Topologies and Algorithms for Data Center Networks

Ljiljana Trajković

Communication Networks Laboratory
http://www.ensc.sfu.ca/~ljilja/cnl/
Simon Fraser University
Vancouver, British Columbia, Canada

Communication Networks Laboratory

- Ph.D. students:
  - Soroush Haeri
  - Hanene Ben Yedder
  - Qingue Ding
  - Zhida Li
  - Umme Zakia

Roadmap

- Data center networks and their topologies
- Network virtualization
- Virtual network embedding algorithms
- Simulation results
- Conclusions and references

Data Center Networks (DCNs)

- Data centers are core infrastructure of cloud computing
- They provide:
  - cost-effective infrastructure for storing data and hosting large-scale service applications
  - infrastructure for providers of Internet applications (Amazon, Google, Facebook)
- DCN architecture needs flexibility to effectively support applications that have diverse resource requirements from the underlying infrastructure:
  - storage, computing power, bandwidth, and latency

DCN topologies

- Switch-Centric:
  - Conventional
  - Fat-Tree
  - F^2Tree
  - Diamond
- Server-Centric:
  - BCube
  - DCell
  - FiConn
- Enhanced topologies (optical or wireless designs)

Virtual network embedding (VNE) algorithms

- Global Resource Capacity Multicommodity Flow (GRC-M)
- Global Resource Capacity (GRC)
- D-ViNE and R-ViNE

- 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 and Systems*, Montreal, Canada, May 2016, pp. 666–669.
- 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.
- 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.

VNE in data center networks

- Data center networks have defined topologies
- Topological features significantly affect quality of the VNE solution
- Goal:
  - Identify the network topology that is better suited for VNE

