

# Modified Method of Traffic Engineering in DCN with a Ramified Topology

As'ad Mahmoud As'ad Alnaser<sup>1</sup>  
Department of Applied Science  
Ajloun University College  
Al-Balqa Applied University, Ajloun, Jordan

Yurii Kulakov<sup>2</sup>, Dmytro Korenko<sup>3</sup>  
National Technical University of Ukraine "Igor Sikorsky  
Kyiv Polytechnic Institute," 37 Peremohy ave.  
03056 Kyiv, Ukraine

**Abstract**—This article consider two main local network topologies. Based on the basic DFS protocol, a mathematical model has been developed for a new method of multipath routing and traffic engineering in data centers with a ramified topology. This method was developed with the features and benefits of SDN in mind. Also, the simulation of the developed method was carried out on two local topologies considered earlier.

**Keywords**—Local networks; traffic engineering; SDN; DCN; DFS; Mininet

## I. INTRODUCTION

The successful operation of the enterprise in a dynamic market and competition is largely determined by the ability to make quick decisions. Timely and high-quality decision-making is possible only if the reliable and productive operation of the company's IT infrastructure. In turn, modern IT systems are evolving dynamically: there are more and more different business applications and automated workplaces, increasing the amount of information, requires continuous provision of IT services.

The basis of building the company's IT infrastructure is the Data Center (DC). The main task of the DC is to ensure guaranteed trouble-free operation of the enterprise's IT infrastructure. In this case, we are talking not only about the automation of business processes, but also about the most reliable data storage and guaranteed constant access to them. Databases can be upgraded by adding computing resources when implementing new business applications, as well as increase the amount of data storage, which means the ability to quickly adapt to changing business requirements.

First of all, data centers are in demand by large organizations, such as banks, insurance and trading corporations, mining companies, telecommunications companies (billing systems, hosting, various Web-services and social services). They all use complex business applications, and their activities depend on the reliability of the IT infrastructure.

The most important advantages of creating a DC are the consolidation of computing power and storage systems. As shown in [1], It is known that centralized management of IT infrastructure and information systems is more efficient than in the case of a distributed heterogeneous solution. In addition, it is easier to provide surveillance of a single complex and protection against possible failures.

At the same time, the growing needs of business must be met in conditions of limited resources: more data must be stored in the allocated space, this will prevent consumption or allocation of too much energy so calculations will be faster, to transmit more information through existing channels of communication, to ensure maximum readiness operated IT systems. And all this with limited funding. In such conditions, the competent design of the DC is a key link to achieve efficient operation of the enterprise, and these limitations determine the choice of technologies and equipment used.

It is known that in the field of high-availability IT systems, the situation is constantly and rapidly changing. The lack of a unified approach to the organization of DC, standards for the design and operation of various processing centers and server rooms pose the problem of developing a systematic approach to infrastructure implementation, as well as developing methods and models of DC to a new level. The emergence of experience in the form of standards, models and methods will unify the implementation and simplify changes in its infrastructure, thereby contributing to the replication and scalability of solutions.

In the absence of such experience, it is easy to be tempted and prefer the latest developments. Obviously, it is a rational choice, justified from the economic and technological point of view of the infrastructure that would provide long-term investment protection and allow the company to perform current tasks and develop. The amount of information in the future will only grow. Further growth of such sectors of the Internet related to business applications as e-commerce, payments, communications require an appropriate infrastructure. It is necessary to make sure that in the future there is an opportunity to ensure cost-effective growth and expansion of diabetes, because disruptions in electronic services can have significant economic consequences, both for individual enterprises and for government agencies and sectors of the economy.

The above indicates the relevance of the study of ways to effectively design traffic in networked data centers with a branched topology, taking into account open standards, which helps to minimize the problems of interaction when scaling the DC [2].

Improvement of quality of service (QoS) of traffic and reduction of it's design time can be hold by using multipath routing in SDN for centralized formation of multiple paths [3],

unlike known methods that have high time complexity. So a modified method was proposed in which formation of multiple paths has less complexity compared to known methods.

This paper considers the modeling proposed in article [4] mathematical model of a modernized method of constructing traffic in networked data centers with a branched topology, focused on software configuration of the network. This method, by taking into account the peculiarities of the organization of SDN, in particular due to the presence in the network of a central controller, reduces the time of formation of many routes of access to network resources [5], [6], [7], [8] and [9].

