Flexible Bandwidth-Based Virtual Network Embedding

Cihangir Beşiktaş[1], Didem Gözüpek[1], Aydın Ulaş[2], Erhan Lokman[2], Özcan Özyurt[2], Kazım Ulusoy[2]
[1]Department of Computer Engineering, Gebze Technical University, Kocaeli, Turkey
[2]Argela Technologies, Istanbul, Turkey
Emails: {cihangirbesiktas, didem.gozupek}@gtu.edu.tr, {aydin.ulas, erhan.lokman, ozcan.ozyurt, kazim.ulusoy}@argela.com.tr

***Abstract*—Virtual network operators may distrust each other and require that their virtual infrastructure is not cohosted on the same physical equipment. In this paper, we formulate a virtual network embedding problem that maximizes the total revenue where conflicting virtual network operators utilize distinct physical devices. Virtual links in our model have the option of selecting among a range of discrete bandwidth values, which have a corresponding price. This way, any revenue function can be realized. We evaluate the performance of our model by comparing it with a fixed bandwidth scheme and demonstrate its superior performance in terms of revenue.**

I. Introduction

Network virtualization addresses the problem of ossification in the Internet by enabling the rapid deployment of new technologies without altering the physical infrastructure. In this approach, the substrate (physical) network provider offers a substrate (physical) network to support virtual networks and a *virtual network operator (VNO)* manages the virtual network [1]. *Virtual Network Embedding (VNE)*, a.k.a. Virtual Network Assignment, problem is to map virtual networks to substrate network resources subject to some constraints like CPU requirements and optimizing a certain objective function such as energy efficiency. Authors in [2] proved that the VNE problem is NP-hard by reduction from the multi-way separator problem. Another work [3] has proved by reduction from the unsplittable flow problem that the problem is still NP-hard even when the virtual nodes are already assigned and the problem is merely to make the virtual link assignments by adhering to the bandwidth requirements.

Works in the literature analyzed many variants of this problem such as CPU, disk, bandwidth, and memory requirements of the substrate and virtual links, maximum length requirement for the virtual paths, requirement on the maximum number of virtual nodes or links that can be assigned to a certain substrate node or link, and economical benefits [4]–[11].

The work in [1] indicated an important open research issue: Different VNOs may distrust one another and require that their virtual infrastructure is not cohosted on the same physical equipment [1]; therefore, VNE problems that handle this constraint is of paramount importance. In this paper, we address this open research issue by providing a virtual network embedding formulation that ensures that the resources allocated to the conflicting virtual networks do not share the same physical resources.

Many studies on VNE problem assume that each link has a fixed bandwidth request [5], [8]–[19]. The work in [20] relaxes this assumption by only probabilistically satisfying the bandwidth requirements of some links. Setting a fixed bandwidth requirement for each virtual link can cause feasibility problems. Furthermore, it is unrealistic since a range of bandwidth values instead of a single one usually better fits the requirement of virtual links. In addition, most works in the literature focus on a linear revenue function of total bandwidth [5], [8]–[19]. The work in [21] mentions that a linear revenue function is unrealistic and therefore proposes an exponential function. On the one hand, the infrastructure provider in our model offers a range of discrete bandwidth values, each with a corresponding price. On the other hand, each virtual link has a minimum and maximum bandwidth requirement depending on the quality of service requirement of its application (email, video etc.). Our VNE formulation maximizes the revenue of the infrastructure provider such that the requirement of the virtual links are satisfied. This flexibility in bandwidth assignment results in higher revenue. Moreover, unlike other works in the literature, our model realizes any revenue function.

The enormous growth of Internet has also incapacitated the current distributed architecture based on the concept of Autonomous Systems (AS). The current Internet routers are responsible for both routing decisions and data transmission. The fact that routing decisions are made in a distributed manner in such a large network causes many inefficiencies such as loops and inconsistent routing tables. Management and control of the network also becomes very difficult. Furthermore, the fact that the control plane which is in charge of controlling the network elements behaves differently depending on the manufacturers of the network elements causes interoperability problems [22], [23].

In order to address these problems, the concept of software defined networking (SDN) has been proposed [23], [24]. SDN is a novel networking paradigm based on (logically) centralizing the control logic in order to better manage the network. In this paradigm, network devices are simple forwarding devices that can be programmed by the SDN controller. This also has a positive impact of making the network devices cheaper. SDN provides a centralized management as well as a fast reshaping of the traffic through

978-1-5090-0223-8/16/$31.00

modifications of the flow tables in the forwarding devices. Moreover, especially because the SDN controller has a global knowledge about the network, it is a proper environment in order to make the resource allocation decisions for network virtualization.

