Dynamic Placement of Virtual Network Functions based on Model Predictive Control

Kota Kawashima, Tatsuya Otoshi, Yuichi Ohsita, and Masayuki Murata
Graduate School of Information Science and Technology, Osaka University
1–5 Yamadaoka, Suita, Osaka, 565–0871, Japan
Email: {k-kawashima, t-otoshi, y-ohsita, murata}@ist.osaka-u.ac.jp

Abstract—**Dynamic placement of the virtual network functions (VNFs) is one of the promising approaches to handling time-varying demands; when demands are small, the energy consumption can be reduced by placing the VNFs to a small number of physical nodes and shutting down unused nodes. If the demands becomes large, the VNFs are migrated to allocate the sufficient resources. In the dynamic placement of the VNFs, it is important to avoid a large number of migrations at each time because the migration requires a large amount of bandwidth. In this paper, we propose a new method to dynamically place the VNFs to follow the traffic variation without migrating a large number of VNFs. Our method is based on the model predictive control (MPC). By applying the MPC to the dynamic placement of the VNFs, our method starts migration in advance by considering the predicted future demands. As a result, our method allocates sufficient resources to the VNFs without migrating a large number of VNFs at the same time even when traffic variation occurs. Through simulation, we demonstrate that our method handles the time variation of the demands without requiring a large number of migration at any time slot.**

I. INTRODUCTION

In recent years, the time variation of the Internet traffic has increased due to the emerging new applications such as cloud computing. Backbone networks must accommodate such time-varying traffic. However, the increase of the time variation of the traffic causes the difficulty in accommodating traffic in a single static backbone network; the network that can handle any possible traffic requires a large amount of resources and consumes a large amount of energy.

The network function virtualization (NFV) is one of the promising approaches to handling such time variations efficiently [1]. In the NFV, the network functions such as routers and firewalls are virtualized. The virtual network functions (VNFs) are hosted by ordinary server computers. The network services are provided through the virtual network constructed of the VNFs. The VNFs can be migrated by using the live migration technologies [2]. By dynamically placing the VNFs to the suitable server, the network services are provided without wasting the resources and energy consumption; the VNFs are hosted by a small number of servers when the demand is small and each VNF requires only small resources. If the demand increases and more resources become required, the VNFs are migrated to new servers [3].

The problem to obtain the suitable servers hosting the VNFs can be formulated as the *virtual network embedding problem*, and has been investigated in many papers [3]–[10]. These methods obtain the optimal locations of the VNFs and the topology of the virtual network that minimize the objective functions such as the amount of the used resources. Then, the network is reconfigured based on the obtained solution by migrating the VNFs or changing the configuration of the routing.

In the dynamical placement of the VNFs, the cost of the migration of the VNFs is important, because the migration consumes network resources. Thus, migrating a large number of VNFs at the same time should be avoided. The method proposed by Blenk et al. [10] considers the cost of the migration by minimizing the weighted sum of the performance metrics and the cost of the migration. This method considers only the currently required resources, and does not perform migration unless the necessity of the migration is detected However, when the necessity of the migration is detected, a large number of migrations may be required. If we can detect the necessity of the migration in the future time slots from the predicted demands, we can start the migration in advance and avoid a large number of migration at each time slot.

In this paper, we propose a new method to dynamically place the VNFs so as to follow the traffic variation without migrating a large number of VNFs. Our method is based on the model predictive control (MPC). In MPC, a controller inputs the system parameters so as to maintain the output of the system at close to a target value. The controller calculates the optimal input values for the future time slots based on the prediction, but implements the input values only for the next time slot. Then, the controller observes the output and corrects the prediction by using the observed output as feedback. After the correction, the controller calculates the optimal input values again by using the corrected prediction. By repeating the above steps, the controller calculate suitable input for the future time slot even when prediction errors occur. We have already applied the MPC to the dynamic route control [11], and showed that dynamic route control based on the MPC avoids the congestion by changing the routes in advance.

By applying the MPC to the dynamic placement of the VNFs, our method starts migration in advance of the change of the required resources by considering the predicted future demands. As a result, our method handles time variations of the required resources without migrating a large number of VNFs at the same time.

