Migrating to SDN for Mobile Core Networks: A Dynamic and Global Perspective

1st Xiaole Li
School of Electronic and Information Engineering
Beijing Jiaotong University
Beijing, China
18120101@bjtu.edu.cn

2nd Hongyun Zheng
School of Electronic and Information Engineering
Beijing Jiaotong University
Beijing, China
hyzheng@bjtu.edu.cn

3rd Yuchun Guo
School of Electronic and Information Engineering
Beijing Jiaotong University
Beijing, China
ychguo@bjtu.edu.cn

***Abstract*—The rapid growth of mobile traffic and the vast variety online activities demand enhancement of mobile networks. Software-Defined Networking (SDN) is considered a promising solution for the core network. But upgrading to SDN from a legacy network is a multi-period process. From both dynamic and global perspectives, during the migration it needs to simultaneously consider optimizing the local goal at each intermediate step and the global goal. This involves in some essential questions that have to be answered: which legacy devices to upgrade and when to, and how to place controllers. Due to interaction between the local goals and the global goal, however, to answer these questions together is challenging. In this paper we study the SDN migration problem and answer all of these questions together. We formulate the SDN migration problem as a time-varying dual-objective dynamic optimization model, in which the timing factor is taken into account, and the objective of optimization varies with time. Then we introduce a penalty item to convert the dual-objective dynamic optimization problem to a series of single step optimization problems that can be solved directly by CPLEX for small scale networks. The simulation results based on real network topology show that our model can obtain a tradeoff between the local goals and global one.**

***Index Terms*—SDN, hybrid SDN, upgrading problem, controller deployment, mobile network**

I. Introduction

In past decade mobile traffic is continuing to grow and contribute to more than half of total Internet traffic [1]. With the pervasiveness of portable mobile intelligent devices like smartphones, tablets, and notebooks, mobile devices account for more and more online activities, including web surfing, video streaming, social networking, and online gaming [2]. Internet has demonstrated a clear shift towards mobile form.

Such a proliferation of mobile data, as well as the vast variety of online activities, imposes a heavy pressure to mobile network. Solutions to enhance mobile networks are requested. From the view of core network, Software-Defined Networking (SDN) has been seen as a promising architecture [3]. With the separation of control and data plane, SDN offers flexible network programmability, and will likely play a crucial role in upgrading the current 3G/4G and designing of the next generation 5G wireless network [4]. However, due to techno-economic constraints, network is usually upgraded to SDN in a way of multiple periods or steps, which will typically span over several years [5]. At each step, only a part of selected legacy switches are replaced by SDN switches. The process ends until all of legacy switches have been upgraded and thus the network migrates to a pure SDN. For example, AT&T upgraded 34%, 55% and 75% of its network to SDN by 2016, 2017 and 2020, respectively [6]. Hence in the upgrade process the legacy devices and SDN switches may coexist. This network is called a hybrid SDN [7].

Since all of legacy network devices cannot be upgraded at once, when making SDN migration plan a natural question is: which legacy devices should upgrade first, and which subsequently, etc, *i.e.*, *which* and *when*. The problem is referred as SDN migration schedule problem [8]. In this paper, we also call it SDN upgrade or migration problem. The incentive upgrade to SDN technology is to enjoy the benefits of network programmability. The selection of switches to upgrade will affect network's programmability. An efficient migration scheme is to spread the benefit of network programmability as far as possible. Existing studies proposed many SDN upgrade schemes with the goal of maximizing network programmability [9], [10]. But these works do not take the impact of SDN controller on the network migration into consideration.

The SDN migration problem will become more complicated when taking the controllers into account. The locations of controllers have impact on the control latency, which is an important metric effecting the performance of SDN [11]. Hence *how* to place controllers (*i.e.*, where to deploy controllers, which controller controls which SDN switches) is an important issue for SDNs [12]. And for the migration it is another essential question has to be addressed. At each intermediate step together considering deployment of controllers and selection of SDN switches to upgrade in a hybrid SDN is complicated. Just as shown in [13], to guarantee the performance of such a partial upgraded network, it needs to jointly deploy controllers and upgrade switches. The objective of the upgrade scheme is dual: maximizing network programmability and minimizing the delay at the same time. We refer to the joint upgrade objective for each step as a single-step or a local objective. It is worth noting that at the last step the migration leads to a pure SDN, where all devices are SDN switches and it does not need to optimize network programmability. Thus from the dynamics of upgrade process point of view, the local goal varies with time, which evolves to a single objective from a dual one.

