

# Dynamic Controller/Switch Mapping in Virtual Networks Service Chains

Chuan Pham\*, Duong Tuan Nguyen\*, Nguyen H. Tran<sup>†</sup>, Kim Khoa Nguyen\*, Mohamed Cheriet\*

\*Synchromedia - École de Technologie Supérieure, Université du Québec, H3C1K3, Canada,

<sup>†</sup> School of Information Technologies, The University of Sydney, NSW, Australia.

**Abstract**—Accelerated Software Defined Networking (SDN) adoption makes SDN paradigm emerging in the state of the art. Especially, the combination of SDN and network functions virtualization (NFV) becomes a promising trend in deploying virtual network services for all network operators. Although SDN can decouple networks into the control plane and the data plane to obtain flexible operation and programmability, there are many open issues that need to be addressed for SDN deployments, such as i) where to place SDN controllers in a given network, ii) how to assign connections from controllers to switches in terms of satisfying multiple objectives (e.g., resource utilization, failure, quality of services). In this work, we focus on the efficient assignment between SDN controllers and switches to guarantee resource efficiency, quality of services, and fault tolerance, in which the complexity of virtual links in network services, an omitted factor in current works, is considered and addressed. We formulate an optimization problem for dynamic controller/switch mapping (DCSM) in the network virtualization. We then proposed approximation algorithms in terms of relaxing the binary variables to solve the NP-hard problem, DCSM, in both centralized and distributed mechanisms. We also create various simulation schemes to evaluate our methods where they outperform state-of-the-art methods.

## I. INTRODUCTION

Recently, several cloud/datacenters have updated their platform to support software-defined networking (SDN) [1], [2], a promising approach that can make datacenters infrastructure more elastic and efficient in control and management. Datacenter networks now can be operated under the adoption of several technologies of virtual networks based on SDN, such as VLAN, vCDNI, virtual machine network [1]. Especially, the combination of SDN and network functions virtualization (NFV) is considered as a novel network architecture for datacenters in order to meet requirements for dynamic, scalable, flexible and efficient implementations of applications and network services [2], [3]. SDN-based architectures can decouple the network control and the data planes underlying proper SDN controllers to manage, control and route service flows in networks. Hence, there is increasing interest to deploy SDN controllers in datacenters network to address joint multiple objectives, such as the quality of services (QoS), failure, and scalability requirements.

In NFV networks, SDN controllers define rules to manage and route flows of virtual networks through switches [2]. Although switches in cloud/datacenter networks can link together via physical links (i.e., links connect among switches), flows cannot traverse through physical links properly from the source node to the destination node without a controller [2]. Different SDN controller

assignments result in different operational costs in the network that are related to flow arrival rates, aggregate traffic, fault tolerant and QoS. A *static assignment* between switches and controllers is not suitable for the dynamic architecture of SDN and NFV network due to following drawbacks. First, it is difficult for the control plane to adapt to traffic load fluctuation in a static assignment since it may result in wasting network resource due to over provisioning or a cascading failure once a controller, which is keeping control a lot of network flows is failed [4], [5], [6]. Second, the network operator needs efficient solutions to map connections between controllers and switches instead of manual configurations of static methods. Furthermore, each controller has a specific configuration to manage an amount of requests from switches, which is not considered in the state-of-the-art of the single controller placement [7], which mainly focuses on the distance between controller-switch [7]–[9]. A dynamic mapping between controllers and switches when traffic varies is necessary to achieve an efficient deployment in datacenters networks.

In this paper, we consider a datacenter network, where service components are deployed as service chains [3]. Each service chain contains a list of virtual network functions (VNFs) in a specific order [3], which are in turn connected with each other via virtual or physical switches. An example of a controller-switch assignment based on the shortest distance from a switch to a controller is shown in Fig. 1. There are three service chains that are placed into the different nodes (e.g., each service chain is represented by VNFs that have the same color). Those service chains are connected through the switches (e.g., SW1, SW2 and SW3). To control the flows of those service chains (represented by the dotted line between VNFs), the network operator needs to deploy a set of controllers that directly control and manage routing information of such flows. With a static assignment based on the shortest distance from switches to controllers, the connections are assigned as shown in Fig. 1. The example demonstrates an inefficient assignment that not only results in resource waste, but also adversely affects service performance in terms of service delay. Without considering the resource utilization, there is a large number of flows from VNFs to controller 2 that leads to the overload at controller 2 (represented by red color), whereas controller 1 and 3 are only responsible for one or no service chain. An assignment that creates too many connections to one controller while others are idle is not a desirable strategy in SDN

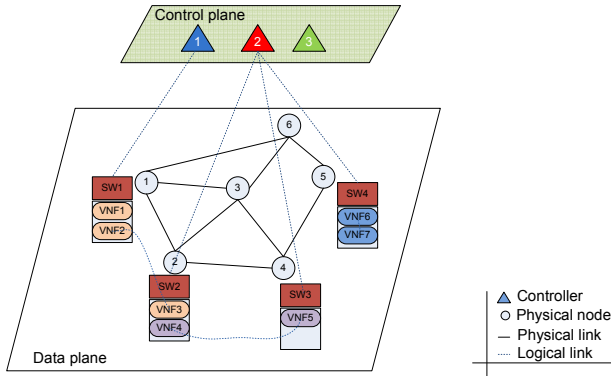


Fig. 1: Controller assignment in the SDN network.

