Latency-aware Load Balancing Virtual Network Embedding Algorithm

Yue Zong [1,2], Dedi Li [1,3], Xinjie Lai [1,4], Yuanlin Luo [1], Han Xu[1]

[1] Power China Huadong Engineering Corporation Limited, Hangzhou, 311122, China
[2] Country College of Information Science &Electronic Engineering, Zhejiang University, Hangzhou, 310027, China
[3] Country School of Mechanical Engineering, Tianjing University, Tianjing, 300072, China
[4] College of Economics and Management, North China Electric Power University, Beijing, 102206, China

Abstract

To overcome network ossification, network slicing has been proposed as a promising method in 5G network. Software defined networking (SDN) and network function virtualization (NFV) as emerging enabled technology support multiple logical networks sharing physical infrastructure. Virtual network embedding (VNE) is one of the main challenges for network slicing. To avoid link congestion and satisfy the latency requirement of virtual network requests, latency model of network and load balancing link weight formulation are proposed. Then we design a load balancing link reconfiguration embedding algorithm and latency-aware load balancing VNE algorithm. Compared with the baseline algorithms, simulation results show that our proposed algorithm has better performance.

Keywords

Virtual network embedding, load balancing, latency-aware, network reconfiguration

1. Introduction

Network slicing[1] is recognized as key technology for 5G network to support multiple diversified vertical markets with efficiency and flexibility. Software defined networking (SDN) and network function virtualization (NFV) as emerging enabled technology support multiple logical networks sharing physical infrastructure[2]. Network virtualization is an enabler for network slicing, where the physical network can be partitioned into different configurable slices in the multi-domain heterogeneous converged networks.

The corresponding virtual network (VN) is essentially the deployment of resource allocations optimization by considering network and computing resources. The mapping from the virtual networks to the substrate infrastructure is one of the main challenges referred as virtual network embedding (VNE)[3][4] which has been intensively studied in the literatures.

5G emerging high-bandwidth and low latency applications have driven efficient VNE strategy to satisfy the Quality of Service(QoS) of user requests. Virtual link scheme without considering load balancing may cause link congestion. Authors in [5] have proposed a method of load balancing-based allocating bandwidth resource and a reconfiguration strategy to improve the network performance. The multi-objective VNE by utilizing node load value as one of the fitness functions has been proposed[6]. A VN mapping strategy based on hybrid genetic algorithm is proposed adopting dynamic calculated cross-probability to increase the flexibility of network[7].

Some critical nodes and links as primary selection to satisfy the latency will cause link congestion, thereby the literatures above addressed the problem by load balancing embedding or reconfiguration. However, they don't consider latency affect for sensitive applications. In this work, we design latency

ISCIPT2022@7th International Conference on Computer and Information Processing Technology, August 5-7, 2022, Shenyang, China
EMAIL: zong_y2@hdec.com (Yue Zong); li_dd3@hdec.com (Dedi Li); lai_xj2@hdec.com (Xinjie Lai); luo_yl2@hdec.com (Yuanlin Luo); xu_h@hdec.com (Han Xu)
ORCID: 0000-0003-3782-5252 (Yue Zong)

model to satisfy the requirement of latency-sensitive virtual network requests and formulate load balancing link weight. Then, we have proposed a load balancing link reconfiguration embedding algorithm (VLR-E) and latency-aware load balancing VNE algorithm (LALB-VNE).

The remainder of this paper is organized as follows. Section 2 introduces the network model and formulates load balancing link weight. In Section 3, we detail the proposed VLR-E and LALB-VNE algorithm. We evaluate our proposed algorithm by simulation in Section 4. Finally, we conclude the paper in Section 5.

2. System model

Substrate network use optical connected by reconfigurable optical add-drop multiplexer (ROADM) to accommodate online virtual network requests. The network uses OXC and data center links establish transparent all-optical without considering router to reduce network latency. In this section, we describe network model and load balancing link weight formulation.

2.1. Network model

The substrate network is directed by an undirected graph $\mathrm{G} = (\mathrm{N}, \mathrm{E})$,where N is the node set $\{1,..\mathrm{n}\}$and E is the link set(m, n). Each node has $\mathrm{Com}(\mathrm{n})$ computing resource, and the location of each node is $(\mathrm{locX}(\mathrm{n}), \mathrm{locY}(\mathrm{n}))$. Each optical fiber (m, n) has W wavelength and C is the capacity of each wavelength. EA_{mn} is the number of EDFA, where, $EA_{mn} = \lfloor L_{mn}/D - 1 \rfloor + 2$, L_{mn} is the length of link (m, n) and D represent the distance of two EDFA.