## II. ALGORITHM MODIFICATION

In terms of routing, we are usually only interested in optimal routes. This applies to both one-path routing (search for the shortest route) and multi-path (search for a set of non-ordinary, partially ordinary routes). However, applying this method to this model is ineffective, because the controller requires a rapid response to changes in the state of communication channels, which leads to changes in the metric. Therefore, it is unwise to list the paths to find the shortest path.

For example, consider dynamic routing protocols such as OSPF and IS-IS [10]. These protocols are similar and designed for dynamic routing and take into account the state of the channels when constructing routes. These protocols use the Dijkstra algorithm to find the shortest route. In time, the complexity of the algorithm is  $O(n^2)$ . From this we can conclude that as the number of nodes increases, the time of finding the shortest path increases quadratically. With frequent updating of the status of communication channels, this approach is ineffective.

In the proposed method, the network topology rarely changes, therefore, it is advisable to consider a one-time calculation of all possible paths and the calculation of the next route to form new connections. Therefore, if it is necessary to find the optimal route, it is not necessary to calculate the route using the adjacency matrix; it is enough to list metrics and to choose an optimum line on the basis of the available information on a condition of knots and channels. This method can be used for different types of metrics. In the model used, the speed of information transmission along the route is used as a metric, so the controller will choose the path at high speed and send it to the first vertex along the route to the router [11].

Using the algorithm of passage of the graph in width makes it possible to identify all possible paths between the vertices. Ideally, this method will be applied once for each pair of vertices, while lists with calculated routes will be stored in the controller.

The method of constructing a route is based on the use of recursive steps. As a result, a set of all possible routes from the final to the initial vertex is formed. To do this, the extreme vertex is selected in each path and, starting from it, new routes are formed, which includes the already formed sub-route from the final vertex to the current one with the inclusion of one of the neighboring vertices. The algorithm terminates for this subroute if a new neighboring vertex is already in the route or

the current vertex is the last. The procedure is complete if all routes have reached the initial peak.

However, a new route will not be formed if the adjacent vertex is already on the desired path. As mentioned, this step is recursively repeated for all routes until the next peak on the way is final. In the case of the formation of new links in the network, the already formed routes are supplemented as follows:

- 1) Select all paths that include vertices connected by a formed channel;
- 2) For each selected path, a sub-route is created from the vertex that participates in the new connection;
- 3) The vertex, which is located on the other side of the new connection, is added to the received sub-routes.
- 4) Recursive steps described for the formation of all routes, starting with the corresponding sub-route, are repeated.

Consider forwarding a packet to the next node on the way to its destination.

In high-load networks, the state of communication lines is constantly changing. These changes are difficult to predict because there are a number of reasons that can cause an unexpected load on a connection that was lightly loaded and guaranteed data transfer while maintaining a high level of service quality.

The packet forwarding algorithm used (MPLS VC) calculates the path during the traffic generation phase and stores the intermediate node labels in the packet headers. This algorithm using a virtual data channel does not allow timely response to changes in the state of communication channels in the network, because the intermediate nodes on the data route are stored as labels in the packet. Therefore, it is possible that the intermediate nodes stored in the packet header will not create the optimal path for the current time. This situation is likely to lead to some delays in data transmission for both the destination node and those nodes along which the routes intersect with the considered.

The proposed modernized method is based on the traditional approach to the routing table. These tables are updated after a certain period of time, each of the paths is updated independently of the others, just when you want to use it, or the time until the next update of the current route. After a request from the router, the controller calculates indicators if the paths for the destination node are already formed, generates these paths, and then sends them to the router, which sent a request to update the path with the number of the next node on the optimal path now.

Consider an unmodified width search algorithm ( DFS ) [12],[13]. It is based on a recursive stroke of the graph's vertices, through the tour, visited and non-visited nodes were marked. In the start point, some nodes are visited. The visited node is the one that receives the control packet and analyze information about neighboring nodes. After that the packet is transmitted to one of the neighboring unlabeled nodes. Otherwise, the control packet will pass back, if there is no unmarked neighboring nodes related to this node.

The disadvantages of main algorithm come from the ignorance of:

- 1) Channels communication capacity.
- 2) Network features topology.

Strategies for channel selection:

- Worst .Fit: In which the highest available bandwidth channel is selected.
- First Fit: In which any available bandwidth Chanel fits the requirements is selected.
- Best Fit: In which the available bandwidth Chanel that best fit the requirements

Taking in account, when DFS method is used to create multiple paths in DCN, the most effective strategy is Worst Fit. Moreover, the efficiency of aeration of many paths using DFS method is increased in association with the peculiarities of the DCN hierarchical organization and the communication channels bandwidth between the nodes of the network [14].