networks. Such an assignment leads to lost balance in not only resource utilization, but also network delay. Services managed by controller 1 are suffered high processing delay because the processing time of the controller depends on the number of incoming service requests and the configuration of the controller. To say the least, if a failure happens at a controller that is holding most of routing policies of networks, it results in the worst case latency. As illustrated via the example, our goal is to find an efficient method to assign connections between controllers and switches regarding the overhead of *delay, failure, and energy consumption* in the control plane given predefined service chains and traffic flows routed via switches.

To address aforementioned issues, we formulate the dynamic controller/switch mapping (DCSM) problem as an optimization problem that dynamically assigns connections between controllers and switches. And we show that DCSM is an NP-hard problem that cannot be solved in polynomial time. In this work, we study a relaxation method to transform DCSM into a tractable approximation problem. Another factor that distinguishes our work from the rest is that we take into account the relaxation of the binary variable of DCSM. This aspect is often omitted or not addressed well in literature, even though there is a big gap between the relaxation result and the optimal solution. Based on the Penalty Successive Upper Bound Minimization (PSUM) method [10], we then propose a centralized approximation algorithm to find the assignment solution. In order to reduce the complexity of computation, we apply the decomposition framework [11] that can solve the relaxation problem DCSM in a distributed manner. We also implement different simulation schemes when evaluating our approaches. Comparing with state-of-the-art methods, our proposed mechanisms outperform others in terms of reducing the optimality gap and the total operational cost.

The rest of this paper is organized as follows. In Section II, we introduce the system model and the problem formulation. Section III presents the designed algorithms in terms of centralized and distributed manners. We make the simulation to evaluate the system in Section IV. Finally, we conclude the paper in Section V.

## II. PROBLEM FORMULATION

In this work, we consider a system model of a cloud network provider, who is deploying a set of given network service chains. Those network service chains are hosted on VNF nodes. Each VNF node can deploy multiple VNFs and the virtual links between VNFs are embedded into physical links. Flows among VNFs defined by controllers are routed through switches. To manage those service chains, the network provider needs to carry out a set of controllers and assign connections between controllers and switches. We formulate the SDN controller assignment problem and the optimization of resource allocation (e.g., CPU, memory, and storage) for handling network service chains.

### A. Constraints and the objective function

We consider a service provider that has to assign connections from a set  $\mathcal{C}$  of controllers to a set  $\mathcal{V}$  of switches. Such an assignment is used to manage flows of a set  $\mathcal{S}$  of service chains. A service chain  $s \in \mathcal{S}$  includes an ordered list of VNFs. We denote  $\mathcal{N}$  as the set of VNFs. To present the connection between VNF  $n \in \mathcal{N}$  and  $n' \in \mathcal{N}$  in the service chain  $s \in \mathcal{S}$ , we denote  $r_{nn'}^s$  as the traffic rate between them. A virtual flow (i.e., a flow connects two VNFs in a service chain) between VNF  $n$  and  $n'$  is handled by switches that VNF  $n$  and  $n'$  are directly connected. Specifically, if VNF  $n \in \mathcal{N}$  connects to switch  $v \in \mathcal{V}$  and VNF  $n' \in \mathcal{N}$  connects to switch  $v' \in \mathcal{V}$ , the virtual flow between VNFs  $n$  and  $n'$  is traversed on the physical link connected by switches  $v$  and  $v'$ . For example, the flow from VNF2 to VNF3 in Fig. 1 is routed based on the physical link from SW1 to SW2. We use  $a_{n,v}^s$  as an indicator to express a connection between VNF  $n$  of service chain  $s$  and switch  $v$  if  $a_{n,v}^s = 1$ , otherwise  $a_{n,v}^s = 0$ .

In order to manage service chains, a switch needs to keep routing policies that are frequently updated by its controller. Depending on the number of service chains that a switch manages, the updating rate from such a switch to its controller is various. For example, as shown in Fig. 1, SW 1 has to keep a routing policy to route flows of the service chain VNF1→VN2→VN3 via SW1 and SW2. This routing policy is frequently updated by controller 1. We assume that the updating rate of service chain  $s$  follows a Poisson process with a rate  $\lambda_s, \forall s \in \mathcal{S}$ . Given an VNF placement, we can calculate the updating rate of a switch  $v \in \mathcal{V}$  as follows:

$$\lambda_v = \sum_{s \in \mathcal{S}} \alpha(s, v) \lambda_s, \forall v \in \mathcal{V}, \quad (1)$$

where

$$\alpha(s, v) = \begin{cases} 1 & \text{iff } \exists a_{v,n}^s = 1, \forall n \in \mathcal{N}, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

**Delay constraints of controllers.** We consider a controller can manage multiple switches, and query requests from these switches are aggregated and served by a controller under a queueing model as shown in Fig. 2. We define a binary variable  $x_{v,c}$  to indicate a connection between switch  $v \in \mathcal{V}$  and the controller  $c \in \mathcal{C}$  if  $x_{v,c} = 1$ ,