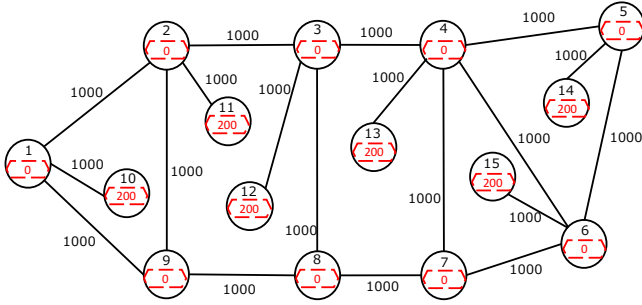This rest of this paper is organized as follows. Section II

Fig. 1. Physical network.

formalizes the problem of the placement of the VNFs and the construction of the virtual network. Section III explains our method for the dynamic placement of the VNFs based on the MPC. Section IV evaluates our method. Finally, Section V presents our concluding remarks.

II. PROBLEM FORMULATION

In this paper, we maps the virtual network to the physical network. This section formulates this problem as an optimization problem.

A. Physical network

Figure 1 shows a physical network. The physical network is constructed of physical nodes and physical links. We denote the set of physical nodes by N^p, and the set of physical links by L^p. The physical network is modeled by a weighted directed graph $G^p = (N^p, L^p)$. The bandwidth of the physical link $l \in L^p$ is denoted by B_l^p. The resource of the physical node $n \in N^p$ is denoted by a vector U_n^p, whose number of elements corresponds to the number of kinds of resources such as CPU and memory. We denote the number of kinds of resources by R, and ith element of U_n^p by $u_{n,i}^p$.

In the physical network, multiple paths exist between two physical nodes. Among them, we consider k shortest paths. We denote the set of the all k shortest paths between all node pairs by P^p. Each path $p \in P^p$ is defined by the set of physical links on the path. n_p^{start} and n_p^{end} indicate the first and last nodes on the path p, respectively.

We also define a matrix A^p whose element $a_{i,j}^p$ is 1 when path i goes through link j; otherwise, 0.

B. Virtual Network

Figure 2 shows a virtual network. The virtual network is constructed of the virtual nodes and virtual links. We denote the set of virtual nodes by N^v, and the set of virtual links by L^v. The virtual network is modeled by a weighted directed graph $G^v = (N^v, L^v)$.

In this paper, we consider two kinds of virtual nodes, *VNF nodes* and *user nodes*. Each VNF node corresponds to a VNF, and must be mapped to the physical node that has sufficient resource to run the VNF. The user nodes correspond to the users who use the network service. The location of each user
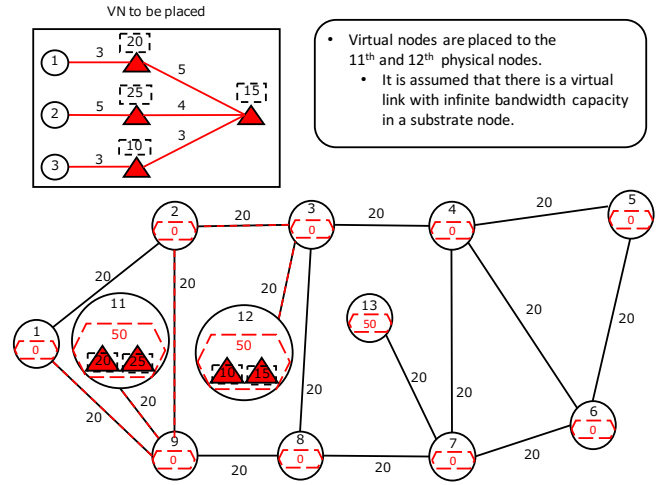

Fig. 2. Overview of placement of virtual network.

node is fixed. We denote the set of the VNF nodes by N^{VNF}, and the set of the user nodes by N^{user}. $N^v = N^{\text{VNF}} \cup N^{\text{user}}$.

Each VNF node $n^{\text{vnf}} \in N^{\text{VNF}}$ requires the resources of the physical node. The required resources change in time. We denote the resource required by n^{vnf} at the time slot t by a vector $U_{n^{\text{vnf}}}^v(t)$. The location of the user node $n^{\text{user}} \in N^{\text{user}}$ is denoted by $n_{n^{\text{user}}}^p \in N^p$.