Node process and transmission of link latency are considered as link latency, $l_{mn} = 2\left(l_{trs} + l_{fec}\right) + L_{mn} \cdot l_{prop} + EA_{mn} \cdot l_{edfa} + 2l_{roadm}$. l_{trs} is the delay of a transponder that converts request into an optical signal for transmission. l_{fec} is the delay of FEC coder-decoder processing module. l_{roadm} is the delay of ROADM node. Optical fiber transmission delay l_{prop}/km is the main components of link delay. The delay of EDFA to enhance signal during transmission is l_{edfa}. The delay of regenerator and dispersion compensation is not considered in this section.

The r^{th}virtual network request is directed by an undirected graph $G_v^r = (N_v^r, E_v^r)$, where N_v^r is the virtual node set and E_v^r is the virtual link set. For virtual node s in N_v^r, the computing resource request is c^{rs}, and the location is $(\mathrm{vlocX}(\mathrm{s}),\ \mathrm{vlocY}(\mathrm{s}))$. The bandwidth request of virtual link $(\mathrm{s}, \mathrm{d}) \in E_v^r$, where the maximize acceptance latency is l_v^{rsd}.

2.2. Load balancing link weight formulation

To avoid link congestion, we design a novel load balancing link weight update mechanism by utilizing standard deviation of link bandwidth resource consumption. The bandwidth resource occupied of substrate link (m, n) is described in Eq.1, where y_{mn}^{rsd} equals 1 if virtual link (s, d) of request r is mapped onto substrate link (m, n).

$$u_{mn} = \sum_{r \in R} \sum_{(s,d) \in E_v^r} y_{mn}^{rsd} \cdot b^{rsd} \tag{1}$$

Standard deviation of link load is described in Eq.2, where $\bar{u}$ is the mean value of link load occupation.

$$\sigma = \sqrt{\frac{\left(\sum_{(m,n) \in E} u_{mn} - \bar{u}\right)^2}{|E|}} \tag{2}$$

The link weight is defined as Eq.3 both considering standard deviation of link load and link latency, where B_{mn} is the initial wavelength and capacity of link (m, n), α and β represent the weight of link unitization and latency ratio, $0 < \alpha, \beta < 1, \alpha + \beta = 1$. The Maximum link latency for all substrate links l_{max} is shown in Eq.4.

$$\varphi_{mn} = \alpha \frac{u_{mn} + \sigma}{B_{mn}} + \beta \frac{l_{mn}}{l_{max}} \tag{3}$$

$$l_{max} = max\{l_{mn}, \forall (m,n) \in E\} \tag{4}$$

3. Latency-aware load balancing VNE algorithm

In this section, for online virtual network requests, latency-aware load balancing VNE algorithm (LALB-VNE) is proposed, which is one-stage embedding strategy. For virtual node embedding, we use Global Topology Resource (GTR) node ranking for substrate node and virtual node embedding according to Ref.[8]. To improve the utilization of link resource and avoid link congestion, load balancing link reconfiguration embedding algorithm (VLR-E) is proposed.

3.1. Load balancing link re-configuration embedding algorithm

Since virtual node embedding results may cause virtual link embedding failure, in this section, VLR-E algorithm is proposed to improve the link embedding performance by considering path splitting, shown in **Algorithm 1.**

Algorithm 1 VLR-E

Input: $G = (N, E)$, (s, d), m, n, b^{rsd}, L_v^{sd};
Output: virtual link embedding results El_v^{sd}
1. update link weight, establish link auxiliary graph;
2. search the shortest path satisfy the bandwidth and latency requirements;
3. **if** successful **then**
4. update bandwidth state of substrate network and embedding state of virtual link;
5. return El_v^{sd};
6. **else**
7. calculate satisfied candidate path set P_{sd}^{mn} between substrate node m and n;
8. search split path satisfying latency in P_{sd}^{mn};
9. **if** successful **then**
10. update bandwidth state of substrate network and embedding state of virtual link;
11. return El_v^{sd};
12. **else**
13. return link embedding failure;
14. **end if**
15 **end if**

3.2. Latency-aware load balancing VNE algorithm

For online virtual network requests, we proposed a LALB-VNE shown in **Algorithm 2**. First, generate a set of upcoming requests according to time window, and release resource if there exists request leave. Then, for each request execute one-stage virtual network embedding according GTR virtual node and VLR-E embedding strategy.

