

# Optimal Network Function Virtualization Realizing End-to-End Requests

Tachun Lin\*, Zhili Zhou<sup>†</sup>, Massimo Tornatore<sup>‡</sup>, and Biswanath Mukherjee<sup>‡</sup>

\*Dept. of Computer Science and Information Systems, Bradley University, Peoria, Illinois, USA.

<sup>†</sup>IBM Research, Singapore.

<sup>‡</sup>Dept. of Computer Science, University of California, Davis, California, USA.

**Abstract**—Network function virtualization provides a new network resource utilization approach which decouples network functions from proprietary hardware and enables adaptive services to end-user requests. In this paper, we present a joint design which optimally deploys network functions and allocates physical resources satisfying end-to-end requests with generated routes. We first discuss the problem behind such design and show its NP-completeness. We then propose a mixed-integer program which simultaneously identifies physical nodes to be deployed with network functions and generates routes sharing common physical resources realizing end-to-end requests. Computational results demonstrate the value of the integrated approach and its ability to allocate network functions supporting end-to-end requests with limited physical resources.

## I. INTRODUCTION

Modern telecommunication networks are composed of a variety of proprietary interconnecting hardware, typically called middleboxes, which provides network functions such as firewall, network address translation, WAN accelerator, quality-of-service analyzers, etc. These middleboxes are deployed both singularly to provide an isolated function and, more commonly, in conjunction with other network functions [1][2]. Due to increasing demands to shorten time-to-market for new network services, scale up/down existing services, and reduce capital and operational expenditure, the concept of network functions virtualization (NFV) attracts more attention as it facilitates the cycle of network function (NF) induction, modification, upgrade, and removal. In general, NFV replaces proprietary networking hardware with services as software running on generalized commercial-off-the-shelf (COTS) equipments such as servers, switches, and storage devices. Hence, virtual network functions (VNFs) can be deployed and removed at runtime on COTS devices at NFV infrastructure's points of presence (NFVI-PoPs), including datacenters, network nodes, and end-user premises [1], to accommodate changes in traffic demands and network states [3]. Current cloud systems consist of geographically-distributed datacenters, servers hosting content/services, and a wide-area network which interconnects them and consumers [4][5]. Network virtualization techniques, which support multiple coexisting virtual networks over shared physical infrastructure and resources [6][7], allow cloud and network service providers to offer a fraction of connection, computation, and storage capacities to one or many tenants through NFV [8][9].

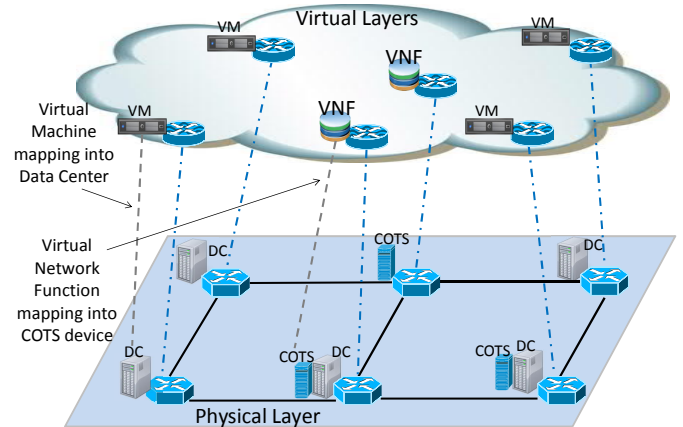


Fig. 1: Network Function Virtualization

In this paper, we study the provision and design of NFV for agile and flexible network services coupling with elastic end users' demands and traffic flows. Figure 1 illustrates VNFs on top of a physical infrastructure. Virtual networks (accessed through IP layer) are mapped over a backbone network (typically optical network), where virtual nodes accessing through IP routers co-exist with some optical switching nodes. The optical/IP backbone provides connectivity among datacenters; and VNFs are deployed on datacenters and/or routers/switches.

Network function virtualization is still an emerging technology. Major network operators and standard setting organizations lead the development of NFV and present it in technical reports and white papers [2][10][11][12]. Research works on NFV focus on the architecture and framework to enable and control VNFs [13][14][15][16][17], and static and dynamic NFs and service chain placement [18][19] through various placement- and scheduling-based mathematical models [3][18][19]. Compared with the cloud infrastructure using proprietary networking hardware, capital expenditure (CAPEX) and operating expense (OPEX) of NFVs are greatly reduced as migrating and scaling-up and down of workload do not require deployment of specialized hardware [3]. An important benefit from the above is that network service operators can provide more flexible and operationally efficient NFVs to end-users [14][20], which is a win-win situation for both service operators and end-users as both CAPEX and OPEX are reduced [2].

