Journal of Computer Science

Original Research Paper

Certain Contributions in Traffic Engineering Based on Software-Defined Networking Technology

As'ad Mahmoud As'ad Alnaser

Department of Applied Science, Ajloun College, Al-Balqa Applied University, Jordan

Article history
Received: 20-04-2019
Revised: 25-05-2019
Accepted: 08-07-2019

Email: Asad1-99@Bau.edu.jo

Abstract: The paper addresses the traffic engineering in Software-Defined Networks (SDN). A brief analysis of the features of the SDN organization, which allows to increase the Traffic Engineering (TE) efficiency, is given. The viability of using multipath routing with TE is substantiated. Taking into account the features of the SDN technology, a modified TE method, which allows to reduce the time complexity of generating a set of paths and to reduce the rerouting time, is proposed. The procedure and an example of the routing tables' formation are given. The results of modeling the TE process with changing communication links load are given. It is shown, that route reconfiguration does not significantly affect its traffic metric.

Keywords: Software-Defined Networking, Multipath Routing, Traffic Engineering

Introduction

Modern computer networks have large sizes and diverse hardware composition. This complicates the process of managing computer networks, in particular, Traffic Engineering (TE). With an increase in the size of computer networks and in the volume of network traffic, the task of evenly loading the network and reducing energy consumption is relevant.

To solve these problems, the technology of Software-Defined Networks (SDN) is currently used (Isong *et al*., 2017; Kumar *et al*., 2017; Sahoo *et al*., 2017). A distinctive feature of Software-Defined Networks is that the network is organized and managed at the software level with the use of virtual switches and a central SDN controller (Shu *et al*., 2016) Fig. 1.

This allows organizing both centralized and decentralized management of network resources, while expanding functionality of TE and of data transmission in the network. With centralized path building, the SDN controller has full information about the structure of the network and its components; this allows the paths to be optimized according to specified metrics in the process of path generation (Nunes *et al*., 2014).

The paper (Yasir *et al*., 2018) provides an analysis of various methods of traffic optimization in SDN networks. The advantages of using SDN technology for solving load balancing problems are given. Using a centralized method to control traffic using an SDN controller allows reducing latency and packet loss in a computer network (Agarwal *et al*., 2013).

Currently, to increase the efficiency of traffic management in computer networks, multipath routing methods are widely used (Liu *et al*., 2014; Singh *et al*., 2015; Anasane and Satao, 2016; Onthachi and Jayabal, 2018). In paper (Alnaser, 2014), a streaming multipath routing algorithm is presented, which allows to simultaneously create a set of paths between different pairs of network nodes. This algorithm has less time complexity in comparison with well-known multipath routing algorithms, but does not consider path metric. This complicates the TE process.

Alnaser (2017), a modified algorithm was proposed for generating the maximum set of disjoint paths, taking their metrics into account. Path optimization is performed by their reconfiguration with an adjacent path. Reconfiguration is performed within subgraphs that include only the vertices of the main and the adjacent paths. This allows reducing the area of path formation and reducing the time complexity of its formation.

The paper (Kulakov *et al*., 2018) presents a multipath routing method in network data centers, which, by taking into account the network topology and centralized management, makes it possible to optimize the procedure for generating a set of disjoint paths.

DOI: 10.3844/jcssp.2019.944.953

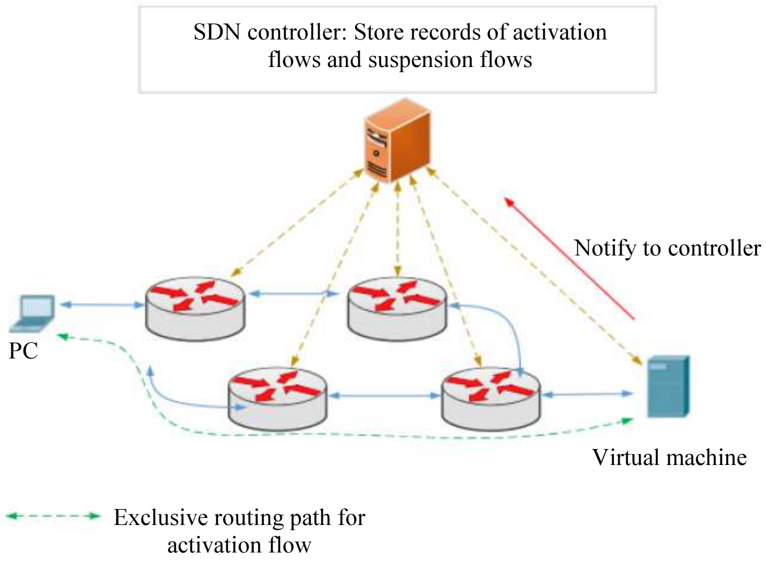

Fig. 1: Structure of a software-defined network

The centralized formation of multiple paths based on multipath routing in SDN allows us to reduce TE time and improve the Quality of traffic Service (QoS) (Chemeritskiy and Smeliansky, 2014).

