Comparison of Virtual Network Embedding Algorithms for Data Center Networks

Hardeep Kaur Takhar, Ana Laura Gonzalez Rios, and Ljiljana Trajković
Simon Fraser University
Vancouver, British Columbia, Canada
{hktakhar, anag, ljilja}@sfu.ca

***Abstract*—Software defined networks are a new Internet architecture paradigm that allows co-existence of heterogeneous network architectures. They optimize network management (maintenance, operability, and effective content delivery) by provisioning a centralized network intelligence. Virtual network embedding (VNE) algorithms improve scalability and utilization of physical resources in data center networks (DCNs). In this paper, we implement various DCN topologies and evaluate performance of VNE algorithms using the VNE-Sim simulator. We compare performance by implementing both server-centric and switch-centric DCN topologies.**

***Index Terms*—Networks virtualization, virtual network embedding, software defined networks, data center networks**

I. Introduction

Virtualization in data centers reduces inefficient resource utilization and addresses high storage and processing demands [5], [26]–[28]. Network virtualization enables flexible network operability and maintenance by sharing the existing physical network resources [21], [31], [43], [44]. This new paradigm addresses the rapidly growing network demands on the Internet infrastructure. The software defined network (SDN) model decouples the network layer control and data planes integrated within the Open Systems Interconnection Internet model and leads to a logically centralized approach that facilitates network management [33], [38], [41].

In the virtualized network model, the Internet service providers (ISPs) are categorized as infrastructure (InPs) and service (SPs) providers. The InPs own and manage the physical infrastructure while SPs establish agreements with InPs to access network resources and offer end-to-end services [14]. Virtual network embedding (VNE) is a process of allocating physical network resources when embedding virtual networks (VNs) onto substrate networks (SNs) [32], [37]. The VNE success rate depends on the substrate and virtual network topologies as well as the embedding algorithms [18].

VNs consist of nodes with central processing units (CPUs) and links with bandwidth capacities [14], [43]. Substrate network resources are allocated to virtual nodes and links through the virtual node (VNoM) and virtual link (VLiM) mappings [16]. Attending to virtual network requests (VNRs) whose arrival rate may vary is a crucial component in the VNE process. The VNRs are served based on deployed VNE algorithms. Mapping limited resources of a substrate network for virtual requests is an NP-hard problem [6].

We evaluate VNE algorithms using server-centric and switch-centric DCN topologies. The paper is organized as follows: An overview of VNE algorithms and their performance metrics is given in Section II while DCN topologies are described in Section III. We discuss the simulation scenarios and results in Section IV and conclude with Section V.

II. Virtual Network Embedding

Based on the approach used to solve the VNoM and VLiM tasks, VNE algorithms are categorized as uncoordinated two-stage, coordinated two-stage, and coordinated one-stage algorithms. In the uncoordinated two-stage algorithms, VNoM is first solved heuristically and the generated node mapping is used to solve VLiM by using the shortest-path or multicommodity flow (MCF) algorithms with or without path splitting. This approach results in a narrow solution space. Path splitting maps a virtual link onto multiple substrate links [10]. Although permitted with MCF algorithms, it is not an option in case of shortest-path algorithms. In case of coordinated two-stage algorithms, VNoM is solved first while considering virtual link constraints [9], [11], [17]. VLiM is then solved by using the shortest-path or MCF algorithms. Two-stage coordinated algorithms include Global Resource Capacity (GRC) [17], GRC with MCF (GRC-M) [22], ViNEYard (D-ViNE, R-ViNE, WiNE) [11], and Monte Carlo Tree Search (MCTS) based VNE algorithms [25]. They simultaneously solve node mapping along with considerations for link constraints. GRC algorithm employs a node-based ranking algorithm that applies large-large and small-small mappings [9] between the substrate and virtual nodes. It solves the VLiM problem by using the breadth-first search (BFS) to find the shortest-path between nodes. GRC-M algorithm relies on MCF for path splitting to maximize substrate resource utilization [22].