The contributions of this paper lie in the following. (1) To the best of our knowledge, this is the first paper which discusses a consolidated design and provision scheme for virtual network function allocation targeting to minimize CAPEX/OPEX by optimally serving end-to-end user requests. (2) We define the problem and propose mathematical models for such scheme in this paper. (3) The proposed approach in this paper can also serve as the foundation of the integrated virtualization scheme for multiple virtual-layer settings in a cloud platform.

The rest of the paper is organized as follows. In Section II, we provide formal problem statements for a network function virtualization scheme realizing end-to-end requests, and prove its NP-completeness. We propose a mixed-integer program for the NFV scheme, which generates exact solutions in Section III. Experiment settings and computation results are given in Section IV.

## II. PROBLEM STATEMENT

In this section, we first conclude the use cases for virtual network functions discussed in [1]. Then, we provide problem definitions and prove its computational complexity.

### A. VNF Mapping Use Cases

Use cases for network function virtualization [1] include: (1) Virtual Network Function as a Service (VNFaaS), which configures the set of VNF instances made available by service providers; (2) Virtual Network Platform as a Service (VNPaaS), with which dedicated Access Point Name (APNs) serves as IP level entry points to private corporate networks of enterprises; and (3) end-to-end services supported by network service providers which involve cross-administrative-boundary operation, interworking, and migration to/from physical network function implementations. Correspondingly, network function virtualization can be categorized into three types: (1) Network-Level VNFs: an entire virtual network is associated with a set of VNFs. For example, all end requests in an enterprise Virtual Private Network (VPN) are required to pass through network functions like authentication and firewall. (2) Node-Level VNFs: to fulfill certain requests, they are routed through virtual nodes associated with a subset of VNFs. For instance, demands requiring user registration or/and load balancing would be routed through either single or multiple nodes with such functions. (3) Link-Level VNFs: similar to (2), but VNFs are associated with virtual links instead of nodes. For instance, end-to-end requests over certain virtual links are required to go through intrusion detection, firewall, and authentication.

In this paper, we focus on the third use case and present **network function virtualization realizing end-to-end requests** (NFV-RR) as follows: given a physical substrate network provided by a network service provider, end-user's requests may be estimated based on their service contracts. Service providers can then deploy network functions required onto VNF-enabled physical nodes to achieve user's end-to-end requests without the need of proprietary networking hardware.

Parameter	Description
$G_P(V_P, E_P)$	Physical substrate network (e.g., optical network connecting data centers) with node set $V_P$ and edge set $E_P$
$i, j, s, t$	Physical node indices, $i, j, s, t \in V_P$
$e$	Physical link index, $e \in E_P$
$C_e$	Capacity of physical link $e$ , $e \in E_P$
$C_i$	Computational capacity of physical node $i$ , $i \in V_P$
$D$	End-user request set, i.e., $D = \{(s, t) : s, t \in V_P\}$
$\mathcal{F}, f$	Network function set $\mathcal{F}$ with network function $f \in \mathcal{F}$
$F_{st}, m_{st}^f$	Network function required by the demand between $(s, t)$ , i.e., $F_{st} = \{(f, m_{st}^f) : f \in \mathcal{F}, m_{st}^f \in \mathbb{Z}^+\}$ , where $m_{st}^f$ is the required instances of function $f$
$d_{st}^c$	Required bandwidth for request between $(s, t)$
$d_{st}^l$	Required computational resources for request between $(s, t)$
$\eta_i^f$	Required computational resources for network function $f$ at node $i$
$d_{st}$	End-to-end request between $(s, t)$ denoted as a triplet $d_{st} = \{F_{st}, d_{st}^c, d_{st}^l\}$
$p_{st}$	Physical path for request $d_{st}$ , $p_{st} \subseteq G_P$ and $(s, t) \in D$
$c_i^f$	Cost to deploy an instance of network function $f$ to node $i$ , $f \in \mathcal{F}$ and $i \in V_P$
$c_{ij}$	Cost to utilize a unit of physical resources on edge $e$ with $e \in E_P$
$c_i$	Cost to utilize a unit of physical resources on node $i$ with $i \in V_P$
Variable	Description
$x_i^{st}$	Binary variable indicating whether the route $p_{st}$ of end-to-end request $(s, t)$ passes through node $i$ or not. If yes, $x_i^{st} = 1$ ; otherwise, $x_i^{st} = 0$ . $(s, t) \in D$ and $i \in V_P$
$y_{ij}^{st}$	Binary variable indicating whether the route $p_{st}$ of end-to-end request $(s, t)$ is routed through $(i, j)$ . If yes, $y_{ij}^{st} = 1$ ; otherwise, $y_{ij}^{st} = 0$ . $(s, t) \in D$ and $(i, j) \in E_P$
$n_i^f$	Number of instances of network function $f$ deployed to physical node $i$ , $f \in \mathcal{F}$ and $i \in V_P$
$\varsigma_{st}^f$	Number of instances of network function $f$ deployed to physical node $i$ for request $(s, t)$

TABLE I: Notations for parameters and variables

### B. Problem Description

Based on the setting of NFV-RR, we introduce the notations for parameters and variables used in this paper in Table I and provide formal definitions for NFV-RR and related problems.