Recently, the idea of using SDN for virtualization has been proposed [25], [26]. For instance, the work in [26] has made a technical and economical analysis of an SDN approach for network virtualization and concluded that this method has substantial benefits for network operators. The work in [27] has focused on the implementation of virtual network embedding algorithm in an SDN network. In this paper, we also advocate the usage of SDN for our proposed virtual network embedding problem and algorithm due to the global topology knowledge of the SDN controller.

The rest of this paper is organized as follows. Section II formulates our optimization problem, while Section III presents our simulation results. Finally, Section IV concludes the paper.

II. Problem Formulation

Substrate Network: We model the substrate network as an undirected graph and denote it by $G^S = (V^S, E^S)$, where V^S is the set of substrate nodes and E^S is the set of substrate links. $loc(v^S)$ represents the location of substrate node $v^S \in V^S$. $B_E(u, w)$ denotes the total bandwidth of substrate link $(u, w) \in E^S$, which is between substrate nodes $u, w \in V^S$.

Virtual Network Requests (VNRs): Let $\mathcal{G}$ be the set of VNRs and g be the index of a VNR such that $g \in \{1, 2, \cdots, |\mathcal{G}|\}$. Each VNR g can be represented by a graph $G_g^V = (V_g^V, E_g^V)$, where V_g^V and E_g^V denote the set of virtual nodes and links, respectively. D_g^V refers to the maximum allowed distance between $loc(v_g^V)$ and the location of the substrate node that v_g^V is mapped to. The value of D_g^V may be determined according to the maximum permissible delay. b_{min}^{gi} and b_{max}^{gi} denote the minimum and maximum requested bandwidth, respectively, of virtual link $i \in E_g^V$ in VNR g. The values of b_{min}^{gi} and b_{max}^{gi} often depend on the application executed by the link. For example, higher bandwidth is usually necessary for a real-time application compared to a non-real-time application.

In this paper, the infrastructure provider offers a discrete set of bandwidth values to the VNRs. Each bandwidth value has a corresponding price. Our problem formulation aims to assign every link i of an accepted VNR g a certain bandwidth level while complying with the b_{min}^{gi} and b_{max}^{gi} values. Each bandwidth level is denoted by an index k. The bandwidth value and price corresponding to bandwidth index k is represented by b_k and r_k, respectively.

Augmented Graph Construction: As in [5], we create an *augmented graph* $G^{S'} = (V^{S'}, E^{S'})$ by using the substrate network topology and the location requirements of virtual nodes. Augmented graph contains a meta-node $\mu(v_g^V)$ corresponding to virtual node v_g^V of each VNR g. Every meta-node is connected to all substrate nodes that are within the radius D_g^V of the virtual node that this meta-node represents. Let $\Omega(v_g^V) = \{v^S \in V^S \mid dist(loc(v_g^V), loc(v^S)) \leq D_g^V\}$, where $dist(a, b)$ denotes the distance between a and b. Then $V^{S'} = V^S \cup \bigcup_g V_g^V$ and $E^{S'} = E^S \cup \bigcup_g \{(\mu(v_g^V), v^S) | v_g^V \in V_g^V, v^S \in V^S\}$.

Integer Linear Programming Formulation (ILP): The input and decision variables of our ILP formulation are in Table I and Table II, respectively. The endpoints of each link i of VNR g are referred to as source node s_{gi} and destination node t_{gi}, where it does not matter which endpoint is selected as source or destination.