Algorithm 2 LALB-VNE

Input: $G = (N, E)$
Output: embedding results
1. **for all** time window t_w **do**
2. generate a set of virtual network requests $G_v(N_v, E_v)$;
3. if there exists requests leave, release resource;
4. **for all** virtual network requests $G_v^r \in G_v$ **do**
5. sort all virtual nodes according to GTR, obtain G_N;
6. **for all** virtual node in G_N **do**
7. sort substrate node according to GTR;
8. calculate the location and computing resource for all substrate nodes, and establish candidate set;
9. select the first node in candidate set for embedding;
10. **if** virtual node embed successful **then**
11. **for** virtual link **do**
12. get the corresponding substrate node m and n for successful embedding;
13. execute **Algorithm 1** for virtual link embedding;
14. update network resource;
15. **end for**
16. **else**
17. refuse the request
18. **end if**
19. **end for**
20. **end for**
21. **end for**

4. Performance Evaluation

In this section, we present simulation result to investigate the performance of the proposed LALB-VNE algorithm. In the simulation, we consider NSFNET and USNET as network topology. The number of wavelength W is in range [50, 80] and computing resource $\mathrm{Com}(n)$ is in range [400, 500] unit. Substrate network parameter is setting according to Ref.[9].

We assume VN request arrival follows Poisson distribution with the mean of [4, 20] requests per 100 time units, and the lifetime of each request follows the exponential distribution with an average lifetime of 1000 time units.

Virtual nodes number of per request is generated in the range of 2 to 4. And the computing requirement is in range [2, 5] units. For the random links generation, the connectivity probability of two virtual nodes is set to 0.5. The virtual links range from 50 to 200 Gbps. Assume the maximum acceptance latency is generated in [10, 30] ms.

In this paper, online GTR-based two-stage VNE algorithm (GTR-T-VNE) and GTR-based one-stage VNE algorithm (GTR-O-VNE) are designed as benchmark.

Figure 1-3 shows the simulation results in NSFNET. The acceptance ratio is shown in Figure 1, proposed LALB-VNE is 20% improvement than GTR-O-VNE and GTR-T-VNE. Revenue to cost ratio(R/C) is shown in Figure 2, proposed LALB-VNE is 1%-2% increased. Average latency is shown in Figure 3, proposed LALB-VNE can decrease 2-2.5ms latency.

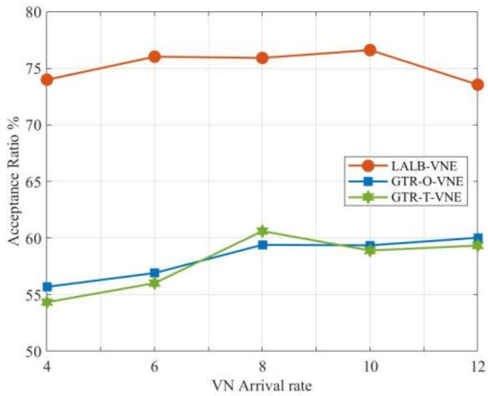

Figure 1 : Acceptance ratio in NSFNET

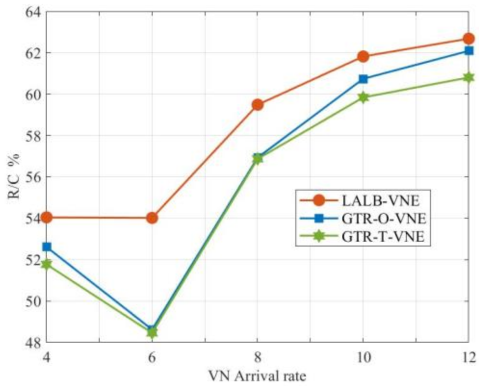

Figure 2: R/C in NSFNET

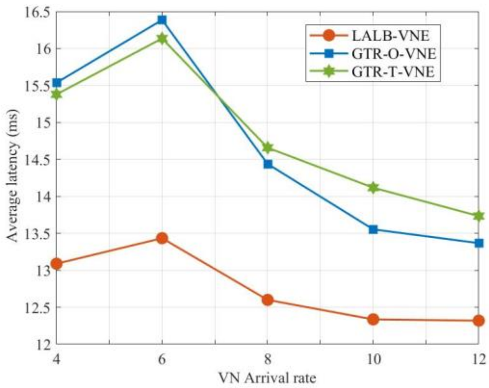

Figure 3: Average latency in NSFNET

Figure 4-6 shows the simulation results in USNET. Figure 4 shows that the acceptance ratio of proposed LALB-VNE has about 2% improvement than GTR-O-VNE and GTR-T-VNE. Revenue to cost ratio of proposed LALB-VNE achieve up to 6% improvement shown in Figure 5. The average latency decrease about 1.35ms for proposed LALB-VNE shown in Figure 6.