Moreover, from the perspective of the whole network upgrade, the ultimate aim is to migrate to a pure SDN with optimal performance. The global goal of upgrade is to optimize the delay in the final pure SDN. That is to say, when the migration ends, the whole network controllers should be deployed in such a way that the delay is minimized. It is worth noting that this global goal is just the same as the local one at the last step of upgrade since the final network is exactly that pure SDN. Hence the global objective and those local ones at intermediate steps are related to each other while they are inconsistent. On the one hand, due to the dynamics of the upgrade process, the final global result is affected by the previous steps. On the other hand, from a global point of view the upgrade result at last step also has a constraint to the upgrades of previous steps. It makes the upgrade process different from classical multi-period dynamic processes where typically only the outputs of forgoing steps affect the successor rather than in bi-direction.

In a summary, the SDN migration problem is complicated. It needs to address essential questions mentioned above: *which*, *when* and *how*. However, to the best of our knowledge, there are not related works that answered all of these questions together. Existing works answered either one or two of these questions. Some works studied SDN upgrade at a fixed point of time ignoring multiple periods or dynamic of the migration progress [14]–[16]. Those works that defined SDN upgrade as dynamic optimization problems, however, just focus on local optimal objective for each single step while not recognizing the global one [17]–[19]. Existing works on controller placement problem for pure SDN examined controllers' deployment from the global perspective, but they work statically without considering timing issues of migration [20], [21].

In this paper, we study the SDN upgrade problem from both dynamic and global point of views, and answer these questions together. Our problem is to find an SDN migration scheme, *i.e.*, make a decision to first select which nodes to upgrade, and which subsequently, etc., and the locations of deployed controllers, and how to connect controllers and upgraded switches for simultaneously guaranteeing the local goal of each upgrade step and the final upgrade goal. We formulate this problem as a time-varying dual-objective dynamic optimization problem, and then propose a method to transfer it to a classical dynamic optimization problem to solve. Finally we evaluate our model with a real network topology, and compare our proposed method with other state-of-art methods.

II. Related work

Based on whether timing issues of migration are considered, existing works on hybrid SDNs can be classified as static or dynamic scheme. A static scheme studied SDN upgrade at a fixed point of time. It just aimed at optimizing network's programmability for one step. Study in [14] maximized hybrid SDN's programmability in terms of programmable traffic. Work [15] selected switches to upgrade by maximizing the number of programmable paths. Some works proposed upgrade schemes with other different motivations, such as traffic engineering [16], link failure recovery [22] and power saving [23]. These static schemes have in common that they ignored the impact of controllers on selection of upgraded switches. The only work that considered the factor of controllers is [13]. The authors formulate the problem as a dual-objective static optimization model by jointly upgrading switches and deploying controllers.

A dynamic scheme studies SDN migration not only for the network state at a given point of time, but considering the entire migration trajectory as a whole. The authors in [8] examined SDN migration for maximizing the programmable routing number. They also extended the work with cost constraints in [9]. Work in [10] aimed at simultaneously maximizing programmable traffic and SDN-enabled routing paths. Work in [24] proposed an optimized migration schedule based on the benefits of customers. However, all of these works did not take the factor of controllers' deployment into consideration; consequently they did not recognize the overall goal at all.

As for works about pure SDN, there are abundant works studying how to place controllers [20], [21]. But in pure SDNs, all devices are SDN switches and no intermediate upgrade step exists, so these works only considered the global controller deployment for minimizing the delay.