The objective function of our ILP formulation maximizes the total revenue as follows:

$$\max \sum_{g=1}^{|\mathcal{G}|} \sum_{i=1}^{|E_g^V|} \sum_{k=1}^{K} r_k z_k^{gi} \tag{1}$$

We first model the requirement of assigning distinct physical resources to conflicting VNRs as follows:

$$t_w^g + t_w^{g'} \leq 2 - C_{gg'} \quad \forall g \neq g', \ \forall w \in V^S, \tag{2}$$

Constraints (3)-(4) are capacity constraints. In particular, constraint (3) ensures that the total flow on each substrate link does not exceed the capacity of the link:

$$\sum_{g=1}^{|\mathcal{G}|} \sum_{i=1}^{|E_g^V|} (f_{uw}^{gi} + f_{wu}^{gi}) \leq B_{uw} \quad \forall u, w \in V^{S'} \tag{3}$$

If VNR g does not utilize a substrate node, i.e., if $t_w^g = 0$, then constraint (4) ensures that the substrate links incident to w do not carry any flow belonging to any virtual link of VNR g. Recall that the decision variable t_w^g is used in constraint (2) to model the requirement that different resources are assigned to the VNRs that request not to be cohosted on the same physical node or link. That is to say, constraint (4) ensures that conflicting VNRs are assigned not only to distinct substrate nodes, but also to distinct substrate links.

$$\sum_{u \in V^S} \sum_{i=1}^{|E_g^V|} (f_{uw}^{gi} + f_{wu}^{gi}) \leq t_w^g \times B_{uw} \quad \forall g, \ \forall w \in V^S \tag{4}$$

Constraints (5)-(7) refer to the flow conservation conditions. In particular, constraint (5) denotes that the net flow to a node must be zero except for source and destination nodes. Besides, constraint (6) ensures that each flow must exit from its source node, whereas constraint (7) guarantees that each flow must enter its destination node.

$$\sum_{w \in V^{S'}} f_{uw}^{gi} - \sum_{w \in V^{S'}} f_{wu}^{gi} = 0 \quad \forall g, \forall i, \text{ and } \forall u \in V^{S'} \backslash \{s_{gi}, t_{gi}\} \tag{5}$$

$$\sum_{w \in V^S} f_{uw}^{gi} - \sum_{w \in V^S} f_{wu}^{gi} = \sum_{k=1}^{K} b_k z_k^{gi} \quad \forall g, \forall i, \text{ and } u = s_{gi} \tag{6}$$

$$\sum_{w \in V^S} f_{uw}^{gi} - \sum_{w \in V^S} f_{wu}^{gi} = -\sum_{k=1}^{K} b_k z_k^{gi} \quad \forall g, \forall i, \text{ and } u = t_{gi} \tag{7}$$

Input Variable	Explanation
V^S	= Set of substrate nodes
V_g^V	= Set of virtual nodes in VNR g
E_g^V	= Set of virtual links in VNR g
$V^{S'}$	= Set of all nodes in the augmented graph $G^{S'}$
b_k	= Bandwidth value corresponding to bandwidth level k
r_k	= Revenue corresponding to bandwidth level k
K	= Total number of bandwidth levels
B_{uw}	= Capacity of the link between substrate nodes u and w
$C_{gg'}$	= $\begin{cases} 1, & \text{if there is a conflict between VNR } g \text{ and } g' \\ 0, & \text{otherwise} \end{cases}$
s_{gi}	= Source node of virtual link i in VNR g
t_{gi}	= Destination node of virtual link i in VNR g
b_{min}^{gi}	= Minimum bandwidth requested by virtual link i of VNR g
b_{max}^{gi}	= Maximum bandwidth requested by virtual link i of VNR g

TABLE I: Table for Input Variables

Decision Variable	Explanation
z_k^{gi}	= $\begin{cases} 1, & \text{if bandwidth level } k \text{ is assigned to virtual link } i \text{ of VNR } g \\ 0, & \text{otherwise} \end{cases}$
f_{uw}^{gi}	= Total amount of flow from node u to w belonging to virtual link i of VNR g in the augmented graph
$f_{uwgi}^{'}$	= $\begin{cases} 1, & \text{if some bandwidth level is assigned to the link between } u \text{ and } w \text{ in the augmented graph in the direction} \\ & \text{from } u \text{ to } w \text{ for virtual link } i \text{ of VNR } g \\ 0, & \text{otherwise} \end{cases}$
x_{uw}^g	= $\begin{cases} 1, & \text{if substrate node } w \text{ is used by virtual node } u \text{ of VNR } g \\ 0, & \text{otherwise} \end{cases}$
y_g	= $\begin{cases} 1, & \text{if VNR } g \text{ is accepted} \\ 0, & \text{otherwise} \end{cases}$
d_{kuw}^{gi}	= $\begin{cases} 1, & \text{if } x_{uw}^g \times z_k^{gi} = 1 \\ 0, & \text{otherwise} \end{cases}$
t_w^g	= $\begin{cases} 1, & \text{if substrate node } w \text{ is used by VNR } g \\ 0, & \text{otherwise} \end{cases}$