ViNEYard algorithms solve the VNoM as a mixed-integer problem (MIP) while the MCF algorithm is used to perform the VLiM. Deterministic (D-ViNE) or randomized (R-ViNE) rounding techniques are used for linear programming relaxation of MIP. They lead to high VNR acceptance ratio and larger revenue because of successful coordination of node and link mappings [11]. VNE algorithms based on MCTS employ MCF (MaVEn-M) or shortest-path (MaVEn-S) [25] to solve the VLiM while Markov decision process (MDP) is

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada under grant R31-611284.

used to perform the VNoM [12], [29]. These VNE algorithms maximize revenue by searching for the most profitable embeddings. At each state of the MDP, rewards are given for each performed action to maximize the expected cumulative rewards [25]. In the coordinated one-stage algorithms, the virtual node and link mapping tasks are solved simultaneously by creating a suitable optimal virtual link between nodes [8].

The substrate network graph is denoted by $G^s(N^s, E^s)$, where $N^s = \{n^s_1, n^s_2, \ldots, n^s_j\}$ and $E^s = \{e^s_1, e^s_2, \ldots, e^s_k\}$ are sets of j substrate nodes (vertices) and k substrate links (edges), respectively. The i^{th} VNR is a triplet $\Psi_i(G^{\Psi_i}, \omega^{\Psi_i}, \xi^{\Psi_i})$, where $G^{\Psi_i}(N^{\Psi_i}, E^{\Psi_i})$ is the VN graph with l virtual nodes $N^{\Psi_i} = \{n^{\Psi_i}_1, n^{\Psi_i}_2, \ldots, n^{\Psi_i}_\ell\}$ and m virtual edges $E^{\Psi_i} = \{e^{\Psi_i}_1, e^{\Psi_i}_2, \ldots, e^{\Psi_i}_m\}$, ω^{Ψ_i} is the VNR arrival time, and ξ^{Ψ_i} is the VNR lifetime [25].

Performance of VNE algorithms is evaluated based on: acceptance ratio, generated revenue, incurred cost, and substrate resource (node and link) utilizations. The goal of VNE algorithms is to increase the revenue by minimizing the embedding cost. Acceptance ratio is defined as:

$$p^\tau_a = \frac{|\Psi^a(\tau)|}{|\Psi(\tau)|}, \tag{1}$$

where $\Psi^a(\tau)$ is the number of accepted VNRs and $\Psi(\tau)$ is the total number of VNRs that arrive over a time interval τ.

InP's revenue $\mathbf{R}(G^{\Psi_i})$ from embedding VNRs is calculated as:

$$\mathbf{R}(G^{\Psi_i}) = w_c \sum_{n^{\Psi}i \in N^{\Psi_i}} \mathcal{C}(n^{\Psi_i}) + w_b \sum_{e^{\Psi}i \in E^{\Psi_i}} \mathcal{B}(e^{\Psi_i}), \tag{2}$$

where w_c and w_b are the weights for the CPU $\mathcal{C}(n^{\Psi_i})$ and bandwidth $\mathcal{B}(e^{\Psi_i})$ requirements, respectively.

The cost $\mathbf{C}(G^{\Psi_i})$ of embedding VNRs is defined as:

$$\mathbf{C}(G^{\Psi_i}) = \sum_{n^{\Psi}i \in N^{\Psi_i}} \mathcal{C}(n^{\Psi_i}) + \sum_{e^{\Psi}i \in E^{\Psi}i} \sum_{e^s \in E^s} f^{e^{\Psi_i}}_{e^s}, \tag{3}$$

where $f^{e^{\Psi_i}}_{e^s}$ is the total bandwidth of substrate link e^s allocated to the virtual link e^{Ψ_i}. The substrate node and link utilizations are calculated based on the available cost and bandwidth resources:

$$\mathcal{U}(N^s) = 1 - \frac{\sum_{n^s \in N^s} \mathcal{C}(n^s)}{\sum_{n^s \in N^s} \mathcal{C}_{max}(n^s)} \tag{4}$$

$$\mathcal{U}(E^s) = 1 - \frac{\sum_{e^s \in E^s} \mathcal{B}(e^s)}{\sum_{e^s \in E^s} \mathcal{B}_{max}(e^s)}, \tag{5}$$

where $\mathcal{C}_{max}(n^s)$ and $\mathcal{B}_{max}(n^s)$ are the maximum available substrate node CPU and substrate link bandwidth, respectively. Revenue increases with higher acceptance ratios and improves by increasing node and link utilizations [25].

