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

AN APPROACH TO THE EVOLUTION FROM AN OSPF NETWORK TO SDN

RUSLAN LEONIDOVICH SMELYANSKIY*, RAMIL AIDAROVICH YANBULATOV

Non-Profit Partnership "Applied Research Center for Computer Networks" (ARCCN), Moscow 142784, Russian Federation

* Address correspondence to this author at the Non-Profit Partnership "Applied Research Center for Computer Networks" (ARCCN), Moscowskiy town, Business Park «Rumyantsevo», Moscow 142784, Russian Federation; Email echemeritskiy@arccn.ru

ABSTRACT

Increasing requirements to the quality of service and the dramatic growth of traffic are the main driving development factors of modern network technology. Traditional networks, such as TCP/IP networks, have a complex structure and it is difficult to control them. Thus, there is a need to shift to a higher-quality technology, such as SDN technology. With centralized network management and programmability of control units, the SDN architecture has the potential, which can reduce COPEX and OPEX, simplify the network administration and maintenance. However, replacement of the traditional networks by a new one is a long and spending process, which is also related to the lack of methodology. This paper discusses the methodology intended for the transition from a traditional OSPF/IS-IS network to a network based on SDN architecture. As a result, the authors developed greedy algorithm, which provides the stepped system for the transition from OSPF to software defined network.

Key words: *SDN, OSPF, hybrid SDN, Incremental Deployment, Network Migration*

1. INTRODUCTION

The end of the first decade of the new century was marked by the awareness of crisis in computer networks [3.36]. Modern information technologies have increased requirements for flexibility and scalability of computer networks. The problem is that traditional routers and switches does not separate traffic transmission function and management function. This challenge predetermined the emergence of a fundamentally new approach to their development – the construction of Software Defined Networks (SDN) [4]. SDN technology provides new opportunities for network management, simplifies automation of its administration and reduces the cost of network equipment, reduces dependency on vendor lock, opens up new opportunities for network virtualization. SDN network, consisting of a package of devices, serves as a logical switch for the application. This allows programming the network as a complete unit instead of dealing with a number of certain switches.

SDN includes three main components – the SDN controller, Northbound API, and Southbound API [21.22.23]:

The SDN controller acts as the "brain" of the network. It is responsible for arranging and displaying the topology, programming network devices, and serves as the single management point for the entire network;

Southbound API is used for the communication between the SDN controller and network devices. The most well-known communication protocol is called OpenFlow;

Northbound API is used to interpret the business logic in the form of network instructions. Using Northbound API, business applications can transfer information to the SDN controller for subsequent network programming. The interface enables administrators to allocate network resources is a flexible way based on the requirements of the applications while abstracting the network infrastructure.

SDN are effective in creating infrastructural cloud services in cases when, according to user requirements, it is necessary to create virtual nodes and allocate virtual network resources thereto automatically and within the shortest time possible [24.25.26].

In addition, it is rational to use SDN in large data processing centers, because they allow cutting network maintenance costs by centralizing management at the program controller and increasing the network resource utilization percentage through dynamic management [27.28.29].

Another promising implementation area of SDN are applications in the "Internet of Things" concept – applications that are based on the computer networks of physical objects, which are equipped with built-in technologies for interacting with each other and the external environment [30.31.32]. The key ideas of SDN are:

- Creation of a Software programmable interface between the network application and the transmission media;
- To switch from managing separate parts of the network equipment to managing the network as a whole.
- Separation of network equipment management from communication control by creating a special software that can run on a standard PC.

In addition, traditional networks are too static, which prevents them from keeping up with the dynamics of modern business, unlike servers, which have this property thanks to virtualization technologies. Nowadays, applications are distributed among multiple virtual machines that exchange data intensively (which leads to increases in west-east traffic, which is beginning to dominate over the north-south traffic that is traditional for client-server architectures). In order to optimize server loads, virtual machines often migrate, which changes traffic control points. Traditional schemes of addressing, logical network partition, and means of setting of traffic processing rules become inefficient in dynamic environments [33.34.35].

However, as shown by practice, introduction of even state-of-the-art technologies is doomed to failure if they require radical changes in the infrastructure [4].

Typically, due to technical and economic reasons (for example, high cost of full network deployment), network transformation into a new architecture does not happen at once. In practice, it always presents step-by-step transformation of the old architecture into the new one.

