Reliable Service Function Chain Provisioning in Software-Defined Networking

Yansen Xu* Ved P. Kafle*†
* Department of Informatics
Graduate School of Informatics and Engineering, University of Electro-Communications (UEC)
† National Institute of Information and Communications Technology (NICT)
E-mail: xu.yansen@net.is.uec.ac.jp kafle@nict.go.jp

Abstract*—A Service Function Chain (SFC) is an ordered Network Function (NF) chain to process flows or packets for the end-to-end delivery of network services. In the context of Network Function Virtualization (NFV) and Software-Defined Networking, which are promising technologies for next generation networks, the Virtualized Network Function (VNF) can be deployed on either generic physical machines or virtual machines. A challenging problem is to determine where and how to place these VNFs of an SFC request in the network. In this paper, we first formulate this VNFs placement problem as an Integer Linear Programing (ILP) model and then propose an enhanced VNF placing scheme based on layered graphs to achieve better reliability. To improve the reliability, our scheme avoids placing more than one VNF of an SFC on the same node to protect the SFC from a single point of failure. We have conducted a numerical analysis and computer simulation for the feasibility validation of our scheme. The performance results, in terms of end to end delay of SFC and computation time cost on different topologies, show that our scheme performs well in different scenarios.

Keywords—SFC;NFV;SDN;Resource Management;Network Management

I. INTRODUCTION

Service Function Chain (SFC) [1] is defined by the Internet Engineering Task Force (IETF) as an ordered set of abstract service functions for processing traffic flows. A traffic flow is supposed to travel through an SFC while going to the final destination. In the context of Network Function Virtualization (NFV), an SFC is used in cloud for network operators or enterprises and Virtual Network Functions (VNFs) in an SFC are deployed on virtual machines to achieve maximum profit and minimum cost.

One of the fundamental problems of SFC in virtualized environment is where to deploy VNFs of an SFC in the network and how to route traffic flow through these VNFs so that the resources such as CPU, memory and link bandwidth capacity are efficiently utilized, and availability and reliability are improved.

Generally, more than one VNFs in an SFC could be deployed on the same physical node to achieve minimum cost. However, this colocation causes an unbalanced resource utilization and increases the risk of server down under high loads [9]. As a result, an SFC will suffer from a single point failure and take a long time to recovery if multiple VNFs in this SFC are deployed on the same node. Furthermore, since a traffic flow is processed by VNFs in SFC in a specific order, if several discontinued VNFs of SFC are deployed on the same physical node, the traffic flow will be loop in the network in order to be processed by all VNFs which increases the end-to-end latency as well as the risk of forming loops.

In this paper, we focus on choosing different physical nodes for placing VNFs of SFC distributedly so that no more than one VNF is placed on the same physical node in order to avoid a single point failure and increase reliability.

We first formulated the problem of service function chain provisioning as an Integer Linear Programming (ILP) model in terms of reliability. Then we proposed an algorithm to improve reliability by extending the previous work [6]. Finally, we conducted numerical analysis to verify that our proposal enhances the network reliability.

II. RELATED WORK

Many prior studies have investigated VNF deployment problems.

An adaptive service function chaining placement by utilizing layered graphs [6] to minimize the network delay has been proposed in [2]. And Huin et al. [3] formulated the problem as ILP for SFC provisioning based on [2]. They investigated the best compromise between bandwidth requirements and the number of possible nodes which can host VNFs, and the number of chain occurrences.

Hsieh et al. [4] proposed a network-aware service function chaining placement by formulating the problem as a bin-pack problem to minimize the network cost and server cost. They treated the problem as a multi-layer bin packing problem and proposed two greedy algorithms for the treelike network topology.

Cohen et al. [5] addressed the VNFs placement problem to minimize overall network cost by formulating the problem as the facility location problem and the generalized assignment problem (GAP). They evaluated how the placement of VNFs affects the performance of the network, reliability and operation cost.