III. Data Center Network Topologies

Data Center Networks (DCNs) are virtualized to meet the increasing network demands and to better utilize limited physical resources [3], [7], [42]. DCNs are categorized as server-centric, switch-centric, and hybrid topologies based on the implementation of packet forwarding [30], [39]. In server-centric topologies (Mesh-of-Torus, FiConn, DCell, CamCube, BCube,) [13], [34], [35], [40], servers (hosts) are used for data forwarding, computation, and storage while in switch-centric topologies (Three-Tier, Spine-Leaf, Fat-Tree, F^2Tree, Diamond, and Collapsed Core) [1], [2], [4], switches are responsible for network interconnections as well as data forwarding and servers are used for computation and storage. Hybrid topologies (star-wired ring) [15], [45] employ both switches and servers for routing tasks.

DCell [19], [36] shown in Fig. 1 (top) is a recursive server-centric DCN topology that consists of servers and mini-switches with each server connected to a single mini switch within a cell. Interconnections between servers are formed recursively. DCell_k, the k^{th} level of a DCell topology, is constructed using the $(k-1)^{th}$ DCell as a building block. BCube [20] shown in Fig. 1 (bottom) is a recursive server-centric DCN topology with k levels and n hosts. BCube_0 is level-0 of the DCN topology and contains all hosts that are connected to n-port switches.

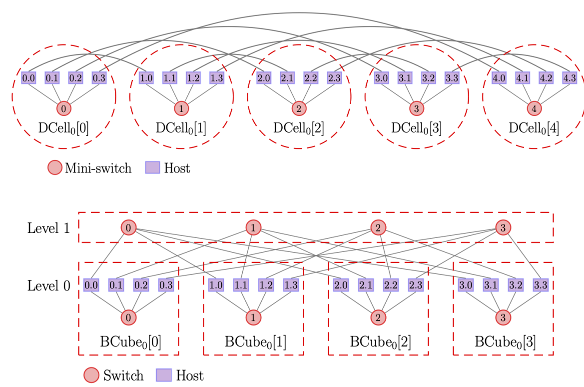

Fig. 1. Server-centric DCN topologies: DCell(1,4) (top) with level $k = 1$, 5 DCell_0s, and 4 hosts in each DCell_0 [19]. BCube(2,4) (bottom) with 4 BCube_0s and 4 hosts in each BCube_0 [20].

The switch-centric Three-Tier topology [1] shown in Fig. 2 (top) is a widely used multi-tiered topology that consists of core, aggregation, and edge layers. The Spine-Leaf topology [4] consists of spine and leaf layers where each switch in the spine layer is connected to all the switches in the leaf layer, as shown in Fig. 2 (middle). Collapsed Core [2] is a cost-effective version of the Three-Tier topology because the core and aggregation layers are combined into a single layer, as illustrated in Fig. 2 (bottom).