The DFS method is dependent on the centralized routing method. In this method all the information that are required to generate routes are found in the central controller, which contains information about network topology and capacity of communication channels, which is used to select the communication channel at a higher level or return to a lower level in the tree. Following the next path formation the communication channel with the allowable bandwidth for the selected path is reduced by a given value.

Based on that the network topology information is found in the central controller, the first step that is determined by the algorithm is the sender and recipient topology level where the nodes are unified. The nodes are connected directly, if the path have many switches to pass through, their number is determined by the nodes relative location.

If the connection level of nodes is determined, routing will be facilitated, there for eliminating the unnecessary transitions between levels.

Fig. 1 shows the sequence of operations for modified DFS algorithm:

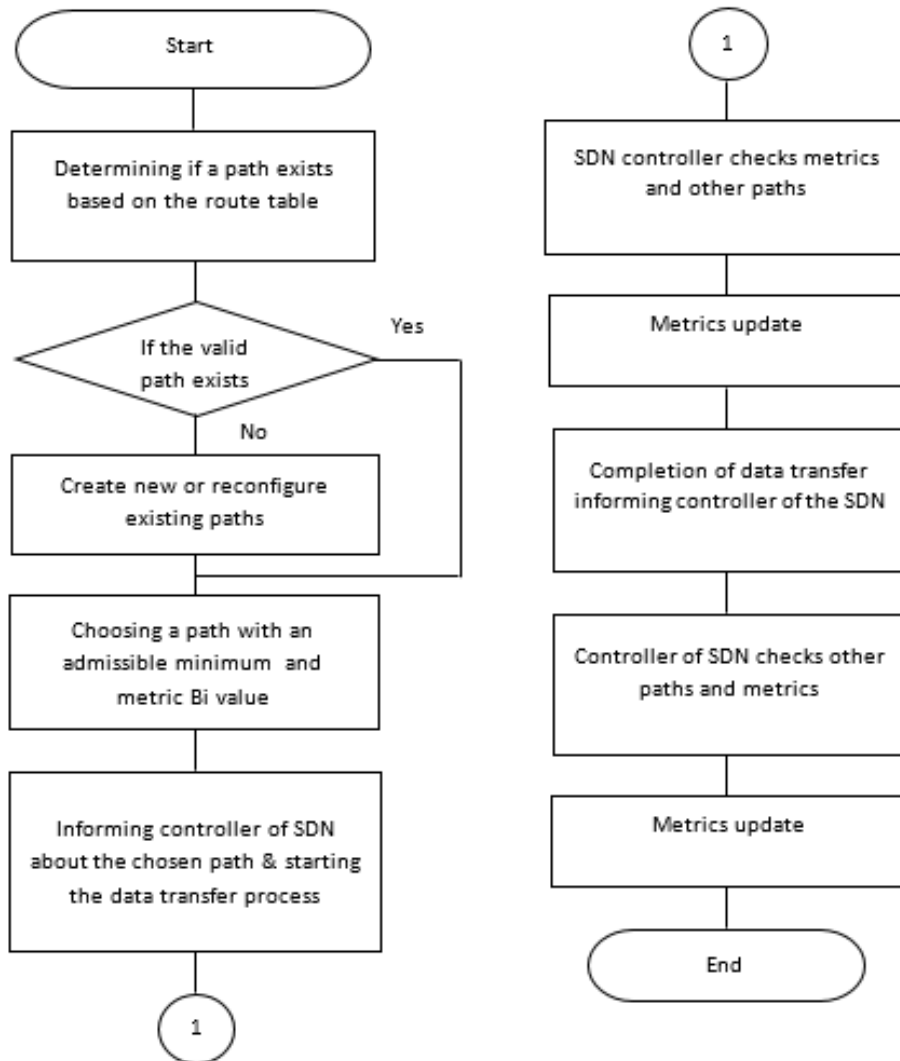


Fig. 1. Modified DFS Algorithm.

An advantage of SDN is that the network organization and management at the software level is carried out by virtual switches [15].

The Open Flow standard, [16], [17] and [18] in which the SDN concept is based, defines the network traffic management flow principles. The implementation of multi-threaded traffic routing in a software - configurable network is proposed due to the ability to configure the route of each individual traffic at the level of software-configurable SDN switches.

Streaming algorithms for multibeam routing have a minimal time complexity with known algorithms [19].