These prior works, however, do not consider reliability issues when provisioning resources to an SFC. Our SFC provisioning scheme is based on layered graphs as used in [6]

978-3-901882-98-2 (c) 2017 IFIP

to enhance SFC reliability by distributedly deploying VNFs in SFC.

III. LAYERED GRAPH SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we first briefly introduce layered graph based SFC provisioning scheme for SFC routing [2][3][6], and then formulate the SFC routing problem as an ILP model.

A. Layered Graph System Model

First, we briefly introduce the layered graph based routing scheme referenced from [2][3][6]. Given a network topology G and a service function chain S composed of t VNFs, the scheme first creates t copies of original network G and each copy's topology is exactly same as the original network's topology. We denote G^0 as the original network and G^i as the i th network. Two neighboring networks G^{i-1} and G^i are connected vertically by the nodes which can host ith VNF in SFC. The source node of the SFC request is remained on the same location in G^0 and the destination node of the SFC request is set on the same location in G^t as it is in G^0. Then these $t+1$ layered graphs compose a new graph. The scheme then finds a shortest path between the source node in the first layer and the destination node in the last layer in the new graph. Each transition node on the shortest path connecting $i-1$ and i layers is the node to host the ith VNF in the SFC.

B. Problem Formulation

A network is modeled as an undirected graph $G=(V,E)$ where V and E represent the set of nodes and edges, respectively. The following notations are used in the model.

F: set of virtual network functions (VNFs) $f \in F$

d_{ij}: delay in a link between node i and node j.

R: service function chain request which contains a service function chain S , source node v_s and destination v_d node, i.e. $\{v_s, v_d, S\}$. Here S contains a series of ordered VNFs $\{f_1, f_2, \dots, f_i, \dots, f_t\}$, where f_i is the ith VNF in S and t is the total number of VNFs in S.

N_f: set of possible nodes for hosting VNF f.

x_{ij}: binary value that represents whether edge e_{ij} is on the path from source node v_s to destination node v_d.

y_v^f: binary value that represents whether VNF f is mapped on node v.

The problem is to determine which VNF in SFC is mapped on to which node to minimize the delay of SFC. We also consider that one node can host at most one VNF to avoid a single point of failure in SFC.

Objective:

Minimize delay

$$min \sum_{(i,j)\in E} x_{ij} d_{ij} \tag{1}$$

Subjected to:

$$\sum_{(i,j)\in E} x_{ij} - \sum_{(j,i)\in E} x_{ji} = 0 \quad i,j \in V, i \neq v_s, i \neq v_d \tag{2}$$

$$\sum_{(i,j)\in E} x_{ij} - \sum_{(j,i)\in E} x_{ji} = 1 \ \ i \in \{v_s, v_d\ \} \tag{3}$$

$$\sum_{v\in N_f} y_v^f \ = 1 \quad for\ \forall f \in S \tag{4}$$

$$\sum_{f\in S} y_v^f \ = 1 \quad for\ \forall v \in V \tag{5}$$

The objective (1) is to minimize the end-to-end delay of the network path configured by the SFC request. Constraint (2) states that for any node i, which is neither a source node nor a destination node, the number of incoming flows is equal to the number of outgoing flows. Constraint (3) states that for any source or destination node, there is always an outgoing flow or an incoming flow. Constraint (4) states that for any network function f in service function chain request, only one

```
Algorithm 1 Node-selection
Variables:
$L_n$: a VNF list that node $n$ can host
$min_l$: the minimum length of $L_n$ for all $n$
$f_{min}$: VNF $f$ that has least number of possible nodes to deploy
$N_{f_{min}}$: : set of possible nodes for hosting VNF
INPUT: A set of nodes which can host a given VNF
  $f$, $N = \{N_f | \forall f \in S\}$;
  The node set $V$ of original network
OUTPUT: A new set of nodes which can host a
  given VNF and any two sets of nodes have no intersection.
$N = \{N_f | \cap_{\forall f \in S} N_f \ = \phi\}$
Begin:
for $n$ in $V$ do
  $L_n \leftarrow None$
  for $N_f$ in $N$ do
    if $n$ in $N_f$ then
      insert $f$ into $L_n$
    end if
  end for
end for
for all $L_n$ do
  $min_l \leftarrow Max\_Integer$
  $f_{min} \leftarrow None$
  if $|L_n| > 1$ then
    for each $f$ in $L_n$ do
      if $min_l > |L_n|$ then
        $min_l \leftarrow |L_n|$
        if $f_{min} \neq None$ and $|N_{f_{min}}| > 1$ then
          delete $n$ from $N_{f_{min}}$
        end if
        $f_{min} \leftarrow f$
      else
        if $|N_f| > 1$ then
          delete $n$ from $N_f$
        end if
      end if
    end for
  end if
end for
return $N_f$
```