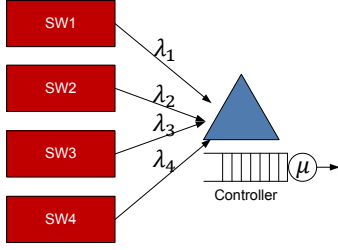


Fig. 2: Processing switch requests at a controller.

otherwise  $x_{v,c} = 0$ . Hence, total requests at the controller  $c$  is aggregated based on all requests from connected switches by

$$\Lambda_c = \sum_{v \in \mathcal{V}} \lambda_v x_{v,c}, \forall c \in \mathcal{C}. \quad (3)$$

We assume that each controller can be modeled as an M/M/1 queue, and hence we formulate the delay constraint as follows:

$$\frac{\Lambda_c}{\mu_c - \Lambda_c} \leq D_c, \forall c \in \mathcal{C}, \quad (4)$$

where  $D_c$  is the delay threshold and  $\mu_c$  is the processing capacity of the controller  $c$ .

**Failure constraints of controllers.** The failure problem in datacenters is very common [12]. Specifically, a controller failure results in the worst case latency due to reassigning switch connections. In order to reduce the risks of failure controllers in assignment, we add the failure constraint to limit the number of connections from switches to a controller. Such a constraint to prevent too many connections that may be fallen into the same failure domain

$$\sum_{v \in \mathcal{V}} x_{v,c} \leq \kappa \omega_c, \forall c \in \mathcal{C}, \quad (5)$$

where  $\kappa \in (0, 1]$  is the decay factor [4], [12] and  $\omega_c$  is the maximum connections that controller  $c$  can connect. This constraint can be considered as the reliability percentage of a controller depending on the physical condition of the node in the network.

**Operational cost.** In this paper, we consider the operational cost of an assignment between controllers and switches under two factors. *First*, we take into account the *energy efficiency* factor based on the utilization of controllers. A controller has a specific resource capacity to manage a number of NFV flows. Following [13], the energy consumption of controller  $c \in \mathcal{C}$  can be calculated based on the utility function in terms of the M/M/1 queueing model as follows  $E_c(\mathbf{x}_c) = P_c^i + (P_c^a - P_c^i) \frac{\Lambda_c}{\mu_c}, \forall c \in \mathcal{C}$ , where  $P_c^i$  is the idle power, and  $P_c^a$  is the fully utilized power of controller  $c \in \mathcal{C}$ . In particular, the formula represents the dependency of energy consumption of a controller on total amount of incoming traffic from switches that are assigned and managed by controller  $c$ .

*Second*, the operational cost is affected by the *communication cost between controllers* [14]. When controllers control the same network service chains, they need to communicate for updating routing information [9], [15]. We define a control function  $\pi : |\mathcal{S}| \times |\mathcal{C}| \rightarrow \{0, 1\}$ , which

returns 1 if controller  $c \in \mathcal{C}$  controls service chain  $s$ , and 0 otherwise. Specifically, we have

$$y_{s,c} = \begin{cases} 1 & \text{iff } \exists s \in \mathcal{S}, v \in \mathcal{V} \text{ such that} \\ & \alpha(s, v) = 1 \text{ and } x_{v,c} = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

Mathematically, we can express this function as follows:

$$y_{s,c} = 1 - \prod_{v \in \mathcal{V}_s} (1 - x_{v,c}), \forall v \in \mathcal{V}, c \in \mathcal{C}, \quad (7)$$

where  $\mathcal{V}_s$  is the set of switches containing VNFs of service vice chain  $s$ . Depending on the updating rate of service chains, we can represent the communication cost of between controller  $c$  and others in the control plane as follows

$$W_c(\mathbf{x}_c) = \sum_{c' \neq c \in \mathcal{C}} \sum_{s \in \mathcal{S}} \lambda_s y_{s,c} y_{s,c'} l_{cc'}, \forall c \in \mathcal{C}; \quad (8)$$

Hence, the operational cost of DCSM can be rewritten as follows:

$$S(\mathbf{x}) = \sum_{c \in \mathcal{C}} S_c(\mathbf{x}_c) = \sum_{c \in \mathcal{C}} \sigma E_c(\mathbf{x}_c) + (1 - \sigma) W_c(\mathbf{x}_c), \quad (9)$$

where  $\mathbf{x} = \{x_{v,c}\}_{v \in \mathcal{V}, c \in \mathcal{C}}$ , and  $\mathbf{x}_c = \{x_{v,c}\}_{v \in \mathcal{V}}, \forall c \in \mathcal{C}$  and  $\sigma$  is the weight factor in the objective function.