A set of continuous paths can be formed in the presence of centralized SDN control based on a network controller, by organizing counter - flows between sender and recipient nodes. Finding paths task is to find the tree's points of connections from the start and end vertices. As in [20], for a route similar to the modified wave algorithm, as the next for a certain path, a vertex with a smaller external degree is selected. So, a divergent paths formation is ensured.

Path trees are built until all the intersecting paths between the start and end vertices are built. Forming a set of disparate paths algorithm is illustrated below.

```
Define nodes set  $X_j = \{U_n\}$ ;  
 $B_i=0$ ;  
 $J=0$ ;  
for  $j = j + 1$  step 1, create a nodes set  $X_{j+1} = \{U_i | i=1, \dots, n\}$   
adjacent to the set nodes of  $Y_i \{U_n, U_i, N_i, c, b_i\}$ , where  $n$  is  
the power's sum of the set of node  $X_{j+1} = \{U_i | i=1, \dots, n\}$ ;  
if  $X_{j+1} = \emptyset$  then go to 10 do  
for  $i=1$  step 1 to  $n$  calculate  
 $Y_i \{U_n, U_i, N_i, c, b_i\}$   
if  $b_j > B_i$  then  $B_i = b_j$   
end;  
go to 4
```

end.

In the centralized formation of a set of independent paths, complete information about the formed paths and the trees that generated them is found in the central controller SDN. This gives us the chance to optimize them in the process of forming paths in accordance with the specified indicators.

### III. MODELING

The Mininet environment was used to demonstrate the operation of the algorithm. Mininet is a computer network emulator. It allows you to quickly set up a network on a personal computer. This network will be almost indistinguishable from the real one, it will just not be able to send pings to external IP addresses.

To model the proposed method of traffic design, you need to build and configure the topologies considered in article [4], namely the double extended star and the double ring.

Two (2) controllers, 6 switches and 12 hosts were used to build the Double Extended Star topology (Fig. 2).

Before the simulation begins, the controllers form a model of the constructed topology. To do this, they build an adjacency matrix and store nodes and connections. As a result of the correct start of modeling of controllers on the console the list of all nodes and their connections will be deduced. The console displays basic information about the nodes and the connections with other nodes, namely the actual node name, bandwidth and channel load (delay). In this step, each connection has a bandwidth of 100Mbit, i.e. it does not contain any load. This message is displayed only once during the initial initialization, then when the topology changes, the network model is updated by the controller without messages of this type.

To simulate the operation of the algorithm, it is necessary to initiate traffic between hosts h7 and h14. To demonstrate, execute the command to compile the dump and start the traffic between hosts h7 and h14 (Fig. 3).

As you can see from the results, the routing table that was created at network startup and updated periodically, controllers immediately know the optimal route for traffic. In our case, we can see in the logs, which are collected during the transmission of the packet, a complete description of the route, and the time of passage of the packet between nodes. We also obtain the IP and MAC addresses of the sending and receiving nodes, which simplifies the analysis of the collected data and allows easy packet routing.

To demonstrate the effective operation of the developed method of traffic design for this topology, a original DFS algorithm was simulated.

As you can see in Fig. 4 the execution time of the first packet transmission takes 0.11 ms, which is 0.058 ms more than the modified one. This is because before the transmission begins, complete information is collected about the state of the nodes and communication channels in the topology, which takes some time. If the topology was static (nodes are always working, non-communication channels have a constant non-variable load) then a slightly longer time to send the first packet did not play an important role, but because it is not possible and the topology is dynamic (nodes fail or topology is supplemented by new, in communication channels are constantly changing the load or they break off altogether) then each time to collect new information about the state of the topology before sending the packet is quite time-consuming operation.

To obtain more detailed information about the operation of the modified method of traffic design, simulations were performed with different numbers of packets (Graph 1 and Table I).

Four (4) controllers, 8 switches and 12 hosts were used to build the Double Ring topology (Fig. 5).

In this simulation, the situation is the same as described earlier. A topology model is formed due to the adjacency matrix and information about nodes and connections between them.