node can host it. Constraint (5) states that for any node in the network, it can host only one VNF.

IV. Reliable SFC Provisioning Scheme

In this section, we propose a reliability enhanced SFC provisioning scheme. The basic idea of our scheme is that if a node can host more than one VNF, we assign this node to hosting only one VNF. If a node can host more than one VNF, we compare the number of nodes for hosting each VNF and find out the VNF which has the least number of nodes. Then we assign the node to this VNF. This node-selection scheme is shown in Algorithm 1.

Then according to the new set of nodes for VNFs $N = \{N_f \mid \forall f \in S\}$ generated by Algorithm 1, we create a new layered graph and find the shortest path between the source node in the first layer and the destination node in the last layer.

Note that our scheme cannot find out a proper solution if the collocated deployment, which more than two VNFs are deployed on a same node, is unavoidable. For example, consider three VNFs, and $N_{f1} = \{A, B\}, N_{f2} = \{C, D\}, N_{f3} = \{A, B\}$, then node A or node B will always host more than one VNF.

V. Evaluation and Analysis

A. Simulation Setup

We conducted numerical simulation to evaluate our scheme. The simulator is written in Python and runs on an Ubuntu 16.04 server with Intel-4600M 2.90GHz CPU and 8G memory. The network topology is generated on basis of Erdős-Rényi graph [7][8].

B. Performance of Reliability

Fig. 1 shows the number of times rate the reliability requirement was met under the condition when the number of nodes of network was fixed at 100 with the edge connection probability 0.5, and the number of VNFs in an SFC was fixed at 5. We varied the number of nodes for hosting one VNFs as 5, 10, 15, and 20, and repeated the experiment with each configuration 1000 times. We compared our scheme with a normal scheme, which only used layered graph system model without considering reliability. The results in Fig. 1 show that 546 out of 1000 times (54.6%) experiments the reliability requirements were met when the number of nodes for hosting one VNF was 5, and 52 out of 1000 times (5.2%) experiments the reliability requirements were met when the number of nodes for hosting one VNF was 20. However, by using our scheme, in all 1000 (100%) experiments of each configuration, the provisioning of SFC met the reliability requirement that no more than one VNF deployed on the same node.

C. Performance of End to End Delay in SFC

For evaluating performance in terms of end to end delay of SFC provisioned by utilizing our scheme, we set each link delay in network to a value uniformly distributed between 1 to 10 units. We considered that the bandwidth of link and the computation capacity of nodes were sufficient so that the processing time in a node could be ignored. We tested on two different network sizes, one with 50 nodes and the other with 100 nodes.

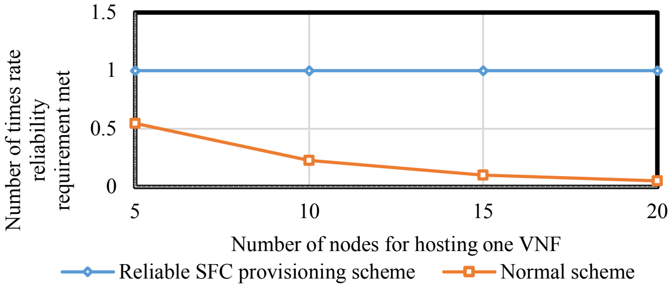

Fig. 1. Performance of reliability
(*100 nodes, edge connection probability 0.5*)