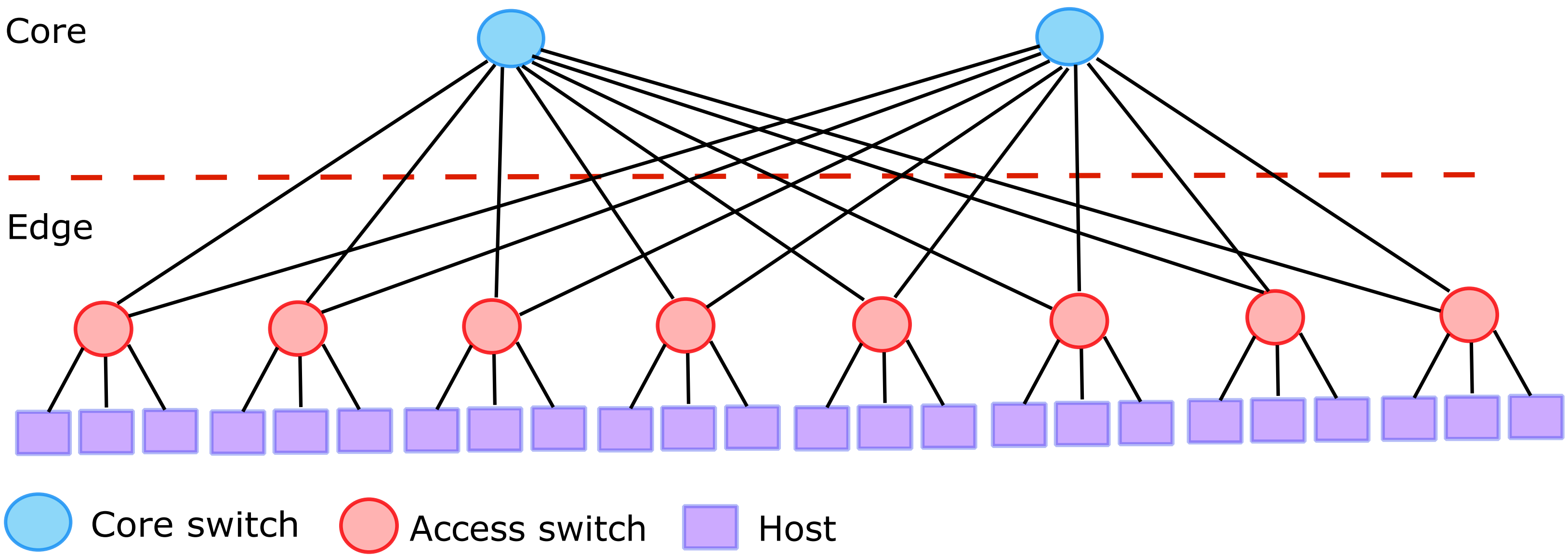
Switch-centric topologies: Conventional

- Based on tree topology:
  - core layer (layer-3) of switches/routers
  - aggregation layer (layer-2) of switches
  - edge layer (layer-1) of top-of-rack access switches
- Core layer: responsible for routing and balancing traffic load between the core and the aggregation layer
- Aggregation layer: provides default gateway redundancy, spanning tree processing, load balancing, and firewall
- Edge layer: each switch is connected to two aggregation switches for redundancy

  - T. Chen, X. Gao, and G. Chen, “The features, hardware, and architectures of data center networks: a survey,” *J. Parallel Distrib. Comput.*, vol. 96, pp. 45–774, Oct. 2016.