Thus, the development of TE methods based on multi-path routing, taking into account the features and advantages of SDN, is relevant (Alnaser, 2018).

Method of Multipath Traffic Engineering Based on SDN

Procedure for Exchanging Control Information Between an SDN Controller and Network Router

This papers proposes a TE method based on combination of the advantages of centralized and decentralized routing methods. Complete set of paths between different routers is formed in the SDN controller using the modified method of forming a set of paths according to the link state. When forming a path between two remote nodes, the paths between their inner nodes are formed. This allows to significantly reduce the time complexity of path formation due to re-formation of paths between inner nodes of a formed path. The SDN controller updates the routing information in SDN switches by adjusting their routing tables. This allows to organize a dynamic reconfiguration of traffic and to ensure the most uniform network load.

Figure 2 shows the process of service information exchange between the sender router R_i and the SDN controller with TE.

At time t_0 router R_s sends a routing request with its number, the destination router R_0 number and the value of the Quality of Service (QoS) parameter. At time t_1 the SDN controllers determines the availability of valid paths. In the absence of valid path, the SDN controller uses a modified routing protocol to form a valid set of paths. At time t_2 the SDN controller sends routing tables to router R_i .Among available paths, the R_i router chooses the path with the minimum load and reports its number to the SDN controller. The data transmission to the chosen adjacent router begins. After the data transmission is completed, the routing tables are adjusted and updated.

The Data Transmission Process

The data transmission is performed sequentially from the sender router R_i to the adjacent router R_j towards the router R_a. The adjacent router R_j is determined by the routing table $Tm(i)$ of router R_i. The data transmission is carried out according to the following algorithm.

Data Transmission Algorithm

1. Assign the initial router R_s as a sender router R_i
2. The sender router determines from its routing table the availability of a path to router R_a with an acceptable QoS value
3. If there are valid paths in the routing table $Tm(i)$, go to step 5
4. If there are no valid paths, the R_i router sends request to the SDN controller to form new paths
5. Among the valid paths, the R_i router chooses paths to the router *R*a with an acceptable metric and the minimum load value
6. Router R_i sends data packet to the chosen adjacent router R_j
7. If the packet is not the last, go to step 2
8. The end of the data transmission process

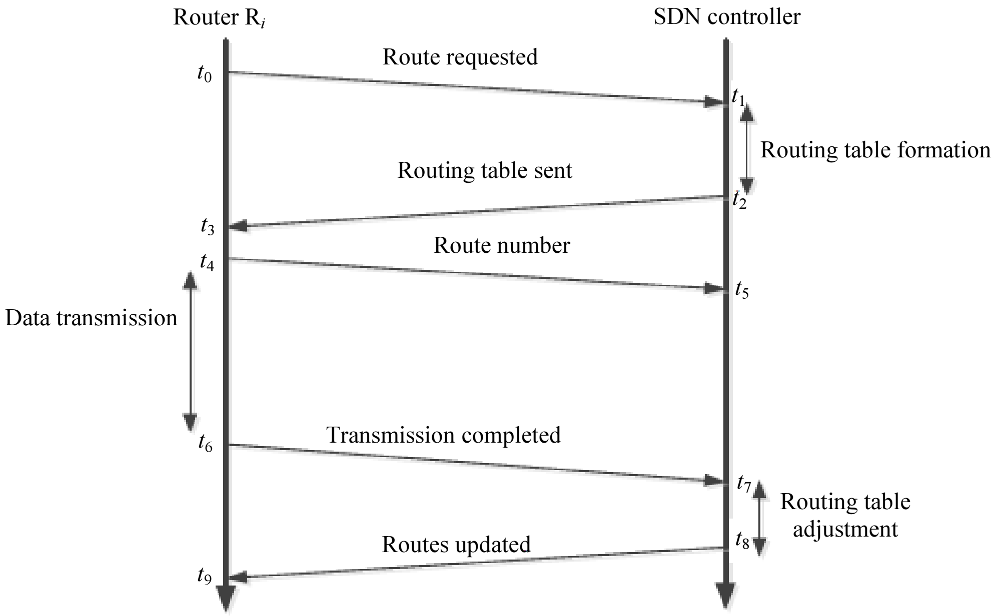

Fig. 2: The procedure for service information exchange between an SDN controller and a network router

Formation of the Routing Information

Alnaser (2018), a centralized wave algorithm for forming a set of paths based on SDN technology is presented. The main advantage of this method is that it is possible to form multiple paths between several network routers at the same time. When forming a path between two remote routers, paths between its internal routers are simultaneously formed. This allows to significantly reduce the time complexity of the formation of a paths set. The number of hops is selected as the metric. This allows creating a path of the minimum length, but does not allow optimizing network traffic and ensuring uniform network load.