The virtual link requires the bandwidth. The required bandwidth changes in time. We denote the bandwidth required by $l \in L^v$ at the time slot t by $B_l^v(t)$.

n_l^{start} and n_l^{end} indicate the source and destination nodes of the virtual link $l \in L^v$.

C. Optimization Problem

In this paper, we map the virtual network to the physical network. Each VNF node must be mapped to the physical node that has the sufficient resources to host the VNF. In addition, the virtual links must be mapped to the physical paths so that all physical links on the paths have the sufficient bandwidth to accommodate the virtual link. Therefore, we calculate the suitable mapping that satisfies these constraints.

We minimize the number of active physical nodes that host the VNFs, though there may be other objective functions. By minimizing the number of active physical nodes and sleeping the servers on the other physical nodes, we can reduce the energy consumption.

When mapping the virtual network to the physical network, the cost of the migration should be considered. Therefore, we also minimize the number of migrated VNFs in addition to the number of active physical node.

We formulate the problem to map the virtual network to the physical network at the time slot t by the optimization problem. In this optimization problem, we set the following variables.

- $M_{v,n}^{Node}$: A binary variable, which is 1 if the virtual node v is hosted by the physical node n; otherwise, 0.

- M_n^{Node} : A binary variable, which is 1 if at least one VNF node is hosted by the physical node n; otherwise, 0.
- $M_{l,p}^{Link}$: The ratio of the traffic amount on the virtual link l passing through the physical path p.

In this optimization problem, the information of the physical network G^p and the information of the virtual network G^v, including the predicted values of the required resources of the VNFs and the required bandwidths, are given as input. In addition, the locations of the VNFs at the previous time step $M_{v,n}^{prev}$ are also given as input. $M_{v,n}^{prev}$ is set to 1 if the VNF node v is hosted by the physical node n, otherwise 0. w is a parameter indicating the weight to the cost of the migration; setting w to a large value avoids migration.

$$minimize \quad (1-w)\sum_{n\in N^p} M_n^{Node} +$$

$$w\sum_{n\in N^p}\sum_{n^{\mathrm{vnf}}\in N^{\mathrm{VNF}}} |M_{n^{vnf},n}^{Node} - M_{n^{vnf},n}^{prev}| \tag{1}$$

$$subject\ \ to$$

$$\forall n \in N^p, \frac{1}{|N^{\mathrm{VNF}}|}\sum_{n^{\mathrm{vnf}}\in N^{\mathrm{VNF}}} M_{n^{\mathrm{vnf}},n}^{Node} \le M_n^{Node} \tag{2}$$

$$\forall n^{\mathrm{vnf}} \in N^{\mathrm{VNF}}, \sum_{n\in N^p} M_{n^{\mathrm{vnf}},n}^{Node} = 1 \tag{3}$$

$$\forall n^{\mathrm{user}} \in N^{\mathrm{user}}, \forall n^p \in N^p, M_{n^{\mathrm{user}},n^p}^{Node} = \begin{cases} 1 & (n_{n^{\mathrm{user}}}^p = n^p) \\ 0 & (otherwise) \end{cases} \tag{4}$$

$$\forall l \in L^v, \sum_{p\in P^p} M_{l,p}^{Link} = 1 \tag{5}$$

$$\forall l \in L^v, \forall p \in P^p, M_{l,p}^{Link} \le M_{n_l^{start},n_p^{start}}^{Node} \tag{6}$$

$$\forall l \in L^v, \forall p \in P^p, M_{l,p}^{Link} \le M_{n_l^{end},n_p^{end}}^{Node} \tag{7}$$

$$\forall n \in N^p, \forall u_{n^{\mathrm{vnf}},i}^v(t) \in U_{n^{\mathrm{vnf}}}^v(t),$$
$$\sum_{n^{\mathrm{vnf}}\in N^{\mathrm{VNF}}} M_{n^{\mathrm{vnf}},n}^{Node} \cdot u_{n^{\mathrm{vnf}},i}^v(t) \le u_{n,i}^p \tag{8}$$

$$\forall l^p \in L^p, \sum_{l\in L^v}\sum_{p\in P^p} a_{p,l}^p \cdot M_{l,p}^{Link} \cdot B_l^v(t) \le B_l^p \tag{9}$$

In the above optimization problems, Eq. (2) defines the relation between $M_{n^{\mathrm{vnf}},n}^{Node}$ and M_n^{Node}. Eq. (3) ensures that each VNFs must be hosted by one of the physical nodes. Eq. (4) sets the locations of the user nodes. Eqs. (5), (6) and (7) ensure that each virtual link must be accommodated by the paths whose source and destination nodes are the nodes hosting the virtual nodes at the both edges of the virtual link. Eq. (8) ensures that the virtual nodes are mapped to the physical nodes with the sufficient resources. Similarly Eq. (9) ensures that the virtual links are mapped to the physical links with the sufficient bandwidths.

