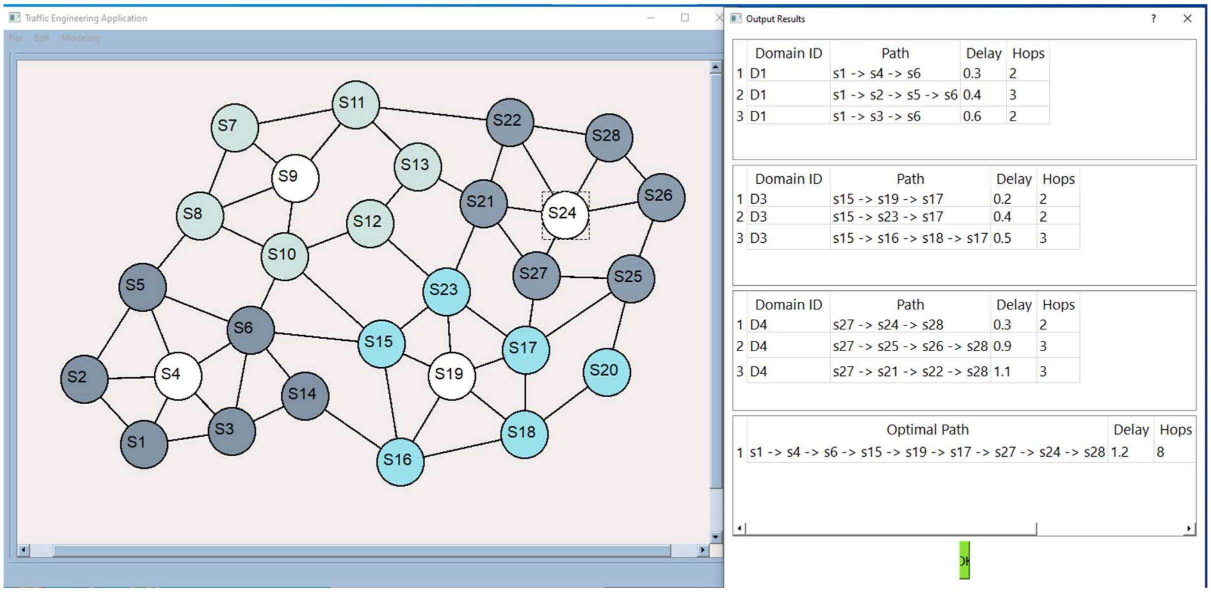


*Figure 3: path formation $P(S_1,S_{28})$*

As a result, for the path $P(S_1,S_{28})$ at the vertex $S_1$, the path vector $R_1(S_1,S_{28})=\{S_4, M_1\}$ with the metric $M_1$= 1.2 is formed.

## 4. PATH RECONFIGURATION

The presence of several vectors of distances to a given vertex at each vertex of the generated path allows dynamic reconfiguration of the path. In case of channel overload $S_6 \rightarrow S_{15}$ by 50%, the this case, a new path will be formed between the vertices $S_1$ and $S_{28}$ (Figure 4) through the domain $D_2$ with a delay equal to 1.3. This is only 10% more than the delay on the path $P_1$ (1,28) before the $S_6$ channel is overloaded$\rightarrow S_{14}$.

