Controller Placement and Flow based Dynamic Management Problem towards SDN

Long Yao*, Peilin Hong*, Wen Zhang*, Jianfei Li†, Dan Ni*
* The Information Network Lab of EEIS Department USTC, Hefei, China, 230027
yaolong@mail.ustc.edu.cn, plhong@ustc.edu.cn, zw1991@mail.ustc.edu.cn, nidan@mail.ustc.edu.cn
† Huawei Technology Corporation Limited
fre.lijianfei@huawei.com

***Abstract*—The SDN architecture decouples the control plane and data plane, and multiple controllers are adopted to solve the scalability and reliability problem in SDN. However, most of the researches are focused on the control architecture and ignore the controller placement problem. Moreover, the mapping between the switches and controllers are static in current proposals, which will lead to the load unbalance of controllers under dynamic flow variations. In this paper, we define a new controller placement metric considering the node weight for a single domain at first, and then propose a dynamic switch migration algorithm to adapt to the flow dynamics and realize controller load balance in multiple SDN domains. Finally, a simple simulation platform is built to verify the proposed scheme.**

I. Introduction

Software Defined Networking(SDN) is a new emerging technology that decouples the control plane from the data plane. It relies on a centralized controller that runs on control plane to manage the network. On one hand, the control plane communicates with the forwarding layer to collect the information and maintains the network topology; On the other hand, it interacts with the application layer to implement various network functions. Despite the technical benefits SDN brings, especially in the field of traffic engineering [1]. However, the separation of control plane and forwarding plane also brings some new problems, such as the controller placement problem [2] and network management problem under flow dynamics [3] .

The controller placement problem focuses on seeking the best controller position to satisfy the optimal target for a give SDN topology. When there is a single domain, a single controller is needed. While for large scale networks, multiple domain is created and multiple controllers are necessary, thus how to deploy multiple controllers is a research point in SDN. For the controller placement problem, it can be regarded as the task in network designing period which relies on the physical network topology to get the best controller position without considering the network dynamics. In current researches, most of the researchers take the propagation latency between switches and controller into consideration, and ignore the weight of switches which is a critical factor in real networks. From this point, we propose a new controller placement metric considering the switch weight and the delay from switches to controller together. The objective of the metric is to minimize the total cost of flow set-up request from switches to controller.

Another problem is the dynamic network management problem, which mainly deals with the flow variations in multiple control domains for large scale networks. The centralized management of SDN confronts the problems of scalability and reliability [4]. Therefore, distributed control architecture is proposed to solve the problem, such as Onix [5] , Hyperflow [6], Devoflow [7], and so on. However, these proposals have the limitation that the mapping between the switches and controllers are static, making it difficult for controllers to adapt to the flow variations. When the switches observe a large number of busty flows, the controller that manages the switches may become overload. Thus, the static mapping may not able to deal with the dynamic flows, and it can easily cause load unbalance of controllers. To address the problem, we propose a flow based dynamic switch migration algorithm to realize the load balance among controllers. When controller becomes overload, it cooperates with the neighbors to migrate the boundary switches to the lightly-load neighbor controllers.

The controller placement problem is a pre-planning problem of SDN and the flow based dynamic switch migration scheme is the complement to realize the optimal network management in SDN. The both constitute an integrated scheme towards large scale SDN network in together. The contributions of this article are summarized as follows:

- A new controller placement metric that considers the switches weight are defined.
- A flow based dynamic network management scheme to realize the load balance of controllers are presented and also the simulation are built to verify the proposal.

The rest of this paper is structured as follows. Section II gives a brief introduction to the related works. In section III, we mainly address the controller placement problem and the dynamic flow management strategy. And Section IV focuses on the evaluation. Finally a summary concludes this paper with an outlook of future works in section VI.