III. PLACEMENT OF VIRTUAL NETWORK FUNCTIONS BASED ON MPC

A. Model Predictive Control

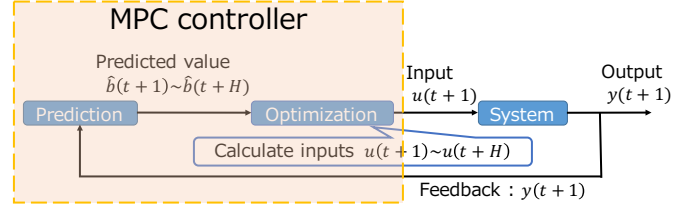Figure 3 shows a overview of MPC [12]. The MPC controller predicts the operation of system for the future time
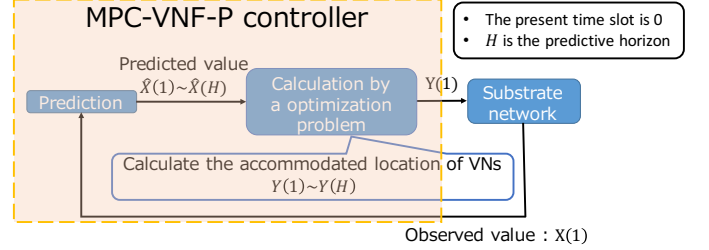

Fig. 3. Overview of MPC.

Fig. 4. Overview of MPC-VNF-P.

slots $[t+1,...t+H]$ called predictive horizon where H is the length of the predictive horizon. Based on the prediction, the controller calculates the inputs for the predictive horizon. However, the controller implements only the calculated inputs for the next time slots $[t+1]$. Then, the controller observes the output and corrects the prediction, using the output value. After the correction, the controller recalculates the inputs for the next time slot with the corrected prediction. Thus, the MPC controller modifies the prediction by using feedback. By using and modifying the prediction, the MPC achieves a prediction–based control that is robust to prediction errors.

B. MPC-VNF-P

In this paper, we apply the MPC to the dynamical placement of the VNFs, which is formulized in Section II-C. We call this method *MPC-VNF-P*. In this method, we consider the predicted future values of the required resources. By considering the predicted future values, we can start migration in advance of the changes of the required resources. As a result, we can follow the time variation of the required resources without a large number of migrations at each time slot.

Figure 4 shows an overview of MPC-VNF-P. The MPC-VNF-P controller (1) predicts the required resources of virtual nodes and virtual links for each time slot t $(1 \le t \le H)$, (2) calculates the physical servers hosting the VNFs and topology of the virtual network for future H time slots, and (3) performs the migration and configures the routes according to the calculated results for the next time slot. Then, at the next time slot, the controller obtains the new information on the required resources, and perform the above steps again. By continuing these steps, the MPC-VNF-P controls the locations of the VNFs considering the future required resources. In addition, even if the prediction errors are included in the predicted future required resources, the impact of the prediction errors is avoided by correcting the prediction at each step.

We formulate the MPC-VNF-P as follows. The MPC-VNF-P uses the predicted values of the required resources for the time slots $[t+1, ..., t+H]$ as input, while the problem formulated in Section II-C uses the predicted values of the required resources only at the next time slot $t+1$. The predicted value of resource required by n^{vnf} at time t is $\hat{U}^v_{n^{\mathrm{vnf}}}(t)$. The predicted value of required bandwidth of the virtual link $l \in L^v$ at the time slot t is $\hat{B}^v_l(t)$.

