

The Energy-Aware Controller Placement Problem in Software Defined Networks

Ying Hu, Tao Luo, *Senior Member, IEEE*, Norman C. Beaulieu, *Fellow, IEEE*, and Chunxue Deng

Abstract—Controller placement is an important problem in software defined networks. Previously, some works addressed the problem by taking into account primary factors, such as propagation delay of control paths and capacity of controllers. Here, the problem is addressed from a novel standpoint of energy consumption, which is modeled as a binary integer program (BIP). In the BIP model, the energy consumption of the network that serves for the control traffic is minimized under the constraints of the delay of control paths and the load of controllers. In consideration of the high complexity of the BIP in large networks, a genetic heuristic algorithm is designed to find an effective sub-optimal solution. Simulation results show that the energy consumption of the heuristic algorithm is close to that of the BIP solution. No more than 4%, additional links are used by the heuristic algorithm if all the links have the same energy consumption.

Index Terms—Controller placement, energy saving, SDN, software defined networks (SDNs).

I. INTRODUCTION

OPENFLOW [1] is an important enabler for software defined networks (SDNs), which are expanding to more and more areas in recent years. Initially, OpenFlow was designed with only one single centralized controller, which results in some drawbacks such as long response delay and poor scalability. Then, a distributed control plane consisting of multi-controllers was proposed to address these drawbacks [2]–[4]. Since the controller placement has a significant effect on the performance of the network, it is important to determine where the controller(s) should be placed, whether in a single controller case or in a multiple controllers case.

The first work on the controller placement problem is [5]. Based on propagation delay, Heller et al. classified the problem into the facility location problem and found that one controller was enough to meet the delay constraint by evaluating over many real topologies. In [6], the authors reported that more than 20% of all nodes should be controllers to assure a well resilient control plane based on Pareto optimization. Reference [7] studied a capacitated controller placement problem, aiming at minimizing the worst delay of the control paths under the condition of satisfying the load constraint of controllers. In [8], the authors proposed a mathematical model for building a SDN with the lowest cost, while considering the capacity of the controllers and the delay of the paths.

Manuscript received July 23, 2016; revised September 25, 2016 and November 18, 2016; accepted December 18, 2016. Date of publication December 28, 2016; date of current version April 7, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant No.61571065. The associate editor coordinating the review of this letter and approving it for publication was T. Ngatched.

The authors are with the Beijing Key Laboratory of Network System Architecture and Convergence, and Beijing Laboratory of Advanced Information Networks, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: yhu@bupt.edu.cn; tluo@bupt.edu.cn; nborm@bupt.edu.cn; dchunx@bupt.edu.cn).

Digital Object Identifier 10.1109/LCOMM.2016.2645558

Hu et al. worked on building a reliable control plane for SDN in [9], aiming at minimizing the failure probability of the control plane.

Owing to both economic and environmental concerns, the energy consumption of the network can no longer be ignored in light of the rapid development of the Internet [10]. Hence, it is of great importance to design an energy-aware SDN. In [11], the authors proposed a controller association algorithm named GreCo to address the assignment of switches to controllers under an energy saving consideration, in which they assumed that the controller placement was already known. Considering that the placement of the controllers also affects the energy consumption of the network, in this letter, we study the controller placement problem from the perspective of energy consumption. Different from [11], we mainly focus on studying how to choose locations for the controllers in the network to maximize the energy saving of the network. The main contributions of our work are as follows.

- To the best of our knowledge, this letter is the first to consider the controller placement problem from the perspective of energy consumption.
- To find the optimal solution, the energy-aware controller placement problem is formulated based on a binary integer program (BIP), minimizing the energy consumption of the network that serves for the control traffic, jointly with propagation delay and the load of controllers.
- Considering the high complexity of the problem in large scale networks, a heuristic algorithm is introduced to find an effective sub-optimal solution.
- The simulation results show that the methods proposed in this letter can attain improved performances on energy saving, and no more than 4% extra links are used by the heuristic algorithm compared to the BIP solution when all links consume the same energy.