III. Problem Statement

We consider a multi-period time horizon to present the SDN upgrade process, where the whole process lasts several periods (time-steps), and a set of legacy switches are upgraded in each step, as shown in Fig. 1. It takes T time-steps for the legacy network to upgrade to a SDN. The network topology remains the same during the whole upgrade process [19]. At the beginning, *i.e.*, at step 1, by selecting some legacy switches to upgrade, the network migrates to a hybrid SDN, in which legacy and SDN switches coexist. Some other legacy switches are selected from the remaining to upgrade at subsequent time-steps until all the switches migrate and a pure SDN is ultimately obtained at the last step, at which the upgrade process ends. The upgrades through all steps form the entire upgrade trajectory.

The snapshots at each time-step are hybrid SDN, as shown in (b) and (c), except the last snapshot is a pure SDN, as shown in (d). For hybrid SDN at any intermediate step, the upgrade aims to optimize both of network programmability and delay. Hence the local goal of upgrade is dual: maximizing network programmability benefits and minimizing the delay simultaneously. However, for the final pure SDN in the last snapshot, it only needs to optimize the network performance; the local goal at the last step is merely to minimize the delay. With time elapsing, the local goal of upgrade is varying.

From the perspective of the whole network, the pure SDN in the last snapshot is exactly the end of the upgrade process. To guarantee its performance, the global objective should be to minimize delays. It is also one point on the time horizon, the dynamic of the upgrade process becomes complex. The interaction between snapshots is bi-direction. Not only those intermediate snapshots effect the last snapshot, but also the last one plays a role of ruling how to create previous snapshots. Due to this relatedness it is difficult to simultaneously obtain optimal local goal at each step and optimal global goal. Even though each step achieves optimum, it will not eventually get the global optimum. We will use a motivation example to show the dilemma.

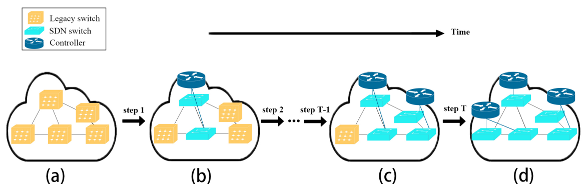


Fig. 1. SDN upgrade trajectory. ((a) legacy network; (b)(c) hybrid SDN; (d) pure SDN)

Fig. 2 shows the network composition: 16 nodes and 23 edges. The number on the edge means the delay between nodes. Assume deploying three controllers for such a pure SDN, the optimal locations are node B, I and L, with minimizing the total delay in SDN. The total delay is $83.5ms$, which is the sum of control delays for all controllers.

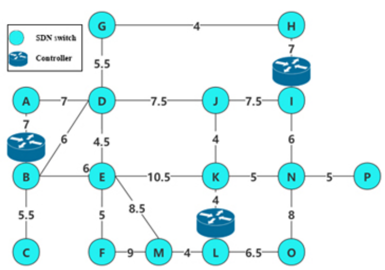


Fig. 2. Optimal layout of controllers for the motivation network.

Now we assume it takes three steps migrating to the pure SDN shown in Fig. 2. Considering upgrade cost, we specify that only 6 switches can be upgraded and 1 controller can be placed at each time-step. For other assumptions we use the same as in [13]. Each controller can only control 6 SDN switches and be placed at the locations of SDN switches. The network programmability is measured by the sum of degrees of the upgraded switches, named degree gain [13]. We use the joint upgrade scheme in [13]. It only considers the optimal layout at each intermediate step with simultaneously maximizing network programmability and minimizing the delay, while not recognizing the global objective.

The migration process is shown in Fig. 3, and the final selected controllers are located on node D, E, and N, with a delay of $93.5ms$, which is longer than that of the global optimal layout in Fig. 2. It shows that the accumulation of local optimum may not exactly be the final global optimum. One may argue that it is possible to achieve the global optimal delay for the pure SDN by rebuilding the existing controllers on the final optimal positions. But it will increase the upgrade cost.

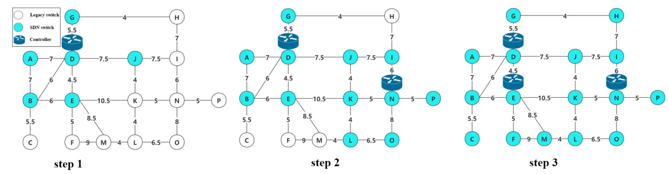


Fig. 3. Upgrade process under the joint deployment.