IV. Simulation Scenarios and Results

Performance of VNE algorithms is evaluated for a range of VNR traffic loads using various DCN topologies (substrate

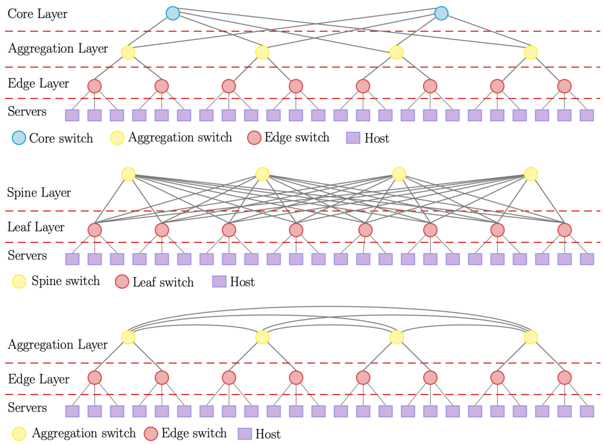

Fig. 2. Switch-centric DCN topologies: Three-Tier (top) with core, aggregation, and edge layers [1]. Spine-Leaf (middle) with spine (aggregation) and leaf (edge) layers [4]. Collapsed Core (bottom) with aggregation and edge layers [2].

networks). We implement the DCell topology using the VNE-Sim discrete event simulator [24] to evaluate performance of MaVEn-M, MaVEn-S, D-ViNE, R-ViNE, GRC, and GRC-M algorithms. DCell performance is then compared to server-centric (BCube) and switch-centric (Three-Tier, Spine-Leaf, and Collapsed Core) topologies. Elements of SNs for various DCN topologies and parameters used to generate VNs are listed in Table I. CPU of substrate nodes and bandwidth of substrate links are 100 units. Each virtual node is connected to a maximum of 3 virtual nodes. The VNR arrival process is assumed to be Poisson with a mean arrival rate of λ requests per unit time and exponential lifetime of $\frac{1}{\mu}$ thus generating traffic of $\lambda \times \frac{1}{\mu}$ Erlangs. Each simulation scenario lasts 50,000 time units. A computational budget (number of simulations) $\beta = 5$ is used for MaVEn-M and MaVEn-S algorithms. Processing times and performance results for VNE algorithms using various DCN topologies are shown in Fig. 3 and Fig. 4, respectively. Experiments are conducted using an MS-Surface platform with 8 GB memory and Intel i5-7200U processor.

TABLE I
ELEMENTS OF DCN TOPOLOGIES AND VIRTUAL NETWORK PARAMETERS

DCN topology	**Servers (hosts)**	**Switches (layer/level k)**	**Links**
DCell(1,4)	20	5 (DCell_0)	30
BCube(2,4)	16	4 (BCube_1), 4 (BCube_0)	32
Three-Tier	90	3 (core), 6 (aggregation), 18 (edge)	126
Spine-Leaf	90	6 (spine), 18 (leaf)	198
Collapsed Core	90	6 (spine), 18 (leaf)	123
Parameter		**Value**	**Distribution**
CPU		[2, 20] units	uniform
Link bandwidth		[1, 10] units	uniform
Link splitting rate		0.1	
Lifetime mean rate		1,000	exponential
Number of nodes		[3, 10]	uniform

The simulation results indicate that DCell topology offers high performance while requiring fewer network elements. Comparable performance is exhibited in case of both MCTS based algorithms while the highest acceptance ratio is achieved using MaVEn-M algorithm. Both MaVEn-M and MaVEn-S algorithms outperform other VNE algorithms in terms of revenue to cost ratio with MaVEn-S algorithm exhibiting the best performance. The highest node and link utilizations are obtained using MaVEn-M algorithm. Although GRC algorithm requires shorter processing times, it is outperformed by GRC-M, D-ViNE, and R-ViNE algorithms for all metrics except revenue to cost ratio. The lowest revenue to cost ratio is observed for D-ViNE and R-ViNE algorithms.

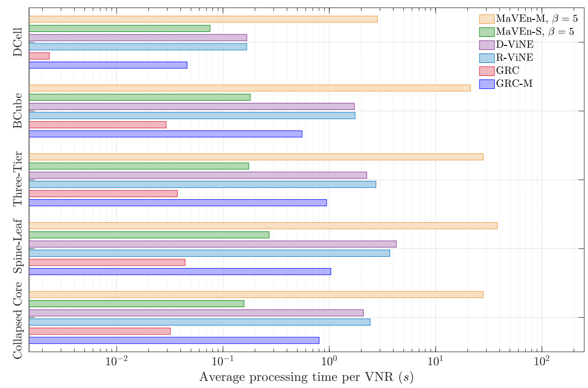

Fig. 3. Processing times for VNE algorithms using various DCN topologies.