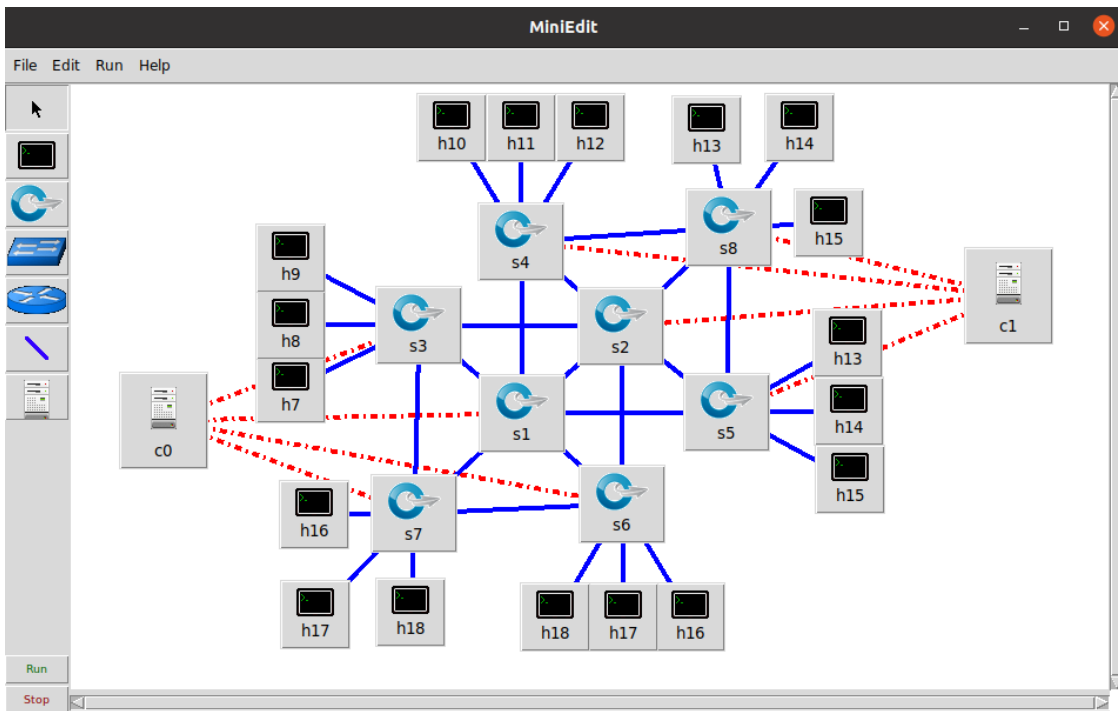


Fig. 2. Topology Double Extended Star in Mininet.

```
mininet> dpctl dump-flows
*** s1 .....
NXST_FLOW reply (xid=0x4):
*** s2 .....
NXST_FLOW reply (xid=0x4):
*** s3 .....
NXST_FLOW reply (xid=0x4):
*** s4 .....
NXST_FLOW reply (xid=0x4):
*** s5 .....
NXST_FLOW reply (xid=0x4):
*** s6 .....
NXST_FLOW reply (xid=0x4):
mininet> h7 ping -c 2 h14
PING 127.0.3.7 (127.0.3.7) 56(84) bytes of data.
64 bytes from 127.0.3.7: icmp_seq=1 ttl=64 time=0.052 ms
64 bytes from 127.0.3.7: icmp_seq=2 ttl=64 time=0.021 ms

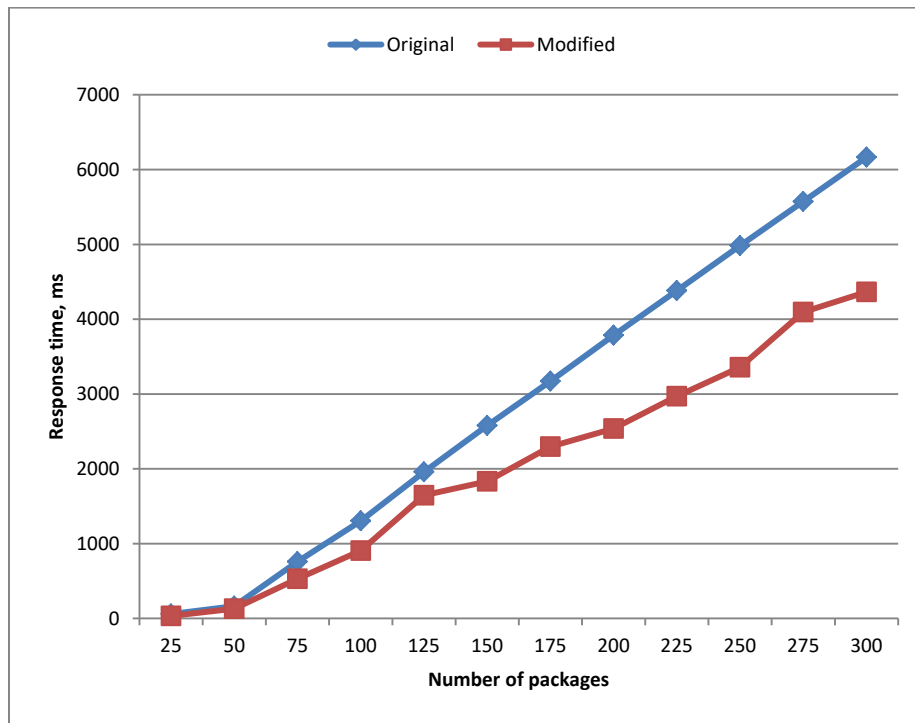
--- 127.0.3.7 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 100ms
rtt min/avg/max/mdev = 0.021/0.026/0.032/0.005 ms
```