We define variables which indicate the location of virtual nodes and virtual links at the time slot t by $M^{Node}_{v,n}(t)$, $M^{Link}_{l,p}(t)$ and $M^{Node}_n(t)$. The placement of the VNFs can be calculated by the following optimization problem.

$$minimize \quad \frac{(1-w)}{H \cdot |N^p_f|} \sum_{0 < k \le H} \sum_{n \in N^p} M^{Node}_n(t) +$$

$$\frac{w}{2|N^v_{j,r}|} \overline{M} \tag{10}$$

$$subject \ \ to$$

$$0 < t \le H, \forall n \in N^p,$$

$$\frac{1}{|N^{\mathrm{VNF}}|} \sum_{n^{\mathrm{vnf}} \in N^{\mathrm{VNF}}} M^{Node}_{n^{\mathrm{vnf}},n}(t) \le M^{Node}_n(t) \tag{11}$$

$$0 < t \le H, \forall n^{\mathrm{vnf}} \in N^{\mathrm{VNF}}, \sum_{n \in N^p} M^{Node}_{n^{\mathrm{vnf}},n}(t) = 1 \tag{12}$$

$$0 < t \le H, \forall n^{\mathrm{user}} \in N^{\mathrm{user}}, \forall n^p \in N^p,$$

$$M^{Node}_{n^{\mathrm{user}},n^p}(t) = \begin{cases} 1 & (n^p_{n^{\mathrm{user}}} = n^p) \\ 0 & (otherwise) \end{cases} \tag{13}$$

$$0 < t \le H, \forall l \in L^v, \sum_{p \in P^p} M^{Link}_{l,p}(t) = 1 \tag{14}$$

$$0 < t \le H, \forall l \in L^v, \forall p \in P^p,$$

$$M^{Link}_{l,p}(t) \le M^{Node}_{n^{start}_l, n^{start}_p}(t) \tag{15}$$

$$0 < t \le H, \forall l \in L^v, \forall p \in P^p,$$

$$M^{Link}_{l,p}(t) \le M^{Node}_{n^{end}_l, n^{end}_p}(t) \tag{16}$$

$$0 < t \le H, \forall n \in N^p, \forall \hat{u}^v_{n^{\mathrm{vnf}},i}(t) \in \hat{U}^v_{n^{\mathrm{vnf}}}(t),$$

$$\sum_{n^{\mathrm{vnf}} \in N^{\mathrm{VNF}}} M^{Node}_{n^{\mathrm{vnf}},n}(t) \cdot \hat{u}^v_{n^{\mathrm{vnf}},i}(t) \le u^p_{n,i} \tag{17}$$

$$0 < t \le H, \forall l^p \in L^p,$$

$$\sum_{l \in L^v} \sum_{p \in P^p} a^p_{p,l} \cdot M^{Link}_{l,p}(t) \cdot \hat{B}^v_l(t) \le B^p_l \tag{18}$$

$$0 < t \le H,$$

$$\sum_{n \in N^p} \sum_{n^{\mathrm{vnf}} \in N^{\mathrm{VNF}}} |M^{Node}_{n^{\mathrm{vnf}},n}(t) - M^{Node}_{n^{\mathrm{vnf}},n}(t-1)| \le \overline{M} \tag{19}$$

In this optimization problem, Eqs. (11)–(18) are the similar constraints to the problem formulated in Section II-C. In Eq. (19), we introduce a variable $\overline{M}$, which indicates the largest number of migrated VNFs within the predictive horizon.

This optimization problem minimizes the weighted sum of the number of active physical nodes and the expected number of migrated VNFs in the future time slot. By this optimization problem, MPC-VNF-P controls the locations of the VNFs without a large number of migrations, so as to minimize the
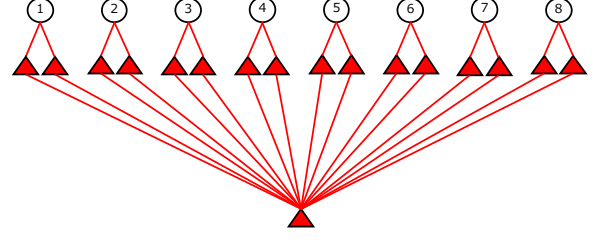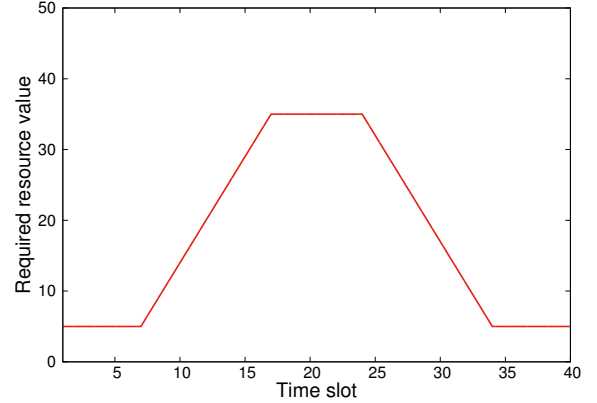