In this paper, a combined metric is used, which allows the construction of traffic based on QoS and link load. The combined metric takes into account the amount of traffic $S(L_{i,j})$ transmitted over each link $L_{i,j}$, the bandwidth capacity $B(L_{i,j})$ and the time delay τ_i of data transmission over this link.

To ensure a uniform network load, the load factor $d_{i,j} = S(L_{i,j})/\mathrm{B}(L_{i,j})$ is taken into account. The value $d_{i,j}$ is used when choosing the transmission paths in order to optimize the load in the network. The load D_i of path P_i is determined by the maximum value of its links load ($max\ d_{i,j}$).

The formation of paths is performed sequentially between routers $R_j \in W_{i+1}$ and $R_k \in W_i$ of adjacent sets. Adjacent router sets are determined as $W_{i+1} = \{R_j\}$ and $W_i = \{R_k\}$ with the shared links $L_{j,k}$, where: $R_j \in W_{i+1}$ and $\mathrm{R}_k \in W_i$. The paths formation begins with the destination router R_s with $i = 1$. In this case the set $W_1 = \{R_s\}$ and the set $W_2 = \{R_j\}$ is the set of routers adjacent to the router R_s. Then, for routers $R_j \in W_2$ adjacent to the router R_s the routing tables towards the router R_a are formed. On the second wave of routing, the formation of routes from routers $R_j \in W_{i+1}$ to routers $R_k \in W$ continues. As a result, tables of routes from routers $R_j \in W_{i+1}$ to router R_a are formed. The process continues until all the paths between routers R_s and R_a are formed. Figure 3 shows the network graph and the result of the formation of the set of paths from router R_s to router R_a. In this case, the relative transmission rate $V_i = m_s/n$ was chosen, where m_s is the path bandwidth; n is the number of links in the transmission path.

With a modified routing algorithm, a set of slightly different adjacent paths is formed. Figure 4 shows the set of formed paths with the maximum metric.

This allows changing the route Fig. 5 during the data transmission without the additional delay and packet loss.

The use of link-state routing protocols like OSPF is not effective when forming a large number of paths. Data transmission over a specific path complicates the rerouting process in case of changed metric. In this case, distance-vector protocols like RIP are more effective. The formation and updating of the routing information is performed centrally in the SDN controller. This allows reducing the time of formation of route information and eliminates the problem of looping routes. A routing table is built for every router R_j (Table 1).

The Procedure of Route Information Formation

1. Set a set of routers $W_1 = \{R_n\}$
2. $D_i = 0$
3. $J = 0$
4. for $j = j+1$ step 1 form a set of routers $W_{j+1} = \{R_i | i = 1,\ldots k\}$ adjacent with the routers of the set W_{j1} =

$\{R_i | i = 1,\ldots k\}$., where k is the sum of powers of routers set $W_{j1} = \{ R_i | i = 1,\ldots k\}$;

5. if $W_{j+1} = \varnothing$ then go to 10 do.
6. for $i = 1$ step 1 until k calculate $Z_i \{V_n, V_l, M_{l,n}, d_{ij}\}$
7. If $d_j > D_i$ then $D_i = d_j$
8. end
9. go to 4
10. end

Let us consider an example of routing table formation when transmitting data from router R_0 to router R_{12} Fig. 3. With the different link loads $L_{i,j} = (R_i, R_j)$ (Table 3).

Using the routing table, adjacent routers are determined toward the destination router, as well as the path metric and load. For example, for router R_5, Fig. 5, the following routing table is formed (Table 2).