Switch-centric topologies: Conventional

- Conventional (two-tier):
  10 switches, 24 hosts, and 40 links



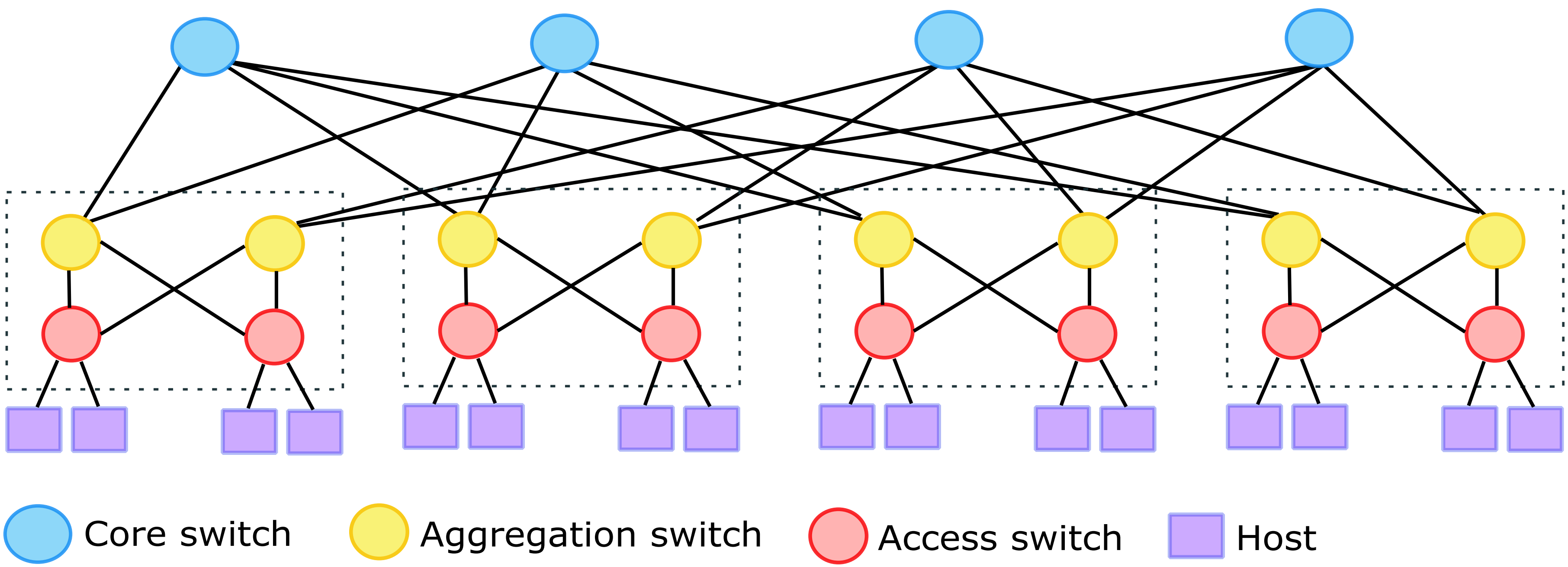
Switch-centric topologies: Fat-Tree

- Notation: Fat-Tree$_K$
- Special Clos architecture
- Initially proposed to interconnect processors of parallel supercomputers
- $(k/2)^2 + k^2$ k-port switches
- Supports $k^3/4$ hosts

- C. E. Leiserson, "Fat-Trees: universal networks for hardware-efficient supercomputing," *IEEE Trans. Comput.*, vol. 30, no. 10, pp. 892–901, Oct. 1985.
- M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 63–74, Oct. 2008.