Fig. 5. Virtual network

Fig. 6. Time variation of the required resources

number of active physical nodes under the constraint that the VNFs are mapped to the physical nodes with sufficient resources and the virtual links are mapped to the physical links with sufficient bandwidths.

IV. EVALUATION

A. Simulation Environment

1) Physical Network: We use the topology shown in Figure 1, which are based on the backbone network of the Internet2 [13]. In this topology, six nodes are connected to the servers. Only the servers have the resources to host the VNFs. For the simplicity, we consider the case that the number of the kinds of the resources is 1, and each server has the resource whose capacity is 200. We set the bandwidth of each link to a sufficiently large value to focus on the impact of the time variation of the resources required by the VNFs. In this evaluation, we set three shortest paths for each node pair as the candidate paths used to accommodate virtual links.

2) Virtual network: In this evaluation, the virtual network includes 8 user nodes and 17 VNFs, shown in Figure 5. This virtual network includes three kinds of the VNFs. Two of them handles the traffic near user and are connected to user nodes. The other VNF is connected to all of the VNFs connected to user nodes.

We generate the time variations of the required resources. To demonstrate our method in the case that the total required resources increases or decreases, we generate the same time variation of the required resources for all VNFs. Figure 6 shows generated the time variation of the required resources for each VNFs.

3) Prediction Method: In our evaluation, we use the following simple prediction method. First, we find a best-fit straight line $l_k = a_k + b$ which minimizes the sum of squared distance from the previous observed required resource values $u^v_{n^{\text{vnf}}}(t-2)$ and $u^v_{n^{\text{vnf}}}(t-1)$ denoted as $\sum^2_{k=1}(u^v_{n^{\text{vnf}}}(t-k) - lt - k)^2$. Then, we obtain the future required resource value as $\hat{u}^v_{n^{\text{vnf}}}(t+k) = l_{t+k}$.

By using this prediction method in our evaluation, the prediction errors occur at the time slot 7, 17, 24, and 34.

4) Compared methods: We compare the following method.

MinActiveNode: This method decides the location of the VNFs so as to minimize the number of active physical nodes hosting the VNFs without considering the cost of migration. This method solves the optimization problem in Section II with $w = 0$. By comparing the MPC-VNF-P with this method, we clarifies the impact of considering the cost of the migration.

NoMPC: This method decides the location of the VNFs considering the cost of migration without considering the future variations of the required resources. This method uses the predicted required resources only at the next time slot, and solves the optimization problem in Section II with $w = 0.03$. By comparing the MPC-VNF-P with this method, we clarifies the impact of considering the future values of the required resources.

In addition to the above methods, we compare of the two cases of parameters in the MPC-VNF-P; the case with $H = 3$ and the case with $H = 5$. In all methods, we solve the optimization problem by using Cplex 11.0 [14].

5) Metrics: In this evaluation, we use the following metrics.

Maximum resource utilization: The largest resource utilization among all physical nodes at each time slot, which is defined by

$$\max_{n^p \in N^p} \left(\frac{1}{u^p_n} \sum_{n^{\text{vnf}} \in N^{\text{VNF}}_{n^p}} u^v_{n^{\text{vnf}}} \right) \quad (20)$$

where $N^{\text{VNF}}_{n^p}$ is the set of virtual nodes hosted by the physical node $n^p \in N^p$.

Number of active physical nodes: the number of physical nodes hosting at least one VNFs.

Number of migrated VNFs the number of VNFs which are migrated at each time slot.

B. Results

Figure 7 shows the time variation of the maximum resource utilization. In this figure, the horizontal axis is the time slot, and the vertical axis is the maximum resource utilization. This figure indicates that all methods map the virtual network properly so that the resource utilizations do not become larger than 1. This is because all methods use the predicted values of the required resources, and migrate VNFs in advance before the lack of the resource occurs.

