

Incremental Switch Deployment for Hybrid Software-Defined Networks

Xuya Jia*, Yong Jiang*, Zehua Guo[†]

{jiaxy14,jiangy}@mails.tsinghua.edu.cn, guolizihao@hotmail.com

*Department of Computer Science, Tsinghua University Shenzhen Graduate School,

Tsinghua National Laboratory for Information Science and Technology(TNList). [†] ChinaCache

Abstract—Software-Defined Networking (SDN) brings great opportunities to improve network performance. However, due to budget constraints and technique limitations, Internet Service Providers (ISPs) can upgrade only a limited number of conventional switches to SDN switches in real backbone networks at one time. In this paper, we propose one heuristic scheme for deploying SDN switches in hybrid SDNs. Our scheme works for two different cases: (1) maximizing the network control ability with a given upgrading budget constraint, and (2) minimizing the upgrading cost to achieve the best network control ability. We evaluate our scheme in real topologies. The results show that our scheme can achieve 95% of flows controlled with only 10% upgrading cost.

Keywords—Software-Defined Networking, Switch deployment, Flow control, Network management.

I. INTRODUCTION

Software-Defined Networking (SDN) technique is widely used in recent years [1][2]. However, in the global Internet environment, the deployment of SDN requires a relatively long period. Therefore, the hybrid SDN, which is composed of SDN devices and traditional network equipment, will exist during a long time [3][4].

In BTSDN [5], it states that the SDN switches and the current Internet can coexist with each other, and the SDN can be incrementally deployed to the Internet. Generally speaking, the network upgrade is a complex operation. ISPs consider not only the budget constraints but also the normal network operation [6]. Thus, it is not practical to upgrade all network devices (e.g., switches and routers) in real networks to SDN devices at one time. Therefore, Internet Service Providers (ISPs) should trade off financial consideration and the network control ability (NCA) to deploy¹ SDN switches in existing networks [2]. To achieve the efficient upgrade, we are faced with two real problems: (1) given the requirement to improve NCA, how to determine the number and the location of nodes in a network to be upgraded to SDN switches? (2) given a limited number of SDN switches, how to place them to achieve the maximum NCA improvement?

In this paper, we propose a heuristic scheme to solve the problem of incremental SDN switch deployment in the hybrid SDNs. First, we formulate an optimization problem to achieve the maximum NCA with little upgrading cost. Due to the high complexity of the problem, we transform it into two problems with real constraints: Max-NCA problem, which aims to

maximize the NCA improvement with a given upgrading cost, and Min-Cost problem, which deploys the minimum number of SDN switches to achieve the maximum NCA improvement in a network. We prove that the two problems are NP-hard, and give a heuristic algorithm to solve the two problems with low complexity. Simulation results show that our scheme can achieve 95% NCA improvement with only 10% upgrading cost.

II. RELATED WORK

Some existing works propose deploying SDN switches to achieve network security, load balancing and traffic scheduling [4][6][7]. Dan Levin *et al.* [8] deploy SDN switches in enterprise networks using VLANs and switch ports to enable each flow to go through at least one SDN switch. The scheme is limited by the number of VLANs, and needs to set the switch ports. In [1], the objective is to develop an SDN deployment scheme that can adaptively and dynamically manage traffic in a network to accommodate different traffic patterns. The solution has high complexity and is not feasible in real networks. Chu *et al.* [4] propose an approach to guarantee traffic reachability in the presence of any single link failure. However, it increases the path stretch of flows and wastes the flow table entries for pre-configured. In [9], the authors focus on the hybrid SDN network scenario, and search for an optimized migration sequence of the routers in the traditional IP network to minimize the maximum link utilization. But it does not give a specific SDN switch deployment plan.

A premise is that we do not modify the flow path of traditional networks. Hence, there is not additional path stretch in our scheme. We deploy the minimum number of SDN switches with a given budget and achieve almost all of the flows go through at least one SDN switch.