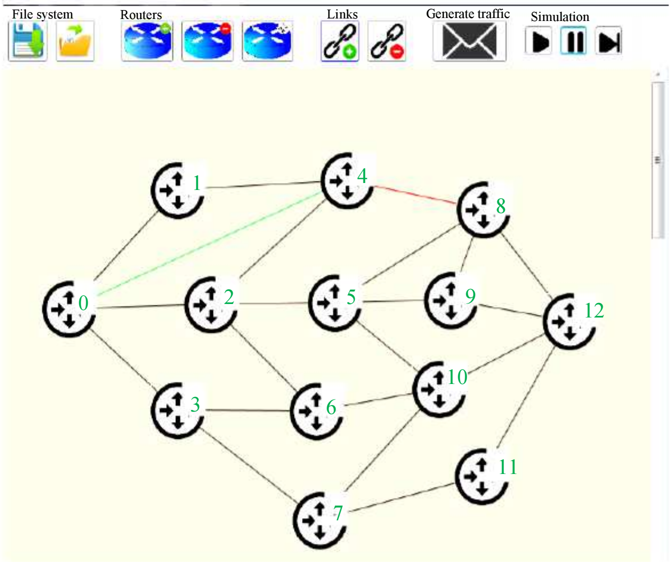

Fig. 3: The computer network graph

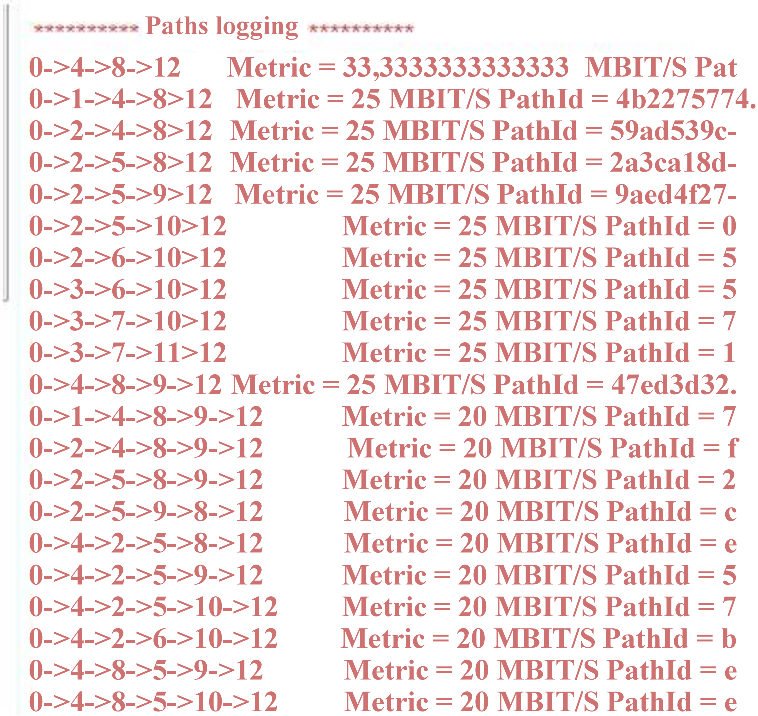

Fig. 4: A set of paths between routers R_o and R_{12}

DOI: 10.3844/jcssp.2019.944.953

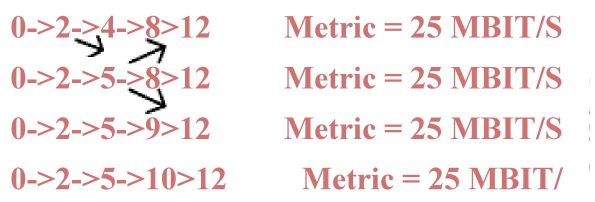

Fig. 5: Changing the route during the data transmission without the additional delay and packet loss

Table 1: Routing table for router R_j

Router

Destination	Adjacent	Path metric	Path load
R_a	R_i	$M_{j,a}$	D_l

Table 2: Routing table for router R_5

Router

Destination	Adjacent	Path metric	Path load
R_{12}	R_8	$M_{j,a}$	D_l
R_{12}	R_9	$M_{j,a}$	D_l
R_{12}	R_{10}	$M_{j,a}$	D_l

The Steps of Routing Tables Formation

1. $i = 1$
2. A set of routers $W_1 = \{R_{12}\}$ is formed
3. Based on Table 3, a set of routers $W_2 = \{R_8, R_9, R_{10}, R_{11}\}$ adjacent to router R_{12} is formed
4. Routing tables (Tables 4-7) for the set of routers $W_2 = \{R_8, R_9, R_{10}, R_{11}\}$ are formed, taking into account the load of the path to router R_{12}

The row and column R_{12} are deleted from the network adjacency Table (3). Next level adjacency table is formed (Table 8):

1. Based on Table 8, a set of routers $W_3 = \{R_4, R_5, R_6, R_7, R_8, R_9, R_{10}\}$ adjacent with the set of routers $W_2 = \{R_4, R_5, R_6, R_7, R_8, R_9, R_{10}, R_{11}\}$ is formed
2. Routing tables (Tables 9-12) for the set of routers $W_3 = \{R_4, R_5, R_6, R_7, R_8, R_9, R_{10}\}$ are formed, taking into account the load of the path to router R_{12}

Routing tables for routers R_8 and R_9 are updated (Tables 13 and 14).

A set of routers $W_2 = \{R_8, R_9, R_{10}, R_{11}\}$ is deleted from the network adjacency Table (8). Next level adjacency table is formed (Table 15).