Switch-centric topologies: Fat-Tree

- Fat-Tree$_4$:
  20 4-port switches, 16 hosts, and 48 links



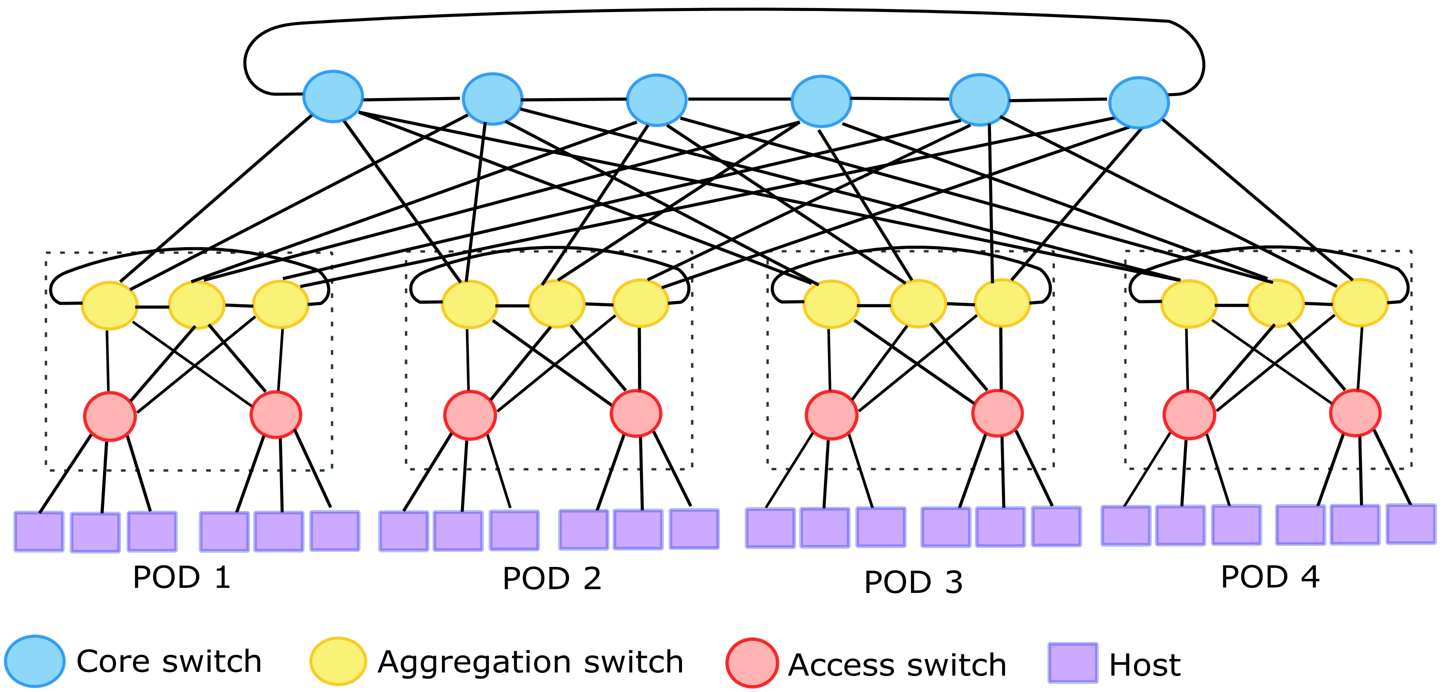
Switch-centric topologies: F^2Tree

- Notation: F^2Tree
- An enhancement of the Fat-Tree topology
- $(5/4) \times k^2 - (7/2) \times k + 2$ switches
- Supports $k^3/4 - k^2 + k$ hosts

- G. Chen, Y. Zhao, D. Pei, and D. Li, "Rewiring 2 links is enough: Accelerating failure recovery in production data center networks," in *Proc. IEEE ICDCS*, Columbus, Ohio, USA, June 2015, pp. 569–578.