Let  $n_i^f$  be the number of instances of network function  $f$  deployed at physical node  $i$ , and  $\eta_i^f$  be the computational resources required to fulfill an instance of network function  $f$  at node  $i$ . We define **network function allocation** as deploying  $n_i^f$  instances of network function  $f \in \mathcal{F}$  to physical node  $i \in V_P$  and determining  $\eta_i^f n_i^f$  as the total physical resources required to fulfill all  $n_i^f$  instances of network function  $f$ .

An **end-to-end virtual function request**, denoted as  $F_{st} = \{(f, m_{st}^f) : f \in \mathcal{F}, m_{st}^f \in \mathbb{Z}^+\}$ , is the network function requirement of the request between nodes  $s$  and  $t$ , where  $m_{st}^f$  denotes the required number of instances of network function  $f$  for the request between  $s$  and  $t$ . An **end-to-end request**  $d_{st}$ , a triplet  $d_{st} = \{F_{st}, d_{st}^c, d_{st}^l\}$ , integrates the network function request, and its required bandwidth ( $d_{st}^c$ ) and computational resources ( $d_{st}^l$ ). With the definitions above, we may now define

the NFV-RR problem.

**Definition 1.** Given a physical substrate network  $G_P = (V_P, E_P)$ , its link capacity  $C_e$ , node capacity  $C_i$ , and demand  $d_{st} = \{F_{st}, d_{st}^c, d_{st}^l\}$  with  $F_{st} = \{(f, m_{st}^f) : f \in \mathcal{F}, m_{st}^f \in \mathbb{Z}^+\}$ , the problem of **network function virtualization realizing end-to-end requests (NFV-RR)** is to determine the placement of network functions  $f \in \mathcal{F}$  which satisfies the following conditions.

- (1) Each demand  $d_{st}$ ,  $(s, t) \in D$ , is realized through a physical route  $p_{st}$ ,  $p_{st} \subseteq G_P$ .
- (2) The route  $p_{st}$  of  $d_{st}$  should pass through physical nodes  $i \in V_P$  deployed with required network functions (specified in  $F_{st}$ ), which satisfies both the types of network functions and their required number of instances. That is,  $\sum_{i \in p_{st}, (s,t) \in D} m_{st}^f \leq n_i^f$  with  $f \in \mathcal{F}$  and  $i \in V_P$ .
- (3) The cumulative bandwidth request for each physical link  $e \in E_P$  should not exceed its capacity, i.e.,  $\sum_{e \in p_{st}, (s,t) \in D} d_{st}^c \leq C_e$ .
- (4) For each physical node  $i \in V_P$ , the cumulative computational resources required to process network functions and flows on node  $i$  should not exceed its capacity, i.e.,  $\sum_{i \in p_{st}, (s,t) \in D} d_{st}^l + \sum_{f \in \mathcal{F}} n_i^f n_i^f \leq C_i$ .

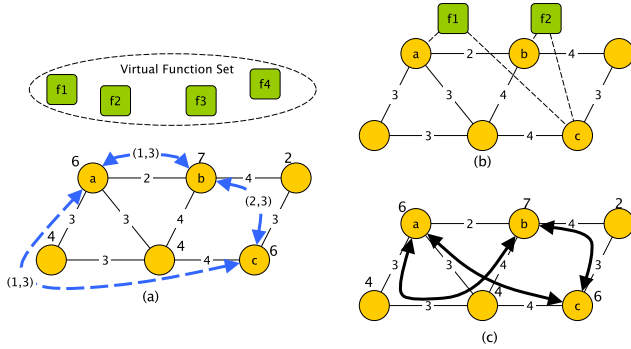


Fig. 2: An instance of NFV for end-to-end requests

We use Fig. 2 to illustrate an instance of VNF allocation realizing end-to-end requests. Figure 2(a) shows a network function set  $\mathcal{F} = \{f_1, f_2, f_3, f_4\}$ , a physical network with node capacity (computation) and link capacity (communication), end-to-end requests (illustrated in dashed blue line), and their computation and communication resources (the two values on the blue line). All requests are required to fulfill network functions  $\{f_1, f_2\}$ . Figure 2(b) presents feasible VNF allocation where function  $f_1$  is deployed to physical nodes  $a$  and  $c$ , and  $f_2$  is assigned to nodes  $b$  and  $c$ . Figure 2(c) illustrates a virtual link mapping which satisfies VNF requests and does not require extra physical resources.