1. Based on Table 15, a set of routers $W_4 = \{R_0, R_1, R_2, R_3\}$ adjacent with the set of routers $W_3 = \{R_4, R_5, R_6, R_7\}$ is formed
2. Routing tables (Tables 16-19) for the set of routers $W_4 = \{R_0, R_1, R_2, R_3\}$ are formed, taking into account the load of the path to router R_{12}

1. Based on Table 20, a set of routers $W_4 = \{R_0,\}$ adjacent with the set of routers $W_3 = \{R_1, R_2, R_3,\}$ is formed
2. Routing table for router R_0 is adjusted (Table 21)

As a result, path for data transmission will be $R_0 \to R_4 \to R_8 \to R_{12}$.

Table 3: Network adjacency table

	R_0	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}	R_{11}	R_{12}
R_0	0.0	0.2	0.1	0.1	0.1								
R_1	0.2	0.0			0.3								
R_2	0,1		0.0		0,3	0.2	0.1						
R_3	0.1			0.0			0.4	0.3					
R_4	0.1	0.3	0.3		0.0				0.1				
R_5			0.2			0.0			0.2	0.1	0.4		
R_6			0.1	0.4			0.0	0.2			0.3		
R_7				0.3			0.2	0.0			0.1	0.4	
R_8					0.1	0.2			0.0	0.5			0.1
R_9						0.1			0.5	0.0			0.2
R_{10}						0.4	0.3	0.1			0.0		0.2
R_{11}								0.4				0.0	0.1
R_{12}									0.1	0.2	0.2	0.1	0.0

Table 4: Routing table for router R_8

Router

Destination	Adjacent	Path metric	Path load
R_{12}	R_{12}	$M_{j,a}$	0.1

Table 5: Routing table for router R_9

Router

Destination	Adjacent	Path metric	Path load
R_{12}	R_{12}	$M_{j,a}$	0.2

DOI: 10.3844/jcssp.2019.944.953

Table 6: Routing table for router *R10*

Router -- Destination	Adjacent	Path metric	Path load
R_{12}	R_{12}	$M_{j,a}$	0.2

Table 7: Routing table for router R_{11}

Router -- Destination	Adjacent	Path metric	Path load
R_{12}	R_{12}	$M_{j,a}$	0.1

Table 8: Network adjacency table

	R_0	R_1	R_2	R_3	$\boldsymbol{R_4}$	$\boldsymbol{R_5}$	$\boldsymbol{R_6}$	$\boldsymbol{R_7}$	$\boldsymbol{R_8}$	$\boldsymbol{R_9}$	R_{10}	R_{11}
R_0	0.0	0.2	0.1	0.1	0.1							
R_1	0.2	0.0			0.3							
R_2	0,1		0.0		0,3	0.2	0.1					
R_3	0.1			0.0			0.4	0.3				
R_4	0.1	0.3	0.3		0.0				0.1			
R_5			0.2			0.0			0.2	0.1	0.4	
R_6			0.1	0.4			0.0	0.2			0.3	
R_7				0.3			0.2	0.0			0.1	0.4
$\boldsymbol{R_8}$					**0.1**	**0.2**			**0.0**	**0.5**		
$\boldsymbol{R_9}$						**0.1**			**0.5**	**0.0**		
$\boldsymbol{R_{10}}$						**0.4**	**0.3**	**0.1**			**0.0**	
$\boldsymbol{R_{11}}$								**0.4**				**0.0**

Table 9: Routing table for router R_4

Router -- Destination	Adjacent	Path metric	Path load
R_{12}	R_8	$M_{j,a}$	0.1

Table 10: Routing table for router R_5

Router -- Destination	Adjacent	Path metric	Path load
R_{12}	R_8	$M_{j,a}$	0.2
R_{12}	R_9	$M_{j,a}$	0.2
R_{12}	R_{10}	$M_{j,a}$	0.4

Table 11: Routing table for router R_6

Router -- Destination	Adjacent	Path metric	Path load
R_{12}	R_{10}	$M_{j,a}$	0.2

Table 12: Routing table for router R_7

Router -- Destination	Adjacent	Path metric	Path load
R_{12}	R_5	$M_{j,a}$	0.2
R_{12}	R_6	$M_{j,a}$	0.2
R_{12}	R_{10}	$M_{j,a}$	0.4

Table 13: Routing table for router R_8

Router -- Destination	Adjacent	Path metric	Path load
R_{12}	R_{12}	$M_{j,a}$	0.1
R_{12}	R_9	$M_{j,a}$	0.5

Table 14: Routing table for router R_9

Router			
Destination	Adjacent	Path metric	Path load
R_{12}	R_{12}	$M_{j,a}$	0.2
R_{12}	R_8	$M_{j,a}$	0.5

Table 15: Network adjacency table

	R_0	R_1	R_2	R_3	R_4	R_5	R_6	R_7
R_0	0.0	0.2	0.1	0.1	0.1			
R_1	0.2	0.0			0.3			
R_2	0,1		0.0		0,3	0.2	0.1	
R_3	0.1			0.0			0.4	0.3
$\boldsymbol{R_4}$	**0.1**	**0.3**	**0.3**		**0.0**			
$\boldsymbol{R_5}$			**0.2**			**0.0**		
$\boldsymbol{R_6}$			**0.1**	**0.4**			**0.0**	**0.2**
$\boldsymbol{R_7}$				**0.3**			**0.2**	**0.0**