From this example we can see that due to the complex dynamics it is not trivial to make the migration decision.

IV. Formulation

Based on the analysis above we formulate the SDN upgrade problem as a time-varying dual-objective dynamic optimization problem.

A. Assumption

We model a network as a weighted undirected graph, $G(N, E, \omega(E))$. The vertex set N consists of n network devices which can be legacy switches or SDN switches. The edge set E includes links. The weight $\omega(E)$ is the delay between the two points of the edge.

Assume that controllers can only be placed at the locations of SDN switches [13]. Each controller has the same capability to control c switches. It takes T time-steps to complete the upgrading. At each step, the number of switches to upgrade and the number of controllers to be deployed are fixed. Each switch can only be upgraded once. Each controller cannot change its position once it is placed, for the change would bring demolition and reconstruction costs, which are not expected. But switch can reconnect to other controller to shorten the control delay. The reason that we set these variables with fixed value is to simplify the problem.

B. Problem Formulation

The first objective is maximizing the network programmability benefits in terms of the sum of degrees of SDN switch nodes, named degree gain [13], [15]. This objective only exists at intermediate steps. To characterize the fact that the goal changes over time, we introduce the fade factor β^t, which decreases to zero as the upgrade progresses. The final step does not involve the degree gain, so $\beta^t = 0$. The first objective is as following.

$$obj_1 : \max \sum_{i \in N} \beta^t * d_i * x_i^t \tag{1}$$

Where, d_i is the degree of node i. And $x_i^t = 1$ means that switch i is an SDN switch at time t, otherwise $x_i^t = 0$. Notations used in the model are shown in Table I.

The second objective shown in (2) is to minimize delay, which is necessary throughout the entire upgrade process.

$$obj_2 : \min \sum_{i,j \in N} w_{ij} * z_{ij}^t \tag{2}$$

where $w_{i,j}$ is the delay between node i and j. And $z_{ij}^t = 1$ denotes that controller i controls the SDN switch j at step t, otherwise, $z_{ij}^t = 0$.

Due to the inconsistence of goals between hybrid SDN and pure SDN, it needs to comprehensively consider the upgrade trajectory as a whole. All of decision variables of any step need to be solved together. Thus, we formulate the upgrade process as a dual-objective optimization problem as following.

$$
\begin{aligned}
obj: \quad & (1)\,(2) \\
\text{s.t.} \quad & \sum\nolimits_{i\in N} x_i^t = s^t, \sum\nolimits_t s^t = n, && \forall t \in [1,T] && (3)\\
& \sum\nolimits_{i\in N} y_i^t = r^t, \sum\nolimits_t r^t = m, && \forall t \in [1,T] && (4)\\
& \sum\nolimits_{j\in N} z_{ij}^t \le c * y_i^t, && \forall i, \forall t \in [1,T] && (5)\\
& \sum\nolimits_{i\in N} z_{ij}^t = x_j^t, && \forall j,\ \forall t \in [1,T] && (6)\\
& y_i^t \le x_i^t, && \forall i,\ \forall t \in [1,T] && (7)\\
& x_i^{t-1} \le x_i^t && \forall t \in [2,T] && (8)\\
& y_i^{t-1} \le y_i^t && \forall t \in [2,T] && (9)\\
& x_i^t, y_i^t, z_{ij}^t \in \{0,1\}, && i,j \in N && (P_1)
\end{aligned}
$$

Constraint (3) and (4) show that s^t switches are upgraded and r^t controllers are placed in step t, and finally, all n SDN switches and m controllers are placed. $y_i^t = 1$ denotes that a controller is deployed at location i in step t, otherwise, $y_i^t = 0$. (5) means the number of switches connected to the controller should not exceed its processing ability c, and only controller can control SDN switches. (6) shows SDN switch must be controlled by exactly one controller. (7) indicates only the location of SDN switch can be the controller candidate position. Constraint (8) means each switch can only be upgraded once. Constraint (9) means the controller cannot be removed during the upgrade process. Besides, the mapping between switch and controller can change at any time, so we don't constrain that. x_i^t, y_i^t, z_{ij}^t are 0-1 decision variables.