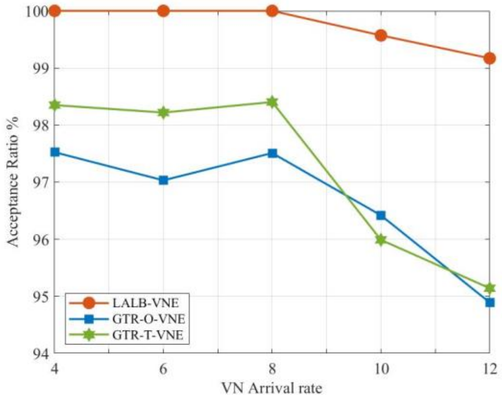

Figure 4: Acceptance ratio in USNET

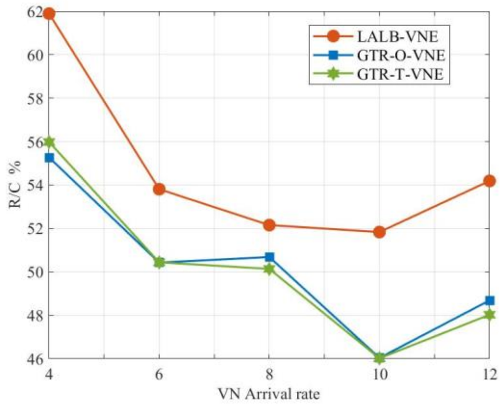

Figure 5: R/C in USNET

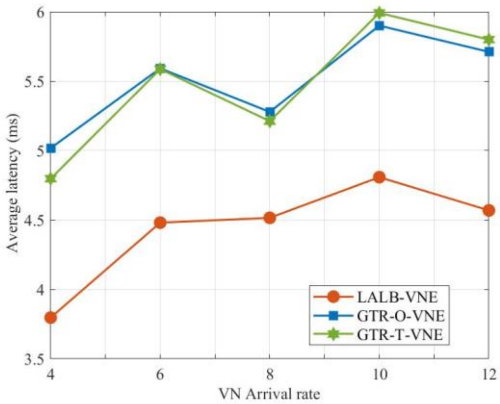

Figure 6: Average latency in USNET

5. Conclusion

This paper has investigated the load balancing VNE problem by considering latency sensitive applications. Specifically, we have described network model and proposed a latency model to calculate the network latency. To avoid link congestion, load balancing link weight formulation has been proposed. Further, VLR-E algorithm has been proposed based proposed load balancing link weight. Then a LALB-VNE algorithm is developed to satisfy the latency sensitive applications of VN. Simulation results have demonstrated that our proposed algorithm has better performance compared with the baseline algorithms. In the future work, the intelligent algorithm such as particle swarm optimization or ant colony optimization algorithm to improve network performance furtherly.

6. Acknowledgements

The authors thank the Power China Huadong Engineering Corporation Limited Research project. And thank the contribution of related colleague.

7. References

[1] S. Wijethilaka, M. Liyanage, Survey on Network Slicing for Internet of Things Realization in 5G Networks, IEEE Communications Surveys and Tutorials (2021) 957-994. doi:10.1109/COMST.2021. 3067807.

[2] Y. Zong, C. Feng, Y. Guan, et al. Virtual Network Embedding for Multi-Domain Heterogeneous Converged Optical Networks: Issues and Challenges, Sensors (2020) 20(9). doi:10.3390/s20092655.

[3] A. Fischer, J. F. Botero, M. T. Beck, et al. Virtual Network Embedding: A Survey, IEEE Communications Surveys and Tutorials (2013) 15(4): 1888-1906. doi: 10.1109/SURV.2013.013013.00155.

[4] F. Esposito, D. D. Paola, I. Matta. On Distributed Virtual Network Embedding with Guarantees, IEEE/ACM Transactions on Networking (2016) 24(1): 569-582. doi: 10.1109/TNET.2014.2375826.

[5] Q. Chen, Y. Wang, X. Qiu, et al. A Survivable Virtual Network Embedding Scheme based on Load Balancing and Reconfiguration, in: proceedings of the IEEE Network Operations and Management Symposium (NOMS), Krakow, POLAND, 2014: 1-7.

[6] Z. Li, F. Yuan, L. A load balancing algorithm for solving multi-objective virtual network embedding, Transactions on Emerging Telecommunications Technologies (2020) e4066.

[7] P. Zhang, F. Liu, C. Jiang, et al. A Multi-Domain VNE Algorithm Based on Load Balancing in the IoT Networks, Mobile Networks and Applications (2021).

[8] Y. Zong, Y. Ou, A. Hammad, et al. Location-Aware Energy Efficient Virtual Network Embedding in Software-Defined Optical Data Center Networks, IEEE/OSA Journal of Optical Communications and Networking (2018) B58-B70. doi: 10.1364/JOCN.10. 000B58.

[9] S. Taeb, N. Shahriar, S. R. Chowdhury, et al. Virtual Network Embedding with Path-based Latency Guarantees in Elastic Optical Networks, In: proceedings of the IEEE International Conference on Network Protocols (ICNP), Chicago, IL, 2019.