Performance of VNE algorithms has been evaluated using server-centric (BCube) [23] and switch-centric (Three-Tier, Spine-Leaf, Collapsed Core) [18] DCN topologies. Reported results indicate that Spine-Leaf and BCube offer higher acceptance ratios than DCell. Similar acceptance ratios are obtained for DCell, Three-Tier, and Collapsed Core topologies. Unlike other topologies, revenue to cost ratios for DCell improve more than 0.05 units with increased traffic load. When using MCTS based algorithms, DCell offers higher revenue to cost ratios than BCube, Three-Tier, Spine-leaf, and Collapsed Core topologies. Higher link and node utilizations are obtained using DCell in comparison to BCube, Three-Tier, Spine-Leaf, and Collapsed Core topologies. Because of DCell's recursive connections between hosts, links and nodes are better utilized while network offers comparable acceptance and revenue to cost ratios using fewer network elements than other topologies. Increasing the the maximum number of connected nodes from 3 to 5 within virtual networks significantly affects acceptance ratio and resource utilizations in server-centric topologies.

V. CONCLUSION

We evaluated performance of VNE algorithms using DCell topology and compared performance between server-centric (BCube) and switch-centric (Three-Tier, Spine-Leaf, Collapsed Core) DCN topologies using the publicly available discrete event simulator VNE-Sim. Performance was evaluated based on acceptance ratio, revenue to cost ratio, and node and link utilizations. MaVEn-M and MaVEn-S algorithms outperformed other algorithms for most performance measures due to their optimized embeddings. Comparable performance was achieved using DCell having fewer network elements than other DCN topologies.