TABLE I
NOTATIONS

Notation	Meaning
n	the number of switches
m	the number of controllers
T	the number of upgrade steps
c	the processing capacity of a controller
N	the set of nodes
i	the name of a switch, $i \in N$
j	the name of a controller, $j \in N$
t	the index of the upgrade step, $t \in [1,T]$
d_i	the degree of node i
q_i	the non-global optimal factor
w_{ij}	the delay between nodes i and j
s^t	the number of upgraded switches in step t
r^t	the number of placed controllers in step t
γ^t	the penalty factor in step t
β^t	the fade factor in step t
η	the preference factor of degree and delay

V. Model Transformation

A. Transformed to Single-objective

Model P_1 is difficult to solve. To simplify the solution, we converted the entire upgrade process into a series of programming problems by fixing the step t and using constraints (8)(9) to connect each programming problem. Then we introduce a preference factor η to convert the dual-objective optimization problem into a single-objective upgrade problem P_1^*.

$$
\begin{aligned}
\max \quad & \sum_{i\in N} \beta^t * d_i * x_i^t - \eta \sum_{i,j\in N} w_{ij} * z_{ij}^t \\
\text{s.t.} \quad & (3)(4)(5)(6)(7)(8)(9) \\
& x_i^t, y_i^t, z_{ij}^t \in \{0,1\},\ i,j \in N \qquad (P_1^*)
\end{aligned}
$$

The preference factor η is a balance of degree gain and delay. The equivalence between dual-objective and transformed single-objective has been demonstrated in [13].

The constraints (8)(9) only specify that the latter step is bound by the preceding steps. However, the intermediate steps are also infected by the final step. The controllers deployed in any intermediate step will exist in the final step, affecting the performance of the final pure SDN. So, problem P_1^* cannot fully represent problem P_1 unless this reversed influence is modeled. We introduce a penalty item to take the influence of the final pure SDN into consideration during the middle steps, and form the model P_1^{**}. The penalty item punishes the non-global optimal controller positions, forcing the middle steps to select the global optimal positions for controller deployment, and reducing the delay of pure SDN.

$$
\begin{aligned}
\max \quad & \sum_{i\in N} \beta^t * d_i * x_i^t - \eta \sum_{i,j\in N} w_{ij} * z_{ij}^t - \sum_{i\in N} \gamma^t * q_i * y_i^t \\
\text{s.t.} \quad & (3)(4)(5)(6)(7)(8)(9) \\
& x_i^t, y_i^t, z_{ij}^t \in \{0,1\},\ i,j \in N \qquad (P_1^{**})
\end{aligned}
$$

In the objective of model P_1^{**}, the third part is the penalty, in which q_i means whether node i is the global optimal location of controller, called non-global optimal factor. If the controller is placed in the non-global optimal position, $q_i = 1$, $y_i^t = 1$, the placement is punished. The calculation of q_i is in V-B.

The penalty factor γ^t is introduced to obtain the trade-off between final benefits and local benefits. The penalty decreases with the upgrade progressing since the controllers deployed in the former step run longer than those in the latter step; it is more desirable that those global optimal controllers are deployed in the former steps. We define γ^t as the ratio of the existence time from step t to the existence time of the whole network, calculated by (10). In (10), t_{SDN} is the lifetime of the SDN network, t_{block} is the time interval of two upgrade steps, and $cost$ is a base penalty.

$$
\gamma^t = cost * \frac{t_{SDN} + t_{block} * (T-t)}{t_{SDN} + t_{block} * (T-1)} \tag{10}
$$

B. Calculation of Non-global Optimal Factor

The non-global optimal factor q_i can be obtained by solving the optimal position of controllers in pure SDN network [12]. Consistent with model P, the decision variable y_i shows whether to place a controller at location i, z_{ij} shows the mapping between controller i and switch j. The optimization model is as following, and the goal is to minimize the delay.