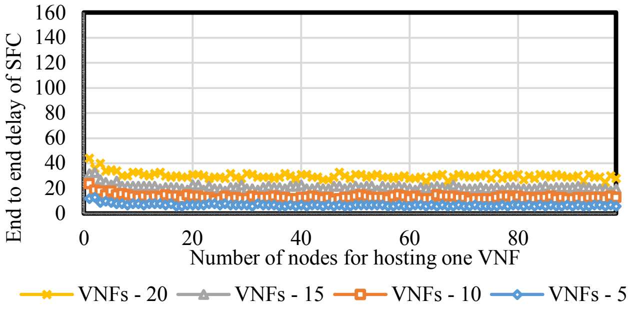

Fig. 2. Performance of end to end delay of SFC
(*100 nodes, edge connection probability 1.0*)

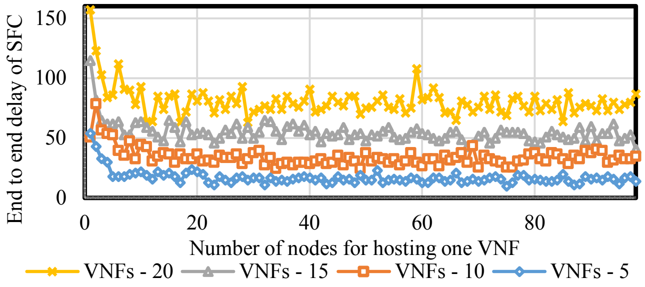

Fig. 3. Performance of end to end delay of SFC
(*100 nodes, edge connection probability 0.1*)

Fig. 2 and 3 show the results of our evaluation on performance of end to end delay of SFC on different physical network topologies consisting of 100 nodes, with probability of edge connection of 1.0, and 0.1. The *x*-axis is the number of nodes for hosting a VNF and the *y*-axis is the end to end delay of SFC provisioned by our scheme. Curves in each figure are end to end delay of SFC consisting of different numbers of VNFs 5, 10, 15, and 20 in the SFC.

By increasing the number of VNFs in SFC, the end to end delay of the SFC provisioned by our scheme increased proportionately as shown in both figures. The reason is that our scheme avoids to placing more than one VNF on the same node, which leads a long path and delay from the source node to destination node in SFC as the number of VNFs in SFC

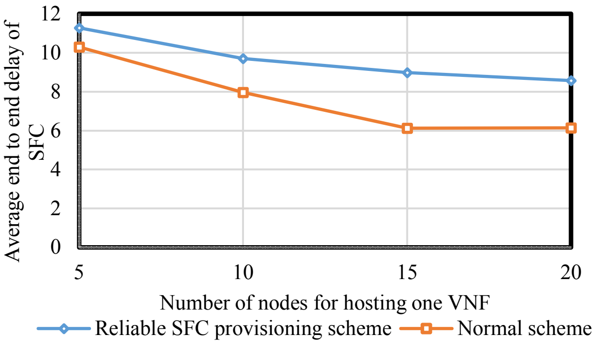

Fig. 4. Performance of average end to end delay of SFC
(*100 nodes, edge connection probability 0.5*)

increases.

Another result we obtained from the evaluation is that as the number of nodes in network and the edge connection probability increase, the end to end delay of SFC decreases. This is because the more nodes in network and the higher the edge connection probability, the more options for our scheme to find a shorter delay routing.

For each specific number of VNFs in the SFC, we evaluated our scheme by varying $|N_f|$. The results shown in Figs. 2 and 3 depict that as $|N_f|$ increases, the end to end delay of the SFC when $|N_f|$ is at low value, but finally the end to end delay of SFC stops decreasing and remains stable even $|N_f|$ increases. This is because that at a low value of $|N_f|$, few nodes can be selected for SFC routing and these nodes may be scattered in the network, thus resulting in a long end to end delay for SFC.