The article presents the methodology aimed at transition from a traditional OSPF network to SDN based network. The paper proposes a method of selecting routers in a traditional network to replace them by SDN switches in order to control over the largest number of data flows.

2. NOTATIONS

Hereafter, we will denote the network as a graph $G=(V,E)$, where G is a set of vertices which are routers that are interconnected by the edges from E. Specifically, $V=\{v_1,...,v_n\}$ is the set of vertices of this graph, $E \subset V \times V$ is the set of undirected edges $e=\{vi,vj\}$, linking a couple of directly adjacent routers, i.e., a channel between them.We use definition of the flow provided in [5], where the flow is defined as a bidirectional packet sequence with the following parameters:

- Source IP address;
- Destination IP address;
- Source port for UDP and TCP;
- Destination port for UDP and TCP;
- Message type and code for ICMP;
- IP protocol;
- Ingress interface (SNMP ifindex);
- IP Type of Service.

The criteria of optimization for OSPF routers selection may be different. For example, it may be the number of flows or the amount of data of all flows passing through the certain OSPF router. To summarize the problem in both cases, we introduce the concept of weight for a flow, a vertex and a route.

Definition 1. The flow weight is measured in bytes if we need to select the most "heavy" flows or it is equal to 1 if the maximum number of flows is needed.
Definition 2. Vertex weight is equal to the total weight of the flows passing through this vertex for a certain period.
Definition 3. We count the weight of the route as the sum of the weights of the flows, which have exactly the same route.

3. STATEMENT OF THE PROBLEM

We are given LAN with TCP/IP architecture, which data-link layer is based on the Ethernet technology. The network has:

a) Computers with running applications;
b) Routers running OSPF (or IS-IS). We call them OSPF routers.

In addition, we are given the fixed number of SDN switches available to replace the OSPF routers. We assume that SDN switches can serve as an Openflow switch as well as an OSPF router. We need to identify, which OSPF routers should be replaced by the

SDN switches in order to control with the SDN controller the largest number of ”heavy” flows.

In this paper, we will consider the problem in three different cases:

1. We know the route and its weight for every flow passed through the network within the certain period of time.
2. We know the weight of every vertex in the network graph.
3. We rely only on information on network topology, the information related to data flows is not used.

4. CASE 1 OF THE PROBLEM

Here we will consider the problem in case 1 where we need the information about every flow route and the routes’ weights.

4.1 The need for information regarding flow routes

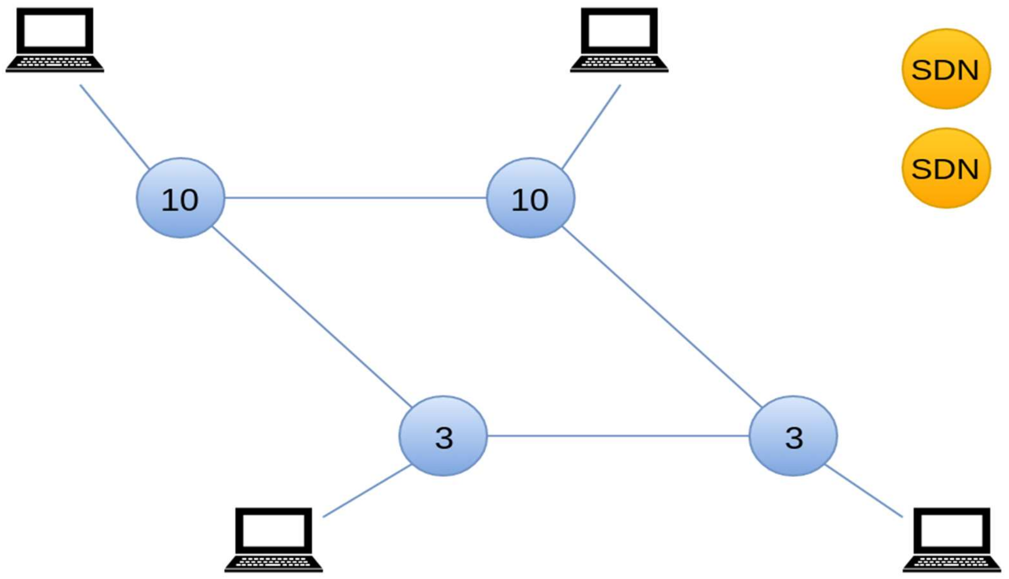

Figure 1. Flow routes are not acquainted

Let us consider the network shown in Figure 1, which includes four routers, two SDN switches for replacement and assume we know how many flows passed through each router. Not taking into account the flow routes, it is worth to replace those two routers, which were passed by 10 flows. However, knowing the routes of the flows (Figure 2) it becomes clear that it is sufficient to replace the routers A and C to take SDN control over all flows in the network, whereas in the above discussion only 10 flows were covered.