Fig. 3. Start Dump Collection and Start Traffic.

```
mininet> h7 ping -c 2 h14
PING 127.0.3.7 (127.0.3.7) 56(84) bytes of data.
64 bytes from 127.0.3.7: icmp_seq=1 ttl=64 time=0.11 ms
64 bytes from 127.0.3.7: icmp_seq=2 ttl=64 time=0.035 ms

--- 127.0.3.7 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 100ms
rtt min/avg/max/mdev = 0.021/0.026/0.032/0.005 ms
mininet>
```

Fig. 4. DFS Modeling.



Graph 1. The Dependence of the Average Response Time on the Number of Packets.

TABLE I. SIMULATION RESULT FOR A LARGE NUMBER OF PACKAGES

Number of packages		25	50	75	100	125	150	175	200	225	250	275	300
Response time, ms	Original algorithm	60	165	760	1305	1958	2580	3171	3785	4382	4982	5573	6166
	Modified algorithm	35	131	531	907	1647	1833	2297	2539	2970	3358	4095	4364

Next, consider modeling a double ring topology:

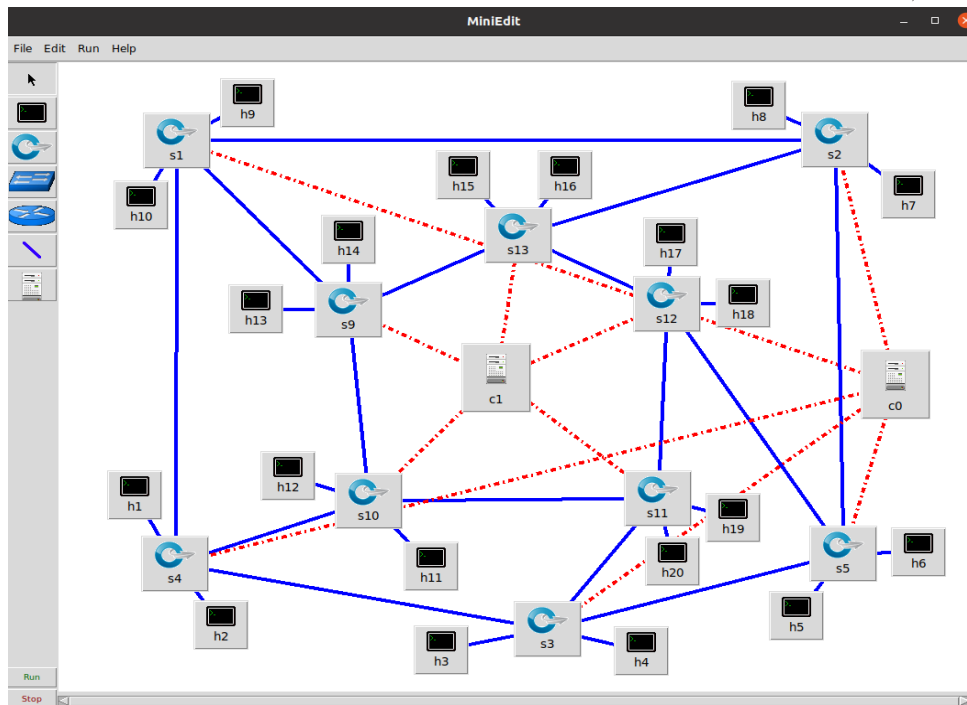


Fig. 5. Topology of a Double Ring in a Mininet.

The simulation follows the same scenario as in the previous section. We initiate traffic between nodes h7 and h14 and collect logs based on simulation results (Fig. 6).

To demonstrate the effective operation of the developed method of traffic design for this topology, an original DFS algorithm was simulated.

As you can see in Fig. 7 the execution time of the first packet transmission takes 0.087 ms, which is 0.057 ms more than the modified one.