Fig. 4 shows the average end to end delay of SFC under the condition that the number of nodes of network was fixed at 100 with the edge connection probability 0.5, and the number of VNFs in an SFC was fixed at 5. We varied the number of nodes for hosting one VNFs 5, 10, 15, and 20, and repeated each experiment 1000 times. We compared our scheme with a normal scheme which only used layered graph system model but not considered reliability. The results show that the average of end to end delay of SFC is 11.287 by using our scheme and 10.292 by using normal scheme, and as the number of nodes for hosting one VNF increased, our scheme had a bit higher delay compared with normal scheme. This is reasonable since our scheme avoids multiple nodes to be deployed on the same node which results in a long path from a source to destination node.

D. Performance of Computation Time

For evaluating performance of computation time of Algorithm 1, we set the number of network nodes as 100, and the edge connection probability is fixed at 0.5. We also fixed $|N_f|$ at 5 and varied the number of VNFs in SFC 5, 10, 15, and 20. We repeated the experiment with each configuration 100 times and measured computation time cost for running each round of Algorithm 1. Fig. 5 shows the cumulative distribution function (CDF) of computation time cost of Algorithm 1.

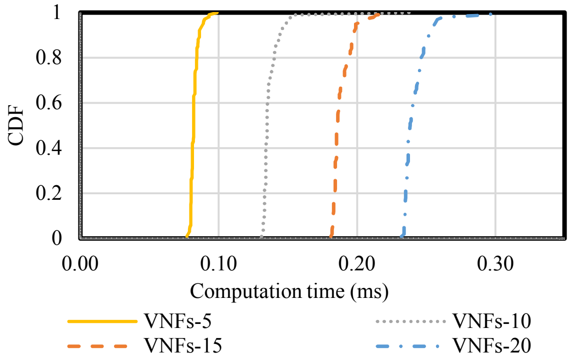

Fig. 5. Performance of end to end delay in SFC
(*100 nodes, edge connection probability 0.5*)

As shown in the figure, the computation time cost increases as the number of VNFs in an SFC increases. This is reasonable since a larger number of VNFs in an SFC increases the complexity of computation. It is clear from the figure that the computation time in different topologies is less than 1 millisecond, which is acceptable for online SFC provisioning.

VI. CONCLUSION

In this paper, we introduced the problem of virtual network function placement in the context of service function chaining, which specifies an order in the processing of different VNFs. We formulated the problem as an ILP problem and briefly described the layered graph based VNFs routing scheme. We proposed a reliable scheme for VNFs placement. The numerical simulation shows that our scheme can avoid deploying more than one VNF on the same nodes, which ensures the protection against a single-point failure and enhance reliability. We also showed the performance of end to end delay and computation time cost of the algorithm in our scheme. In future work, we will extend our scheme by considering more constraints such as limited bandwidth and computation capabilities.

REFERENCES

[1] P. Quinn, and Tom Nadeau, “Problem statement for service function chaining,” IETF RFC 7498 (2015).

[2] A. Dwaraki and T. Wolf, “Adaptive service-chain Routing for virtual network functions in software-defined networks,” Workshop on Hot topics in Middleboxes and Network Function Virtualization (HotMIddlebox), 2016.

[3] N. Huin, B. Jaumard, and F. Giroire, “Optimization of network service chain provisioning,” IEEE ICC 2017.

[4] C. Hsieh, J. Chang, C, Chen, and S. Lu, “Network-aware service function chaining placement in a data center,” IEEE APNOMS 2016.

[5] R. Cohen, L. Lewin-Eytan, J. S. Naor, and D. Raz, “Near optimal placement of virtual network functions,” IEEE INFOCOM 2015.

[6] S. Choi, J. Turner, and T. Wolf, “Configuring sessions in programmable networks,” Computer Networks 41.2 (2003): 269-284.

[7] P. Erdős and A. Rényi, “On random graphs,” Publ. Math. 6, 290 (1959).

[8] E. N. Gilbert, “Random graphs,” Ann. Math. Stat., 30, 1141 (1959).

[9] J. Lee et al., “Application-driven bandwidth guarantees in datacenters,” ACM SIGCOMM Computer Commun. Review, Vol. 44, No. 4, 2014.