II. Related Works

The controller placement problem was first proposed in [2], where the authors considered the best controller positions from the view of minimized average latency and minimized maximum latency from switches to controllers. However, the author didn't consider the weight of nodes and regard them as the same. In [8] [9] [10] [11], the authors defined the optimal

978-1-4673-6305-1/15/$31.00 ©2015 IEEE

controller placement problem from the network resilience, both the control plane and data plane. We do approval the view of resilience, however, the disruption possibility in network is rare and we may not take this into consideration in the pre-planning of network. In [12], the authors proposed a zone-based distributed network optimization and gave several heuristic solutions. The authors in [13] considered the controller capacity in the placement problem. And in [14] the authors gave a complete modulation for the controller placement problem from the view of deployment price. However, the proposals mentioned before are all static deployment without considering the flow dynamics in real network.

Recently, Bari in [15] provided a dynamic controller provision strategy based on the flow dynamics. They formulated the optimal controller provisioning problem as an integer linear problem and proposed two heuristic algorithms. And in [3], which was most related with our works, the author proposed the detailed migration mechanism when dynamically assigning switches to controller. However, the author did not give any algorithms to determine the reassigned switches. In recent researches, the authors gave solution from game theory [16] [17] maybe a good solution towards the dynamic management in SDN.

III. Controller Placement and Flow based Dynamic Management Problem

A. Overview

As previously discussed, the topic we discuss mainly deals with the following two parts. The first defines optimal placement for controller instances, of which we will focus on the single domain and multiple domains. While the second supplement the static deployment with switch migration under the flow dynamics.

B. Controller Placement Strategy towards SDN

For the controller placement problem, it depends on the network topology. For a small network, only a single controller is needed. While for large-scale networks, multiple controller are desired to deal with network scales.

1) Controller Placement for a single domain: In a single domain, we suppose that the network is managed by a centralized controller. For any given network topology, it is constituted by nodes and links. Therefore, these two issues should be considered simultaneously in the optimal controller placement problem. However, the previous works in controller placement problem only considered the latency from switch to controller [2][8][9][10]. This means that all nodes are considered to be equal in SDN networks, and this is obviously unreasonable. In our controller placement strategy, we take both of these two issues into consideration and make the following two assumptions:

- In real deployment, the controller and the switches are co-located, that is, the controller and switches are deployed at the same position.
- The connection between controller and switches adopts the in-band paradigm, which means that the control messages and data messages are transmitted via the same channel.

We believe that the weight of switch nodes are different from each other in most of the network topologies, and some nodes will be the hot spots, as the work done in [18]. From one side, the weight of switch node can represent its importance in the network; On the other side, it reflects the maximum throughput it can achieve. Therefore, in the determine metric for controller placement problem, both the delay from switches to controller and weight of switch nodes should be considered simultaneously.

In SDN, the main function of controller is dealing with the new flow set-up requests. When a packet arrives at the switch, it will match against the flow entries. In the situation of a matched flow entry, the corresponding action is taken. Whereas if there is no matching entries in switch, the Packet-in message will be send to controller, then the controller will determine the rule to handle the packets. In the controller placement strategy, we mainly focus on the cost of flow set-up requests (Packet-in messages) from switches to controller. It consists of the following two issues:

- The number of Packet-in message.
- The routing cost from the switches to controller.

In our controller placement strategy, we consider these two issues mentioned before together and propose a new metric for the controller placement problem. The physical meaning of the metric is the total cost of packet-in events generated from switches to controller.

We abstract the SDN network topology as a graph $G(V, E)$. The former issue can be reflected by the weight of switch node. Since the weight of switch node represents the node importance and the flow generated by the node, we adopt the degree d_v of switch node to measure its weight in this paper. That is, the weight $w(v)$ is the function of node degree: $w(v) = f(d_v)$. We use the normalized node degree to determine the node weight in this paper,

$$w(v) = 1 + \frac{d_v}{d_{\max}} \tag{1}$$

where $d_{\max} = \max_{v \in V} d_v$. The later issue can be reflected by the routing costs from the switch node to controller. This is a routing problem, the cost of shortest path can be obtained with routing algorithm such as Dijkstra's algorithm [19].