### B. DCSM: Dynamic controller/switch mapping problem

We formulate the optimization problem for the dynamic controller/switch mapping problem (DCSM) as follows.

$$\min \quad S(\mathbf{x}) \quad (10)$$

$$\text{s.t.} \quad \sum_{c \in \mathcal{C}} x_{v,c} = 1, \forall v \in \mathcal{V}, \quad (11)$$

$$\frac{\Lambda_c}{\mu_c - \Lambda_c} \leq D_c, \forall c \in \mathcal{C}, \quad (12)$$

$$\sum_{v \in \mathcal{V}} x_{v,c} \leq \kappa \omega_c, \forall c \in \mathcal{C}, \quad (13)$$

$$x_{v,c} \in \{0, 1\}, \forall v \in \mathcal{V}, c \in \mathcal{C}. \quad (14)$$

The equality constraint (11) ensures that a switch has a unique connection with only one controller. This assumption is reasonable since the overhead of synchronization among controllers and message passing will be very high if there are multiple connections from one switch to many controllers. Constraint (14) is used to express the binary variable in our problem. The objective function of DCSM is to minimize the operational cost including energy consumption and the communication cost.

To make the solution for DCSM feasible in practice, the challenge is to find a tractable solution to solve DCSM because of its NP-hardness. In this work, we study an approximation approach based on the PSUM method, where the binary variables are relaxed to be continuous variables. Having said that, the proposed algorithm can enforce the relaxed variables converge to the binary value at the end of loops.

## III. PROPOSED ALGORITHMS

In this section, we propose algorithms for DCSM in terms of centralized and distributed manners.

### A. R-DCSM: A centralized approximation optimization for DCSM

As aforementioned discussions, there are two major challenges to solve DCSM, including the non-convex, non-linear objective function and the binary assignment variable. We first tackle the first challenge by using an approximation to transform the objective function into a convex function.

Since  $e^{-x} \approx 1 - x$  for small  $x$  (e.g.,  $x < 1$ ), using Taylor approximation, we have  $y_{s,c} \approx 1 - e^{-\sum_{v \in \mathcal{V}_s} x_{v,c}} \approx \sum_{v \in \mathcal{V}_s} x_{v,c}, \forall s \in \mathcal{S}, c \in \mathcal{C}$  with the additional constraint  $0 \leq y_{s,c} \leq 1, \forall s \in \mathcal{S}, c \in \mathcal{C}$  to enforce that exactly one switch containing service chain  $s$  is connected to controller  $c$ . We then re-write the communication cost in the control plane incurred by sharing routing information between controllers as follows

$$W(\mathbf{x}) = \sum_{c \in \mathcal{C}} W_c(\mathbf{x}_c) = \sum_{c \in \mathcal{C}} \sum_{s \in \mathcal{S}} \lambda_s (\mathbf{y}_s^T \mathbf{L} \mathbf{y}_s), \quad (15)$$

where matrix  $\mathbf{L} = (l_{c,c'})_{c,c' \in \mathcal{C}}$ , with  $l_{c,c} = 0, \forall c \in \mathcal{C}$  and the vector  $\mathbf{y}_s = \{y_{s,c}\}_{c \in \mathcal{C}}$ . The communication cost function is formed in a quadratic and convex function if  $\mathbf{L}$  is positive semidefinite matrix.

For the second challenge, we approximately solve DCSM based on PSUM, in which  $x_{v,c} \in [0, 1], \forall v \in \mathcal{V}, \forall c \in \mathcal{C}$ .

Let  $\mathbf{x}_v = \{x_{v,c}\}_{c \in \mathcal{C}}, \forall v \in \mathcal{V}$ . The idea of relaxation is from the assignment constraint (11), where we consider the optimization for any  $v \in \mathcal{V}$  as follows

$$\begin{aligned} \min_{\mathbf{x}_v} \quad & \|\mathbf{x}_v + \xi \mathbf{1}\|_p^p \\ \text{s.t.} \quad & \|\mathbf{x}_v\|_1 = 1, \\ & x_{v,c} \in [0, 1], \forall c \in \mathcal{C}, \end{aligned} \quad (16)$$

where  $p \in (0, 1)$  and  $\xi$  is a nonnegative constant.

The interesting result of (16) is that this optimization problem guarantees a binary optimal solution with only one element is 1, and all others are 0. With that optimal solution, the optimal value of (16) is [10]:

$$\psi_{\xi,v} = (1 + \xi)^p + (|\mathcal{C}| - 1)\xi^p, \forall v \in \mathcal{V}. \quad (17)$$

Therefore, we add a penalty term to the objective function of DCSM when relaxing the binary constraint (14) as follows

$$\min_{\mathbf{x}} S(\mathbf{x}) + \delta P_{\xi}(\mathbf{x}), \quad (18)$$

where  $\delta > 0$  is the penalty parameter, and

$$P_{\xi}(\mathbf{x}) = \sum_{v \in \mathcal{V}} \sum_{c \in \mathcal{C}} (\|x_{v,c} + \xi \mathbf{1}\|_p^p - \psi_{\xi,v}). \quad (19)$$

Based on the fact that the problem (16) reaches to the optimal solution  $\psi_{\xi,v}$  with a binary solution, the designed of the penalty term is used to force (18) to a binary solution that makes this penalty term approaches to zero