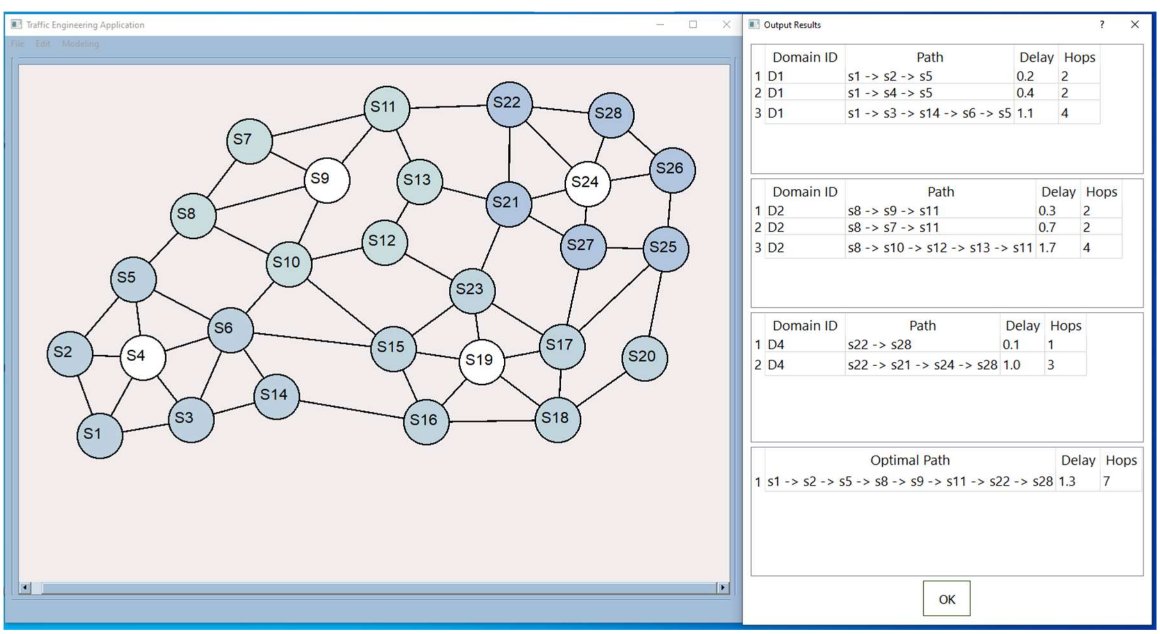


*Figure 4: path reconfiguration $P(S_1,S_{28})$ when channel $S_6$ is overloaded$\rightarrow S_{14}$*

As a result, for the path $P(S_1,S_{28})$ at the vertex $S_1$, the path vector $R_1(S_1,S_{28})=\{S_2, M_1\}$ with the metric $M_1$= 1.3 is formed.
For example, when overloading the link $L(S_{22},S_{28})$ paths $R_1(S_1,S_{28})$ the data stream will be redirected according to the distance vector $R_2(S_s,S_{28})=\{S_{24}, M_2\}$ along the path $S_{22}\rightarrow S_{24}\rightarrow S_{28}$. The transmission delay will increase by 0.1 and become 1.4 instead of 1.3. In this case, the recalculation of the entire path $R_1(S_1,S_{28})$ is excluded in the process of information transfer.

Route correction in the network controller and domain controllers can be done statically or dynamically, depending on the type of network.

## 5. CONCLUSION

The paper proposes a method for multi-path routing of multi-domain software-defined networks, which, due to the presence of a central SDN controller in the network, reduces the time for generating multiple routes. The time complexity of path formation is reduced due to the proposed method of dividing the network into the optimal number of routing domains using the k-means method, which allows reducing the time complexity of routing. The proposed algorithm for multipath routing along the distance vector allows you to dynamically rebuild the route in the process of information transfer.

In the future, when forming a route, the fuzzy logic method can be used [22], [23] to ensure uniform loading of computer network channels.

## REFERENCES:

[1] K.S.Sahoo, S.K.Mishra, S.Sahoo, B.Sahoo, "*Software defined network: the next generation internet technology,*" Int. J. Wirel. Microwave Technol. (IJWMT) 7(2), 13–24, 2017. https://doi.org/10.5815/ijwmt.2017.02.02

[2] A.M.Yasir, H.A.Chasib, Q.M.Zainab, "*Analyzing Methods and Opportunities in Software-Defined (SDN) Networks for Data Traffic Optimizations,*" Int. Inter. J. Rec. Inn. Trends Comp. Com. (IJRITCC) 6 (1), 75 – 82, 2018.

[3] B.Augustin, T. Friedman, R. Teixeira, "*Measuring Multipath Routing in the Internet*," IEEE/ACM Transactions on Networking Volume: 19, Issue: 3, pp 830-840, June 2011

[4] K. Poularakis, Q. Qin, E. Nahum, M. Rio, L. Tassiulas, "*Flexible SDN control in tactical ad hoc networks,*" *Ad Hoc Networks* Volume 85, pp 71-80, 15 March 2019.

[5] P.Kumar, R.Dutta, R.Dagdi, K. Sooda, A. Naik, "*A programmable and Managed Software Defined Network,*" Int. J. Comput. Netw. Inf. Secur. (IJCNIS) 12, 11–17, 2017. https://doi.org/10.5815/ijcnis.2017.12.02. In MECS http://www.mecs-press.org/. Accessed Dec 2017.

[6] Zohaib Latif, Kashif Sharif, Fan Li, Md Monjurul Karim, Sujit Biswas, Madiha Shahzad, Member, IEEE, and Saraju P. Mohanty, "*DOLPHIN: Dynamically Optimized and Load Balanced Path for Inter-domain SDN Communication,*" IEEE Transactions On Network And Service Management, VOL. 18, NO. 1, pp. 331 – 346, January 2021.