The optimal objection of controller placement problem is to minimize the total cost of flow set-up requests generated from switches to controller. The metric is defined as the multiply of flow requests number and the delay from each switch to the controller, as the following equation shows:

$$f(c) = \sum_{v \in V} [w(v) \cdot f(v, c)] \tag{2}$$

where $w(v)$ is the weight of switch node v, $f(v, c)$ is the routing cost from switch v to controller c, and $f(c)$ means the cost of deploying controller at position c.

Meanwhile, we consider that not all nodes are the candidate deploying nodes of controller. We define the candidate nodes set $C \subseteq V$, and the principle of set C is the node whose degree is larger than the predefined constant value W. The objection is to find an optimal position c from set C that minimize the total cost $f(c)$.

Therefore, the optimal controller placement can be defined as the following optimization equation:

$$c = \underset{c \in C}{\arg} \quad \min f(c) \tag{3}$$

If there are several nodes whose costs are the same, one of them is chosen in random.

The algorithm procedure to calculate the best controller position is shown at *Algorithm* 1. Given the network topology $G(V, E)$, the distance matrix $\tilde{D}$ and the threshold W of node degree, the algorithm returns the optimal controller position c with minimum flow set-up cost.

Algorithm 1 Controller Placement in a single domain

Input: Topology: $G(V, E)$;
Distance Matrix: $\tilde{D}$;
Node degree Threshold W;
Output: Best controller position c;

for each $v \in V$ **do**
  Get the degree d_v of each node;
  if $d_v > threshold$ **then**
    Add node v to candidate controller position C;
  end if
end for
Calculate the maximum node degree: $d_{\max} = \max\limits_{v \in V} d_v$
for each $c \in C$ **do**
  Get $f(v, c)$ with routing algorithm for each $v \in V$;
end for
Determining the optimal controller position with Eq. (3).
return c;

2) Multiple Controller Placement problem: A centralized controller of SDN confronts the scalability and reliability problem. Therefore, researchers have proposed distributed multiple controllers to realize logical centralized and physical distributed control towards the SDN network. In the multiple controller placement problem, we propose a near-optimum method which refers to the idea of diver-and-conquer. Firstly, the network topology is divided into several sub-domains with topology partition algorithm such as the Multilevel k-way partition[20]. Afterwards, it is demanded to deploy a single controller in each domain, which is the work we have done before. Finally, the controller maintains the mapping relationship with its corresponding switches. The only problem to be solved is how to evaluate the number of partitioned sub-graph. For this problem, we can estimate k with the following equation:

$$\lceil k \rceil = \frac{\sum_{i=1}^{M} l_i}{C} \tag{4}$$

where C is the capacity of the controller(The capacity of controller can be estimated with the ability to deal with the flow requests), and l_i is the requests of each switch.

C. Flow base dynamic scheduling strategy

The controller placement problem is the work during network initialization, thus the best controller position can only be derived with the parameters of physical network, and it will not consider the dynamic flow variations in real network. Once the positions of controllers are chosen, they are constant and static. Each controller manages the switches in its own domain, and the switches send the "packet-in" messages to its master controller. Meanwhile, the controller will communication with other controllers periodically.

However, real networks may have significant variations in both temporal and spatial traffic characteristics, such as the burst flow in the shopping mall or convention hall. Therefore, the static control towards network management will confront some problems. For example, when some nodes have busty flows, they will generate a mass of flow set-up requests to the controller, causing the overload of controller.

The flow base dynamic scheduling strategies can be summarized as the following three categories according to the various flow variations:

1) There observe a burst flow variation in a control domain, leading to the overload the controller while the boundary domains remain the same load as before. In this scenario, we consider a dynamic switch migration scheme to release the overload problem and realize the load balancing of controllers.
2) There observe a burst flow variation in several domains and the switch migration is not available. Therefore, the solution towards this problem is the complement controller provision.
3) There observe a decreasing flow variation in several domains and we also consider migrating all the switches in a domain, and close the controller to realize energy conservation.