Graph 2 and Table II show the simulation results for the Double Ring topology of the original and modified method of constructing traffic for a large number of packets.

```
mininet> dpctl dump-flows
*** s9 -----
NXST_FLOW reply (xid=0x4):
*** s12 -----
NXST_FLOW reply (xid=0x4):
*** s11 -----
NXST_FLOW reply (xid=0x4):
*** s10 -----
NXST_FLOW reply (xid=0x4):
*** s1 -----
NXST_FLOW reply (xid=0x4):
*** s4 -----
NXST_FLOW reply (xid=0x4):
*** s3 -----
NXST_FLOW reply (xid=0x4):
*** s2 -----
NXST_FLOW reply (xid=0x4):
mininet> h7 ping -c 2 h14
PING 127.0.2.6 (127.0.2.6) 56(84) bytes of data.
64 bytes from 127.0.2.6: icmp_seq=1 ttl=64 time=0.028 ms
64 bytes from 127.0.2.6: icmp_seq=2 ttl=64 time=0.038 ms

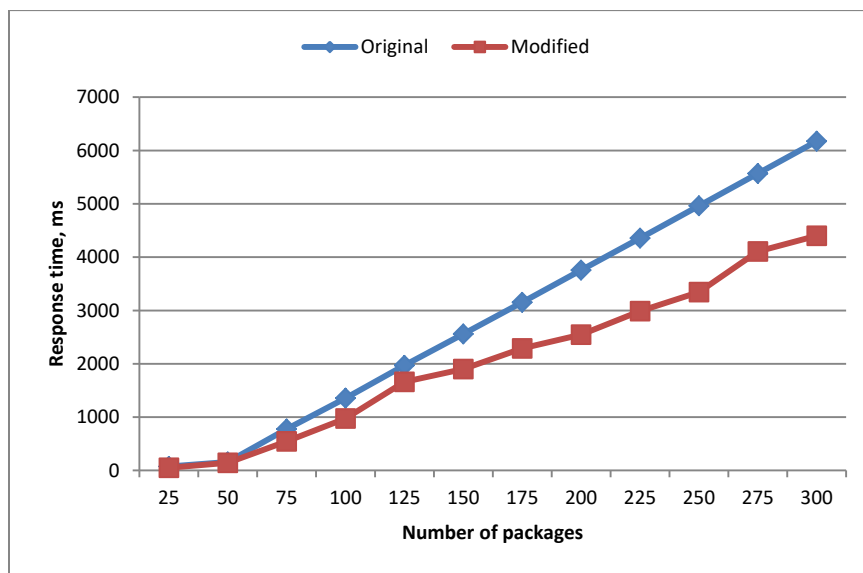
--- 127.0.2.6 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 0.028/0.033/0.038/0.005 ms
```

Fig. 6. Start Dump Collection and Start Traffic.

```
mininet> h7 ping -c 2 h14
PING 127.0.3.7 (127.0.3.7) 56(84) bytes of data.
64 bytes from 127.0.3.7: icmp_seq=1 ttl=64 time=0.087 ms
64 bytes from 127.0.3.7: icmp_seq=2 ttl=64 time=0.041 ms

--- 127.0.3.7 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 0.021/0.026/0.032/0.005 ms
mininet>
```

Fig. 7. DFS Modeling.



Graph 2. The Dependence of the Average Response Time on the Number of Packets.

TABLE II. SIMULATION RESULT FOR A LARGE NUMBER OF PACKAGES

Number of packages		25	50	75	100	125	150	175	200	225	250	275	300
Response time, ms	Original algorithm	76	158	776	1357	1972	2559	3153	3757	4357	4962	5570	6175
	Modified algorithm	51	143	547	974	1663	1901	2289	2547	2989	3346	4107	4401

#### IV. CONCLUSION

This paper simulates the operation of a modified method of constructing traffic in networked data centers with a branched topology, which, given the peculiarities of the organization of SDN, reduces the time of forming a set of routes to access network resources and simplify the procedure for changing the route.

Topology data using this method [6] allow to virtually eliminate the delay or loss of packets in the process of traffic reconstruction. At the same time, the more paths formed in the topologies, the less likely it is that packets will be delayed or lost.

The results, based on the two most popular topologies for large businesses, showed a 30-50 percent acceleration of packet transmission between nodes due to the collection of information about communication channels by controllers and the construction of a contiguity matrix. As'ad et al [21], [22], [23] and [24] enter new concept on networks and computer science which it bipolar intuitionistic fuzzy sets and he used it in many of his articles.