TABLE II: Table for Decision Variables

Constraint (8) ensures that the bandwidth assigned to each virtual link of an accepted VNR is within the minimum and maximum bandwidth that the link requested:

$$b_{min}^{gi} \times y_g \leq \sum_{k=1}^{K} b_k z_k^{gi} \leq b_{max}^{gi} \times y_g \quad \forall g, \forall i \tag{8}$$

Constraints (9)-(11) makes the term $d_{kuw}^{gi} = x_{uw}^g \times z_k^{gi}$ linear:

$$d_{kuw}^{gi} \leq z_k^{gi} \quad \forall g, \forall i, \forall k, \forall u, w \in V^{S'} \tag{9}$$

$$d_{kuw}^{gi} \leq x_{uw}^g \quad \forall g, \forall i, \forall k, \forall u, w \in V^{S'} \tag{10}$$

$$d_{kuw}^{gi} \geq z_k^{gi} + x_{uw}^g - 1 \quad \forall g, \forall i, \forall k, \forall u, w \in V^{S'} \tag{11}$$

Constraints (12)-(13) guarantee that the amount of flow used by a virtual link i of VNR g in the direction from a node u to a node w in the augmented graph equals the bandwidth assigned to that virtual link. Moreover, these constraints also ensure that virtual nodes that are not endpoints of a virtual link do not appear in the augmented graph on the path corresponding to that particular virtual link.

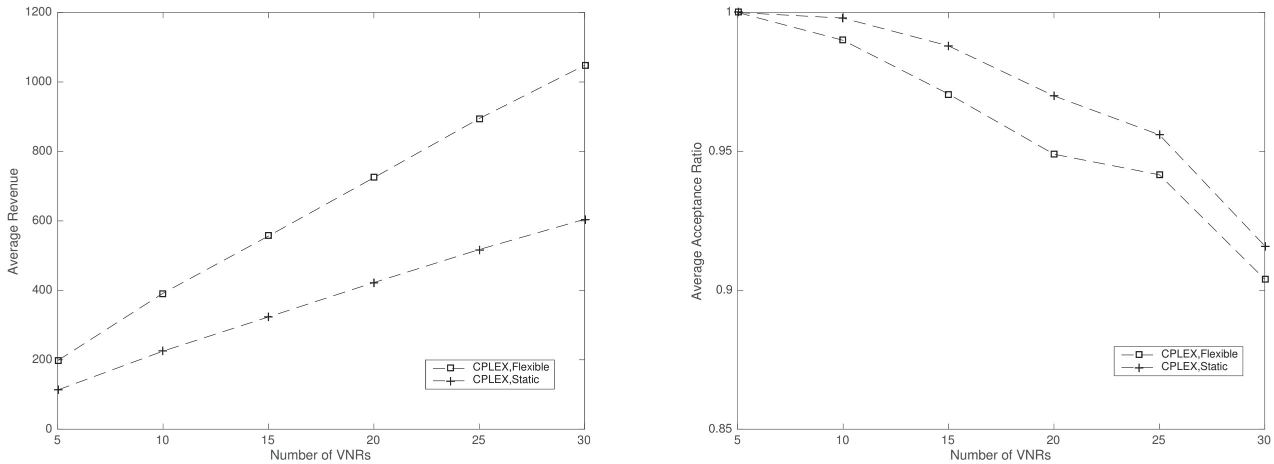
$$f_{uw}^{gi} = \begin{cases} \sum_{k=1}^{K} b_k d_{kuw}^{gi} & \forall g, i, \ \forall w \in V^S, \ u = s_{gi} \\ 0, & \forall g, i, \ \forall w \in V^S, \ \forall u \in V^{S'} \setminus V^S \text{ and } u \neq s_{gi} \end{cases} \tag{12}$$

$$f_{wu}^{gi} = \begin{cases} \sum_{k=1}^{K} b_k d_{kuw}^{gi} & \forall g, i, \ \forall w \in V^S, \ u = t_{gi} \\ 0, & \forall g, i, \ \forall w \in V^S, \ \forall u \in V^{S'} \setminus V^S \text{ and } u \neq t_{gi} \end{cases} \tag{13}$$