In this section, we focus on the first scenario and propose a dynamic controller management strategy based on the flow dynamics. When one or more controllers become overload, they can cooperate with their neighbor controllers, and migrate the boundary switches to the neighbor domain to reduce the load of overload controller. With this method, we can realize the dynamic management towards the network and the load balancing of controllers.

In order to specify the dynamic scheduling strategy, we introduce the following notations:

- K : The number of controllers
- $U =< \mu_1, \mu_2, \ldots, \mu_K >$: The capacity of controllers
- α : Threshold trigger facto of controller
- $d(i, j)$: The routing cost from switch i to controller j
- λ_i : The demands of switch i
- L_j : The loads of controller j

The key idea of our proposal is that, when the controller load goes beyond its capacity(related with the facto α), it will adjust the boundary switches with its neighbor controller to relieve its load and realize the load balancing of controllers. That is, the switch migration scheme is triggered. We propose a switch migration algorithm that minimize the flow set-up request cost of the migrating switches under the capacity constraint of neighbor controllers. This equals the reassignment of boundary switches and its new controller. There are two metrics related to the algorithm.

1) The flow set-up request cost P_i for switch i. It reflects the service level each switch could get from the controller. The lower the cost, the higher the service level, and vice versa. The cost P_i can be represented by the multiply of flow numbers generated at switch i and the delay from switch to controller j:

$$P_i = \lambda_i \cdot d(i,j) \tag{5}$$

2) The load L_j at the controller j. We use the average arrival number of flow set-up requests to denote the controller load, and it can be signified with the following equation

$$L_j = \sum_{i \in j's \ Control \ domain} \lambda_i \tag{6}$$

Then, we focus on the algorithm design of switch migration(the messages exchanged during migration can be seen from [3]). The migration algorithm procedure is shown at Fig. 1.

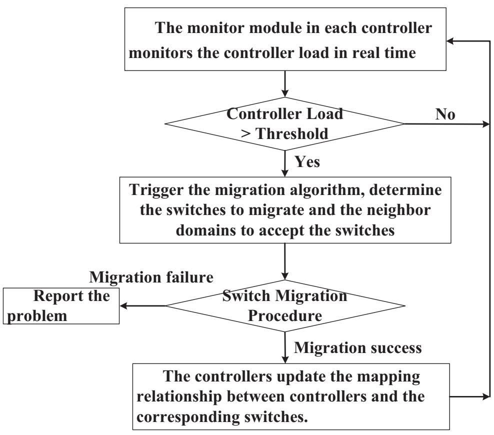

Fig. 1. Switch migration algorithm procedure

The monitor module at each controller monitors the controller load in real time. In the case of current controller load exceeds its capacity($L_j > \alpha\mu_j$), the switch migration algorithm is triggered. And the algorithm adjusts the network iteratively via determining the boundary switch to migrate and the neighbor domain to accept the switch. The algorithm contains the following three steps:

1) Determine the overload control domain.
The control domain that exits the controller capacity becomes the overload domain.
2) Choose the boundary switch to migrate.
Let the boundary switches of the overload domain be the candidate migrating boundary switch set I. From the view of whole network, the algorithm sorts the set I in descending order with their flow arrival rate λ and chooses the switch i^M with largest rate λ, because we want to release the controller load as fast as possible.

$$i^M = \arg_{i \in I} \ \max\{\lambda_i\} \tag{7}$$

3) Determine the target domain.
Since each boundary switch will connect to several neighbor domains, and it will also connect to several switches for each neighbor domain. In order to maintain the minimized flow set-up cost, the migrating boundary switch will choose the target domain j^D with smallest cost P_i, meanwhile, the target domain should have sufficient capacity to serve these flows. The process can be expressed as a optimization problem as follows:

$$\begin{aligned} j^D = & \arg_{j \in i's \, neighbor} \ \min\{P_i\} \\ s.t. \ & L_{j^D} + \lambda_{i^M} \leq \alpha \cdot \mu_{j^D} \end{aligned} \tag{8}$$

The switch migration algorithm in overload controller is shown at *Algorithm* 2.