#### REFERENCES

[1] Aguado, M. Davis, S. Peng, M.V. Álvarez, V. LÁpez, T. Szyrkowicz, A. Autenrieth, R. Vilalta, A. Mayoral, R. Muoz, R. Casellas, R. Martnez, N. Yoshikane, T. Tsuritani, R. Nejabati, D. Simeonidou, Dynamic virtual network reconfiguration over sdn orchestrated multitechnology optical transport domains, *J. Lightwave Technol.* 34 (8) (2016).

[2] Y. Han, S. Seo, J. Li, J. Hyun, J. Yoo, J. W. Hong, Software Defined Networking-based Traffic Engineering for Data Center Networks: In *Asia-Pacific Network Operations and Management Symposium* (2014), <https://ieeexplore.ieee.org/document/6996601>.

[3] E. Chemerinsky, R. Smeliansky, On QoS Management in SDN by Multipath Routing. In: *Proceedings International Science and Technology Conference (Modern Networking Technologies) (MoNeTeC)* (2014) <https://ieeexplore.ieee.org/document/6995581>.

[4] Y. Kulakov, D. Korenko, Traffic engineering in DCN with a ramified topology. 2020 6th High Performance Computing Conference, 2020.

[5] ZHAOGANG SHU, JIAFU WAN, JIAXIANG LIN, Traffic Engineering in Software-Defined Networking: Measurement and Management, *IEEE Access*. – 2016. – №4. – C. 3246–3256.

[6] 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>.

[7] B. Isong, T. Kgogo, F. Lugayizi , Trust establishment in SDN: controller and applications. *Int. J. Comput. Netw. Inf. Secur. (IJCNIS)* 9(7), 20–28 (2017) . <http://dx.doi.org/10.5815/ijcnis.2017.07.03>.

[8] 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>.

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

[10] Katz, Dave (2000). OSPF and IS-IS, A Comparative Anatomy. North American Network Operators Group NANOG 19. Albuquerque. Archived from the original on June 20, 2018.

[11] P. Kumar, R. Dutta, R. Dagdi, K. Sooda, A. Naik, A programmable and managed softwaredefined 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.

[12] Kleinberg, Jon; Tardos, Éva (2006), *Algorithm Design*, Addison Wesley, pp. 92–94.

[13] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein., *Introduction to Algorithms*, Second Edition. MIT Press and McGraw-Hill, 2001. ISBN 0-262-03293-7. Section 22.3: Depth-first search, pp. 540–549.

[14] B. Isong, T. Kgogo, F. Lugayizi, Trust establishment in SDN: controller and applications. *Int. J. Comput. Netw. Inf. Secur. (IJCNIS)* 9(7), 20–28 (2017). <https://doi.org/10.5815/ijcnis.2017.07.03>.

[15] M. Luo, Y. Zeng, J. Li, W. Chou, An adaptive multi-path computation framework for centrally controlled networks. *Comput. Netw.* 83, 30–44 (2015).

[16] 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>. In *MECS* <http://www.mecs-press.org/>. Accessed may 2016.

[17] Z. Shu, J. Wan, J. Lin, S. Wang, D. Li, S. Rho, C. Yang, Traffic engineering in softwaredefined networking: measurement and management. *IEEE Access* 4, 3246–3256 (2016). [http://www.ieee.org/publications\\_standards/publications/rights/index.html](http://www.ieee.org/publications_standards/publications/rights/index.html).

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

[19] A. M. A. Alnaser, Streaming algorithm for multipath secure routing in mobile networks, *IJCSI International Journal of Computer Science Issues*, Vol. 11, Issue 4, No 1, July 2014 pp 112 – 114.

[20] M. R. Abbasi, A. Guleria, M.S.Devi, Traffic engineering in software defined networks: a survey. *J. Telecommun. Inf. Technol.* 4, 3–13 (2016).

[21] A. M. A. Alnaser, Novel Properties for Total Strong - Weak Domination Over Bipolar Intuitionistic Fuzzy Graphs.

[22] Ahlam Fallatah, As'ad Alnaser, Mourad Oqla Massa'deh , Bipolar Intuitionistic Fuzzy Graph Over Cayley Groups , *J. Math. Comput. Sci.* 11 (2021), No. 5, 6403-6419.

[23] Ahlam Fallatah, Mourad Oqla Massa'deh, As'ad Mahmoud As'ad Alnaser , Some Contributions on Operations and Connectivity Notations in Intuitionistic Fuzzy Soft GRAPHS , *Advances and Applications in Discrete Mathematics* , Volume 23, Number 2, 2020, Pages 117-138.

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