$$\begin{aligned}
\min \quad & \sum_{i,j \in N} w_{ij} * z_{ij} \\
\text{s.t.} \quad & \sum_{i \in N} y_i = m, \qquad \forall i \qquad (11) \\
& \sum_{j \in N} z_{ij} < c * y_i, \qquad \forall i \qquad (12) \\
& \sum_{i \in N} z_{ij} = 1, \qquad \forall j \qquad (13) \\
& y_i, z_{ij} \in \{0,1\}, \qquad i,j \in N \qquad (P_2)
\end{aligned}$$

Constraint (11) stipulates that there are exactly m controllers placed. Constraint (12) stipulates the switches controlled by each controller cannot exceed its capacity c, and only controller can control SDN switches. Constraint (13) specifies each switch is exactly controlled by one controller.

By solving P_2, we get the optimal y_i, *i.e.*, the optimal locations for controllers. If $y_i = 1$, node i is the global optimal location for controller, so $q_i = 0$. Thus, q_i can be obtained by (14).

$$q_i = 1 - y_i \tag{14}$$

VI. EVALUATION

A. Simulation Setup

We evaluate our proposed method using a real network topology, Atmnet, from Topology Zoo [25], shown in Fig. 4, which contains 21 nodes and 22 edges. We specify there are 3 upgrade steps, the upgrade duration would be ignored, while the span between two steps is 1 year, and the SDN network can maintain 10 years. At each step, 7 switches and 1 controller can be upgraded and the controllers' capacity is 10.

Fig. 4. Atmnet topology. ((a) with map; (b) without map)

B. Implemented Methods

a) Naive upgrade method (Naive deployment): First select the switches in degree descending order, then deploy the controllers in the sub SDN network to minimize control delay.

b) Joint Switch Upgrade and Controller Deployment (Joint deployment): To maximize the degree gain and minimize the control delay simultaneously in each step [13].

c) Global deployment: To maximize the degree gain, minimize control delay, and minimize the penalty simultaneously in each step. This is our proposed method by using CPLEX to solve the model P_1^{**}.

C. Upgrade Results

The final optimal controllers are node I (New York), node K (Los Angeles) and node M (St Louis) by solving the problem P_2. The upgrade results are shown in Fig. 5.

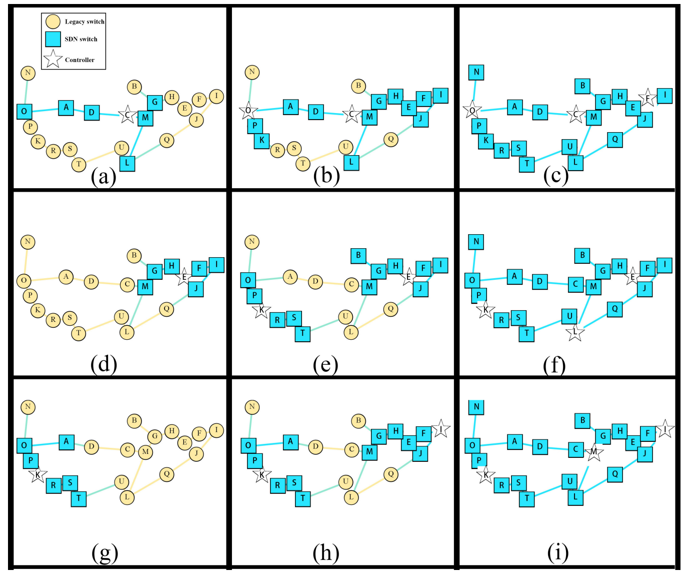


Fig. 5. Upgrade results of the three upgrade methods. ((a), (b), (c) are the results after three upgrade steps in the Naive deployment; (d) (e) (f) are the upgrade results after three steps in the Joint deployment; (g), (h), (i) are three upgraded networks of the Global deployment.)