**Theorem 1.** The network function virtualization problem for end-to-end request realization is NP-complete.

Based on the problem definition, two-commodity integer flow problem [21] is a special instance of the network function virtualization problem in which no NF is allocated and no node

capacity is considered. Since the two-commodity integer flow problem is NP-complete, our claim holds.

### III. MATHEMATICAL FORMULATION

The NFV-RR problem requires the allocation of resources to physical routes and network functions for end-to-end requests. In this section, we present a mixed-integer program aiming at optimizing NFV-RR problem which satisfies all conditions in Definition 1 as well as minimizing the CAPEX for VNF provisioning and assignment. All variables used in the mathematical formulations are listed in Table I.

First, as discussed in [22], each end-to-end request is realized through a physical route generated by flow conservation constraints as follows.

$$\sum_{(i,j) \in E_P} y_{ij}^{st} - \sum_{(j,i) \in E_P} y_{ji}^{st} = \begin{cases} 1, & \text{if } i = s, \\ -1, & \text{if } i = t, \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

$$(s, t) \in D, i \in V_P$$

Different from [22], we introduce node capacity and network function allocation in our formulation which determine (1) the consumption of physical node resources for end-to-end requests, and (2) whether physical routes  $p_{st}, (s, t) \in D$ , travel through physical node  $i$  or not. We introduce an auxiliary variable  $x_i^{st}$  which represents whether request  $d_{st}$  is routed through physical node  $i$  or not. If yes,  $x_i^{st} = 1$ ; otherwise,  $x_i^{st} = 0$ .

**Proposition 1.** Variable  $x_i^{st}$  indicates whether the physical route  $p_{st}$  of  $d_{st}$  visits physical node  $i$  if and only if the following node-based routing constraints lead  $x_i^{st}$  to 0 or 1.

$$x_i^{st} \leq \sum_{(i,j) \in E_P} (y_{ij}^{st} + y_{ji}^{st}), \quad (s, t) \in D, i \in V_P \quad (2)$$

$$x_i^{st} \geq y_{ij}^{st} + y_{ji}^{st}, \quad (s, t) \in D, i \in V_P, (i, j) \in E_P \quad (3)$$

Please see Appendix A for proof of correctness. Note here that constraints (2) and (3) build the connection between indicator  $x_i^{st}$  and route indicator  $y_{ij}^{st}$ .

We now discuss the allocation of network functions which fulfills the network function requirement for each  $d_{st}$ . Intuitively, the network function allocation problem is a facility location problem which selects physical nodes to place network functions. We provide a counterexample in Fig. 3 and demonstrate that location-based methods cannot solve the network function allocation problem satisfying required network functions for demand  $d_{st}$ .

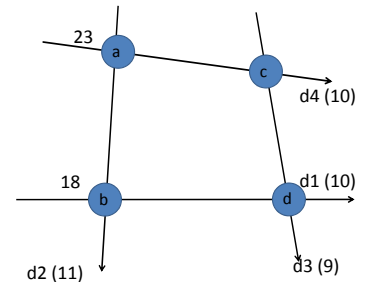


Fig. 3: Network function allocation through a location-based method

A single network function is considered in this example, where network functions are deployed to node  $a$  and  $b$  with computational capacity 23 and 18. The total

computational capacity of the network function is 41 which is greater than the sum of all demands, 40; moreover, except demand  $d_3$ , the computational capacities of the NF at node  $a$  and  $b$  are greater than all demands. It shows that without considering the relationship between network function placement (location and capacity) and demands, location-based methods may not fulfill the network function requirement for each end-to-end request. To solve the problem, we propose an assignment-based method and introduce an auxiliary variable  $\varsigma_{st}^{fi}$  which represents how many instances of network function  $f$  required by  $d_{st}$ ,  $(s, t) \in D$ , are deployed at physical node  $i$ . The corresponding constraints are introduced as follows.

**Proposition 2.** *The following constraints determine the allocation of network functions which fulfills the network function request of  $F_{st}$ .*

$$m_{st}^f = \sum_{i \in V_P} \varsigma_{st}^{fi} x_i^{st}, \quad f \in \mathcal{F}, (s, t) \in D \quad (4)$$

$$n_i^f = \sum_{(s,t) \in D} \varsigma_{st}^{fi}, \quad f \in \mathcal{F}, i \in V_P \quad (5)$$

In constraint (4),  $\varsigma_{st}^{fi} x_i^{st} \neq 0$  means that demand  $d_{st}$ 's route  $p_{st}$  passes through physical node  $i$ , and network function  $f$  required by  $d_{st}$  is deployed at  $i$ . Constraint (5) cumulates all instances of network function  $f$  placed at  $i$  for all requests in  $D$ . Here, we reformulate the nonlinear constraint (4) to linearize it without introducing extra variables.