II. NETWORK MODEL

We model the topology of the network as an undirected graph $G(V, E)$, where V represents the set of the nodes (switches), and E represents the set of the edges (links among the switches). $|V|$ and $|E|$ denote the cardinalities of V and E , respectively. Each node is a candidate location for the controllers. If a node has a controller, the controller is attached to the switch in the node. Each switch must be mapped to one and only one controller. Then the links in the control paths (the selected paths between switches and the controllers that manage them) and all the nodes constitute a control network. Note that the paths for inter-controller communication are neglected in this letter. We assume all controllers are fully connected and independent of the paths from switches to controllers. During the operating time of the network, traffic varies all the time. For energy saving, some

idle links can be turned to sleep by turning the corresponding switch ports into sleep mode. In SDN, considering that the control message must be answered quickly, the links belonging to the control network should stay in the working state all the time. Hence, the smaller the energy consumption of the control network, the larger the possible energy saving for the network.

Assuming that all controllers have the same capacity and the control traffic of each switch is the same, the number of switches that a controller manages can be used as a measure to represent the load of the controller. Denote the number of controllers that will be placed in the network as N . It can be readily shown that distributing all the switches over all the controllers equally is the optimal solution for assuring that the load of the controllers will meet their capacities, i.e., the optimal number of switches that a controller manages equals $\lceil \frac{|V|}{N} \rceil$, where $\lceil x \rceil$ is the integer ceiling function of x . Denote the optimal number of switches as L .

As reported in [5], considering pure delay, one controller is enough to satisfy the response time when it is placed in the optimal place. Thus, we set the delay threshold, D_t , as

$$D_t = \min_{\forall c \in V} \max_{\forall s \in V} \{d_{sc}\} \quad (1)$$

where d_{sc} denotes the delay of the shortest path from node s to node c .

III. PROBLEM FORMULATION

Denote the set of the first K shortest paths from node s to node c as P_{sc} , and the k^{th} path as p_k^{sc} . For simplicity, the notation p_k^{sc} is also used as a binary random variable to indicate whether the said path is selected to be a control path. For each link e , we use a set S_e to represent all the paths using link e . Let x_e be a binary random variable indicating whether link e belongs to the control network, and denote the power consumption of link e as ρ_e . Then our objective is

$$\text{minimize } \sum_{e=1}^{|E|} \rho_e x_e \quad (2)$$

subject to the following constraints:

- Ensure that the number of controllers is N . That is,

$$\sum_{c=1}^{|V|} y_c = N \quad (3)$$

where $y_c \in \{0, 1\}$ equals 1 if there is a controller placed in node c , and otherwise equals 0.

- Each switch is mapped to one and only one controller, which means that in all the paths from node s to the other nodes, only one path is selected, viz.

$$\sum_{c=1}^{|V|} \sum_{k=1}^K p_k^{sc} = 1 \quad \forall s \in V \quad (4)$$

where $p_k^{sc} \in \{0, 1\}$ equals 1 if the k^{th} path from node s to node c is selected to be a control path, and otherwise equals 0.

- The switch must be connected to a node where a controller resides, i.e.,

$$\sum_{k=1}^K p_k^{sc} \leq y_c \quad \forall s, c \in V. \quad (5)$$

- The selected paths must meet the delay constraint

$$\sum_{k=1}^K d_k^{sc} p_k^{sc} \leq y_c D_t \quad \forall s, c \in V \quad (6)$$

where d_k^{sc} represents the delay of the k^{th} path from node s to node c .

- Each controller manages L switches at most, thus

$$\sum_{s=1}^{|V|} \sum_{k=1}^K p_k^{sc} \leq y_c L \quad \forall c \in V. \quad (7)$$

- When link e belongs to at least one selected path, x_e equals 1, and otherwise equals 0. That is,

$$x_e \geq p_k^{sc} \quad \forall e \in E, \forall p_k^{sc} \in S_e. \quad (8)$$

Solving the model above yields the optimal solution for energy saving. However, the BIP model can only be used for small and medium scale networks due to its high complexity. For large scale networks, we introduce a heuristic algorithm that finds an effective sub-optimal solution in the next section.

IV. HEURISTIC ALGORITHM FOR LARGE NETWORKS

Here, an improved genetic algorithm, named the improved genetic controller placement algorithm (IGCPA), is used to solve the problem. In IGCPA, each node is encoded by its respective index. A chromosome corresponds to a placement solution of the controllers. After getting a placement of the controllers, address the allocation problem of switches using GreCo [11]. The fitness of a chromosome is the number of the links used in the solution. To calculate the fitness of a chromosome, similarly use GreCo in the fitness function. According to the selected paths between the switches and the corresponding controllers, the number of the links used in the control network can be determined.