[7] L. Zhu, M. M. Karim, K. Sharif, C. Xu, F. Li, X. Du, and M. Guizani, "*SDN Controllers: A Comprehensive Analysis and Performance Evaluation Study,*" ACM Computing Surveys, vol. 53, no. 6, pp. 1–40, Dec 2020.

[8] S. Saraswat, V. Agarwal, H. P. Gupta, R. Mishra, A. Gupta, and T. Dutta, "*Challenges and solutions in software defined networking: A survey,*" Journal of Network and Computer Applications, vol. 141, pp. 23–58, 2019.

[9] K.Rajasekaran, K.Balasubramanian, "*Energy Conscious based Multipath Routing Algorithm in WSN*". Int. J. Comput. Netw. Inf. Secur. (IJCNIS), 1, 27-34, 2016. https://doi.org/10.5815/ijcnis.2016.01.04 in MECS (http://www.mecs-press.org/).

[10] M. Caria, A. Jukan, and M. Hoffmann, "*SDN Partitioning: A Centralized Control Plane for Distributed Routing Protocols,*" Volume: 13, Issue: 3, pp 381-393, Sept. 2016

[11] M.Moza, S.Kumar, "*Analyzing multiple routing configuration,*" Int. J. Comput. Netw. Inf. Secur. (IJCNIS) 5, 48–54, 2016. https://doi.org/10.5815/ijcnis.2016.05.07.

[12] Y.Kulakov, A. Kohan, S. Kopychko "*Traffic Orchestration in Data Center Network based on Software-Defined Networking Technology,*" International Conference on Computer Science, Engineering and Education Applications ICCSEEA 2019. Advances in Computer Science for Engineering and Education II pp 228-237

[13] V. Kotronis, A. Gämperli, X. Dimitropoulos, "*Routing centralization across domains via SDN: A model and emulation framework for BGP evolution,"* arxiv, 8 Nov 2016.

[14] Y.Kulakov, S.Kopychko, V.Gromova, "*Organization of Network Data Centres Based on Software-Defined Networking,*" In: Proceedings International Conference on Computer Science, Engineering and Education Applications ICCSEEA 2018, pp.447-455 /https://link.springer.com/book/10.1007/978-3-319-91008-6

[15] C. Chen, F. Xue, Z. Lu, Z. Tang, C. Li, *"Reinforcement Learning BasedMultipath Routing for SDN,"* Wireless Communications and Mobile Computing, Volume 2022, DOI:org/10.1155/2022/5124960

[16] E. Akin, T. Korkmaz, "*Comparison of Routing Algorithms With Staticand Dynamic Link Cost in Software DefinedNetworking (SDN*)," IEEE Access Volume 7, pp 148629 – 148644, 11 October 2019. DOI: 10.1109/ACCESS.2019.2946707

[17] O. Blial, M. B. Mamoun, R. Benaini, "*An Overview on SDN Architectures with Multiple Controllers,"* Journal of Computer Networks and Communications, Volume 2016, DOI.org/10.1155/2016/9396525

[18] A. M. A. Alnaser, "*Certain Contributions in Traffic Engineering Based on Software-Defined Networking Technology,*" Journal of Computer Science 15 (7), pp 944.953, 2019, DOI: 10.3844/jcssp.2019.944.953.

[19] A. Abugabah,A. A. AlZubi, O. Alfarraj, M. Al-Maitah, W. S. Alnumay, " *Intelligent Traffic Engineering in Software-Defined Vehicular Networking Based on Multi-Path Routing*," IEEE ACCESS Volume 8, pp 62334 – 62342, 25 March 2020, DOI: 10.1109/ACCESS.2020.2983204

[20] A.M.A. Alnaser, "*A method of multipath routing in SDN networks,*" Advances in Computer Science and Engineering Volume 17, Number 1, pp 11-17, 2018.

[21] A. M. A. Alnaser, Y. Kulakov, D. Korenko, "*Modified Method of Traffic Engineering in DCN with a Ramified Topology,*" (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 12, No. 12, 2021. pp. 439-446. DOI : 10.14569/IJACSA.2021.0121258.

[22] A. M. A. Alnaser, "*Novel Properties for Total Strong - Weak Domination Over Bipolar Intuitionistic Fuzzy Graphs,*" (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 12, No. 4, pp 672-679, 2021.

[23] A. M. A. Alnaser1, W. A. AlZoubi1 and M. O. Massadeh, "*Bipolar Intuitionistic Fuzzy Graphs and it's Matrices,*" Applied Mathematics & Information Sciences 14(2), pp 205-214, 2020.