Algorithm 2 Switch Migration Algorithm

Input: Overload controller : j;
$j's$ boundary switches set: I;
$j's$ neighbor domain: N ;
Output: New mapping between controllers and switches

while $L_j > \alpha\mu_j$ & $I \neq \emptyset$ **do**
  Determine the migrating switch i^M with Eq. 7;
  Determine the target domain j^D with Eq. 8;
  Updating the information: $I = I \backslash i^M$;
  if $j^D = \emptyset$ **then**
    L_j remains unchanged;
  else
    $L_j = L_j - \lambda_{i^M}$
  end if
end while
if $L_j > \alpha\mu_j$ **then**
  The network is overload, report the problem.
else
  Return new mapping between controllers and switches.
end if

Actually, the switch migration algorithm can also be adopt in the scene when the controller load is such small that we can migrate the switches it manage to other domains and close the light-controller.

IV. EVALUATION

In this section, we do the evaluation for both the controller placement metric and the flow base dynamic scheduling strategy we proposed.

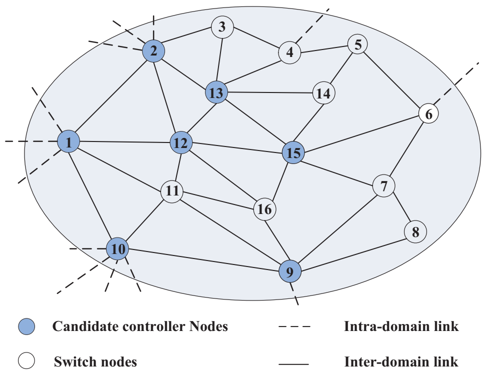

Fig. 2. An instance of Controller Placement Problem

A. Controller Placement Illustration in a single SDN Control domain

To understand the controller placement strategy more intuitively, we give an instance of controller placement problem in a real network. Fig. 2 shows a logical SDN network view. The topology is a single domain network including 16 nodes, some of which are boundary nodes connecting other domains. We will determine the best controller position for this domain.

The node degree can be get directly from the topology, and the maximum degree is $d_{\max} = \max\limits_{v \in V} d_v = 7$ with nodes number of 1,2 and 10. And the normalized degree can be get with Eq. (1). Then we add the node whose degree is equal or greater than 6 (here we assume that the threshold $W = 6$) to the candidate controller nodes set C, in this case $C = 1, 2, 9, 10, 12, 13, 15$.

To give an intuitionist view, the hop count is adopted here to denote the routing cost. For example, the cost from switch 1 to 16 is 2 hops with the shortest path 1-12-16. With *Algorithm* 1, the cost of each candidate node is calculated and the node 12 is chosen as the controller position because the cost we defined at Eq. (2) is minimal. This result also satisfies with our intuition. The nodes in the left have higher weight, which means they will generate more flow set-up requests, therefore the controller should be placed closer to these nodes. However, if we adopt the rule defined at [2], that is the average-case latency without considering the node weights, node 15 is chosen as the best controller node. To compare these two methods, we calculate the average flow set-up cost, and the result [1] is show at Table I.

TABLE I
THE AVERAGE COST AT EACH CANDIDATE NODE

Measurement Model	Controller Node	Average flow set-up cost
Proposed method	node 12	1.4545
Average case	node 15	1.5844

It is obviously from the table that our proposed controller placement strategy has smaller average flow set-up cost compared with the strategy at [2].

[1]Here, the average flow set-up cost is calculated by the division of the total flow set-up cost and the total flows. In our proposed method, we adopt the sum of node degree to denote the total number of generated flows.