The following constraint models the relationship between f'_{uwgi} and f_{wu}^{gi}:

$$f'_{uwgi} \leq f_{wu}^{gi} \leq f'_{uwgi} \times B_{uw} \quad \forall g, \forall i, \forall u, w \in V^{S'} \tag{14}$$

Constraint (15) avoids path splitting, i.e., it ensures that the flow corresponding to a virtual link is passes through a



(a) Average revenue with varying number of VNRs

(b) Average acceptance ratio with varying number of VNRs

Fig. 1: Comparison of CPLEX outputs of flexible and static bandwidth schemes for varying number of VNRs

single path in the substrate network:

$$\sum_{w \in V^{S'}} f'_{uwgi} \leq 1; \forall g, \forall i, \forall u \in V^{S'} \tag{15}$$

The following constraints are related to the admission control of VNRs. If VNR g is not accepted, then constraint (16) guarantees that VNR g does not utilize any substrate node:

$$x^g_{uw} \leq y^g; \forall g, \forall u, w \in V^{S'} \tag{16}$$

Likewise, if a VNR g is rejected, then constraint (17) ensures that virtual links of VNR g are not allocated any bandwidth level:

$$z^{gi}_k \leq y^g; \forall g, \forall i, \forall k \tag{17}$$

Similarly, if VNR g is rejected, then constraint (18) ensures that no flow is assigned in the substrate links for the virtual links of VNR g:

$$f^{gi}_{uw} \leq B_E(u,w) y^g; \forall g, \forall i, \forall u, w \in V^{S'} \tag{18}$$

Constraint (19) ensures that if VNR g is accepted, then each virtual link i of VNR g for which $b^{gi}_{max} > 0$ is assigned exactly one bandwidth level. If VNR g is rejected, this constraint also ensures that virtual links of VNR g are not assigned any bandwidth level. As input to our ILP, incident links of the nodes that correspond to the virtual nodes in the augmented graph are assigned $b^{gi}_{max} = 0$ since they are not substrate links.

$$\sum_{k=1}^{K} z^{gi}_k = y_g \quad \forall g, \forall i \text{ such that } b^{gi}_{max} > 0 \tag{19}$$

Constraint (20) guarantees that each virtual link of an accepted VNR is assigned to exactly one substrate node:

$$\sum_{w \in V^S} x^g_{uw} = y_g \quad \forall g, \forall u \in V^{S'} - V^S \tag{20}$$

Constraint (21) ensures that at most one virtual link of an accepted VNR is assigned to a particular substrate node:

$$\sum_{u \in V^{S'} - V^S} x^g_{uw} \leq y_g \quad \forall g, \forall w \in V^S \tag{21}$$

Constraint (22) prevents the allocation of any flow between the virtual nodes. This constraint is necessary because a virtual node may be connected to multiple substrate nodes in the augmented graph and without this constraint, it may appear on a path corresponding to another pair of virtual links, which is not a desired situation.

$$f^{gi}_{uw} = 0 \quad \forall g, \forall u, w \in V^{S'} - V^S \tag{22}$$

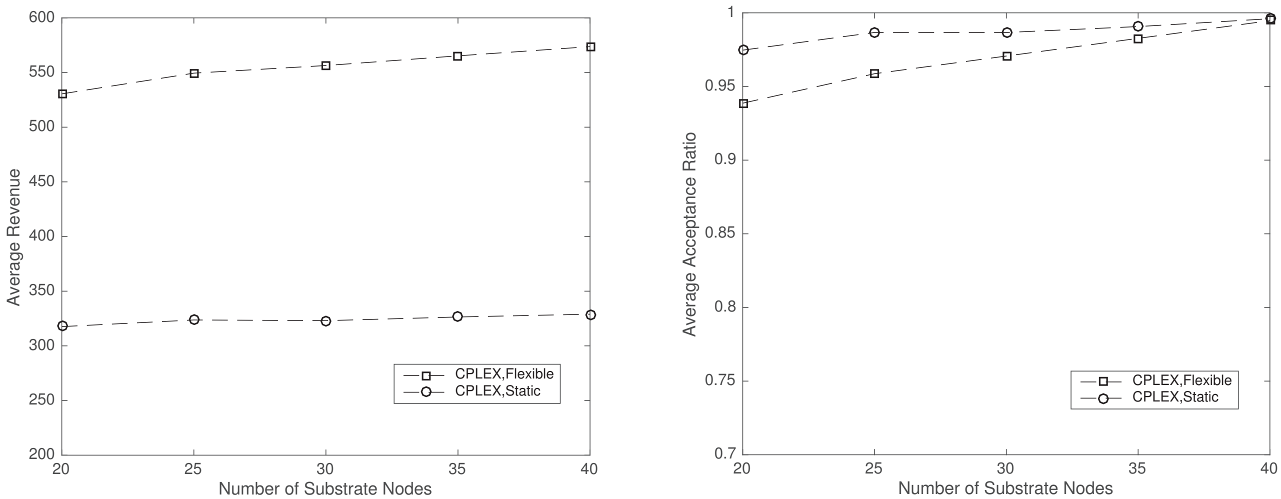
Finally, we model the constraints related to the range of values that each decision variable can take as follows:

