

VNF Management with Model Predictive Control for Service Chains

Masaya Kumazaki¹, Masaki Ogura² and Takuji Tachibana¹
 Graduate School of Engineering, University of Fukui, Japan¹

Division of Information Science, Nara Institute of Science and Technology, Japan²

Abstract—By using Network Function Virtualization (NFV), several kinds of network functions can be provided in commercial off-the-shelf (COTS) servers. Such network function is called Virtual Network Function (VNF). In terms of the utilization of VNFs, an emerging technology called service chaining enables network operators to provide network services dynamically and flexibly. Here, the number of VNF instances should be managed dynamically according to the amount of traffic. In this paper, we propose a dynamic management of VNF instances with Model Predictive Control (MPC) for multiple service chains. In the proposed method, the number of VNF instances is changed dynamically according to the predicted amount of traffic. We evaluate the performance of the proposed method and threshold method with simulation. In numerical examples, it is shown that the proposed method can avoid data loss with a small number of VNF instances.

I. INTRODUCTION

By using Network Function Virtualization (NFV), several kinds of network functions such as Intrusion Detection System (IDS), Firewall, switching router, WAN optimize, and Evolved Packet Core (EPC) can be provided in commercial off-the-shelf (COTS) servers [1]-[3]. Such network function is called Virtual Network Function (VNF), and VNF can be utilized to provide network functions [4].

Here, the amount of data that can be processed with a VNF depends on the number of VNF instances. In order to increase the data throughput, the number of VNF instances should be increased. A larger number of VNF instances can also avoid packet losses in a buffer at each node. However, a large number of VNF instances wastes a larger amount of node resources such as CPU and memory. Therefore, the number of instances should be managed dynamically according to the amount of traffic.

In this paper, we propose a dynamic management of VNF instances with Model Predictive Control (MPC) approach for service chains. In the proposed method, the number of VNF instances is changed dynamically by adding and removed VNF instances according to the estimated amount of traffic. In order to change the number of VNF instances appropriately, MPC is utilized for each service chain. We evaluate the performance of the proposed method with simulation and investigate the effectiveness of our proposed method.

The rest of this paper is organized as follows. Section II explains our proposed method including MPC formulation and Section IV shows some numerical examples. Finally, Section V presents our conclusions.

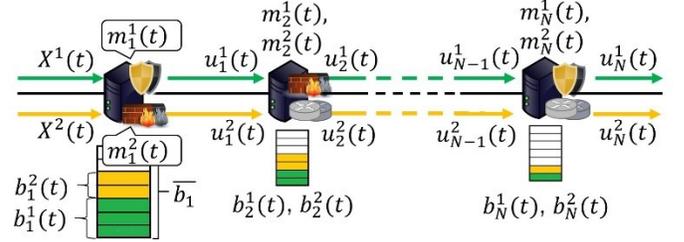


Fig. 1. System model.

II. MPC FORMULATION

Now, we focus on K service chains consisting of N nodes and the k th service chain is denoted as f_k (Fig. 1). The number of VNF instances can be added and removed dynamically. Let \bar{m}_i denote the maximum number of VNF instances that can be placed on the node n_i , and let $m_i^k(t)$ denote the number of VNF instances for the service chain f_k on the node n_i at time t . We also denote by $X^k(t)$ the amount of traffic flowing into the first node at time t and by C_i^k the amount of data of f_k that the node n_i can process from t to $t+1$. In addition, a buffer in each node can store up to \bar{b}_i data. Finally, let $b_i^k(t)$ denote the amount of data for f_k on the node n_i at time t and $u_i^k(t)$ denote the amount of data for f_k processed in n_i from $t-1$ to t .

Then, the amount of stored data dynamically evolves by the following difference equation:

$$b_1^k(t+1) = b_1^k(t) - \delta_1^k(t) + X^k(t+1) - u_1^k(t+1), \quad (1)$$

$$b_i^k(t+1) = b_i^k(t) - \delta_i^k(t) + u_{i-1}^k(t+1) - u_i^k(t+1), \quad (2)$$

$$u_1^k(t+1) = \min(b_1^k(t) - \delta_1^k(t) + X^k(t+1), C_1^k m_1^k(t)), \quad (3)$$

$$u_i^k(t+1) = \min(b_i^k(t) - \delta_i^k(t) + u_{i-1}^k(t+1), C_i^k m_i^k(t)). \quad (4)$$

In these equations, $u_i^k(t+1)$ denotes the amount of data for f_k processed in node n_i from t to $t+1$ and $\delta_i^k(t)$ denotes the amount of rejected data for f_k in the node n_i at the time t .

We also assume that an instance of VNFs can be added or removed each time for each node. The addition (removal) of an instance takes S_i^k (R_i^k , respectively) units of time. Therefore, if we introduce the following binary variables

$$x_i^k(t) = \begin{cases} 1, & \text{a VNF instance is added,} \\ 0, & \text{otherwise,} \end{cases}$$

$$y_i^k(t) = \begin{cases} 1, & \text{a VNF instance is removed,} \\ 0, & \text{otherwise,} \end{cases}$$

then the number of VNF instances obeys the following difference equation.

$$m_i^k(t+1) = m_i^k(t) + x_i^k(t+1 - S_i^k) - y_i^k(t+1 - R_i^k). \quad (5)$$

Our objective in this paper is to minimize the number of used VNFs while avoiding data losses during the operation of the service chains. For this purpose, we introduce the variable $\zeta_i(t)$ defined by

$$\zeta_i(t) = \max\left(0, \sum_{k=1}^K b_i^k(t) - \kappa_i \bar{b}_i\right), \quad (6)$$

where κ_i is set to determine the acceptable amount of data stored in each buffer. (6) indicates the amount of data flowed beyond a certain amount of buffer storage that is determined by the network administrator.