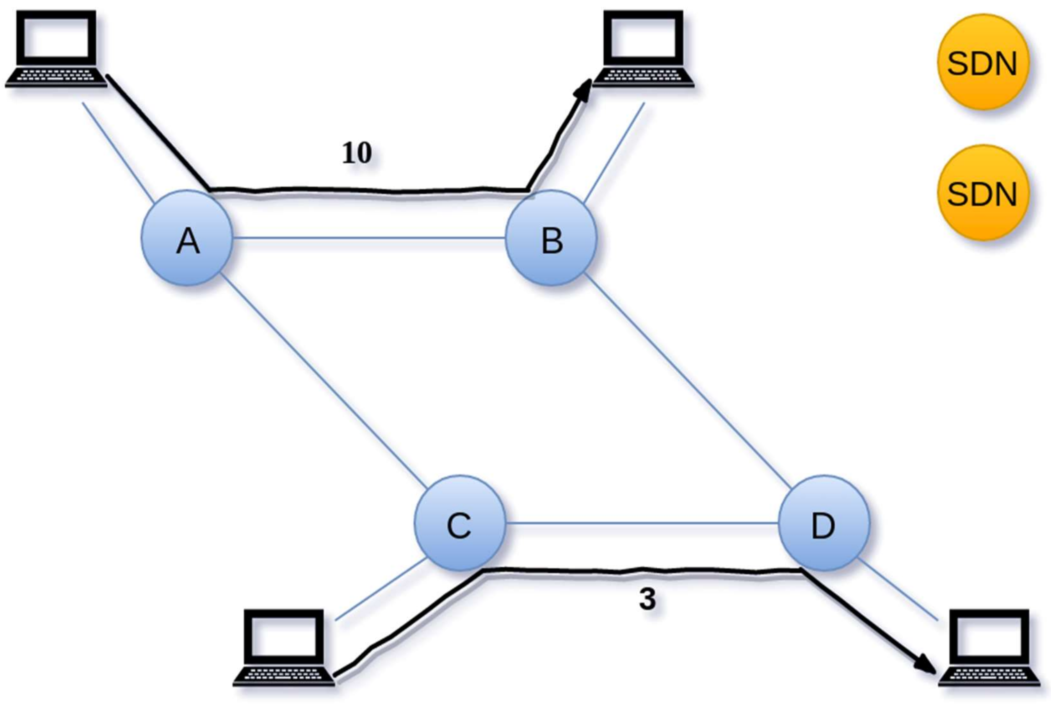

Figure 2. Flow routes are acquainted

Let us consider convenient representation of flow distribution in the form of a matrix for the same example.

4.2 The brute force algorithm

Let us have an arbitrary flow i, assume $Path_i = (v_{i_1}, ..., v_{i_n})$ is the flow's route and $D = \{Path_1, ..., Path_l\}$ is the set of all flows' routes in the network. By $N_0^{s,n}$ we denote the set of matrices with s rows, n columns and elements belonging to $N_0 = \{0,1,2,3,...\}$. Let us present the pair (V, D) as the matrix $M \in N_0^{|V|,|D|}$, for which the value M_{ij} equals to the route weight if and only if $v_i \in Path_j$ and to 0 in other case. Let us say that i-row of the matrix M covers its j-column if and only if $M_{ij} > 0$, i.e. $v_i \in Path_j$.

In this way, for the given example the following matrix is obtained:

	$Path_1$	$Path_2$
A	10	0
B	10	0
C	0	3
D	0	3

The rows corresponding to the vertices A and C cover all the columns in this table.