$$f^{gi}_{uw} \geq 0 \quad \forall u, w \in V^{S'} \tag{23}$$

$$z^{gi}_k, f'_{uwgi}, x^g_{uw}, y_g, d^{gi}_{kuw}, t^g_w \in \{0,\ 1\} \quad \forall g, \forall i, \forall k,\ \forall u, w \in V^{S'} \tag{24}$$

III. Simulation Results

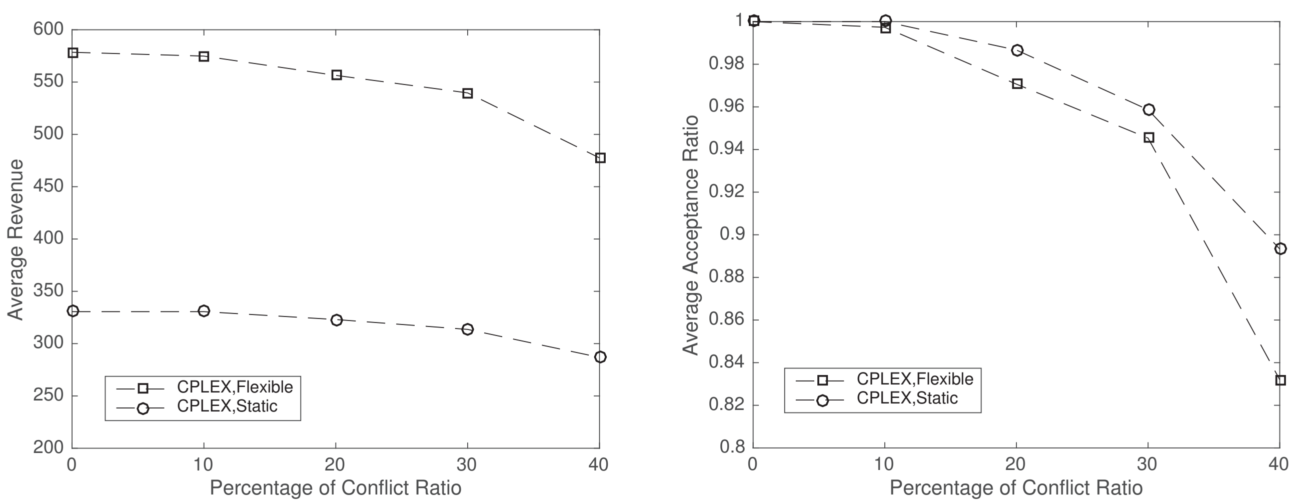
As in [5], [9], [28], we use GT-ITM to generate substrate and virtual network topologies. The capacity of each substrate link is uniformly distributed between 15 and 30. We compare the simulation results of our flexible bandwidth scheme with the fixed bandwidth scheme. In the flexible bandwidth scheme, each virtual link has $b_{min} = 6$ and $b_{max} = 14$, whereas in the static bandwidth scheme, teach virtual link has $b_{max} = b_{min} = 10$. The bandwidth levels offered by the substrate network provider are $\{6, 8, 10, 12, 14\}$ and the revenue values corresponding to



(a) Average revenue with varying substrate network size

(b) Average acceptance ratio with varying substrate network size

Fig. 2: Comparison of CPLEX outputs of flexible and static bandwidth schemes for varying substrate network size (number of substrate nodes)



(a) Average revenue with varying conflict ratio

(b) Average acceptance ratio with varying conflict ratio

Fig. 3: Comparison of CPLEX outputs of flexible and static bandwidth schemes for varying conflict ratio

them are $\{2, 5, 8, 11, 14\}$, respectively. In each experiment, we take the mean value of 50 different simulations, where each simulation consists of a randomly generated substrate network such that 50% of the nodes are connected via an edge. The number of virtual nodes in each VNR is uniformly distributed between 2 and 10. The work in The number of virtual nodes in each VNR is uniformly distributed between 2 and 10 presents a similar simulation scenario.