The main process of IGCPA is described in Algorithm 1. First, encode the nodes using their indices (Line 2). Then, generate an initialized population including all genes (Line 3). For example, when $|V| = 12$, the initialized population should include all the indices of nodes (1,2,3,...,12). For each member in the population, get its fitness according to GreCo (Line 4). The loop (Lines 6-18) describes the genetic process. If the best solution isn't replaced after $|V| * N^2$ iterations, end the loop. Then take the current best member as the optimal solution.

The genetic process is described as follows. Select the parent members by using the roulette method. Different from the common genetic algorithm, the crossover operator is replaced by a new generation operator (Line 9 in Algorithm 1) which was proposed in [12]. The detailed process of the generation operator is described in Algorithm 2. Denote the set of the same nodes in the parent members as V_{fix} (Line 2), and the set of the different nodes as V_{free} (Line 3). For each node in V_{free} , delete it from V_{free} , and form a new draft solution

Algorithm 1 IGCPA**Input:** $G(V, E)$ **Output:** Placement solution;

```

1: procedure IGCPA
2:   Encode nodes by their indices
3:   Initialize population
4:    $fits \leftarrow Fitness(population)$ 
5:    $m = 0$ 
6:   while  $m \leq |V| * N^2$  do
7:      $fit_{best} = \min(fits)$ 
8:      $members \leftarrow Selection(population)$ 
9:      $members_{new} \leftarrow Generation(members)$ 
10:     $members_{new} \leftarrow Mutation(members_{new})$ 
11:    Update Population
12:     $fits \leftarrow Fitness(population)$ 
13:    if  $fit_{best} \leq \min(fits)$  then
14:       $m \leftarrow m + 1$ 
15:    else
16:       $m = 0$ 
17:    end if
18:  end while
19:  Get the best member from the population
20: end procedure

```

M_{draft} by taking the union of V_{free} and V_{fix} (Line 8). Calculate the fitness of the current draft solution (Line 9). By comparison, get the node which makes the current draft show the best fitness after deleting it (Lines 10-13). Then delete it from V_{free} (Line 15). Do the loop from line 5 to line 17 until the number of genes in the union set consisting of V_{fix} and V_{free} equals N . The mutation operator is common. After generating new members using the generation operator, replace a gene in the new members with the set mutation probability.

The main source of time complexity in IGCPA is the genetic process. In the genetic process, the largest cost is the generation operator. To evaluate the fitness of a solution, only the shortest path needs to be calculated in *Fitness*, which has a complexity of $O(|V|(|E| + |V|\log|V|))$. The *Fitness* function will be called at most $2N^2$ times in the generation operator, and the genetic process will do at most $(|E| - |V| + N) * |V| * N^2$ executions. $|E|$ is at most $|V|^2$, and N is at most $|V|$. Thus, the complexity of IGCPA is $O(|V|^8(|E| + |V|\log|V|))$.

V. EVALUATION

The simulations are carried out using Matlab. Yen's algorithm is used to calculate the first K shortest paths [13], and IBM ILOG CPLEX Optimizer [14] is adopted to solve the BIP model. In IGCPA, we choose 3 pairs of parents solutions each time, and the mutation probability is set at 0.02. The simulations were run on a computer that has 2 Intel Xeon E5-2430 processors with 8 cores, equipped with 16 GB of RAM.

Four topologies with different scales from SND-lib are chosen [15], including Abilene (12 nodes, 15 links), Janos-us (26 nodes, 42 links), Pioro (40 nodes, 89 links), and

Algorithm 2 Generation Operator**Input:** $member_1, member_2$ **Output:** $member_{new}$;

```

1: procedure GENERATE
2:    $V_{fix} \leftarrow$  the same nodes in  $member_1, member_2$ 
3:    $V_{free} \leftarrow$  the different nodes in  $member_1, member_2$ 
4:    $Num = \text{length}(V_{fix} \cup V_{free})$ 
5:   while  $Num > N$  do
6:      $fit_{flag} = 0$ 
7:     for Each node  $\in V_{free}$  do
8:        $M_{draft} \leftarrow (V_{fix} \cup V_{free}) - \text{current node}$ 
9:        $fit_{draft} \leftarrow fitness(M_{draft})$ 
10:      if  $fit_{draft} > fit_{flag}$  then
11:         $fit_{flag} \leftarrow fit_{draft}$ 
12:         $V_{worst} \leftarrow \text{current node}$ 
13:      end if
14:    end for
15:    Remove  $V_{worst}$  from  $V_{free}$ 
16:     $Num = Num - 1$ 
17:  end while
18:   $Member_{new} \leftarrow V_{fix} \cup V_{free}$ 
19: end procedure