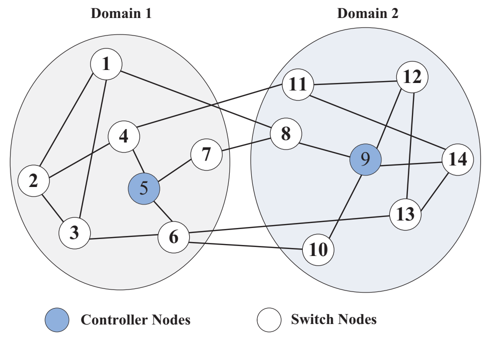

Fig. 3. NSFNET topology

B. Simulation of Flow based Dynamic Management Problem

We evaluate the performance of our proposed migration algorithm via a simple network topology. The system simulation platform is built up to simulate the migration behavior. The topology we adopted is the NSFNET shown at Figure 3, it is managed by two controllers, each controls 7 nodes.

The controller monitors its load in real-time. the controller capacity is assumed to be $\mu_1 = \mu_2 = 1000$, We suppose the overload coefficient is $\alpha = 0.9$.

For a moment, the controller in domain 2 becomes overload due to flow burst. Then the switch migration mechanism is triggered. The controller in domain 2 cooperates with its neighbor controller and determines the switches to migrate. Eventually node 10 is migrated from domain 2 to domain 1. However, we should notice that, after the migration, node 10 is more far away from the new controller compared with the previous controller.

For convenience, we draw the bar graph of controller load before and after migration in Fig. 4. It is obviously from Fig. 4 that the switch migration algorithm realizes the controller load balancing (both of the controllers' load are under the threshold). Therefore the accuracy and applicability of our proposal are verified.

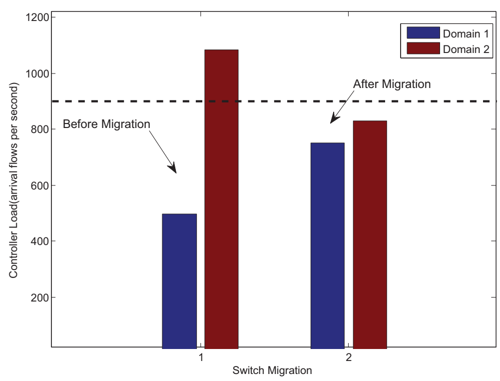

Fig. 4. Switch Migration effection

V. Conclusion and future work

In this paper, we propose a new metric for the controller placement problem in a single domain, the metric considers the weight of switch nodes and the delay from switches to controller simultaneously to minimized cost of flow set-up requests during network planning stage. Nevertheless, the controller placement problem does not consider the flow dynamics. Therefore, we also propose a dynamic switch migration algorithm based on flow variations. When the controller becomes overload due to flow bursts, the controller can cooperate with its neighbors to migrate the boundary switches to reduce the load, therefore, the load balance of controllers are achieved. The simulation results also verify our proposal. However, the topology we adopt is not large enough, and we do not consider the QoS loss of the migrated switches. Our next step is extending our strategy to large scale networks and research the multiple controller placement problem of SDN deeply.

Acknowledgment

This work is sponsored by Huawei Innovation Research Program.

References

[1] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, A. Vahdat, “B4: Experience with a globally-deployed software defined WAN”. in *Proc. ACM SIGCOMM’12*, Hong Kong, China, 12-16 Aug. 2013, pp. 3-14.

[2] B. Heller, R. Sherwood, and N. McKeown, “The controller placement problem,” in *Proc. ACM SIGCOMM HoTSDN’12*, Helsinki, Finland, 13-17 Aug. 2012, pp. 7-12.

[3] A. Dixit, F. Hao, S. Mukherjee, R. Kompella, “Towards an elastic distributed sdn controller”. in *Proc. ACM SIGCOMM HoTSDN’13*, Hong Kong, China, 12-16 Aug. 2013, pp. 7-12.

[4] S. H. Yeganeh, A. Tootoonchian, Y. Ganjali, “On scalability of software-defined networking”. *IEEE Commun. Magazine*, vol. 51, no. 2, pp. 136-141, 2013.