In order to calculate the number of flows that pass through a subset of vertices $X = \{v_{i_1}, ..., v_{i_n}\}, X \subseteq V$, one could use the following algorithm consisting of n steps, where n is the power of set X:

Step 1. In the matrix M, we select the row i_1, which corresponds to the vertex v_{i_1} from X. The sum of all values in this row is added to the result. We eliminate the row i_1 from the matrix M along with all columns, which have such intersections with the row i_1 that contain the values other than 0; thus, we get the matrix M_1.

. . .

Step s. In the matrix M_{s-1} we select the row i_s, which corresponds to the vertex v_{i_s} from X. The sum of all values in this row is added to the result. We eliminate the row i_s from the matrix M_s along with all columns, which have such intersections with the row i_s that contain values other than 0; thus, we get the matrix M_s.

Let us assume that the route of each flow and the weight of each route are known. Then, using the brute force algorithm, the number of possible solutions is estimated as

$$C_n^k = \frac{n!}{k!(n-k)!}, \qquad (1)$$

where k is the number of routers for replacement and n is the number of routers in the network. For every potential solution from the space one should count the number of flows passed through the routers that should be replaced and chose the maximum number. It is clear that the brute force algorithm generally does not work in a polynomial time. Thus, we have shown that this problem cannot be solved in a polynomial time in principle, unless $P \neq NP$.

4.3 Prove of NP-complexity of the problem

We denote by $\mathrm{B}^{s,n}$ the set of matrices with s rows, n columns and elements belonging to $\mathrm{B} = \{0,1\}$. In order to justify the properties of the problem being under study we first formulate a number of supporting questions.

Question 1. What is the maximum number of flows that can be controlled by k SDN switches?
Question 2. Will k SDN switches be enough to control all the specified flows in a given network?
Question 3. Will k rows in the matrix $M \in \mathrm{B}^{p,s}$ be enough to cover all the columns in this matrix?
Question 4. Will k columns in the matrix $M \in \mathrm{B}^{p,s}$ be enough to cover all the rows in this matrix?
Question 5. For a given finite set $\mathrm{N} = \{\alpha_1,...,\alpha_s\}$, family of its subsets $\mathrm{S} = \{\mathrm{N}_1,...,\mathrm{N}_p\}$ and for the integer number k the question is whether there is a family $\mathrm{C} \subseteq \mathrm{S}$ with power k such that $\bigcup_{\mathrm{N}_i \in \mathrm{C}} \mathrm{N}_i = \mathrm{N}$.

Note that the fifth question refers to solvability of a well-known set cover problem, which is proved to be NP-hard, and the relevant decision problem is NP-complete [14]. Next, using a series of sequential steps of convergences from the fifth to the first question, it is easy to show [15] that the question 5 problem is polynomially reduced to our problem.

4.4 Gradient algorithm

Keeping in mind that finding the exact solution is a time consuming problem, let us use the greedy algorithm, which implies selecting vertices with maximum weight at each step.

Algorithm:

1. Recalculate the weight of each vertex.
2. Replace the router corresponding to the vertex with maximum weight. Remove from the set of flows such flows that pass through the vertex.
3. Assign $s := s-1$. Go to step 1 until $s \neq 0$.

For the correctness of the proposed algorithm usage, we will not consider those vertices, which do not transmit any flow. Let us present the pair (V, D) as the matrix $M \in \mathrm{B}^{|V|,|D|}$, for which $M_{ij} = 1$ if and only if $v_i \in Path_j$. In such representation at each step of the gradient algorithm, we select the row, which covers the largest number of uncovered columns and include this row into the cover of the matrix.

The following theorem gives an upper bound for the gradient algorithm of the number of steps needed to cover all the columns.

Theorem 1 ([16]). Assume for the real value γ, 0<γ≤1, $M \in \mathrm{B}^{p,s}$ is not less than γp digits in each column of the matrix M. In this case, coverage length of the matrix M, obtained by gradient algorithm does not exceed

$$\left\lceil \frac{1}{\gamma} \ln^+(\gamma s) \right\rceil + \frac{1}{\gamma}, \qquad (2)$$

$$ln^+ x = \begin{cases} lnx, & x \geq 1 \\ 0, & 0 < x < 1 \end{cases} \qquad (3)$$

where .

For the considered problem $M \in \mathrm{B}^{|V|,|D|}$ this theorem gives us an upper bound for the number of routers that should be replaced by SDN switches to take control over all the flows, where the number of hops in the route is above $\gamma \cdot |V|$.