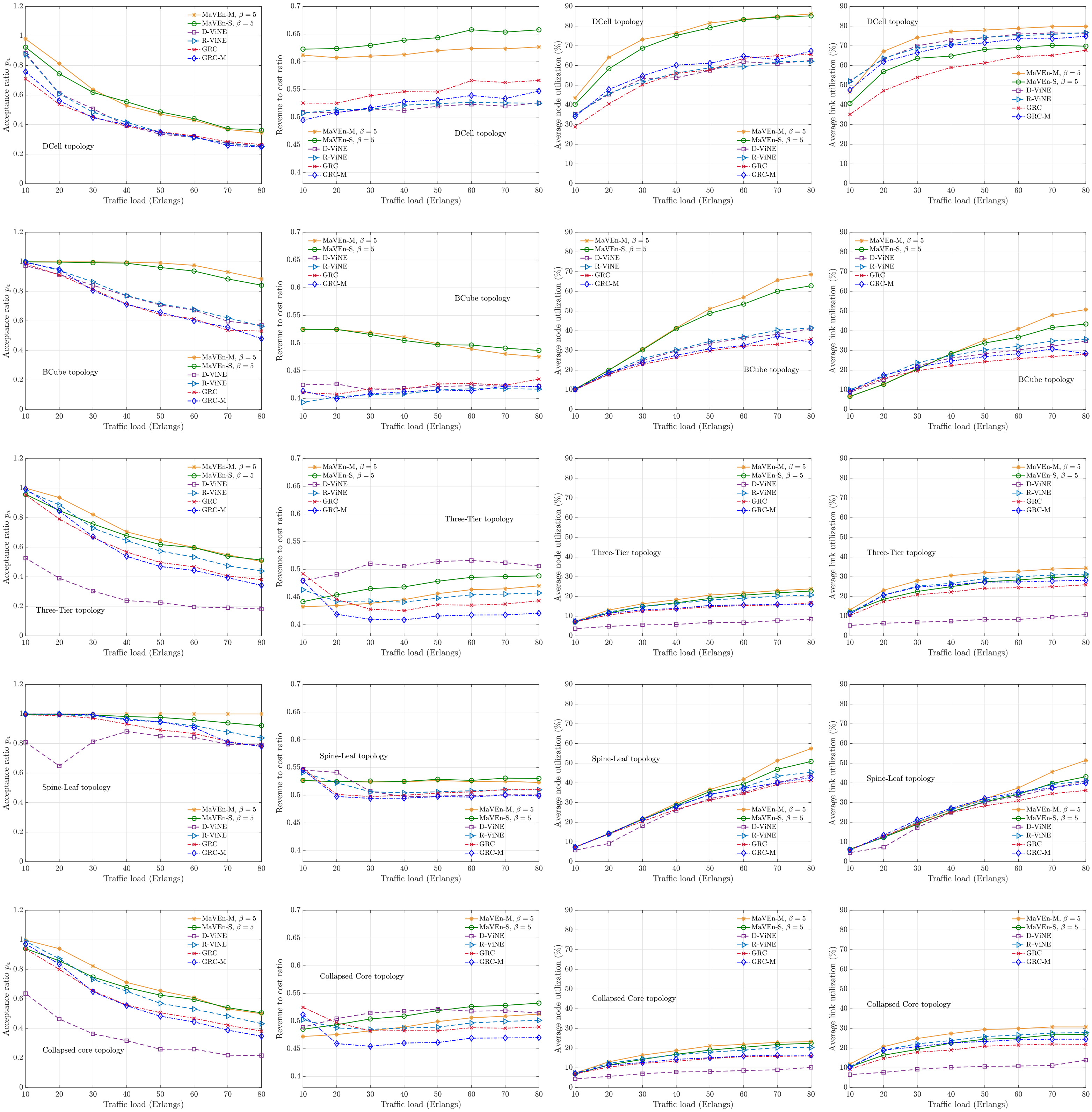

Fig. 4. Comparison of server-centric and switch-centric DCN topologies: DCell (1st row), BCube (2nd row), Three-Tier (3rd row), Spine-Leaf (4th row), and Collapsed Core (5th row). Shown are acceptance ratio, revenue to cost ratio, and average node and link utilizations as functions of VNR traffic load. Generated results indicate that recursive connections between the hosts in DCell lead to better link and node utilizations with comparable acceptance and revenue to cost ratios while using fewer network elements than in other topologies.

References

[1] Data center multi-tier model design, Cisco. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Data_Center/DC_Infra2_5/DCInfra_2.html. Accessed: Feb. 15, 2022.

[2] Cisco networking academy connecting networks companion guide: hierarchical network design. [Online]. Available: https://www.ciscopress.com/articles/article.asp?p=2202410&seqNum=4. Accessed: Feb. 15, 2022.

[3] Cisco global cloud index: forecast and methodology, 2016–2021. [Online]. Available: https://virtualization.network/Resources/Whitepapers/0b75cf2e-0c53-4891-918e-b542a5d364c5_white-paper-c11-738085.pdf. Accessed: Feb. 15, 2022.

[4] Cisco data center spine-and-leaf architecture: design overview white paper. [Online]. Available: https://www.cisco.com/c/en/us/products/collateral/switches/nexus-7000-series-switches/white-paper-c11-737022.html. Accessed: Feb. 15, 2022.

[5] Why companies need data center virtualization right now. [Online]. Available: https://www.newhorizons.com/article/why-companies-need-data-center-virtualization-right-now. Accessed: Feb. 15, 2022.

[6] D. G. Andersen, "Theoretical approaches to node assignment," Dec. 2002, unpublished manuscript.

[7] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Gener. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, May 2012.

[8] H. Cao, H. Zhu, and L. Yang, "Collaborative attributes and resources for single-stage virtual network mapping in network virtualization," *J. Comm. Netw.*, vol. 22, no. 1, pp. 61–71, Feb. 2020.