III. PROBLEM FORMULATION

We assume that the SDN forwarding elements are a subset of the nodes in the network. The rest nodes in the network run standard routing protocol, such as OSPF [1]. We formulate a network as a graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes, and $\mathcal{E}$ is the set of links. Let $\mathcal{F}$ denote the set of traffic flows. We use v_i to represent a node in the network, l_{ij} to represent the link between nodes v_i and v_j . Let C_{ij} denote the capacity of link l_{ij} , and f_{ij}^k ($f_{ij}^k \in \mathcal{F}$) be the size of the k -th flow on link l_{ij} . The notations of this paper are listed in Table I.

¹In this paper, we use deploy and upgrade interchangeably.

Table I
NOTATIONS.

$\mathcal{V}$	The set of nodes in the network.
M	The number of nodes in the set $\mathcal{V}$.
$\mathcal{Q}$	A sub-subset of $\mathcal{V}$ ($\mathcal{Q} \subseteq \mathcal{V}$).
$\mathcal{E}$	The set of links in the network.
$\mathcal{F}$	The set of flows in the network.
K	The number of flows in the set $\mathcal{F}$.
v_i	The i -th node in the network, $v_i \in \mathcal{V}$ ($0 < i \leq M$).
l_{ij}	The link between nodes v_i and v_j , $l_{ij} \in \mathcal{E}$, $v_i \in \mathcal{V}$, $v_j \in \mathcal{V}$ ($0 < i \leq M$, $0 < j \leq M$, $i \neq j$).
C_{ij}	The capacity of link l_{ij} .
f_{ij}^k	The size of the k -th flow on link l_{ij} , $f_{ij}^k \in \mathcal{F}$.
s^k	The source node of flow f_{ij}^k , $s^k \in \mathcal{V}$.
d^k	The destination node of flow f_{ij}^k , $d^k \in \mathcal{V}$.
$\mathcal{F}_{v_i}$	The set of flows passing through node v_i ($\mathcal{F}_{v_i} \subset \mathcal{F}$).
δ_i	$\delta_i = 1$ if node v_i is upgraded to an SDN switch; $\delta_i = 0$, otherwise.
p_i	The number of forwarding ports in node v_i .
s_i	The traffic processing speed of node v_i .
Φ	A given budget for deploying SDN switches.

A. Objective function The goal of incremental SDN switch deployment is to improve the network control ability (NCA) by upgrading a small number of switches to SDN switches at little cost. In [4], the authors show that the hybrid SDN can achieve fast recovery if the flows pass through at least one designated SDN switch. In [8][10], it becomes possible to operate the enterprise network if the source-destination paths include at least one SDN switch. Therefore, the flow can be monitored or flexibly scheduled if it traverses at least one SDN switch in hybrid networks. To quantify the performance of a network, we introduce some definitions below:

Definition 1. Flow control ability: Since an SDN switch can flexibly control all flows that traverse it [5], we assume that if node v_i is an SDN switch (e.g., $\delta_i = 1$), its flow control ability equals to the total size of flows that pass through the node; otherwise (e.g., $\delta_i = 0$), its flow control ability is zero. For node v_i ($v_i \in \mathcal{V}$), the flow control ability is formulated as follows:

$$\mathcal{F}_{v_i} = \begin{cases} \sum_{i: l_{ij} \in \mathcal{E}} \sum_{k=1}^K f_{ij}^k, & \delta_i = 1 \\ 0, & \delta_i = 0 \end{cases} \quad (1)$$

Definition 2. Network control ability (NCA): The network control ability is defined as the flows that can be flexibly managed by deploying SDN switches in the network. In other words, the network control ability is the total flow control ability of nodes in the network. Therefore, for $\mathcal{Q}$, a sub-set of $\mathcal{V}$, we have its network control ability:

$$w(\mathcal{Q}) = \sum_{\mathcal{F}_{v_i} \subseteq \mathcal{F}} \bigcup_{v_i \in \mathcal{Q}} \mathcal{F}_{v_i}. \quad (2)$$

Definition 3. Upgrading cost: The upgrading cost of a network is the total cost for upgrading conventional switches to SDN switches in the network. For switch v_i , its cost depends on many factors, such as p_i , the number of forwarding ports in the node, and s_i , the flow processing speed in the switch [9]. We assume the cost of switch v_i is c_i , $c_i = q(p_i, s_i) =$