Figure 8 shows the number of active physical nodes. In this figure, the vertical axis indicates the number of active physical nodes, and the horizontal axis indicates the time
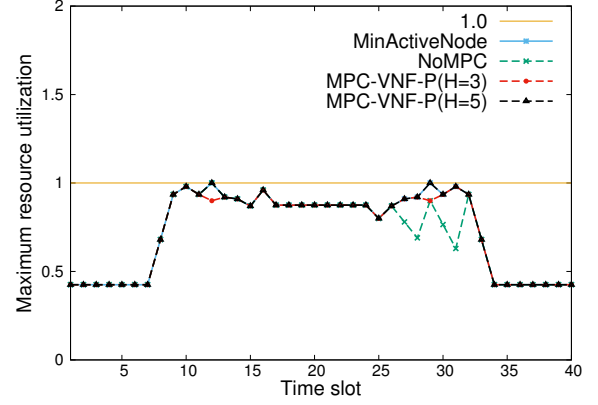

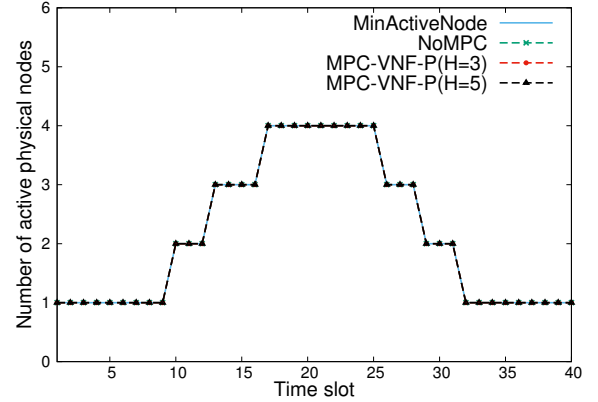Fig. 7. The maximum resource utilization.

Fig. 8. The number of active physical nodes.

slot. This figure demonstrates that all methods increases or decreases the number of active physical nodes according to the time variation of the required resources; the number of active physical nodes increases from time slot 10 to 17, and decreases from time slot 26 to 32. The number of active physical nodes achieved by the MPC-VNF-P is as small as that by NoMPC. That is, the MPC-VNF-P achieves the smallest number of active physical nodes, though the MPC-VNF-P uses the predicted future values of the required resources. Especially at time slot 17, the future required resources are predicted to increase continuously, while the actual required resources stop increasing. Due to this prediction errors, the MPC-VNF-P plans to migrate the VNFs so as to handle the predicted increase. But, at the next time slot, the MPC-VNF-P corrects the prediction errors by using the newly obtained information on the required resources, and calculates the locations of the VNFs again. As a result, even though the MPC-VNF-P uses the future predicted values of the required resources, whose prediction errors becomes large as the prediction target is far ahead, the MPC-VNF-P avoids the increase of the number of active physical nodes.

Finally, we compare the number of migrated VNFs at each time slot. Figure 9 shows the results. In this figure, the vertical axis is the number of migrated VNFs at each time slot, and the horizontal axis is the time slot. This figure shows
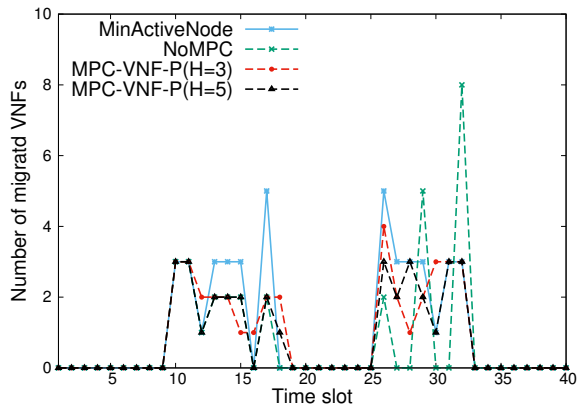
Fig. 9. The number of migrated VNFs.

that MinActiveNode requires a large number of migrations at several time slot, because MinActiveNode does not consider the cost of the migration. NoMPC also requires a large number of migration especially when the required resources decrease. This is because NoMPC does not consider the future values of the required resources and considers only the required resources at the next time slot. As a result, NoMPC does not start migration even when the required resources decreases, unless the migration decreases the number of active physical nodes. On the other hand, the MPC-VNF-P considers the future values of the required resources. Therefore, the MPC-VNF-P starts the migration if the migration is expected to decrease the number of active physical nodes at the future time slots, even when the migration cannot decrease the number of active physical nodes at the next time slot. As a result, the MPC-VNF-P decreases the number of active physical nodes according to the time variation of the required resources without a large number of migrations at any time slot.