```

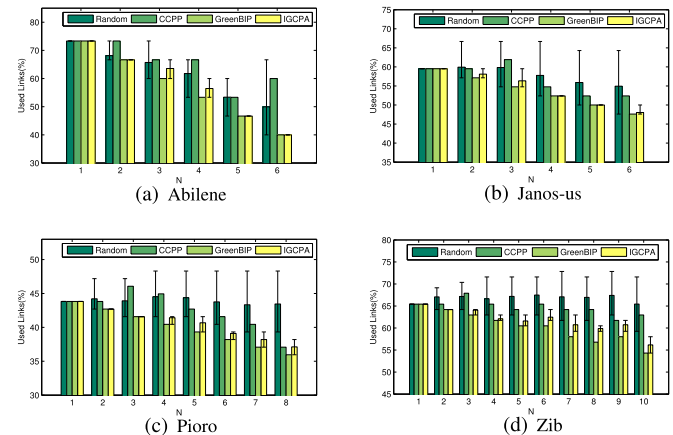


Fig. 1. The percentages of links used for different topologies. The 95% confidence intervals are also shown.

Zib (54 nodes, 81 links). The link weights are set to be the propagation delay calculated from the distance.

Simulations are conducted to test the behavior of the networks as the number of controllers varies. To assess the improvement in energy saving brought by the controller placement, we compare the BIP (dubbed as GreenBIP in the results) and IGCPA with a random placement, in which GreCo is still used for the switch assignment problem. To examine the improvement in energy saving by considering energy consumption of the control network, we compare the green methods with the work in [7] (dubbed as CCPP in the results), in which the authors considered a capacitated controller placement problem, aiming at minimizing the maximum delay of the control paths with consideration of the load of controllers.

For simplicity, we assume that each link consumes the same energy when in the working state. Then for energy saving, the number of links used in the control network should be as few as possible. Fig. 1 shows the results for the percentages of

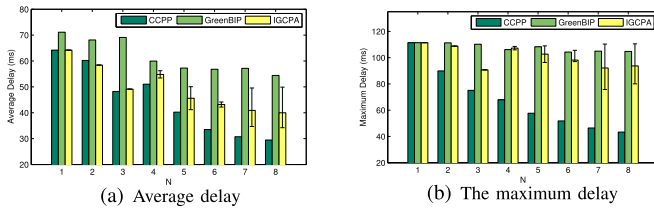


Fig. 2. The results for delay of the Pioro topology.

TABLE I
THE EXECUTION TIMES OF GreenBIP AND IGCPA (UNITS: s)

Topology	N	GreenBIP	IGCPA	N	GreenBIP	IGCPA
Abilene	1	0.54	0.09	4	1.54	7.33
	2	0.96	0.82	5	1.41	6.92
	3	1.38	1.71	6	1.19	14.6
Janos-us	1	181	0.34	4	1819	22.88
	2	790	3.61	5	1521	41.56
	3	1441	14.56	6	1290	82.14
Pioro	1	948	0.72	5	4987	193.5
	2	4020	7.63	6	4250	389.1
	3	4707	31.88	7	3890	534.3
	4	5822	84.18	8	4241	751.6
Zib	1	4496	2	6	17026	1062
	2	23939	17	7	11880	1540
	3	42127	69	8	11942	1630
	4	21060	205	9	13429	2617
	5	17855	391	10	11451	3815

links used for the topologies. The heights of bars for random placement and IGCPA present the averages over 100 independent runs, and the error bars show the 95% confidence intervals. Comparing with the results for random placement and CCPP in the four figures, the two green methods use fewer links in most cases. In the $N = 1$ cases, they perform the same. Compared with GreenBIP, about 4% more links for the abilene topology, 2% more links for the Janos-us topology, 2% more links for the Pioro topology, and 4% more links for the Zib topology at most are used by the IGCPA. For all the topologies, as the number of controllers increases, the number of the links used by both the GreenBIP and IGCPA decreases. Note that there is an exception in the results obtained from the GreenBIP for the Zib topology in Fig. 1. The number of the links used with $N = 9$ is a little more than that with $N = 8$. This is caused by the load constraint. In the $N = 9$ case, the 54 switches are divided into 9 groups on average strictly, which means each group has exactly 6 switches. While, when $N = 8$, the number of the switches that a controller manages is limited to at most 7.