$a_1 \cdot p_i + a_2 \cdot s_i$, where a_1 and a_2 are constant. Therefore, the upgrade cost of the network is formulated as:

$$Cost(\mathcal{Q}) = \sum_{i=1}^M c_i \cdot \delta_i = \sum_{i=1}^M q(p_i, s_i) \cdot \delta_i, \quad (3)$$

In [10], the authors choose the next switch u to upgrade by computing the max *degree-cost ratio* $\Delta(u)/\gamma(u)$, where $\Delta(u)$ denotes the number of links incident to u and $\gamma(u)$ denotes the cost of upgrading switch u . Similarly, we formulate the objective function to max *NCA-cost ratio* as follows:

$$\frac{w(\mathcal{Q})}{Cost(\mathcal{Q})} = \frac{w(\mathcal{Q})}{\sum_{i=1}^M q(p_i, s_i) \cdot \delta_i}, \quad (4)$$

B. Problem formulation The problem of the incremental switch deployment for a hybrid SDN is formulated as below:

$$\max \frac{w(\mathcal{Q})}{\sum_{i=1}^M q(p_i, s_i) \cdot \delta_i} \quad (5a)$$

subject to

$$\sum_{i=1}^M q(p_i, s_i) \cdot \delta_i \leq \Phi \quad (5b)$$

$$\sum_{j: l_{ij} \in \mathcal{E}, i \neq s^k, d^k} f_{ij}^k = \sum_{j: l_{ji} \in \mathcal{E}, i \neq s^k, d^k} f_{ji}^k \quad (5c)$$

$$\bigcup_{v_i \in \mathcal{V}} \delta_i \cdot v_i = \mathcal{Q} \quad (5d)$$

$$\sum_{k=1}^K f_{ij}^k \leq C_{ij} \quad (5e)$$

$$w(v_i) = \begin{cases} \sum_{i: l_{ij} \in \mathcal{E}} \sum_{k=1}^K f_{ij}^k, & \delta_i = 1 \\ 0, & \delta_i = 0 \end{cases} \quad (5f)$$

$$\delta_i \in \{0, 1\}, \quad v_i \in \mathcal{V}, \quad \mathcal{Q} \subseteq \mathcal{V}, \quad 0 < k \leq K,$$

$$f_{ij}^k \geq 0, \quad 0 < i \leq M, \quad 0 < j \leq M, \quad s^k \in \mathcal{V}, \quad d^k \in \mathcal{V}.$$

Constraint (5b) ensures that the cost is less than or equal to a given upgrading budget. Equation (5c) indicates that the total size of incoming flows of a node equals to the total size of outgoing flows of the node except the source and destination nodes. Equation (5d) represents the set of nodes which are upgraded to SDN switches. Constraint (5e) states that the traffic load of a link cannot exceed its capacity. Equation (5f) describes the flow control ability of a node.

C. Complexity analysis Since the cost function of a switch, $q(p_i, s_i)$, is non-linear to p_i and s_i , the function $Cost(\mathcal{Q})$ is a non-linear function [11]. Considering variable δ_i is binary, the proposed model is an Integer Non-Linear Programming problem, and its complexity is $O(2^M)$, where M is the number of nodes in the network. Typically, a backbone network consists of a large number of nodes [7]. Thus, it is not feasible to solve the problem by the simple method of brute force enumeration.

IV. HARDNESS ANALYSIS AND HEURISTIC ALGORITHMS

Due to the high complexity of the problem, we take into account two real constraints to convert the problem into two

general sub-problems: the Max-NCA problem and the Min-Cost problem. In this section, we first prove the hardness of the two problems and then propose a heuristic algorithm to solve the two problems.

A. The hardness of Max-NCA The Max-NCA problem is to maximize the NCA improvement under the constraint that the upgrading cost is less than or equal to a given upgrading budget: $\sum_{i=1}^M q(p_i, s_i) \cdot \delta_i \leq \Phi$.

Theorem 1: The Max-NCA problem is an NP-hard problem.