**Proposition 3.** *Constraints (6) and (7) are equivalent to constraint (4). (See Appendix A for proof of correctness.)*

$$\varsigma_{st}^{fi} \leq m_{st}^f x_i^{st}, \quad i \in V_P, f \in \mathcal{F}, (s, t) \in D \quad (6)$$

$$m_{st}^f = \sum_{i \in V_P} \varsigma_{st}^{fi}, \quad (s, t) \in D, f \in \mathcal{F} \quad (7)$$

Next, we present a mixed integer linear program (MILP) for the NFV-RR problem with a single end-to-end physical route for each  $d_{st}$  as follows.

$$\min \sum_{f \in \mathcal{F}} \sum_{i \in V_P} c_i^f n_i^f + \sum_{(s,t) \in D} \left( \sum_{(i,j) \in E_P} c_{ij} y_{ij}^{st} d_{st}^c + \sum_{i \in V_P} c_i x_i^{st} d_{st}^l \right)$$

s.t. Constraints (1)–(3) and (5)–(7)

$$\sum_{(s,t) \in D} y_{ij}^{st} d_{st}^c \leq C_{ij}, \quad (i, j) \in E_P \quad (8)$$

$$\sum_{(s,t) \in D} x_i^{st} d_{st}^l + \sum_{f \in \mathcal{F}} \eta_i^f n_i^f \leq C_i, \quad i \in V_P \quad (9)$$

$$y_{ij}^{st}, x_i^{st} \in \{0, 1\},$$

$$n_i^f, \varsigma_{st}^{fi} \in \mathbb{Z}^+, \quad (s, t) \in D, s, t, i, j \in V_P, f \in \mathcal{F} \quad (10)$$

Constraints (6) and (7) correspond to condition (2) in Definition 1. Constraints (8) and (9) provide capacity limitation on physical links and nodes introduced as conditions (3) and (4) in Definition 1. Constraint (8) restricts the cumulative traffic flow routed through a physical link to be less than or equal to its capacity. Constraint (9) guarantees the computational resources required by network functions and traffic flows to be less than or equal to the capacity of physical nodes.

## IV. SIMULATION RESULTS

In this section, we propose experimental settings and present their computation results for the optimal NFV-RR problem. We select NSF network, denoted as ‘‘NSF’’ and illustrated in Fig. 4, as the physical infrastructure [23] which has 14 nodes and 21 edges. We consider a network function set  $\mathcal{F} = \{1, 2, 3\}$  with three network functions.

As shown in Table II, two scenarios, denoted as ‘‘ $D_1$ ’’ and ‘‘ $D_2$ ’’, are presented. Each scenario is composed of a set of demand pairs representing the end nodes of a demand, and the corresponding network function required by each demand (denoted as ‘‘NFs’’).

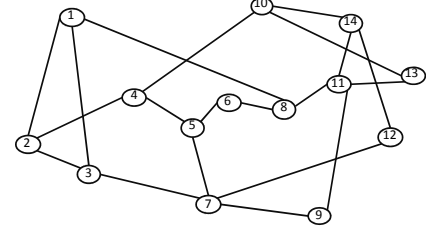


Fig. 4: NSF network

Index	$D_1$	$D_2$	NFs
1	(2, 4)	(1, 2)	{1}
2	(2, 10)	(1, 8)	{2}
3	(4, 7)	(2, 7)	{3}
4	(7, 11)	(7, 12)	{2,3}
5	(10, 14)	(8, 14)	{1,3}
6	(13, 14)	(13, 14)	{1,2,3}

TABLE II: Test scenarios with end-to-end requests and their corresponding network function requirement

In Table III, ‘‘ $P_1$ ’’ and ‘‘ $P_2$ ’’ are the parameter sets representing link and node capacities, demands (for computational resources and bandwidth), and the computational resources consumed by network functions, which are generated in uniform distributions with corresponding intervals.

	$P_1$	$P_2$
Demand (resource)	[ 5, 15]	[ 5, 15]
Demand (bandwidth)	[10, 20]	[10, 20]
Network function (source)	[ 5, 15]	[10, 30]
Capacity (node)	[60,100]	[90,120]
Capacity (edge)	[20, 40]	[40, 60]

TABLE III: Parameter sets used in the test scenarios

Given the above scenarios and parameters, we now present our computational results. To understand how the allocation of network functions affects end-to-end requests under the same network topology, we first consider a relaxed scenario where all edges and nodes have no capacity and computational limitation, and each end-to-end demand (computational and communication resources) and the resources consumed by network functions are all with a single unit. The costs for network function placement and resource consumption are both single unit as well. The objective for this setting is to minimize the total costs of routings and network function allocation.