---

### Algorithm 1 R-DCSM: A centralized algorithm for DCSM

---

- 1: Initialize  $\xi, \eta$ , and  $\gamma$  satisfying  $0 < \gamma < 1 < \eta$ ;
  - 2: Solve DCSM by relaxing (14) to obtain  $\mathbf{x}^{(0)}$ ;
  - 3: Initialize  $k = 0, \delta^{(0)}$  and  $\xi^{(0)}$ ;
  - 4: **repeat**
  - 5:     Solve (24) with the initial value  $\mathbf{x}^{(k)}$  to obtain  $\mathbf{x}^{(k+1)}$ ;
  - 6:     **if**  $\mathbf{x}^{(k+1)}$  is not binary **then**
  - 7:         Update  $\delta^{(k+1)} = \gamma \delta^{(k)}$  and  $\xi^{(k+1)} = \eta \xi^{(k)}$ ;
  - 8:     **end if**
  - 9:      $k \leftarrow k + 1$ ;
  - 10: **until**  $\mathbf{x}^{(k+1)}$  is binary.
- 

with any positive  $\delta$ . We re-formulate DCSM to the new approximation problem (named R-DCSM):

$$\text{R-DCSM: } \min \quad S(\mathbf{x}) + \delta P_{\xi}(\mathbf{x}) \quad (20)$$

$$\text{s.t.} \quad (11) - (13),$$

$$y_{s,c} = \sum_{v \in \mathcal{V}_s} x_{v,c}, \forall s \in \mathcal{S}, c \in \mathcal{C}, \quad (21)$$

$$0 \leq y_{s,c} \leq 1, \forall s \in \mathcal{S}, c \in \mathcal{C}, \quad (22)$$

$$0 \leq x_{v,c} \leq 1, \forall v \in \mathcal{V}, c \in \mathcal{C}. \quad (23)$$

The new problem R-DCSM formed in a convex problem is more tractable compared to the original problem. Thus, we apply the successive upper bound minimization (SUM) method to solve R-DCSM in an iterative manner [16] that can find an optimal solution for R-DCSM. Due to the concavity of  $P_{\xi}(\mathbf{x})$  [10], the first order approximation of  $P_{\xi}(\mathbf{x})$  follows an upper bound trend  $P_{\xi}(\mathbf{x}) \leq P_{\xi}(\mathbf{x}^{(k)}) + \nabla P_{\xi}(\mathbf{x}^{(k)})^T (\mathbf{x} - \mathbf{x}^{(k)})$ , where  $k$  is the index of iterations. Therefore, at the  $(k+1)$ <sup>th</sup> iteration of solving the optimization R-DCSM, the problem in the  $(k+1)$ <sup>th</sup> iteration can be rewritten as follows

$$\min \quad S(\mathbf{x}) + \delta^{(k+1)} \nabla P_{\xi}(\mathbf{x}^{(k)})^T \mathbf{x} \quad (24)$$

$$\text{s.t.} \quad (11) - (13), (21) - (23).$$

With the above discussion, we propose the approximation centralized algorithm (named R-DCSM). The steps of the R-DCSM are represented in Algorithm 1. In particular, we firstly solve DCSM with relaxed binary constraint (14) to find the initial value  $\mathbf{x}^{(0)}$ . We next iteratively solve (24) and update  $\delta^{(k+1)} = \gamma \delta^{(k)}$  until  $\mathbf{x}^{(k+1)}$  is binary, where  $\gamma$  is the predefined constant [10]. To enhance the convergence, we can use a dynamic update the step size  $\xi$  by  $\xi^{(k+1)} = \eta \xi^{(k)}$ , where  $\eta$  follows the constraint  $0 < \gamma < 1 < \eta$  [10].

### B. D-DCSM: A distributed algorithm for DCSM

With respect to DCSM, we recognize that minimizing  $S(\mathbf{x})$  is equivalent to minimizing all  $S_c(\mathbf{x}_c)$ , where  $S_c(\cdot)$  is the cost function of controller  $c \in \mathcal{C}$ . Based on the intuition of the Bender decomposition framework [11], we can solve R-DCSM in a distributed manner as follows. Suppose we obtain a real optimal solution  $\bar{\mathbf{x}}^*$  by solving DCSM with the relaxation constraint (14). Each controller can optimize its operational cost if it relaxes the coupling