Proof: We can convert the Max-NCA problem to a Weighted Set Cover problem. In the Weighted Set Cover problem, there is a link-set E with n elements: $e_1, e_2, \dots, e_n$. Link-set E can be decomposed into m subsets: $S_1, S_2, \dots, S_m$, and the subsets are associated with their weights $w_1, w_2, \dots, w_m$. We treat the flow set $\mathcal{F}$ as the set of E , and each node v_i in the network as the subset S_i . Of course, the flows passing through node v_i correspond to the elements of the subset S_i , the cost of the switch v_i corresponds to the weight w_i of the respective subset S_i . By construction, it is easy to verify that there exists the maximum number of flows controlled under the budget, if and only if there exists the optimal solution to the Weighted Set Cover problem under the resource constraints. The construction can be done in polynomial time and the Weighted Set Cover is an NP-hard problem [12]. Hence, the Max-NCA problem is also NP-hard.

B. The hardness of Min-Cost The Min-Cost problem is to deploy the minimum number of SDN switches to control all flows in the network.

Theorem 2: The Min-Cost problem is NP-hard.

Proof: We can convert the Min-Cost problem to the Minimum Weighted Vertex Cover problem. Given an example of the Minimum Weighted Vertex Cover problem. It includes the point set N , which has k elements $n_1, n_2, \dots, n_k$, and the link set E which has m elements $e_1, e_2, \dots, e_m$, the points have corresponding weight $w_1, w_2, \dots, w_k$. All points and links form a graph. We treat the flow set $\mathcal{F}$ as the link set E , and each node v_i in the network as the point n_i in the graph. Of course, the flows go through node v_i corresponding to the links connected to point n_i , the cost of each switch corresponding to the weight of the point. Through construction, it is easy to prove that there exists the minimum cost to deploy SDN switch to control all the flows, if and only if there exists the optimal solution to the Minimum Weighted Vertex Cover problem. The construction can be done in polynomial time and the Minimum Weighted Vertex Cover is an NP-hard problem [12]. Hence, the Min-Cost problem is also NP-hard.

C. Heuristic algorithm We propose a heuristic algorithm for the Max-NCA and the Min-Cost problems. In Algorithm 1, line 4 guarantees that the cost of the deployment is under budget or the set of flows is not empty. Line 5 greedily selects the optimal node to deploy SDN switch based on the flows and the cost of the node. For the Max-NCA problem, it adds the value of the selected node to $Cost(\mathcal{Q})$ in line 7 of the algorithm. For the problem Min-Cost, $\mathcal{F}$ subtracts the flows of the selected node in line 10. For the Max-NCA problem, the time complexity of the algorithm is $O(km)$. For the Min-Cost

Algorithm 1 Greedy upgrading

```

1: Input: graph  $G = (\mathcal{V}, \mathcal{E})$ , budget  $\Phi$ , flow-set  $\mathcal{F}$ 
2: Output: the selected SDN switch set  $\mathcal{Q}$ 
3:  $\mathcal{Q} = \emptyset$ ;  $Cost(\mathcal{Q}) < \Phi$ 
4: while  $Cost(\mathcal{Q}) < \Phi$  or  $\mathcal{F} \neq \emptyset$  do
5:   Greedily select node  $v_i$  ( $v_i \in \mathcal{V}$ ) from all unselected switch
   to update;  $v_i = \text{maximize } w(v_i)/Cost(v_i)$ ;
6:   if Max-NCA problem then
7:      $Cost(\mathcal{Q}) \leftarrow Cost(\mathcal{Q}) + Cost(v_i)$ ;
8:   end if
9:   if Min-Cost problem then
10:     $\mathcal{F} \leftarrow \mathcal{F} - \mathcal{F}_{v_i}$ ;
11:   end if
12:    $\mathcal{Q} \leftarrow \mathcal{Q} + \{v_i\}$ ;
13: end while
14: return  $\mathcal{Q}$  and  $Cost(\mathcal{Q})$ 

```

problem, the time complexity of the algorithm is $O(mn)$.

V. SIMULATION

In this section, we evaluate our solution against the baseline scheme in real network topologies. The simulation results of the following algorithms are compared with each other to evaluate the algorithm 1.

- **Heuristic:** The scheme uses Algorithm 1 to update nodes to SDN switches.
- **Degree-first:** The scheme in [10], which chooses the next switch u to upgrade by computing the maximum number of links incident to the SDN switch.
- **Random:** The scheme randomly selects nodes to upgrade to SDN switches in the network.