Index	$D_1$		$D_2$	
	Routes	NF-Node	Routes	NF-Node
1	2-4	2	1-2	1
2	2-4-10	2	1-8	1
3	4-5-7	4	2-3-7	2
4	7-9-11	7	7-12	7
5	10-14	10	8-11-14	8
6	13-10-14	13	13-11-14	13

TABLE IV: Optimal routes and network function allocation in an uncapacitated network with unit demands

The results of routings and network function allocation are reported in Table IV for scenarios “ $D_1$ ” and “ $D_2$ ”, in which “NF-Node” represents the optimal locations of network function requests corresponding to each demand. We observe that in the uncapacitated network, all routes for end-to-end requests are still the shortest paths between demands’ end nodes, and network functions are all placed at the source node of each end-to-end request.

Node		2	4	5	7	9	10	11	13	14
Relaxed S1	Routes	2	3	1	2	1	3	1	1	2
	NF	2	1	-	2	-	2	-	3	-
Relaxed S2	Routes	2	2	1	2	2	2	1	1	2
	NF	2	1	2	2	-	-	3	-	-

TABLE V: Physical nodes utilized in optimal routes and network function allocation

Table V presents the results for the optimal routes and NF placement. For both scenarios, the same number of nodes (in NSF network) are visited to generate end-to-end routes and deploy network functions. Note here that “-” means that the corresponding node is not utilized for routes or network function placement in NSF network.

We now present the computational results for the original NFV-RR problem with two demand sets in Table II and two sets of parameters in Table III. The combinations of the demand and parameter sets (4 in total) were tested and indexed as “S1” ( $D_1, P_1$ ), “S2” ( $D_1, P_2$ ), “S3” ( $D_2, P_1$ ), and “S4” ( $D_2, P_2$ ). The computation results for “S1” to “S4” are reported in Table VI. Note here that: (1) “-” in Table VI means no resource consumption for that specific node in the corresponding testing scenario; and (2) network functions’ resource consumption in “S3” and “S4” is twice of that in “S1” and “S2” on average.

We observe that network functions in “S3” and “S4” consume more node resources than “S1” and “S2” as network functions in the former require more resources. Meanwhile, even with the same demands, different resource consumptions for network functions can actually cause the generation of different end-to-end routes and network function placement.

To further analyze these results, we compare the resource consumption of nodes for both network functions and end-to-end requests, and illustrate them in Fig. 5. The resource consumption of network functions in “S3” and “S4” (for indices 1-3 in Table II) are on average double of that in “S1” and “S2”. Figure 5 shows that when the resource consumption of network functions is higher (as in “S3” and “S4”), more

Node	1	2	3	4	5	6	7
Capacity	24	22	33	32	22	21	36
NF (S1)	-	-	12	9	-	6	15
Rqt (S1)	22	22	10	20	12	12	21
NF (S2)	-	-	-	6	-	-	9
Rqt (S2)	22	20	12	20	20	0	27
Capacity	44	42	53	52	42	41	56
NF (S3)	-	17	-	11	-	-	24
Rqt (S3)	20	22	8	30	29	19	31
NF (S4)	11	-	24	11	-	-	17
Rqt (S4)	28	26	16	36	22	12	33
Node	8	9	10	11	12	13	14
Capacity	38	35	27	26	31	38	29
NF (S1)	-	21	-	-	9	12	6
Rqt (S1)	22	13	26	23	22	24	22
NF (S2)	18	-	-	-	21	27	9
Rqt (S2)	18	0	0	14	7	6	14
Capacity	58	55	47	46	51	58	49
NF (S3)	24	-	11	17	24	41	11
Rqt (S3)	27	6	26	29	16	14	32
NF (S4)	17	24	17	-	11	24	24
Rqt (S4)	36	15	24	31	33	31	24

TABLE VI: Node capacity and resource consumption by end-to-end requests and network function allocation

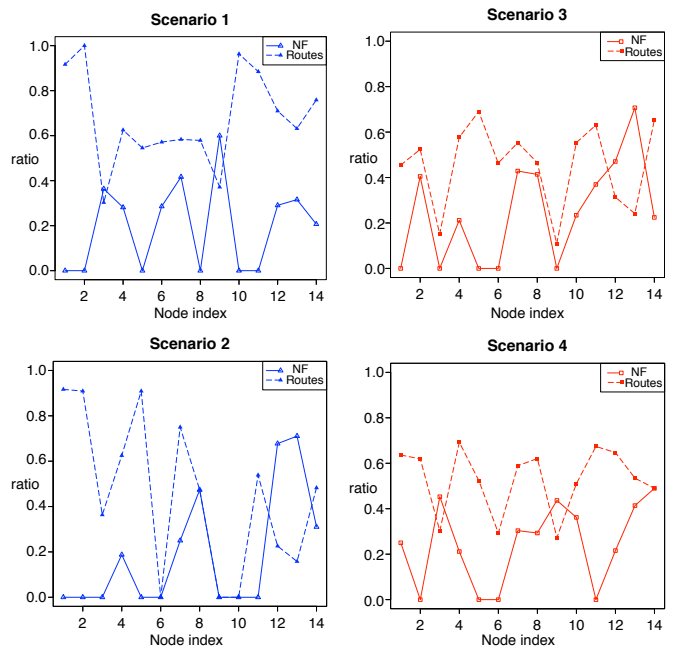


Fig. 5: The ratio of resource consumption on nodes for network functions and end-to-end-requests