Table 16: Routing table for router R_0

Router			
Destination	Adjacent	Path metric	Path load
R_{12}	R_4	$M_{j,a}$	0.1

Table 17: Routing table for router R_1

Router			
Destination	Adjacent	Path metric	Path load
R_{12}	R_4	$M_{j,a}$	0.3

Table 18: Routing table for router R_2

Router			
Destination	Adjacent	Path metric	Path load
R_{12}	R_4	$M_{j,a}$	0.3
R_{12}	R_5	$M_{j,a}$	0.2
R_{12}	R_6	$M_{j,a}$	0.2

Table 19: Routing table for router R_3

Router			
Destination	Adjacent	Path metric	Path load
R_{12}	R_6	$M_{j,a}$	0.4
R_{12}	R_7	$M_{j,a}$	0.3

Table 20: Network adjacency table

	R_0	R_1	R_2	R_3
R_0	0.0	0.2	0.1	0.1
$\boldsymbol{R_1}$	**0.2**	**0.0**		
$\boldsymbol{R_2}$	**0,1**		**0.0**	
$\boldsymbol{R_3}$	**0.1**			**0.0**

Table 21: Routing table for router R_0

Router			
Destination	Adjacent	Path metric	Path load
R_{12}	R_4	$M_{j,a}$	0.1
R_{12}	R_1	$M_{j,a}$	0.2
R_{12}	R_2	$M_{j,a}$	0.1
R_{12}	R_3	$M_{j,a}$	0.1

DOI: 10.3844/jcssp.2019.944.953

Dynamic Rerouting

The rerouting procedure consists in dynamically changing routing tables of the routers that make up a specific path. The SDN controller receives from the network switches the information about changes in link states, adjusts the routing tables of the corresponding network switches. Having several paths in every routing table allows a quick reconfiguration of paths during data transmission. This almost does not lead to data transmission delays and packets loss.

Figure 6 shows the set of optimal paths between routers R_0 and R_{12} when the link $L_{8,12}$ is overloaded.

The choice of route depends on the routing table of the current router R_i in which the packet is currently located. If packet is currently in the router R_8, then the path $R_8 \rightarrow R_9 \rightarrow R_{12}$ or $R_8 \rightarrow R_5 \rightarrow R_{10} \rightarrow R_{12}$ is chosen depending on their load. Having several paths eliminates the need for recalculation of the routes and simplifies the rerouting procedure.

When the path $R_8 \rightarrow R_{12}$ changes to the path $R_8 \rightarrow R_9 \rightarrow R_{12}$, its load becomes equal to 0.6 (Table 22).

Then, the SDN controller makes changes to the routing tables of routers R_0, R_1, R_2, R_4, R_5, located on the path between routers R_0 and R_8 Fig. 6 (Tables 23-27).

Traffic is transferred over to the minimally loaded ($D_l = 0.2$) path: $R_0 \rightarrow R_2 \rightarrow R_5 \rightarrow R_9 \rightarrow R_{12}$.

Disconnecting the router also does not significantly affect the routing time.

Figure 7 shows the network graph and the result of the formation of a set of paths from router R_0 to router R_{12} when the router R_8 is disconnected.

Figure 8 shows the set of optimal paths between the routers R_0 and R_{12} when the router R_8 is disconnected.

In this case, the packet sent by router R_4 is returned to router R_2. At the same time, the packet transmission delay increases slightly. The following packets from R_0 are forwarded without delay to router R_2 or router R_3, depending on their load.

As the simulation results showed, the proposed TE method allows us to basically eliminate the packet delay and loss in the rerouting process.

```
********** Paths logging **********
0->2->5->9->12   Metric = 25 MBIT/S PathId = 727d3d16
0->2->5->10->12       Metric = 25 MBIT/S PathId =
0->2->6->10->12       Metric = 25 MBIT/S PathId =
0->3->6->10->12       Metric = 25 MBIT/S PathId =
0->3->7->10->12       Metric = 25 MBIT/S PathId =
0->4->7->11->12       Metric = 25 MBIT/S PathId =
0->4->8->9->12   Metric = 25 MBIT/S PathId = f741c3c7.
0->1->4->8->9->12     Metric = 20 MBIT/S PathId =
0->2->5->8->9->12     Metric = 20 MBIT/S PathId =
0->4->8->5->9->12     Metric = 20 MBIT/S PathId =
0->4->8->5->10->12    Metric = 20 MBIT/S PathId =
```