V. Conclusion

In this paper, we proposed a new method to dynamically place the VNFs based on the MPC. By applying the MPC to the dynamic placement of the VNFs, our method starts migration in advance of traffic variation by considering the predicted future demands. As a result, our method allocates sufficient resources to the VNFs without migrating a large number of VNFs at the same time even when traffic variation occurs. Through simulation, we demonstrated that our method handles the time variation of the demands without requiring a large number of migration at any time slot.

Our future work includes the evaluation of our method using the actual traffic traces. In addition, we plan to establish a distributed algorithm of the dynamic placement of the VNFs to handle frequent changes in required resource in a large network.

References

[1] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. D. Turck, and R. Boutaba, "Network Function Virtualization: State-of-the-art and Research Challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 236–262, First Quarter 2016.

[2] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in *Proceedings of the 2Nd Conference on Symposium on Networked Systems Design & Implementation - Volume 2*, ser. NSDI'05. Berkeley, CA, USA: USENIX Association, 2005, pp. 273–286. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251203.1251223

[3] H. Moens and F. D. Turck, "VNF-P: A model for efficient placement of virtualized network functions," in *Network and Service Management (CNSM), 2014 10th International Conference on*, Nov. 2014, pp. 418–423.

[4] A. Fischer, J. F. Botero, M. T. Beck, H. de Meer, and X. Hesselbach, "Virtual Network Embedding: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 1888–1906, Fourth Quarter 2013.

[5] M. Bienkowski, A. Feldmann, D. Jurca, W. Kellerer, G. Schaffrath, S. Schmid, and J. Widmer, "Competitive analysis for service migration in vnets," in *Proceedings of the Second ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures*, ser. VISA '10. New York, NY, USA: ACM, 2010, pp. 17–24. [Online]. Available: http://doi.acm.org/10.1145/1851399.1851403

[6] Z. Shun-li and Q. Xue-song, "A Novel Virtual Network Mapping Algorithm for Cost Minimizing," *Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT)*, vol. 2, no. 1, pp. 1–9, Jan. 2011.

[7] C. C. Marquezan, L. Z. Granville, G. Nunzi, and M. Brunner, "Distributed Autonomic Resource Management for Network Virtualization," in *IEEE/IFIP Network Operations and Management Symposium*, Apr. 2010, pp. 463 – 470.

[8] G. Schaffrath, S. Schmid, and A. Feldmann, "Optimizing Long-Lived CloudNets with Migrations," in *Proc. 5th IEEE/ACM International Conference on Utility and Cloud Computing (UCC)*, 2012.

[9] I. Fajjari, N. Aitsaadi, G. Pujolle, and H. Zimmermann, "VNR Algorithm: A Greedy Approach For Virtual Networks Reconfigurations," in *Proceedings of IEEE Globecom 2011*, Dec. 2011, pp. 1–5.

[10] A. Blenk and W. Kellerer, "Traffic pattern based virtual network embedding," in *Proceedings of the 2013 Workshop on Student Workhop*, ser. CoNEXT Student Workhop '13. New York, NY, USA: ACM, 2013, pp. 23–26. [Online]. Available: http://doi.acm.org/10.1145/2537148.2537151

[11] T. Otoshi, Y. Ohsita, M. Murata, Y. Takahashi, N. Kamiyama, K. Ishibashi, K. Shiomoto, and T. Hashimoto, "Traffic engineering based on model predictive control," *IEICE Transactions on Communications*, vol. E98.B, no. 6, pp. 996–1007, Jun. 2015.

[12] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, Jul. 2003.

[13] "Internet2 Network NOC." [Online]. Available: http://noc.net.internet2.edu/i2network/index.html

[14] "IBM ILOG CPLEX Optimizer," optimization software. [Online]. Available: http://www-01.ibm.com/software/integration/optimization/cplex-optimizer