Switch-centric topologies: F²Tree

- F²Tree:
  26 switches, 24 hosts, and 90 links



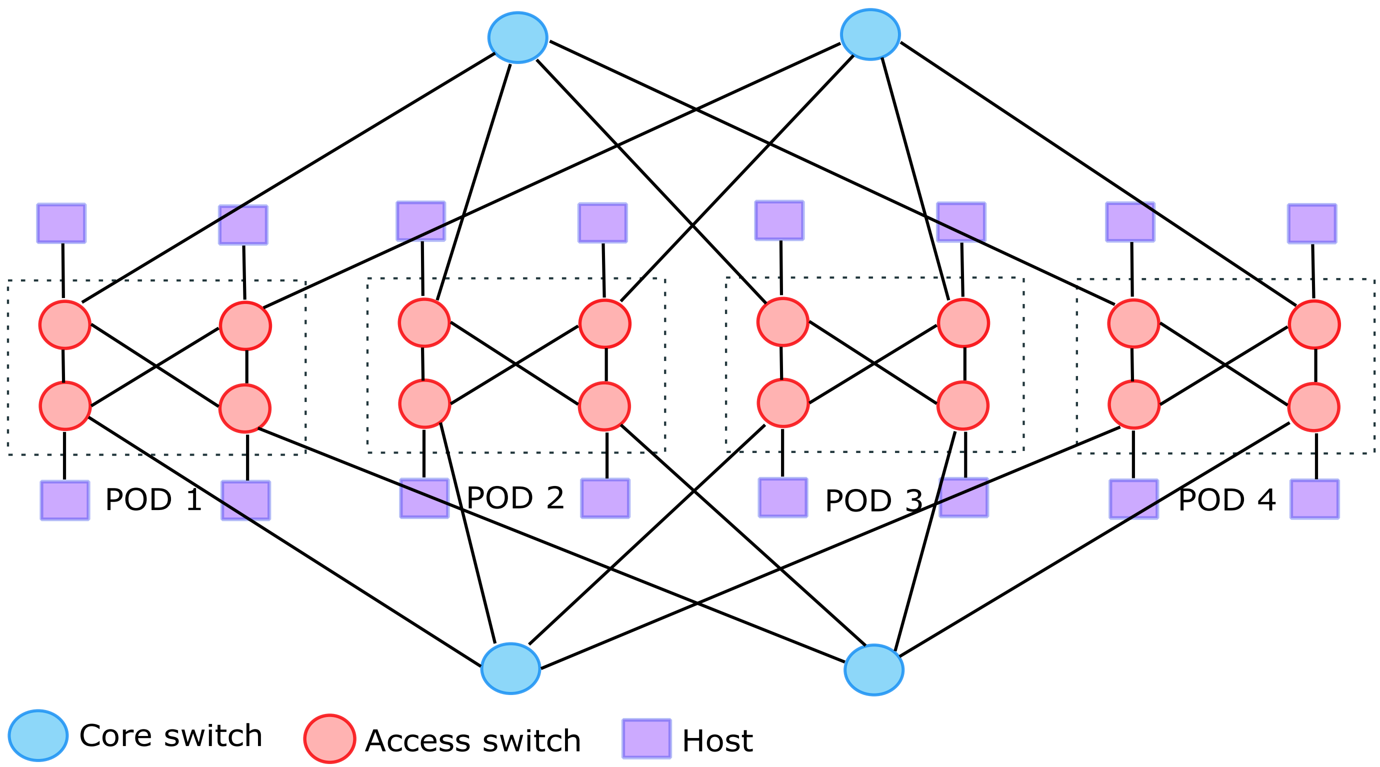
Switch-centric topologies: Diamond

- Notation: Diamond
- Variation of the Fat-Tree topology
- It has symmetrical architecture where the core switches are divided in two layers
- $k^2/4$ switches
- Supports $k^3/4$ hosts

- Y. Sun, J. Chen, Q. Lu, and W. Fang, "Diamond: an improved fat-tree architecture for large-scale data centers," *J. Commun.*, vol. 9, no. 1, pp. 91–98, Jan. 2014.