nodes would be involved in routing and network function placement compared with the scenarios (as in “S1” and “S2”) that require less resources for network functions. For example, in “S2”, network functions are deployed to only 6 nodes compared to 10 nodes in “S4”, and the routes in “S2” only pass through 11 nodes compared to 14 nodes in “S4”. This explains why the resource consumption in the first two scenarios has peak values (0.8 – 1) for certain nodes, and the latter two scenarios in general have values in the range of 0.2 – 0.6. Similar to the uncapacitated NFV-RR problem, when provided with different end-to-end requests, routes and network function allocation are different, and the consumption of their physical

node capacity is also different. We wish to note here that the optimal routes for NFV-RR problems are no longer the shortest paths between demands' end nodes.

## V. CONCLUSION

In this paper, we studied network function virtualization with end-to-end request realization (NFV-RR) based on a use case in [1], and evaluated the performance of the placement of virtual network functions in terms of its ability to support end-to-end requests with limited physical resources. We proposed a mixed-integer program, and solved the NFV-RR problem in its original and relaxed forms. Our computational results demonstrate the value of proposed integrated approach. We will further explore efficient algorithms to solve the proposed problem.

## APPENDIX: PROOF OF PROPOSITIONS 1 AND 3

We first prove the correctness of Proposition 1.

*Proof.* We prove the necessary and sufficient conditions for the proposition. First, we observe that if and only if at least one adjacent arc of a physical node is visited, the physical node is visited. Meanwhile, the physical route for demand  $(s, t)$  is a simple path. Therefore, constraint  $y_{ij}^{st} + y_{ji}^{st} \leq 1$  forbids a cycle connecting nodes  $i$  and  $j$  with parallel arcs.

Sufficient condition: with constraint (1) selecting a route for each  $d_{st}$ ,  $y_{ij}^{st}$  indicates whether  $(s, t)$ 's physical route travels through link  $(i, j)$ . According to constraint (2), if the physical route  $p_{st}$  of  $(s, t)$  does not visit any of  $i$ 's adjacent arcs, then  $x_i^{st} = 0$ ; otherwise,  $x_i^{st} \leq 1$  because of its boundary  $0 \leq x_i^{st} \leq 1$ . Since we require a simple physical route for each  $d_{st}$ , we have  $y_{ij}^{st} + y_{ji}^{st} = 1$ . And constraint (3) makes sure that if a physical route of  $d_{st}$  travels through arcs  $(i, j)$  or  $(j, i)$  adjacent to node  $i$ ,  $x_i^{st} \geq y_{ij}^{st} + y_{ji}^{st} = 1$ ; otherwise,  $x_i^{st} \geq 0$ . Therefore, constraints (2) and (3) force variable  $x_i^{st}$  to have the value following  $x_i^{st}$ 's definition.

Necessary condition: given the definition of  $x_i^{st}$ , if  $x_i^{st} = 1$  in constraint (2), at least an arc adjacent to  $i$  is visited. Hence,  $\sum_{(i,j) \in E_P} (y_{ij}^{st} + y_{ji}^{st}) \geq 1$ ; otherwise,  $x_i^{st} = 0$  and  $\sum_{(i,j) \in E_P} y_{ij}^{st} + y_{ji}^{st} \geq 0$ . If  $x_i^{st} = 1$  in constraint (3),  $y_{ij}^{st} + y_{ji}^{st} \leq 1$ ; otherwise,  $y_{ij}^{st} + y_{ji}^{st} = 0$ , which indicates that no arc adjacent to node  $i$  is visited.

According to the necessary and sufficient conditions above, the proposition holds.  $\square$

Next, we prove the correctness of Proposition 3.

*Proof.* We prove by demonstrating that these two set of constraints provide the same feasible regions for end-to-end request  $(s, t)$ 's network function placement. There are two conditions for  $\zeta_{st}^{fi} x_i^{st}$ : if  $x_i^{st} = 1$ ,  $\zeta_{st}^{fi} x_i^{st} = \zeta_{st}^{fi}$ ; otherwise,  $\zeta_{st}^{fi} x_i^{st} = 0$ . With constraint (6), when  $x_i^{st} = 0$ ,  $\zeta_{st}^{fi} = 0$ , which forces  $\zeta_{st}^{fi} x_i^{st} = 0$ ; otherwise,  $\zeta_{st}^{fi}$  is not forced to be 0, and the summation of  $\zeta_{st}^{fi}$  for all  $i \in V_P$  equals  $m_{st}^f$ . Thus, the proposition holds.  $\square$