We first evaluate the impact of the number of VNRs. The substrate network has 30 nodes and the conflict ratio is 20%; i.e., 20% of the VNR pairs have conflict. Figure 1a shows that the average revenue increases linearly as the number of VNRs increases. Moreover, flexible bandwidth scheme we propose in this paper results in significantly better results than the fixed bandwidth scheme. Figure 1b displays the acceptance ratio of VNRs in the same scenario. The average acceptance ratio decreases as the number of VNRs increases. Figure 1b also shows that the static bandwidth scheme may result in higher acceptance ratio since the flexible bandwidth scheme may reject a certain set of VNRs in order to accept a VNR that generates higher revenue.

We then evaluate the impact of the substrate network size. We set the number of VNRs to 15 and the conflict ratio to 20%. Figure 2a demonstrates that the average revenue increases as the substrate network size increases. However, when compared with Figure 1a, the impact of substrate network size on revenue is less significant than the impact of the number of VNRs. Besides, flexible bandwidth scheme yields significantly higher revenue than the fixed bandwidth scheme. Figure 2a shows the acceptance ratio of VNRs in the same simulation scenario. The average acceptance ratio increases as the substrate network size increases since a larger network can accommodate more VNRs.

We then evaluate the impact of the conflict ratio. We set the number of VNRs to 15 and the substrate network size to 30. Figure 3a demonstrates that the revenue drops as the conflict ratio increases. Moreover, flexible bandwidth scheme results in significantly higher revenue than the fixed bandwidth scheme irrespective of the conflict ratio. Figure 3b displays the acceptance ratio of VNRs in the same simulation scenario. The average acceptance ratio naturally decreases as the conflict ratio increases.

IV. Conclusion

In this paper, we have formulated a novel virtual network embedding problem as an integer linear program. Our formulation ensures that the virtual networks of conflicting virtual network operators are not cohosted on the same physical equipment. Our formulation also offers flexible bandwidth management by enabling the virtual links to select among a range of possible bandwidth values, each having a different revenue and thereby realizing any possible revenue function. Our simulation results demonstrate that our flexible bandwidth management scheme results in higher revenue compared to a fixed bandwidth scheme consisting of only a single bandwidth level requested by each virtual link. As a future work, we plan to propose a heuristic algorithm with low computational complexity.

Acknowledgments

This work has been supported by Argela Technologies, Istanbul, Turkey.

References

[1] A. Fischer, J. F. Botero, M. Till Beck, H. De Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 1888–1906, 2013.

[2] D. G. Andersen, "Theoretical approaches to node assignment," Computer Science Department, Carnegie Mellon University, Tech. Rep., 2002. [Online]. Available: http://repository.cmu.edu/cgi/viewcontent.cgi?article=1079&context=compsci

[3] C. Guo, G. Lu, H. J. Wang, and S. Yang, "Secondnet: a data center network virtualization architecture with bandwidth guarantees," Microsoft Research, Tech. Rep. MSR-TR-2010-81, 2010.

[4] A. Haider, R. Potter, and A. Nakao, "Challenges in resource allocation in network virtualization," in *20th ITC Specialist Seminar*, vol. 18, 2009, p. 20.

[5] N. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," in *IEEE INFOCOM*, 2009, pp. 783–791.

[6] N. F. Butt, M. Chowdhury, and R. Boutaba, *Topology-awareness and reoptimization mechanism for virtual network embedding*. Springer, 2010.

[7] S. Zhang, Z. Qian, J. Wu, S. Lu, and L. Epstein, "Virtual network embedding with opportunistic resource sharing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 3, pp. 816–827, 2014.

[8] A. Xiao, Y. Wang, L. Meng, X. Qiu, and W. Li, "Topology-aware remapping to survive virtual networks against substrate node failures," in *IEEE Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2013, pp. 1–6.

[9] M. Chowdhury, M. R. Rahman, and R. Boutaba, "Vineyard: Virtual network embedding algorithms with coordinated node and link mapping," *IEEE/ACM Transactions on Networking*, vol. 20, no. 1, pp. 206–219, 2012.