Switch-centric topologies: Diamond

- Diamond:
  20 switches, 16 hosts, and 48 links



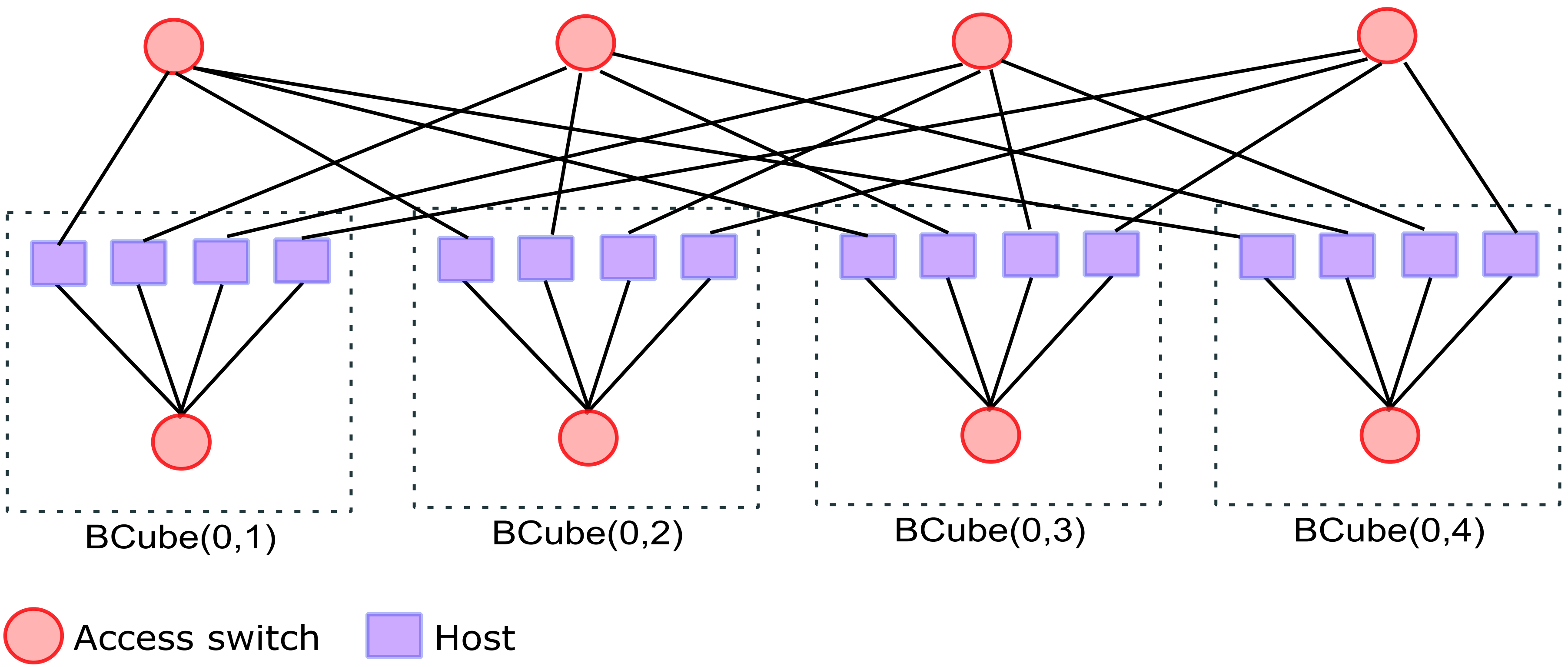
Server-centric topologies: BCube

- Notation: BCube(k,n)
  - k: BCube level
  - n: number of hosts in the level-0 BCube
- Recursively structured
- Switches are not directly interconnected
- Hosts perform packet forwarding functions

- 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.