Fig. 6: Set of paths

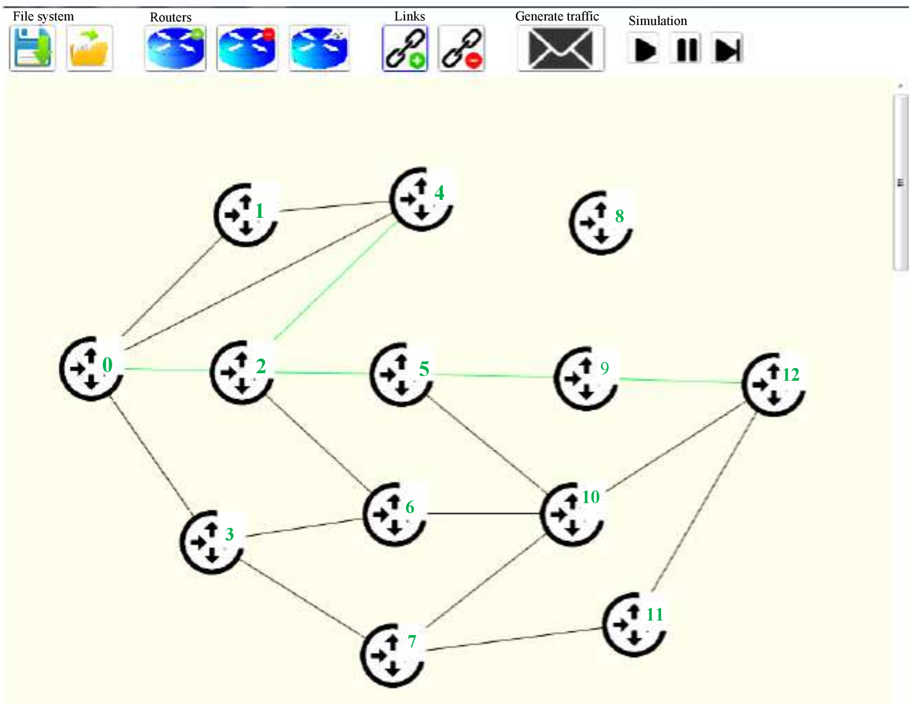

Fig. 7: Network graph with the router R_8 disconnected

DOI: 10.3844/jcssp.2019.944.953

```
**********   Paths logging   **********
0->2->5->9->12   Metric = 25 MBIT/S PathId = fa0af67b
0->2->5->10->12        Metric = 25 MBIT/S PathId =
0->2->6->10->12        Metric = 25 MBIT/S PathId =
0->3->6->10->12        Metric = 25 MBIT/S PathId =
0->3->7->10->12        Metric = 25 MBIT/S PathId =
0->3->7->11->12        Metric = 25 MBIT/S PathId =
0->4->2->5->9->12      Metric = 20 MBIT/S PathId =
0->4->2->5->10->12     Metric = 20 MBIT/S Pathid =
0->4->2->6->10->12     Metric = 20 MBIT/S Pathid =
```

Fig. 8: Set of optimal paths when the router R_8 is disconnected

Table 22: Routing table for router *R8*

Router -- Destination	Adjacent	Path metric	Path load
$\boldsymbol{R_{12}}$	$\boldsymbol{R_9}$	$\boldsymbol{M_{j,a}}$	**0.5**

Table 23: Routing table for router R_4

Router -- Destination	Adjacent	Path metric	Path load
$\boldsymbol{R_{12}}$	$\boldsymbol{R_8}$	$\boldsymbol{M_{j,a}}$	**0.5**

Table 24: Routing table for router R_5

Router --- Destination	Adjacent	Path metric	Path load
$\boldsymbol{R_{12}}$	$\boldsymbol{R_8}$	$\boldsymbol{M_{j,a}}$	**0.5**
R_{12}	R_9	$M_{j,a}$	0.2
R_{12}	R_{10}	$M_{j,a}$	0.4

Table 25: Routing table for router R_1

Router --- Destination	Adjacent	Path metric	Path load
$\boldsymbol{R_{12}}$	$\boldsymbol{R_4}$	$\boldsymbol{M_{j,a}}$	**0.5**

Table 26: Routing table for router R_2

Router --- Destination	Adjacent	Path metric	Path load
$\boldsymbol{R_{12}}$	$\boldsymbol{R_4}$	$\boldsymbol{M_{j,a}}$	**0.5**
R_{12}	R_5	$Mj_{,a}$	0.2
R_{12}	R_6	$M_{j,a}$	0.2

Table 27: Routing table for router R_0

Router --------------------------------			
Destination	Adjacent	Path metric	Path load
$\boldsymbol{R_{12}}$	$\boldsymbol{R_4}$	$\boldsymbol{M_{j,a}}$	**0.5**
$\boldsymbol{R_{12}}$	$\boldsymbol{R_1}$	$\boldsymbol{M_{j,a}}$	**0.5**
R_{12}	R_2	$M_{j,a}$	0.2
R_{12}	R_3	$M_{j,a}$	0.3

Conclusion

This paper proposes a method for traffic engineering, which, by taking into account the specifics of SDN, in particular the presence of a centralized network controller, reduces the time required to form a set of access routes to networks resources and simplifies the rerouting process. Having multiple routes allows us to basically eliminate the packet delay and loss in the process of traffic rerouting. At the same time, the more paths are formed in the SDN controller, the less is the possibility of packet delay or loss.