5. CASE 2 OF THE PROBLEM

5.1 Partitioning algorithm

The main idea of this approach, proposed in [17], implies the fact that SDN switches are placed in the network such a way to split OSPF domain into two or more sub-domains. Then all the flows, which are local in the subdomains, remain intact. We take into account only those flows, which pass through several sub-domains. Such flows pass at list through one SDN switch.

In a graph theory, this problem is known as a Vertex separator problem. Based on [18], we developed the network separation model, which is proposed in terms of Boolean linear programming problem (BLP). In this model, we used only the weight of each vertex (recall that it can indicate both the number of flows passing through this vertex or packet size of them). It is worth to remark that one can build a model that does not depend even on these data, being based only on the network topology. In this form of the model, the algorithm will provide graph partitioning into sub-graphs of comparable size.

5.2 Statement of the problem in terms of graph theory

Suppose we are given a connected undirected graph $G=(V,E)$:

- $V=\{v_1,...,v_n\}$ — set of its vertices; |V| = n
- $E \subset V \times V$ — set of undirected edges $\{v_i, v_j\}$
- c_j — weight of j-vertex
- $s \le n$ — restriction to the number of substitutable vertices

Problem

Find partition of the set V into 3 disjoint sets $\{A,S,B\}$, such that:

1. For $\forall v_i \in A$ and $\forall v_j \in B$ the edge $\{v_i, v_j\}$ does not belong to E.
2. $|S| \le s$ — restriction to the cut length;
3. $\sum_{v_j \in S} c_j$ is maximum.

Condition 3 is determined by the fact that we want to maximize the weight of the flows passing through the SDN switches.

In [15] one can find appropriate ILP model solving this problem.

6. CASE 3 OF THE PROBLEM

6.1 Cycle removal algorithm

Definition 4. Cycle basis is a minimum set of cycles that allows depicting any Eulerian subgraph as a symmetric difference of basis cycles.

Definition 5. Circuit rank [6] of an undirected graph is the minimum number of edges that must be removed in order to provide acyclic transformation of this graph.

The idea of this algorithm, proposed in [19], is based on the hypothesis that vertices that belong to many cycles correspond to the routers, which transmit a lot of traffic.

Because each graph $G=(V,E)$ has at least one basis cycle, any cycle in G can be expressed as a linear combination of the fundamental cycles that make up this basis.

The strategy of this algorithm is based on the assertion that determination of the set of fundamental cycles (cycles that make up the basis of the cyclic graph space) can be provided efficiently.

In order to build a basis of cycles, one should allocate a certain spanning tree. A cycle can be formed by adding any edge that is not included into the spanning tree. The family of these cycles form the basis of a cyclic space of the graph. After constructing the cycle basis, vertices, with the largest number of basis cycles, are replaced by the next algorithm.

Algorithm ([19]):

1. Build a cycle basis B.
2. Replace the vertex v with the largest number of basis cycles.
3. Exclude from the consideration this vertex (replace it with an SDN switch).
4. Remove from B all the cycles passing through v.
5. Finish, if the set of SDN switches is empty.
6. Provided any cycles left in B return to step 2.
7. Provided any cycles left in the graph, return to step 1.

6.2 The problem of cycle removal in terms of integer programming

There is an ambiguity in previously mentioned algorithm in construction of the cycle basis: spanning trees can be constructed in different ways (the number of spanning trees in a complete graph on n vertices is n^{n-2}). Therefore, the choice of vertices for replacement is ambiguous.

The problem related to the minimum number of vertices that require replacement in order to remove all the cycles in the graph is known in the graph theory as the Feedback vertex set problem, which is NP-complete [20]. In order to find optimal solution of this problem, we build the corresponding ILP model.

For each vertex $v \in G$ we denote the variable b_v, being equal to 1 if the vertex should be replaced and 0 in other case. In order to make the graph acyclic, these variables should comply with the following constraint:

$$\text{For } \forall \text{ cycle } C \subseteq G \quad \sum_{v \in C} b_v \geq 1$$

Therefore, in any cycle belonging to G at least one vertex should be replaced in order to provide "decomposition" of that cycle, while minimizing the number of vertices for the replacement.