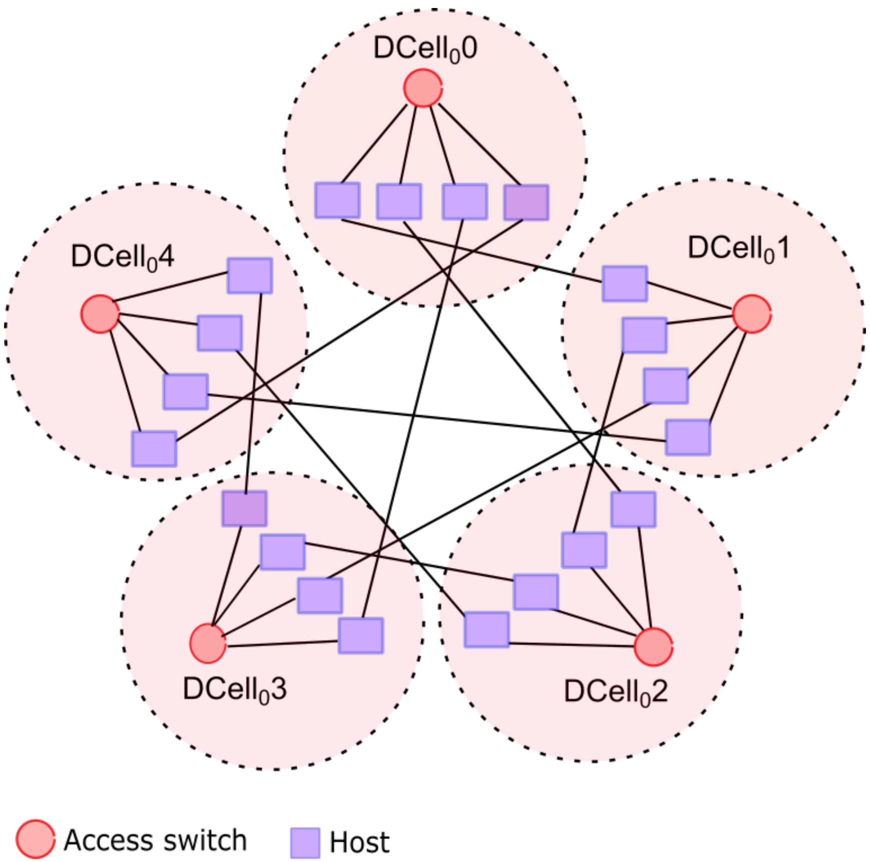
Server-centric topologies: BCube

- BCube (2,4):
  8 switches, 16 hosts, and 32 links



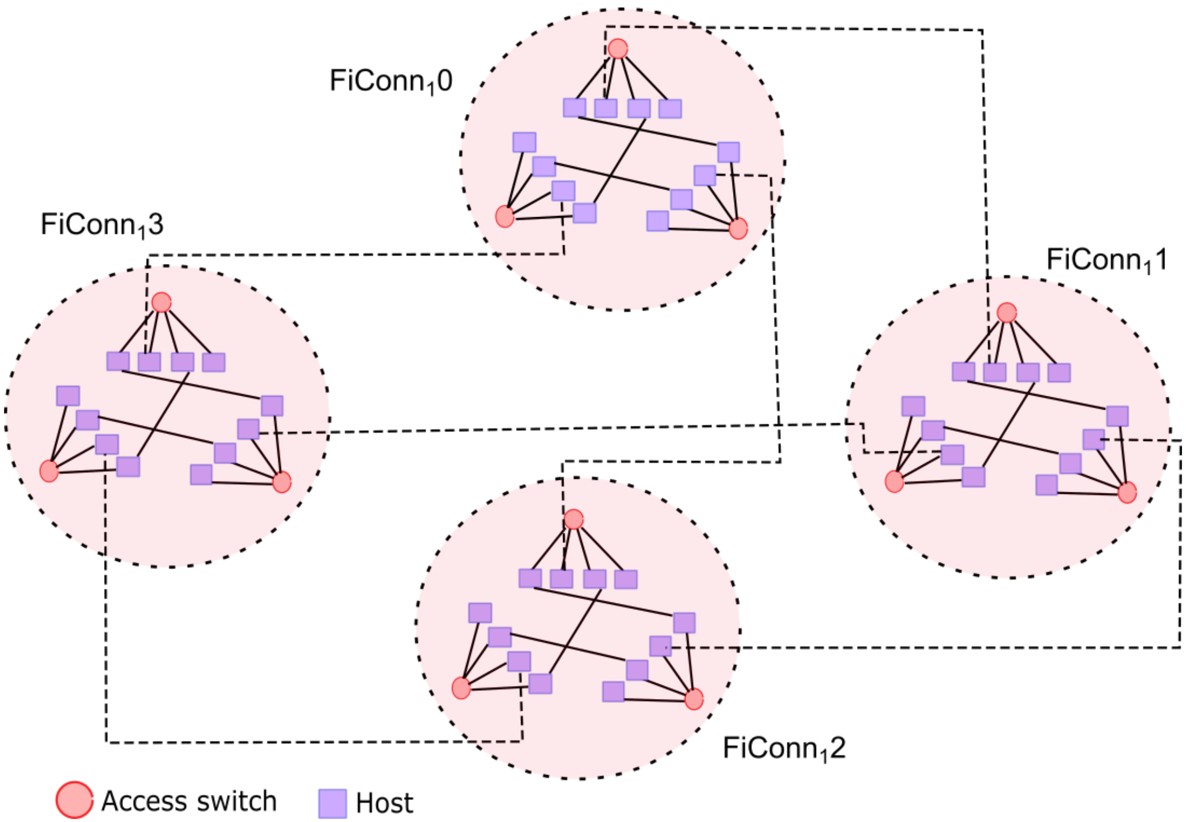
Server-centric topologies: DCell

- DCell$_1$ structure with 4 ports consisting of 5 DCell$_0$s



Server-centric topologies: FiConn

- FiConn$_2$ structure with 4 ports. Each Ficonn$_1$ contains 3 Ficonn$_0$ ports



Network virtualization

- Network virtualization enables coexistence of multiple virtual networks on a shared physical infrastructure
- Provides:
  - flexible management
  - lower implementation cost
  - higher network scalability
  - increased resource utilization, and
  - improved energy efficiency

Network virtualization

- Virtualized network model divides the role of Internet Service Providers (ISPs) into:
  - Infrastructure Providers (InPs)
    - manage the physical infrastructure
  - Service Providers (SPs)
    - aggregate resources from multiple InP into multiple Virtual Networks (VNs)