The proposed method of forming the routing information allows to increase the efficiency of the TE procedure and to ensure a more uniform load of the data transmission links with given QoS parameters.

Author’s Contributions

As'ad Mahmoud As'ad Alnaser: Main author collecting data, run the experiments and write the manuscripts.

Acknowledgement

The author thanks the referees for careful reading of the paper and suggestions for improvement.

Ethics

The author declare that there are no ethical issues associated with this work.

References

Agarwal, S., M. Kodialam and T.V. Lakshman, 2013. Traffic engineering in software defined networks. Proceedings of the IEEE INFOCOM, (PII’ 13), IEEE, pp: 2211-2219.

Alnaser, A.M.A., 2017. A method of forming the optimal set of disjoint path in computer networks. J. Applied Comput. Sci. Math., 11: 9-12. DOI: 10.4316/JACSM.201701001

Alnaser, A.M.A., 2018. A method of multipath routing in SDN networks. Advances Comput. Sci. Eng., 17: 11-17.

Alnaser, A.M.A., 2014. Streaming algorithm for multi-path secure routing in mobile networks. Int. J. Comput. Sci. Issues, 11: 112-114.

Anasane, A.A. and R.A. Satao, 2016. A survey on various multipath routing protocols in wireless sensor networks. Procedia Comput. Sci., 79: 610-615. DOI: 10.1016/j.procs.2016.03.077

Chemeritskiy, E. and R. Smeliansky, 2014. On QoS management in SDN by multipath routing. Proceedings of the Science and Technology Conference (Modern Networking Technologies), (MNT’ 14), IEEE, pp: 1-6. DOI: 10.1109/MoNeTeC.2014.6995581

Isong, B., T. Kgogo and F. Lugayizi, 2017. Trust establishment in SDN: Controller and applications. Int. J. Comput. Netw. Inform. Security, 9: 20-28. DOI: 10.5815/ijcnis.2017.07.03

Kulakov, Y., S. Kopychko and V. Gromova, 2018. Organization of network data centers based on software-defined networking. Proceedings of the International Conference on Computer Science, Engineering and Education Applications, (EEA’ 18), Springer, Cham, pp: 447-455.

Kumar, P., R. Dutta, R. Dagdi, K. Sooda and A. Naik, 2017. A programmable and managed software defined network. Int. J. Comput. Netw. Informat. Security, 10: 11-17. DOI: 10.5815/ijcnis.2017.12.02

Liu, X., S. Mohanraj, M. Pióro and D. Medhi, 2014. Multipath routing from a traffic engineering perspective: How beneficial is it? Proceedings of the IEEE 22nd International Conference on Network Protocols, (CNP’ 14), IEEE, pp: 143-154. DOI: 10.1109/ICNP.2014.34

Nunes, B.A.A., M. Mendonca, X.N. Nguyen, K. Obraczka and T. Turletti, 2014. A survey of software-defined networking: Past, present and future of programmable networks. IEEE Communicat. Surveys Tutorials, 16: 1617-1634.

Onthachi, D. and S. Jayabal, 2018. An optimized QoS-based multipath routing protocol for wireless sensor networks. Int. J. Intell. Eng. Syst., 11: 49-56. DOI: 10.22266/ijies2018.0430.06

Sahoo, K.S., S.K. Mishra, S. Sahoo and B. Sahoo, 2017. Software defined network: The next generation internet technology. Int. J. Wireless Microwave Technol., 7: 13-24. DOI: 10.5815/ijwmt.2017.02.02

Shu, Z., J. Wan, J. Lin, S. Wang and D. Li *et al*., 2016. Traffic engineering in software-defined networking: Measure. Manage. IEEE Access, 4: 3246-3256.

Singh, S.K., T. Das and A. Jukan, 2015. A survey on internet multipath routing and provisioning. IEEE Communicat. Surveys Tutorials, 17: 2157-2175.

Yasir, A.M., C.H. Abooddy and Z.Q. Mohammed, 2018. Analyzing methods and opportunities in Software-Defined (SDN) networks for data traffic optimizations. Int. J. Recent Innovat. Trends Comput. Communicat., 6: 75-82.