---

**Algorithm 2** D-DCSM: A distributed algorithm for DCSM

---

- 1: Initialize  $\eta, \epsilon, \varphi^{(0)}$  and  $\gamma$  satisfying  $0 < \gamma < 1 < \eta$ ;
  - 2: Solve DCSM by relaxing (14) to obtain  $\mathbf{x}^{(0)}$ ;
  - 3: Set  $t = 0$ ;
  - 4: **repeat**
  - 5:     **for each**  $c \in \mathcal{C}$  **do**
  - 6:         Initialize  $k = 0, \delta^{(0)}, \xi^{(0)}$  and  $\mathbf{x}_c^{(0)}$  by  $\mathbf{x}^{(0)}$ ;
  - 7:         **repeat**
  - 8:             Solve (26) with the initial value  $\mathbf{x}_c^{(k)}$  to obtain  $\mathbf{x}_c^{(k+1)}$ ;
  - 9:             **if**  $\mathbf{x}_c^{(k+1)}$  is not binary **then**
  - 10:                 Update  $\delta^{(k+1)} = \gamma\delta^{(k)}$  and  $\xi^{(k+1)} = \eta\xi^{(k)}$ ;
  - 11:             **end if**
  - 12:              $k \leftarrow k + 1$ ;
  - 13:             **until**  $\mathbf{x}_c^{(k+1)}$  is binary;
  - 14:         **end for**
  - 15:         Set  $\mathbf{x}_c^{(t+1)} \leftarrow \mathbf{x}_c^{(k+1)}$ ;
  - 16:         Collect  $\mathbf{x}_c^{(t+1)}$  from all  $c \in \mathcal{C}$ ;
  - 17:         Update the dual variable  $\varphi^{(t+1)}$  by (27);
  - 18:          $t \leftarrow t + 1$ ;
  - 19: **until**  $\|\varphi^{(t+1)} - \varphi^{(t)}\|_2 \leq \epsilon$
- 

constraint (11). Thus, the partial Lagrangian function can be calculated as follows:

$$\mathcal{L}(\mathbf{x}) = \sum_{c \in \mathcal{C}} S_c(\mathbf{x}_c) + \delta P_\xi(\mathbf{x}_c) + \sum_{v \in \mathcal{V}} \varphi_v \left( \sum_{c \in \mathcal{C}} x_{v,c} - 1 \right), \quad (25)$$

where  $\varphi = \{\varphi_v\}_{v \in \mathcal{V}}$  is the dual variable. The main problem R-DCSM is now decomposed into  $|\mathcal{C}|$  sub-problems that can be iteratively solved at each controller. Hence, the sub-problem of each controller  $c \in \mathcal{C}$  that needs to be solved at each iteration  $t$  can be rewritten again as follows:

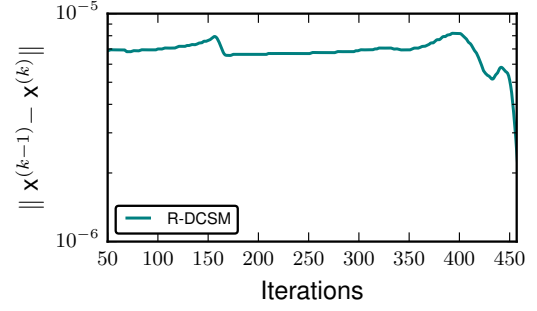
$$\begin{aligned} \min \quad & \mathcal{L}_c(\mathbf{x}_c) = S_c(\mathbf{x}_c) + \delta P_\xi(\mathbf{x}_c) + \sum_{v \in \mathcal{V}} \varphi_v^{(t)} x_{v,c}, \quad (26) \\ \text{s.t.} \quad & (11) - (13), (21) - (23). \end{aligned}$$

To tackle the coupling constraint (11), the dual variable  $\varphi$  is used as the control variable to adjust  $\mathbf{x}_c$ . The dual update is calculated as follows [11]:

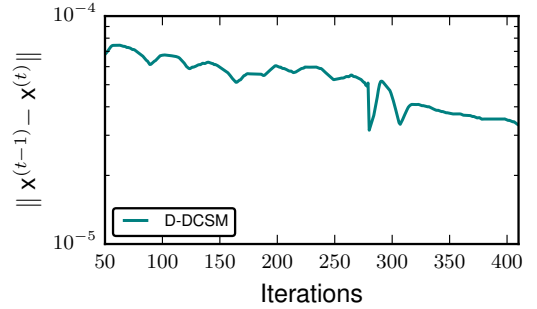
$$\varphi_v^{(t+1)} = \varphi_v^{(t)} + \beta \sum_{c \in \mathcal{C}} (x_{v,c}^{(t+1)} - 1), \forall v \in \mathcal{V}, \quad (27)$$

where  $\beta$  is the step size [11].

We next present a distributed algorithm (named D-DCSM), which is shown in Algorithm 2. The algorithm is operated with an iterative mechanism, in which the computation is decomposed to be executed in a loop at controllers according to adjustments of the control variable (i.e., the dual variable). At the first step, we find the initial value by solving DCSM with relaxing (14) (Line 2). Then, each controller solves (26) to get its optimal solution (Lines 4-14). This step requires an inner loop to find a binary solution with the similar steps to R-DCSM. All the results of controllers are collected to update the dual variable for the next step (Lines 15-18). The result of the dual variable  $\varphi_v^{(t+1)}$  is used to calculate again in



(a) R-DCSM.



(b) D-DCSM.

Fig. 3: Convergence evaluation.

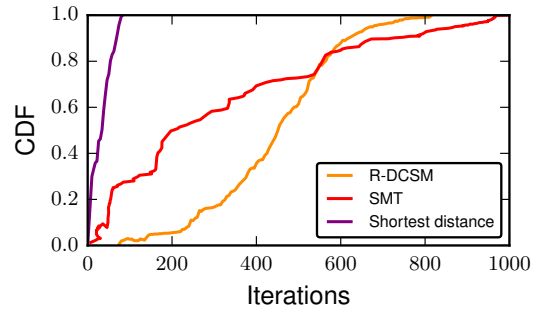


Fig. 4: Evaluation of the total cost.

the next iteration at all controllers. The algorithm will stop when it meets the critical condition.