A. Simulation setup In our simulation, we use two real network topologies the ISP 1755 and the ISP 3967 from the Rocketfuel [13]. In real networks, some nodes do not communicate with others. Thus, we consider two cases. In Case 1, each node sends flows to all other nodes, while in Case 2, only a part of node pairs communicate with each other. Since the flows between node pairs are dynamically generated, and the size of each flow is also dynamic, we use σ to indicate the percentage of the average number of controlled flows [8].

B. Simulation results

1) *Case 1: flows among all nodes.*: Fig. 1 shows the results of the problem Max-NCA in ISP 1755 and ISP 3967. Among the three schemes, Random performs the worst. In Fig. 1(a), Random requires 14% cost to improve 55% of σ , while Heuristic needs only 2% cost. Heuristic needs only 10% cost to realize 98% of σ improvement. The performance of the heuristic is obviously better than the Degree-first when the percentage of upgrading cost is less than 8% in Fig. 1(a) or less than 14% in Fig. 1(b). Compared Fig. 1(a) to Fig. 1(b), we can conclude that as network scale extends, Heuristic's advantage is more apparent than Random.

2) *Case 2: flows between certain node pairs.*: We compare these results in two instances that 1/3 node pairs have flows to send in Fig. 2, and 2/3 node pairs have flows to send in Fig. 3. For the two cases, we randomly generate node pairs 100 times and ran our algorithms 100 times to get the average value on

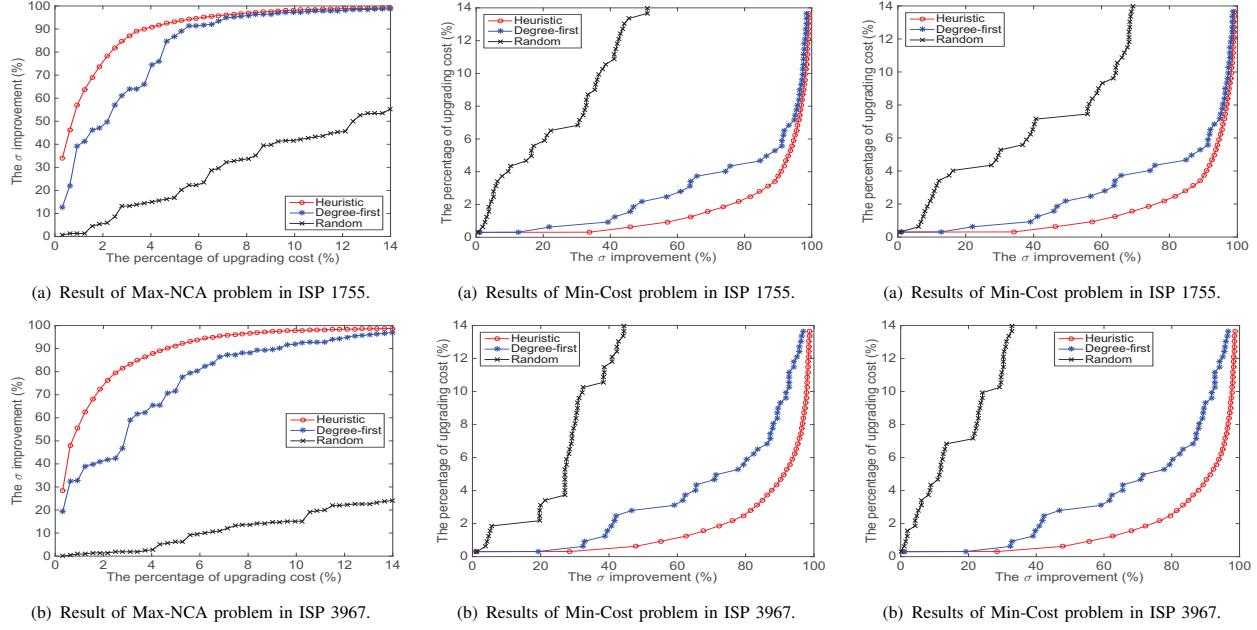


Figure 1. The percentage of controlled flows changes with the percentage of SDN deployment gies where about 1/3 node pairs have flows. cost.