[9] X. Cheng, S. Su, Z. Zhang, H. Wang, F. Yang, Y. Luo, and J. Wang, "Virtual network embedding through topology-aware node ranking," *Comput. Commun. Rev.*, vol. 41, pp. 38–47, Apr. 2011.

[10] L. Chen, Y. Feng, B. Li, and B. Li, "Promenade: proportionally fair multipath rate control in datacenter networks with random network coding," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 11, pp. 2536–2546, Nov. 2019.

[11] M. Chowdhury, M. R. Rahman, and R. Boutaba, "ViNEYard: Virtual network embedding algorithms with coordinated node and link mapping," *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 206–219, Feb. 2012.

[12] R. Coulom, "Efficient selectivity and backup operators in Monte-Carlo tree Search," in *Proc. 5th Int. Conf. Comput. Games*, Turin, Italy, May 2006, pp. 72–83.

[13] P. Costa, A. Donnelly, G. O'Shea, and A. Rowstron, "CamCube: a key-based data center," Microsoft Research, Technical Report MSR TR-2010-74, 2010.

[14] N. Feamster, L. Gao, and J. Rexford, "How to lease the Internet in your spare time," *Comput. Commun. Rev.*, vol. 37, no. 1, pp. 61–64, Jan. 2007.

[15] Z. Feng, W. Sun, J. Zhu, J. Shao, and W. Hu, "Resource allocation in electrical/optical hybrid switching data center networks," *IEEE J. Opt. Commun. Netw.*, vol. 9, no. 8, pp. 648–657, Aug. 2017.

[16] A. Fischer, J. F. Botero, M. T. Beck, H. de Meer, and X. Hesselbach, "Virtual network embedding: a survey," *IEEE Commun. Surveys Tut.*, vol. 15, no. 4, pp. 1888–1906, 2013.

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

[18] A. L. Gonzalez Rios, K. Bekshentayeva, M. Singh, S. Haeri, and Lj. Trajković, "Virtual network embedding for switch-centric data center networks," in *Proc. IEEE Int. Symp. Circuits Syst.*, Daegu, Korea, May 2021.

[19] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, "DCell: a scalable and fault-tolerant network structure for data centers," in *Proc. ACM SIGCOMM*, Seattle, WA, USA, Aug. 2008, pp. 75–86.

[20] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, "BCube: a high performance, server-centric network architecture for modular data centers," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 63–74, Oct. 2009.

[21] A. Hammad, R. Nejabati, and D. Simeonidou, "Cross-layer optimization of network resource virtualization in IP over O-OFDM networks," *J. Opt. Commun. Netw.*, vol. 8, no. 10, pp. 765–776, 2016.

[22] S. Haeri, Q. Ding, Z. Li, and Lj. Trajković, "Global resource capacity algorithm with path splitting for virtual network embedding," in *Proc. IEEE Int. Symp. Circuits Syst.*, Montreal, QC, Canada, May 2016, pp. 666–669.

[23] S. Haeri and Lj. Trajković, "Virtual network embeddings in data center networks," in *Proc. IEEE Int. Symp. Circuits and Systems*, Montreal, QC, Canada, May 2016, pp. 874–877.

[24] S. Haeri and Lj. Trajković, "VNE-Sim: a virtual network embedding simulator," in *Proc. Simutools*, Prague, Czech Republic, Aug. 2016, pp. 112–117.

[25] S. Haeri and Lj. Trajković, "Virtual network embedding via Monte-Carlo tree search," *IEEE Trans. Cybernetics*, vol. 48, no. 2, pp. 510–521, Feb. 2018.

[26] A. Hammadi and L. Mhamdi, "A survey on architectures and energy efficiency in data center networks," *Comput. Commun.*, vol. 40, pp. 1–21, Mar. 2014.

[27] A. Hbaieb, M. Khemakhem, and M. Ben Jemaa, "A survey and taxonomy on virtual data center embedding," *J. Supercomput.* vol. 75, no. 10, pp. 6324–6360, Oct. 2019.