#### IV. NUMERICAL RESULTS

In this section, we present the numerical results with various evaluation schemes. Specifically, we focus on quantifying the convergence result and the operational cost of our proposed mechanisms compared to the state-of-the-art methods.

##### A. Settings

**Network settings.** We create a physical network as a graph with 30 nodes with 50 physical links to connect physical nodes. Based on the given physical network, we generate an NFV network, in which switches, VNFs and controllers are placed on physical nodes. Each physical node has a resource capacity to host from 3 to 5 VNFs. To measure the scalability of our proposed methods, we conduct with the number of VNFs scaling from 30 to 110. An VNF placement is created randomly along the

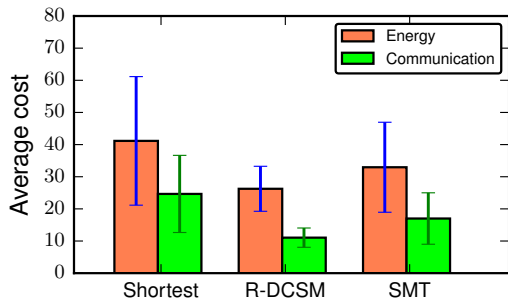


Fig. 5: Evaluation of energy and communication costs.

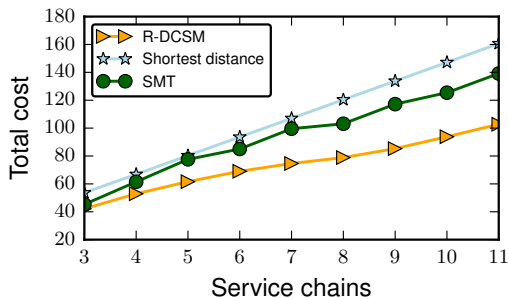


Fig. 6: Comparison of the total cost.

physical network, where virtual links between VNFs are created over physical links. We set the cost  $l_{cc'}$  is set from 0.1 to 0.5. Those settings are changeable depending on the network topology without loss of generality. For example, the cost between controllers can be set based on the number of hops between them.

**Optimization settings.** For the optimization, we set  $\sigma = 0.14$  to emphasize the communication cost. The active energy consumption of a controller is set by  $P_c^i = 50$  kWh and  $P_c^a = 100$  kWh. Due to the lack of access to the real data of the SDN controller in datacenters, we set the energy of a controller according to the average power of a server and the average number of controllers in a server. the service rates of controllers are set from 100 to 200 (i.e., the processing capacity is from 100 to 200 requests per second). Also, the arrival rate service chains  $\lambda_s$  is created randomly in range from 20 to 40. For other settings, we initialize  $\xi = 0.003$ ,  $\delta = 2$ ,  $\gamma = 0.7$ ,  $\eta = 1.1$ ,  $\varphi = 0.3$ , and  $\phi = 0.2$ .

## B. Results

To analyze the performance of R-DCSM and D-DCSM, we carry out evaluation schemes and compare them to state-of-the-art methods, including SMT [4] and the closest distance method (i.e., assigning connection from a switch to the closest controller) mentioned in [17].

**Convergence.** To evaluate the convergence of R-DCSM and D-DCSM, we measure the norm value of  $\|\mathbf{x}^{(k-1)} - \mathbf{x}^{(k)}\|$  corresponding to R-DCSM and  $\|\mathbf{x}^{(t-1)} - \mathbf{x}^{(t)}\|$  corresponding to D-DCSM. The convergence evaluation is shown in Fig. 3. With the centralized approach, R-DCSM can converge better than D-DCSM does. Furthermore, we compare the convergence with prior works as shown in Fig. 4. This figure shows the result in case of the

biggest setting (i.e., the number of VNFs is scaling up to 110). With the simple algorithm, the shortest distance completes the calculation after 78 iterations. To obtain the optimal solution for assigning all connections, R-DCSM needs more than 800 iterations. Meanwhile, SMT converges around 1000 iterations (due to the looping of the coalition games), even though it can not guarantee an optimal solution (the optimal result will be shown later). SMT can achieve the good convergence when the number of VNFs is not huge. The complexity of the coalition game makes SMT worse when enlarging the size of network. In addition, our work consider multiple objectives that also increases the complexity of DCSM compared to [4]. Such a complex consideration causes the long convergence of SMT compared to the report in [4]. With the decomposition method, D-DCSM, we cannot reduce the number of iterations compared to R-DCSM, but the complexity to solve DCSM is reduced, in which each controller only needs to solve its small optimization. Nevertheless, the network system suffers the message-passing overhead in D-DCSM.