These arguments imply the following ILP problem:

$$\text{Minimize: } \sum_{v \in G} b_v$$

$$\text{So that: For } \forall \text{ cycle } C \subseteq G, \sum_{v \in C} b_v \geq 1$$

CONCLUSION

In this paper different methods for the transition of OSPF networks into SDN were considered. Those methods differ by strategies and input data (namely, the statistics collected in a network).

For the problem, where a full path of each flow is known:

- The NP-complexity of the problem was demonstrated.
- Relevant greedy algorithm was proposed.
- The upper bound of the required number of SDN switches to control all flows was provided.

Thus, the authors of this study considered the theoretical transition from the traditional OSPF/IS-IS network to the SDN network architecture. This article can serve as a theoretical source for further development and optimization of SDN network. The proposed methods make the transition to SDN possible without significant costs. **Partitioning algorithm** is a more simplified transition option; however, **cycle removal algorithm** allows avoiding system errors after checking each cycle. The transition to SDN will have a significant impact on network infrastructure management in DPC, as well as on enterprise network management and WAN-network management.

Thus, SDN enable programming the network as a whole, which, in turn, allows system administrators to not bother with separate devices. All tasks are performed by the controller, which will give network devices instructions in regards to traffic processing, which will make it so the devices themselves will no longer have to do this work, which, in turn, will improve their stability and performance.

In addition, maximum automation of the network operation will cut material and financial costs of infrastructure maintenance, allowing specialists to focus on developing new services.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

This research is supported by the Ministry of Education and Science of the Russian Federation, Unique ID RFMEFI60914X0003.

REFERENCES:

[1] D. Levin, M. Canini, S. Schmid, and A. Feldmann, "Incremental SDN Deployment in Enterprise Networks", *ACM SIGCOMM Comput. Commun. Rev.*, Vol. 43, No. 4, pp. 473–474, 2013.

[2] T. Das, M. Caria, A. Jukan, and M. Hoffmann, "Insights on SDN migration trajectory", in *2015 IEEE International Conference on Communications (ICC)*, 2015, pp. 5348–5353.

[3] R. L. Smelyanskiy, "Problems of modern computer networks.Proceedings of the XIX All-Russia Scientific Conference on Telematics.", 2012 [Russian].

[4] R. L. Smelyanskiy, "Software defined networks", Open systems, 2013 [Russian].

[5] Hofstede, Rick; Celeda, Pavel; Trammell, Brian; Drago, Idilio; Sadre, Ramin; Sperotto, Anna; Pras, Aiko. "Flow Monitoring Explained: From Packet Capture to Data Analysis with NetFlow and IPFIX". IEEE Communications Surveys & Tutorials. IEEE Communications Society. Vol. 16, No. 4, pp. 28.

[6] Gregory Berkolaiko; Peter Kuchment (2013). Introduction to Quantum Graphs. American Mathematical Soc. p. 4.

[7] D. Levin, M. Canini, S. Schmid, F. Schaffert, and A. Feldmann, "Panopticon: Reaping the Benefits of Incremental SDN Deployment in Enterprise Networks", 2013.

[8] M. Yang, Y. Li, D. Jin, L. Zeng, X. Wu, and A. V. Vasilakos, "Software-Defined and Virtualized Future Mobile and Wireless Networks: A Survey", *Mob. Networks Appl.*, Vol. 20, No. 1, pp. 4–18, Feb. 2015.

[9] G. Michael and D. Johnson, "Computers and Intractability", *New York: wh freeman*, 2002.

[10] S. V. Yablonskiy and O. B. Lupanov, "Discrete Mathematics and Mathematical Problems of Cybernetics", *M. Nauk.*, Vol. 1, 1974 [Russian].

[11] N. M. Novikova, "Fundamentals of optimization. Lecture course at Moscow State University", 1999 [Russian].

[12] W. Han, H. Hu, Z. Zhao, A. Doupé, G.-J. Ahn, K.-C. Wang, and J. Deng, "State-aware Network Access Management for Software-Defined Networks", in *Proceedings of the 21st ACM on Symposium on Access Control Models and Technologies - SACMAT '16*, 2016, pp. 1–11.

[13] S. Sezer, S. Scott-Hayward, P. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao, "Are we ready for SDN? Implementation challenges for software-defined networks", *IEEE Commun. Mag.*, Vol. 51, No. 7, pp. 36–43, Jul. 2013.

[14] B. Korte and J. Vygen, "Combinatorial optimization", *Springer.*, 2002.