[10] X. Cheng, S. Su, Z. Zhang, K. Shuang, F. Yang, Y. Luo, and J. Wang, "Virtual network embedding through topology awareness and optimization," *Computer Networks*, vol. 56, no. 6, pp. 1797–1813, 2012.

[11] L. Richter Bays, R. Ruas Oliveira, L. S. Buriol, M. Pilla Barcellos, and L. P. Gaspary, "A heuristic-based algorithm for privacy-oriented virtual network embedding," in *IEEE Network Operations and Management Symposium (NOMS)*, 2014, pp. 1–8.

[12] X. Cheng, S. Su, Z. Zhang, H. Wang, F. Yang, Y. Luo, and J. Wang, "Virtual network embedding through topology-aware node ranking," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 2, pp. 38–47, 2011.

[13] G. Sun, H. Yu, V. Anand, and L. Li, "A cost efficient framework and algorithm for embedding dynamic virtual network requests," *Future Generation Computer Systems*, vol. 29, no. 5, pp. 1265–1277, 2013.

[14] I. Fajjari, N. Aitsaadi, G. Pujolle, and H. Zimmermann, "Adaptive-vne: A flexible resource allocation for virtual network embedding algorithm," in *IEEE Global Communications Conference (GLOBECOM)*, 2012, pp. 2640–2646.

[15] S. Abdelwahab, B. Hamdaoui, and M. Guizani, "Bird-vne: Backtrack-avoidance virtual network embedding in polynomial time," in *IEEE Global Communications Conference (GLOBECOM)*, 2014, pp. 4983–4989.

[16] A. Jarray and A. Karmouch, "Decomposition approaches for virtual network embedding with one-shot node and link mapping," *IEEE/ACM Transactions on Networking*, vol. 23, no. 3, pp. 1012–1025, 2015.

[17] L. Gong, Y. Wen, Z. Zhu, and T. Lee, "Toward profit-seeking virtual network embedding algorithm via global resource capacity," in *IEEE INFOCOM*, 2014, pp. 1–9.

[18] S. B. Masti and S. V. Raghavan, "Vna: An enhanced algorithm for virtual network embedding," in *IEEE Computer Communications and Networks (ICCCN)*, 2012, pp. 1–9.

[19] I. Fajjari, N. Aitsaadi, G. Pujolle, and H. Zimmermann, "Vne-ac: Virtual network embedding algorithm based on ant colony metaheuristic," in *IEEE International Conference on Communications (ICC)*, 2011, pp. 1–6.

[20] T. Trinh, H. Esaki, and C. Aswakul, "Quality of service using careful overbooking for optimal virtual network resource allocation," in *International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2011, pp. 296–299.

[21] H. Kim and S. Lee, "Greedy virtual network embedding under an exponential cost function," in *International Conference on Information Networking (ICOIN)*, 2012, pp. 216–221.

[22] "Open networking foundation," 2011. [Online]. Available: https://www.opennetworking.org/

[23] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. Van Der Merwe, "The case for separating routing from routers," in *ACM SIGCOMM Workshop on Future Directions in Network Architecture*, 2004, pp. 5–12.

[24] B. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, T. Turletti *et al.*, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014.

[25] V. D. Philip, Y. Gourhant, and D. Zeghlache, "Openflow as an architecture for e-node b virtualization," in *e-Infrastructure and e-Services for Developing Countries*. Springer, 2012, pp. 49–63.

[26] B. Naudts, M. Kind, F.-J. Westphal, S. Verbrugge, D. Colle, and M. Pickavet, "Techno-economic analysis of software defined networking as architecture for the virtualization of a mobile network," in *IEEE European Workshop on Software Defined Networking (EWSDN)*, 2012, pp. 67–72.

[27] R. Guerzoni, R. Trivisonno, I. Vaishnavi, Z. Despotovic, A. Hecker, S. Beker, and D. Soldani, "A novel approach to virtual networks embedding for sdn management and orchestration," in *IEEE Network Operations and Management Symposium (NOMS)*, 2014, pp. 1–7.

[28] Y. Zhu and M. H. Ammar, "Algorithms for assigning substrate network resources to virtual network components." in *IEEE INFOCOM*, vol. 1200, no. 2006, 2006, pp. 1–12.