a high-performance computer. The Fig. 2 shows that, when we want to control 95% of the number of flows, we need to upgrade only about 6% of the nodes to SDN switches. Similar to the results of Fig. 1, the Heuristic algorithm in Fig. 2 also outperforms Degree-first algorithm. Compared Fig. 2 to Fig. 3, the heuristic algorithm can achieve good results both 1/3 node pairs have flows and 2/3 node pairs have flows. In addition, the Fig. 2 and Fig. 3 show that the results of the algorithms are similar even in different topologies. Moreover, the Heuristic is always better than Degree-first in any case. Therefore, our algorithms are universal and extensive.

VI. CONCLUSION

In this paper, we propose a scheme to deploy SDN switches incrementally in hybrid SDNs. Taking into account the real constraints, we propose two general problems: the Max-NCA problem and the Min-Cost problem. We propose a heuristic algorithm for the two problems. The simulation shows that our scheme can control 95% of flows in the network with only about 10% upgrading cost.

ACKNOWLEDGMENT

This work is supported by the National Research Program of China (973) under grant No. 2012CB315803, the National Natural Science Foundation of China under grant No. 61402255, the R&D Program of Shenzhen under grant No. ZDSYS20140509172959989, No. JSGG20150512162853495, No. Shenfagai[2015]986, and JCYJ20150630170146830.

REFERENCES

- [1] S. Agarwal, M. Kodialam, and T. Lakshman, "Traffic engineering in software defined networks," in *Proceedings of IEEE INFOCOM*. IEEE, 2013, pp. 2211–2219.
- [2] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, and Wanderer, "B4: Experience with a globally-deployed software defined wan," in *Proceedings of ACM SIGCOMM*. ACM, 2013, pp. 3–14.
- [3] Z. Guo, Y. Xu, M. Cello, J. Zhang, Z. Wang, M. Liu, and J. C. H., "Jumpflow: Reducing flow table usage in software-defined networks," in *Computer Networks*, 2015 Elsevier, pp.300-315.
- [4] C.-Y. Chu, K. Xi, M. Luo, and H. J. Chao, "Congestion-aware single link failure recovery in hybrid sdn networks," in *Computer Communications (INFOCOM)*, 2015 IEEE Conference on. IEEE, 2015, pp. 1086–1094.
- [5] P. Lin, J. Bi, and H. Hu, "Btsdn: Bgp-based transition for the existing networks to sdn," in *Ubiquitous and Future Networks*, 2014 Sixth International Conf on. IEEE, 2014, pp. 419–424.
- [6] Z. Li, Q. Li, L. Zhao, and Xiong, "Openflow channel deployment algorithm for software-defined afdx," in *Digital Avionics Systems Conference*, 2014 IEEE 33rd. IEEE, 2014, pp. 4A6–1.
- [7] Y. Guo, Z. Wang, X. Yin, X. Shi, and J. Wu, "Traffic engineering in sdn/ospf hybrid network," in *Network Protocols*, 2014 IEEE 22nd International Conference on. IEEE, 2014, pp. 563–568.
- [8] D. Levin, M. Canini, and Schmid., "Panopticon: Reaping the benefits of incremental sdn deployment in enterprise networks," in *Proceedings of USENIX ATC 14*, 2014, pp. 333–345.
- [9] Y. Guo, Z. Wang, X. Yin, and X. Shi, "Incremental deployment for traffic engineering in hybrid sdn network," in *Computing and Communications Conference*. IEEE, 2015, pp. 1–8.
- [10] D. Levin, S. Canini, and A. Feldmann, *Panopticon: Reaping the benefits of partial sdn deployment in enterprise networks*. Die Professoren der Fakult, Elektrotechnik und Informatik, 2013.
- [11] "The switches of official quoted." [Online]. Available: <http://product.yesky.com/switchboard/list2.html>
- [12] T. H. Cormen, *Introduction to algorithms*. MIT press, 2009.
- [13] N. Spring, R. Mahajan, and D. Wetherall, "Measuring isp topologies with rocketfuel," in *ACM SIGCOMM Computer Communication Review*. ACM, 2002, pp. 133–145.