[15] R. A. Yanbulatov, "A study of different ways of evolution from a traditional network to software-defined", Course paper. M.V. Lomonosov Moscow State University, 2016 [PDF, Russian] (https://cloud.lvk.cs.msu.su/index.php/s/xV8isRmPNtHcFP1).

[16] S. A. Lozhkyn, "Fundamentals of Cybernetics.", 2003 [Russian].

[17] M. Caria, T. Das, A. Jukan, and M. Hoffmann, "Divide and Conquer: Partitioning OSPF networks with SDN", pp. 467–474, Oct. 2015.

[18] E. Balas and C. C. de Souza, "The vertex separator problem: a polyhedral investigation", *Math. Program.*, Vol. 103, No. 3, pp. 583–608, Jul. 2005.

[19] S. Markovitch and M. Schmid Ben, "SHEAR: A Highly Available and Flexible Network Architecture Marrying Distributed and Logically Centralized Control Planes"

[20] G. Even, J. Naor, B. Schieber, and M. Sudan, "Approximating Minimum Feedback Sets and Multi-Cuts in Directed Graphs", *Algorithmica.*, Vol. 20, No. 2, pp. 151–174, 1998.

[21] Ali-Ahmad H. et al. An SDN-based network architecture for extremely dense wireless networks. Future Networks and Services (SDN4FNS), 2013 IEEE SDN for. IEEE, 2013. pp. 1-7.

[22] Zhou R. et al. Study on authentication protocol of SDN trusted domain. Autonomous Decentralized Systems (ISADS), 2015 IEEE Twelfth International Symposium on. IEEE, 2015. pp. 281-284.

[23] López-Rodríguez F., Campelo D. R. A robust SDN network architecture for service providers. Global Communications Conference (GLOBECOM), 2014 IEEE. IEEE, 2014. pp. 1903-1908.

[24] Lorenz C. et al. An SDN/NFV-Enabled Enterprise Network Architecture Offering Fine-Grained Security Policy Enforcement. IEEE Communications Magazine. 2017. Vol. 55. No. 3. pp. 217-223.

[25] Basta A. et al. A virtual SDN-enabled LTE EPC architecture: A case study for S-/P-gateways functions. Future Networks and Services (SDN4FNS), 2013 IEEE SDN for. – IEEE, 2013. – pp. 1-7.

[26] Martins J. S. B., Campos M. B. A security architecture proposal for detection and response to threats in SDN networks. ANDESCON, 2016 IEEE. IEEE, 2016. pp. 1-4.

[27] Haleplidis E. et al. Software-defined networking (SDN): Layers and architecture terminology. 2015. No. RFC 7426.

[28] Schlinker B. et al. Try Before you Buy: SDN Emulation with (Real) Interdomain Routing. ONS. 2014.

[29] Scott-Hayward S., O'Callaghan G., Sezer S. SDN security: A survey. Future Networks and Services (SDN4FNS), 2013 IEEE SDN For. IEEE, 2013. pp. 1-7.

[30] Adami D. et al. Towards an SDN network control application for differentiated traffic routing. Communications (ICC), 2015 IEEE International Conference on. IEEE, 2015. pp. 5827-5832.

[31] Dai W. et al. Extending SDN network with recursive architecture. Wireless Personal Multimedia Communications (WPMC), 2014 International Symposium on. IEEE, 2014. pp. 491-496.

[32] Hu Z. et al. A comprehensive security architecture for SDN. Intelligence in Next Generation Networks (ICIN), 2015 18th International Conference on. IEEE, 2015. pp. 30-37.
[33] Sezer S. et al. Are we ready for SDN? Implementation challenges for software-defined networks. IEEE Communications Magazine. 2013. Vol. 51. No. 7. pp. 36-43.
[34] Zhang G. et al. Research on communication network architecture of energy internet based on SDN. Advanced Research and Technology in Industry Applications (WARTIA), 2014 IEEE Workshop on. IEEE, 2014. pp. 316-319.
[35] Guo Y. et al. Traffic engineering in SDN/OSPF hybrid network. Network Protocols (ICNP), 2014 IEEE 22nd International Conference on. IEEE, 2014. pp. 563-568.
[36] Song H. Protocol-oblivious forwarding: Unleash the power of SDN through a future-proof forwarding plane. Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking. ACM, 2013. pp. 127-132.