We then formulate the following optimization problem for dynamically and optimally managing the number of VNF instances.

$$\min_{x,y} \sum_{\tau=t}^{t+T} \left\{ (1 - \alpha - \beta) \frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K \left[\frac{m_i^k(\tau)}{\bar{m}_i} \right] + \alpha \max_{1 \leq i \leq N} \frac{[\zeta_i(\tau)]}{(1 - \kappa_i) \bar{b}_i} + \beta \frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K [\psi_i^k(\tau)] \right\}, \quad (7)$$

subject to:

$$x_i^k(\tau) y_i^k(\tau) = 0, \forall i, \forall k, \forall \tau, \quad (8)$$

$$K \leq \sum_{k=1}^K m_i^k(\tau) \leq \bar{m}_i, \forall i, \forall \tau, \quad (9)$$

$$\sum_{k=1}^K x_i^k(\tau) \leq 1, \sum_{k=1}^K y_i^k(\tau) \leq 1, \forall i, \forall \tau. \quad (10)$$

In this optimization problem, (2), (4), and (5) are also used as constraint conditions. Here, T is the length of predictive horizon. In addition, $\psi_i^k(t)$ denote the number for the addition and removal of VNF instance, and it obeys the following equation.

$$\psi_i^k(t) = \mu x_i^k(t) + (1 - \mu) y_i^k(t). \quad (11)$$

Here, μ is a weighting factor for the addition of VNF instances. In this optimization problem, α is the weighting factor for data loss and β is the weighting factor for the number of times for the addition and removal of VNF instances. In addition, (8) guarantees that a VNF instance cannot be added and removed at the same time and (9) denotes a constraint for the number of VNF instances for each node n_i . Moreover, (10) guarantees that the number of times for the addition and removal of VNFs is less than one for all nodes n_i at the same time.

III. NUMERICAL EXAMPLE

In this section, we evaluate the performance of the proposed method for a network where the number of service chains is from two to four.

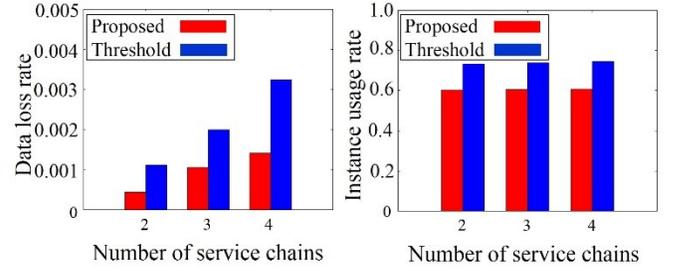


Fig. 2. Performance of our proposed method against number of service chains

For the each service chain, the amount of data flowing into the first node n_1 at time t is determined by the sum of multiple sine functions. Here, our optimization problem was solved with a software called CPLEX. We also evaluate the performance of another method called threshold method. In this method, VNF instances are added and removed when the amount of data in each buffer becomes larger (smaller) than the upper (lower) threshold.

Figure 2 shows the data loss rate and the usage rate of VNF instances. In this figure, $T = 7$, $S_i^k = 4$, and $R_i^k = 1$. From this figure, we can find that the data loss rate for the proposed method is smaller than the threshold method regardless of the number of service chains. VNF instance usage rate for the proposed method is also smaller than that for the threshold method. As a result, even if the number of service chains changes, our proposed method can decrease the data loss rate by utilizing VNF instances.

IV. CONCLUSION AND FUTURE WORK

In this paper, we proposed a dynamic VNF management with MPC for multiple service chains. In the proposed method, a VNF instance can be added and removed dynamically by using MPC so as to avoid the data loss while using VNF instances effectively. In our future work, we will manage the placement of VNF with MPC.

REFERENCES

- [1] K. Shiimoto, "Research Challenges for Network Function Virtualization - Re-Architecting Middlebox for High Performance and Efficient, Elastic and Resilient Platform to Create New Services -," *IEICE Transactions on Communications*, vol. E101-B, no. 1, pp. 96-122, Jan. 2018.
- [2] N. Herbaut, D. Negru, Y. Chen, P. A. Frangoudis, and A. Ksentini, "Content Delivery Networks as a Virtual Network Function: A Win-Win ISP-CDN Collaboration," in *Proc. IEEE Global Communications Conference 2016 (GLOBECOM 2016)*, Dec. 2016.
- [3] Y. Kanizo, O. Rottenstreich, I. Segall, and J. Yallouz, "Optimizing Virtual Backup Allocation for Middleboxes," *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 2759-2772, Oct. 2017.
- [4] P. Chi, Y. Huang, and C. Lei, "Efficient NFV Deployment in Data Center Networks," in *Proc. IEEE International Conference on Communications 2016 (ICC 2016)*, Jun. 2015.