D. Performance Analysis

We choose degree gain and time loss to evaluate network programmability benefit and delay performance, shown in (15) and (16), respectively.

$$degree\_gain = \sum\nolimits_{i \in N} d_i * x_i^t \tag{15}$$

$$time\_loss = \sum\nolimits_{i,j \in N} w_{ij} * z_{ij}^t \tag{16}$$

These indicators reflect the performance of each step. Further, we need another metric to represent the overall benefits and loss. The contribution of each upgrade step can be measured by its lifetime, so we do time integration for the entire upgrade process. The calculation method is to multiply the *gain* (or the *loss*) brought by each upgrade by the duration of the upgrade, and then integrate, shown in (17) and (18).

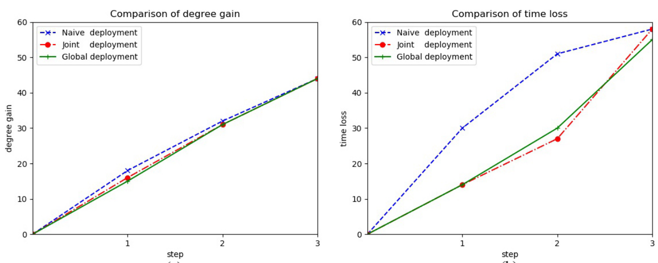
$$total\_degree\_gain = \int_1^T gain = \sum\nolimits_{t=1}^T degree\_gain \tag{17}$$

$$total\_time\_loss = \int_1^T loss = \sum\nolimits_{t=1}^T time\_loss \tag{18}$$

Computational results are shown in TABLE II and Fig. 6. Combine Fig. 6, the Global deployment model gets the best performance at the final step, while intermediate performance may not be the best. But the effect of the last step is the most important because the final operation effect lasts longer than the middle construction process.

TABLE II
DEGREE GAIN AND TIME LOSS COMPARISON

t	Degree gain			Time loss		
	Naive	**Joint**	**Global**	**Naive**	**Joint**	**Global**
1	18	16	15	30	14	14
2	32	31	31	51	27	30
3	44	44	44	58	58	55
total	**402**	**399**	**398**	**545**	**505**	**484**



(a) (b)

Fig. 6. The comparison in each step. ((a) degree gain; (b) time loss)

To address the inconformity of units of degree gain and time loss, (19) and (20) are used to calculate the degree sacrifice and the delay reduction between method a and b. They show how much performance method a has improved over method b.

$$degree\_sacrifice = \frac{degree\_gain_a - degree\_gain_b}{degree\_gain_b} \quad (19)$$

$$delay\_reduction = \frac{time\_loss_a - time\_loss_b}{time\_loss_b} \quad (20)$$

The Global method uses 0.2% of the total degree sacrifice in exchange for a 4% total delay reduction. Compared with the Joint deployment, our method can sacrifice minor total degree gain in exchange for more total delay reduction, and get a tradeoff between them.

VII. Conclusion

In this paper, we studied the SDN migration problem from both dynamic and global points of view. We not only aim at the local optimization of network state at a given time, but the global optimization for the entire migration trajectory as a whole. We formulated a time-varying dual-objective dynamic optimization problem that captures the progressive and simultaneously optimizes the local objective at each intermediate step and the global one viewing the migration as a whole. We proposed a method to solve this problem by converting it to multiple related single objective optimization problems. The simulation results based on real network showed that our method could get tradeoff between the local and global goals.

References

[1] Mobile traffic by application category. Accessed: Nov. 2019. [Online]. Available: https://www.ericsson.com/en/mobility-report/reports/november-2019/mobile-traffic-by-application-category.

[2] Z. Cao, S. Panwar, M. Kodialam and T. V. Lakshman, "Enhancing mobile networks with software defined networking and cloud computing," in IEEE/ACM Transactions on Networking, vol. 25, no. 3, pp. 1431–1444, June 2017.

[3] R. Masoudi and A. Ghaffari, "Software defined networks: a survey," in Network and Computer Applications, vol. 67, pp. 1–25, May 2016.

[4] S. Sun, M. Kadoch, L. Gong and B. Rong, "Integrating network function virtualization with SDR and SDN for 4G/5G networks," in IEEE Network, vol. 29, no. 3, pp. 54–59, May–June 2015.

[5] S. Jain et al., "B4: Experience with a globally-deployed software defined WAN," ACM SIGCOMM 2013, pp. 3–14.