**Incurred cost.** In order to illustrate the efficiency of our proposed methods, we make a comparison of the incurred cost between our proposed algorithms and traditional methods, including SMT and the short distance approaches. We only use the result from R-DCSM in evaluation due to the similar results of R-DCSM and D-DCSM (as mentioned in the previous evaluation). First, we show the average energy and the communication cost when applying the shortest distance, SMT and our methods. The average cost evaluation for each service chain is shown in Fig. 5. The shortest distance method only considers the distance from controllers and switches so that it optimizes neither the energy cost nor the communication cost. Meanwhile, SMT can optimize the communication cost underlying the matching game. However, it cannot guarantee an optimal solution. In average, our methods outperform others in both the energy and communication costs. We next show the gap between them in terms of total cost as depicted in Fig. 6. The shortest distance can converge very fast (as shown in Fig. 4) but its result is the worst. SMT can obtain a better result compared to the shortest distance, where it reduces the total cost up to 11.2% in average. However, SMT cannot guarantee an optimal solution. Therefore, the gap between R-DCSM increases when enlarging the number of service chains. Compared to the shortest distance method, our methods can reduce the total cost up to 31.7%.

## V. CONCLUSION

In this paper, we have presented the dynamic controller/switch mapping problem in the NFV network. We have formulated an optimization problem for coupling multiple objectives, including energy, latency and failure that have not been addressed well in the state of the art. We name the optimization as DCSM and find approximation algorithms to solve it due to its NP-hard problem. We relax the binary variable based on the PSUM method that can approximate DCSM to the linear program. We then propose two algorithms to solve DCSM named R-DCSM and D-DCSM in terms of centralized and distributed

manner. Through an extensive simulation, we evaluate the convergence and the efficiency of our model compared to existing methods. The simulation results show that our methods converge well in both centralized and distributed scenarios. The results also show the efficiency of our method that can reduce significantly the system cost compared to considered approaches.

However, our proposed approaches need a centralized solver at the first step to find the initial values. This step will increase the overhead of computation when scaling up the size of the network. We keep this point to extend our work in future that can solve DCSM in complete distributed manner and reduce the computational cost.

#### REFERENCES

- [1] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.
- [2] D. Kreutz, F. M. V. Ramos, P. E. Verssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan 2015.
- [3] J. d. J. Gil Herrera and J. F. B. Vega, "Network functions virtualization: A survey," *IEEE Latin America Transactions*, vol. 14, no. 2, pp. 983–997, Feb 2016.
- [4] T. Wang, F. Liu, J. Guo, and H. Xu, "Dynamic sdn controller assignment in data center networks: Stable matching with transfers," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, April 2016, pp. 1–9.
- [5] V. Sridharan, M. Gurusamy, and T. Truong-Huu, "On multiple controller mapping in software defined networks with resilience constraints," *IEEE Communications Letters*, vol. 21, no. 8, pp. 1763–1766, Aug 2017.
- [6] X. Ye, G. Cheng, and X. Luo, "Maximizing sdn control resource utilization via switch migration," *Computer Networks*, vol. 126, pp. 69 – 80, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1389128617302669>
- [7] G. Wang, Y. Zhao, J. Huang, and W. Wang, "The controller placement problem in software defined networking: A survey," *IEEE Network*, vol. 31, no. 5, pp. 21–27, 2017.
- [8] A. Blenk, A. Basta, J. Zerwas, M. Reisslein, and W. Kellerer, "Control plane latency with sdn network hypervisors: The cost of virtualization," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 366–380, Sept 2016.
- [9] G. Ishigaki, R. Gour, A. Yousefpour, N. Shinomiya, and J. Jue, "Cluster leader election problem for distributed controller placement in sdn," pp. 1–6, 12 2017.
- [10] N. Zhang, Y. F. Liu, H. Farmanbar, T. H. Chang, M. Hong, and Z. Q. Luo, "Network slicing for service-oriented networks under resource constraints," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2512–2521, Nov 2017.
- [11] E. C. A.J. Conejo and R. Garca-Bertrand, *Decomposition in Linear Programming: Complicating Variables*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. chapter 3, 107–139.
- [12] X. Li and C. Qian, "Traffic and failure aware vm placement for multi-tenant cloud computing," in *2015 IEEE 23rd International Symposium on Quality of Service (IWQoS)*, June 2015, pp. 41–50.
- [13] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska, "Dynamic Right-sizing for Power-proportional Data Centers," in *Proceedings IEEE INFOCOM*, Shanghai, China, apr 2011.
- [14] G. Ishigaki, R. Gour, A. Yousefpour, N. Shinomiya, and J. Jue, "Cluster leader election problem for distributed controller placement in sdn," pp. 1–6, 12 2017.
- [15] Project proposals:odl-sdni app. [Online]. Available: "https://wiki.opendaylight.org/view/Project\_Proposals:ODL-SDNi\_App"
- [16] D. R. Hunter and K. Lange, "A tutorial on mm algorithms," *The American Statistician*, vol. 58, no. 1, pp. 30–37, 2004.
- [17] S. Lange, S. Gebert, T. Zinner, P. Tran-Gia, D. Hock, M. Jarschel, and M. Hoffmann, "Heuristic approaches to the controller placement problem in large scale sdn networks," *IEEE Transactions on Network and Service Management*, vol. 12, no. 1, pp. 4–17, March 2015.