[28] R. Hintemann, "Consolidation, colocation, virtualization, and cloud computing: the impact of the changing structure of data centers on total electricity demand," *ICT Innovations for Sustainability. Advances in Intelligent Systems and Computing*, L. M. Hilty and B. Aebischer, Eds., Springer, 2015, vol. 310, pp. 125–136.

[29] L. Kocsis and C. Szepesvári, "Bandit based Monte-Carlo planning," in *Lecture Notes in Computer Science: Proc. 17th European Conference on Machine Learning (ECML)*, J. Furnkranz, T. Scheffer, and M. Spiliopoulou, Eds., Springer, 2006, vol. 4212, pp. 282–293.

[30] M. Manzano, K. Bilal, E. Calle, and S. U. Khan, "On the connectivity of data center networks," *IEEE Commun. Lett.*, vol. 17, no. 11, pp. 2172–2175, Nov. 2013.

[31] M. R. Rahman and R. Boutaba, "SVNE: survivable virtual network embedding algorithms for network virtualization," in *IEEE Trans. Netw. Service Manage.*, vol. 10, no. 2, pp. 105–118, June 2013.

[32] A. Razzaq and M. Rathore, "An approach towards resource efficient virtual network embedding," in *Proc. Internet 2010*, Valencia, Spain, Sept. 2010, pp. 68–73.

[33] T. G. Robertazzi, "Software-defined networking," *Introduction to Computer Networking*, 2017, pp. 81–87.

[34] T. Wang, Z. Su, Y. Xia, J. Muppala, M. Hamdi, "Designing efficient high performance server-centric data center network architecture," *Comput. Netw.*, vol. 79, pp. 283–296, Mar. 2015.

[35] B. Wang, Z. Qi, R. Ma, H. Guan, and A. V. Vasilakos, "A survey on data center networking for cloud computing," *Comput. Netw.*, vol. 91, pp. 528–547, Nov. 2015.

[36] X. Wang, "Fault-tolerant routing in DCell networks," in *Proc. 2021 IEEE Asia-Pacific Conf. Image Process., Electron., Comput.*, Apr. 2021, pp. 32–36.

[37] H. Wu, F. Zhou, Y. Chen, and R. Zhang, "On virtual network embedding: paths and cycles," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 3, pp. 1487–1500, Sept. 2020.

[38] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 27–51, First Quarter 2015.

[39] W. Xia, P. Zhao, Y. Wen, and H. Xie, "A survey on data center networking (DCN): infrastructure and operations," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 640–656, First Quarter 2017.

[40] P. Xie, H. Gu, K. Wang, X. Yu, and S. Ma, "Mesh-of-Torus: a new topology for server-centric data center networks," *J. Supercomput.*, vol. 75, no. 9, pp. 255–271, Sept. 2019.

[41] M. Yang, H. Rastegarfar, and I. B. Djordjevic, "Physical-layer adaptive resource allocation in software-defined data center networks," *IEEE J. Opt. Commun. Netw.*, vol. 10, no. 12, pp. 1015–1026, Dec. 2018.

[42] J. Zhang, F. R. Yu, S. Wang, T. Huang, Z. Liu, and Y. Liu, "Load balancing in data center networks: a survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2324–2352, Third Quarter 2018.

[43] P. Zhang, H. Yao, and Y. Liu, "Virtual network embedding based on computing, network, and storage resource constraints," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3298–3304, Oct. 2018.

[44] K. Zhu, Z. Yao, N. He, D. Li, and L. Zhang, "Toward full virtualization of the network topology," *IEEE Syst. J.*, vol. 13, no. 2, pp. 1640–1649, June 2019.

[45] Y. Zong, Y. Ou, A. Hammad, K. Kondepu, R. Nejabati, D. Simeonidou, Y. Liu, and L. Guo, "Location-aware energy efficient virtual network embedding in software-defined optical data center networks," *J. Opt. Commun. Netw.*, vol. 10, no. 7, pp. 58–70, July 2018.