## REFERENCES

- [1] "Network function virtualization (NFV): Use cases," ETSI, Tech. Rep. ETSI GS NFV 001, Oct. 2013.
- [2] "Network functions virtualisation (NFV): Network operator perspectives on industry progress," ETSI, Tech. Rep., 2014.
- [3] S. Mehrotra, M. Keller, and H. Karl, "Specifying and placing chains of virtual network functions," *arXiv preprint arXiv:1406.1058*, 2014.
- [4] C. Devellder, M. De Leenheer, B. Dhoedt, M. Pickavet, D. Colle, F. De Turck, and P. Demeester, "Optical networks for grid and cloud computing applications," *Proceedings of the IEEE*, vol. 100, no. 5, pp. 1149–1167, 2012.
- [5] L. M. Contreras, V. López, O. G. De Dios, A. Tovar, F. Muñoz, A. Azañón, J. P. Fernandez-Palacios, and J. Folgueira, "Toward cloud-ready transport networks," *Communications Magazine, IEEE*, vol. 50, no. 9, pp. 48–55, 2012.
- [6] N. Chowdhury, M. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," in *INFOCOM 2009, IEEE*, 2009, pp. 783–791.
- [7] A. Fischer, J. F. Botero, M. Till Beck, H. De Meer, and X. Hesselbach, "Virtual network embedding: A survey," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 4, pp. 1888–1906, 2013.
- [8] B. P. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on*, 2009, pp. 44–51.
- [9] I. Abbadi, "Clouds infrastructure taxonomy, properties, and management services," *Advances in Computing and Communications*, vol. 193, pp. 406–420, 2011.
- [10] "AT&T vision alignment challenge technology survey," AT&T, Tech. Rep., 2013.
- [11] "Network functions virtualization," Hewlett Packard, Tech. Rep., 2014.
- [12] "Value-added services and service chaining: Deployment considerations and challenges," Sandvine, Tech. Rep., 2014.
- [13] H. Xie, Y. Li, J. Wang, D. R. Lopez, T. Tsou, and Y. Wen, "vRGW: Towards network function virtualization enabled by software defined networking," in *ICNP*, 2013, pp. 1–2.
- [14] G. Xilouris, E. Trouva, F. Lobillo, J. Soares, J. Carapinha, M. McGrath, G. Gardikis, P. Paglierani, E. Pallis, L. Zuccaro *et al.*, "T-NOVA: A marketplace for virtualized network functions," in *Networks and Communications, 2014 European Conference on*, 2014, pp. 1–5.
- [15] A. Gember-Jacobson, R. Viswanathan, C. Prakash, R. Grandl, J. Khalid, S. Das, and A. Akella, "OpenNF: Enabling innovation in network function control," in *Proceedings of the 2014 ACM conference on SIGCOMM*, 2014, pp. 163–174.
- [16] A. Al-Shabibi, M. De Leenheer, M. Gerola, A. Koshibe, W. Snow, and G. Parulkar, "OpenVirteX: A network hypervisor," *Open Networking Summit*, 2014.
- [17] J. Soares, M. Dias, J. Carapinha, B. Parreira, and S. Sargento, "Cloud4NFV: A platform for virtual network functions," in *Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on*, 2014, pp. 288–293.
- [18] M. Xia, M. Shirazipour, Y. Zhang, H. Green, and A. Takacs, "Network function placement for NFV chaining in packet/optical data centers," in *Optical Communication, 2014 European Conference on*, 2014, pp. 1–3.
- [19] M. Bagaa, T. Taleb, and A. Ksentini, "Service-aware network function placement for efficient traffic handling in carrier cloud," in *Wireless Communications and Networking Conference (WCNC), 2014 IEEE*, 2014, pp. 2402–2407.
- [20] M. Scholler, M. Stiernerling, A. Ripke, and R. Bless, "Resilient deployment of virtual network functions," in *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2013 5th International Congress on*, 2013, pp. 208–214.
- [21] A. Itai, "Two-commodity flow," *J. ACM*, vol. 25, no. 4, pp. 596–611, Oct. 1978.
- [22] T. Lin, Z. Zhou, K. Thulasiraman, G. Xue, and S. Sahni, "Unified mathematical programming frameworks for survivable logical topology routing in IP-over-WDM optical networks," *Optical Communications and Networking, IEEE/OSA Journal of*, vol. 6, no. 2, pp. 190–203, 2014.
- [23] C. Colman-Meixner, F. Dikbiyik, M. F. Habib, M. Tornatore, C.-N. Chuah, and B. Mukherjee, "Disaster-survivable cloud-network mapping," *Photonic Network Communications*, vol. 27, no. 3, pp. 141–153, 2014.