To examine the tradeoff between delay and energy consumption, we further observe the improved delay of the control paths for energy saving. Fig. 2 shows the results for average delay of the Pioro topology. It is observed that the average delays of GreenBIP and IGCPA are always worse than those of CCPP. The gap of the average delay between CCPP and IGCPA is less than that between CCPP and GreenBIP. Comparing these delay results with the results in Fig. 1, it can be seen that, for energy saving, the delay is improved. Yet all the delays of the control paths still can satisfy the delay threshold as shown

in Fig. 2. The results for the other topologies are similar to those of Pioro. Due to length restrictions, they are omitted.

To compare the time saving achieved by IGCPA, the execution times of the two green methods are presented in TABLE I. The results of IGCPA are statistical averages. It is observed that except for the smallest Abilene topology, the execution times of IGCPA are always much shorter than those of GreenBIP.

VI. CONCLUSION

In this letter, we studied how to place the controllers for a distributed control plane to maximize the possibility of energy saving in SDN. The problem was modeled as a BIP problem, which is a superior model for small scale networks. For large scale networks, we used a heuristic algorithm called IGCPA to find an effective sub-optimal solution for the energy aware controller placement problem. Evaluation results indicate that IGCPA can find a solution close to the optimal one. In our future work, we plan to study the energy saving problem on the data forwarding plane based on this energy-aware control plane for a complete energy-aware SDN.

REFERENCES

- [1] N. McKeown *et al.*, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Apr. 2008.
- [2] T. Koponen *et al.*, "Onix: A distributed control platform for large-scale production networks," in *Proc. OSDI*, 2010, pp. 1–14.
- [3] A. Tootoonchian and Y. Ganjali, "HyperFlow: A distributed control plane for openflow," in *Proc. INM/WREN*, 2010, pp. 1–6.
- [4] D. Levin, A. Wundsam, B. Heller, N. Handigol, and A. Feldmann, "Logically centralized?: State distribution trade-offs in software defined networks," in *Proc. 1st Workshop Hot Topics Softw. Defined Netw.*, Aug. 2012, pp. 1–6.
- [5] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in *Proc. 1st Workshop Hot Topics Softw. Defined Netw.*, Aug. 2012, pp. 7–12.
- [6] D. Hock, M. Hartmann, S. Gebert, M. Jarschel, T. Zinner, and P. Tran-Gia, "Pareto-optimal resilient controller placement in SDN-based core networks," in *Proc. ITC*, Sep. 2013, pp. 1–9.
- [7] G. Yao, J. Bi, Y. Li, and L. Guo, "On the capacitated controller placement problem in software defined networks," *IEEE Commun. Lett.*, vol. 18, no. 8, pp. 1339–1342, Aug. 2014.
- [8] A. Sallahi and M. St-Hilaire, "Optimal model for the controller placement problem in software defined networks," *IEEE Commun. Lett.*, vol. 19, no. 1, pp. 30–33, Jan. 2015.
- [9] Y. Hu, W. Wendong, X. Gong, X. Que, and C. Shiduan, "Reliability-aware controller placement for software-defined networks," in *Proc. IFIP/IEEE IM*, May 2013, pp. 672–675.
- [10] M. Gupta and S. Singh, "Greening of the Internet," in *Proc. SIGCOMM*, Aug. 2003, pp. 19–26.
- [11] A. Ruiz-Rivera, K. W. Chin, and S. Soh, "GreCo: An energy aware controller association algorithm for software defined networks," *IEEE Commun. Lett.*, vol. 19, no. 4, pp. 541–544, Apr. 2015.
- [12] O. Alp, E. Erkut, and Z. Drezner, "An efficient genetic algorithm for the p -median problem," *Ann. Oper. Res.*, vol. 122, no. 1, pp. 21–42, Sep. 2003.
- [13] Y. Y. Jin, "Finding the k shortest loopless paths in a network," *Manage. Sci.*, vol. 17, no. 11, pp. 712–716, 1971.
- [14] CPLEX: IBM's Linear Programming Solver, accessed on 2014. [Online]. Available: <http://www.ilog.com/product/cplex/>
- [15] SNDlib: Library of Test Instance for Survivable Fixed Telecommunication Network Design, accessed on 2014. [Online]. Available: <http://sndlib.zib.de/home.action>