[5] T. Koponen, M. Casado, et. al. “Onix: A Distributed Control Platform for Large-scale Production Networks”. in *Proc. OSDI’10*, Vancouver, BC, Canada, 4-6 Oct. 2010, pp. 1-6.

[6] A. Tootoonchian, Y. Ganjali, “HyperFlow: A distributed control plane for OpenFlow”. in *Internet Network Management Conference on Research on Enterprise Networking*, USENIX Association, 2010, pp. 1-6.

[7] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, S. Banerjee, “Devoflow: scaling flow management for high-performance networks”. *SIGCOMM Comput. Commun. Rev.* Vol. 41, No. 4, pp. 254-265. Aug. 2011.

[8] Y. Zhang, N. Beheshti, M. Tatipamula, “ On resilience of split-architecture networks”. in *Proc. Globecom’11*, Houston, Texas, USA, 5-9 Dec. 2011, pp. 1-6.

[9] Y. Hu, W. Wang. X. Gong, X. Que, S. Cheng, “Reliability-aware controller placement for Software-Defined Networks”. in *Proc. IM’13*, Ghent, Belgium, 27-31 May, 2013, pp. 672-675.

[10] D. Hock, M. Hartmann, S. Gebert, M. Jarschel, T. Zinner, P. Tran-Gia, “Pareto-optimal resilient controller placement in SDN-based core networks”, in *Proc. ITC’13* Shanghai, China, 10-12 Sep. 2013, pp. 1-9.

[11] L. F.Mller, R. R. Oliveira, M. C. Luizelli, L. P. Gaspary, M. P. Barcellos, “Survivor: an Enhanced Controller Placement Strategy for Improving SDN Survivability”. in *Proc. GLOBECOM’14* Austin, USA, 8-12 Dec. 2014, pp. 1944-1950.

[12] X. Li, P. Djukic, H. Zhang,. “Zoning for hierarchical network optimization in software defined networks”. in *Proc. NOMS’14*, Krakow, Poland, 5-9 May. 2014, pp. 1-8.

[13] G. Yao, J. Bi, Y. Li, L. Guo, “On the Capacitated Controller Placement Problem in Software Defined Networks”, *IEEE Commun. Letters*, vol 18, no 8, pp. 1339-1342, Aug. 2014

[14] A. Sallahi, M. St-Hilaire, “Optimal Model for the Controller Placement Problem in Software Defined Networks”. *IEEE Commun. Letters*, vol 19, no 1, pp. 30-33, Jan. 2015

[15] M. F. Bari, A. R. Roy, S. R. Chowdhury, Q. Zhang, M. F. Zhani, R. Ahmed, “Dynamic Controller Provisioning in Software Defined Networks,” in *Proc. IEEE CNSM’13*, University of Zrich, Switzerland, 14-18 Oct. 2013, pp. 18-25.

[16] H. K. Rath, V. Revoori, S. M. Nadaf, A. Simha, “Optimal controller placement in Software Defined Networks (SDN) using a non-zero-sum game”, in *Proc. WoWMoM’14* Sydney, Australia, 16-19, Jun. 2014, pp. 1-6.

[17] G. Cheng, H. Chen, “Game model for switch migrations in software-defined network”. *Electronics Letters*, vol 50, no 23, pp. 1699-1700, Nov. 2014

[18] L. Qiu, V. N. Padmanabhan, G. M. Voelker, “On the placement of web server replicas”, in *Proc. Infocom’01*, Anchorage, Alaska, 22-26 Apr. 2001, pp. 1587-1596.

[19] E. W. Dijkstra, “A note on two problems in connexion with graphs”. Numerische Mathematik, 1959, pp. 269-271, 1959.

[20] A. Buluc, H. Meyerhenke, I. Safro, P. Sanders, C. Schulz, “Recent advances in graph partitioning”, in *Theor. Comput. Sci.*, vol. 1, pp. 237-267, 2013.