[6] As Providers Push NFV/SDN, 3 Key Issues Remain. Accessed: Apr. 23, 2017. [Online]. Available:https://www.rcrwireless.com/20170423/opinion/readerforum/reader-forum-nfv-sdn-issues-tag10.

[7] Sandhya, Y. Sinha, K. Haribabu, "A survey: hybrid SDN," in Network and Computer Applications, vol. 100, pp. 35–55, December 2017.

[8] M. Caria, A. Jukan and M. Hoffmann, "A performance study of network migration to SDN-enabled traffic engineering," IEEE GLOBECOM 2013, pp. 1391–1396.

[9] T. Das, M. Caria, A. Jukan and M. Hoffmann, "Insights on SDN migration trajectory," IEEE ICC 2015, pp. 5348–5353.

[10] K. Poularakis, G. Iosifidis, G. Smaragdakis and L. Tassiulas, "Optimizing gradual SDN upgrades in ISP networks," in IEEE/ACM Transactions on Networking, vol. 27, no. 1, pp. 288–301, February 2019.

[11] B. Killi and S. Rao, "Controller placement in software defined networks: A comprehensive survey," in Computer Networks, vol. 163, no. 106883, pp. 1–16, November 2019.

[12] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in Acm Sigcomm Computer Communication Review, vol. 42, no. 4, pp.473–478, September 2012.

[13] Z. Guo, W. Chen, Y. Liu, Y. Xu and Z. Zhang, "Joint switch upgrade and controller deployment in hybrid Software-Defined Networks," in IEEE Journal on Selected Areas in Communications, vol. 37, no. 5, pp. 1012–1028, May 2019.

[14] Barakat, L. Osamah, T. Emadinia, D. Koll and X. Fu. "Paving the way towards enterprise SDN adoption - new selection strategies for hybrid networks," ICIN 2019, pp. 322–328.

[15] D. Levin, M. Canini, S. Schmid and A. Feldmann, "Panopticon: Reaping the benefits of partial SDN deployment in enterprise networks," USENIX ATC 2014, pp. 333–346.

[16] D. K. Hong, Y. Ma, S. Banerjee and Z. M. Mao, "Incremental deployment of SDN in hybrid enterprise and ISP Networks," ACM SOSR 2016, pp. 1–7.

[17] M. Tanha, D. Sajjadi, R. Ruby and J. Pan, "Traffic engineering enhancement by progressive migration to SDN," in IEEE Communications Letters, vol. 22, no. 3, pp. 438–441, March 2018.

[18] Y. Guo, Z. Wang, X. Yin, X. Shi, J. Wu and H. Zhang, "Incremental deployment for traffic engineering in hybrid SDN network," IEEE IPCCC 2015, pp. 1–8.

[19] K. Poularakis, G. Iosifidis, G. Smaragdakis and L. Tassiulas, "One step at a time: Optimizing SDN upgrades in ISP networks," IEEE INFOCOM 2017, pp. 1–9.

[20] A. Kumari and A. S. Sairam, "A Survey of controller placement problem in software defined networks," ArXiv : 1905.04649, 2019.

[21] V. Huang, C. Chen, Q. Fu and E. Wen, "Optimizing controller placement for Software-Defined Networks," ArXiv : 1902.09451, 2019.

[22] C. Chu, K. Xi, M. Luo and H. J. Chao, "Congestion-aware single link failure recovery in hybrid SDN networks," IEEE INFOCOM 2015, pp. 1086–1094.

[23] X. Jia, Y. Jiang, Z. Guo, G. Shen, and L. Wang, "Intelligent path control for energy-saving in hybrid SDN networks," in Computer networks, vol. 131, pp. 65–76, February 2018.

[24] T. Yuan, X. Huang, M. Ma and P. Zhang, "Migration to software-defined networks: The customers' view," in China Communications, vol. 14, no. 10, pp. 1–11, October 2017.

[25] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden and M. Roughan, "The internet topology zoo," in IEEE Journal on Selected Areas in Communications, vol. 29, no. 9, pp. 1765–1775, October 2011.