Substrate network vs. virtual network

- InPs operate physical substrate networks (SNs)
- SN components:
  - physical nodes (substrate nodes)
  - physical links (substrate links)
- Substrate nodes and links are:
  - interconnected using arbitrary topology
  - used to host various virtualized networks with arbitrary topologies
- Virtual networks are embedded into a substrate network

Virtual network embedding

- Virtual Network Embedding (VNE) allocates SN resources to VNs
- InP's revenue depends on VNE efficiency
- VNE problem may be reduced to the multi-way separator:
  - NP-hard
  - optimal solution may only be obtained for small instances

- M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: substrate support for path splitting and migration," *Comput. Commun. Rev.*, vol. 38, no. 2, pp. 19–29, Mar. 2008.

VNE solution

- Two subproblems:
  - Virtual Node Mapping (VNoM): maps virtual nodes to substrate nodes
  - Virtual Link Mapping (VLiM): maps virtual links to substrate paths
- VNE algorithms address the VNoM while solving the VLiM using:
  - Shortest-Path (SP) algorithms

  or

  - Multicommodity Flow (MCF) algorithm

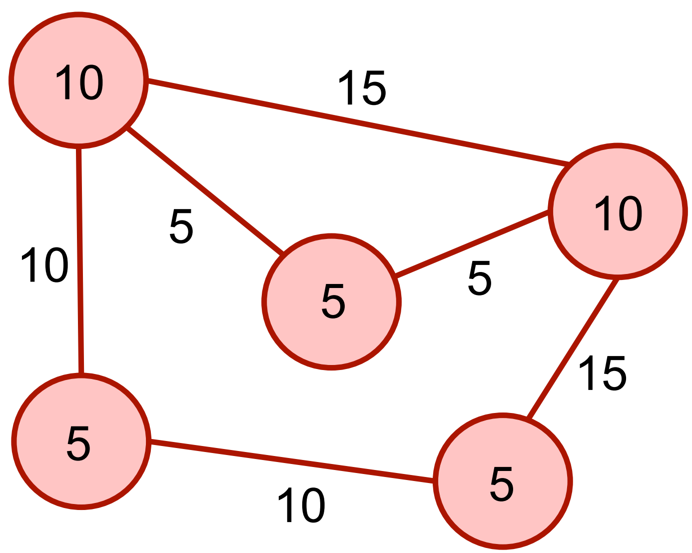
VNE solution: VLiM and path splitting

- The shortest-path algorithms do not permit path splitting:
  - stricter than the MCF algorithm
- MCF enables path splitting:
  - a flow may be divided into multiple flows with lower capacity
  - flows are routed through various paths

- D. G. Andersen, “Theoretical approaches to node assignment,” Dec. 2002, Unpublished Manuscript. [Online]. Available: http://repository.cmu.edu/compsci/86/.

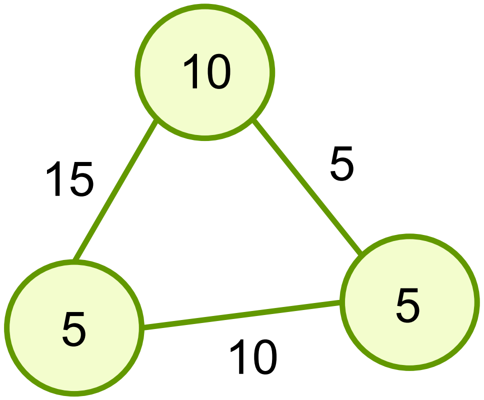
VNE formulation: constrains

- Substrate network graph: $G^s(N^s, E^s)$
- Resources:
  - substrate nodes: CPU capacity $\mathcal{C}(n^s)$
  - substrate links: bandwidth $\mathcal{B}(e^s)$

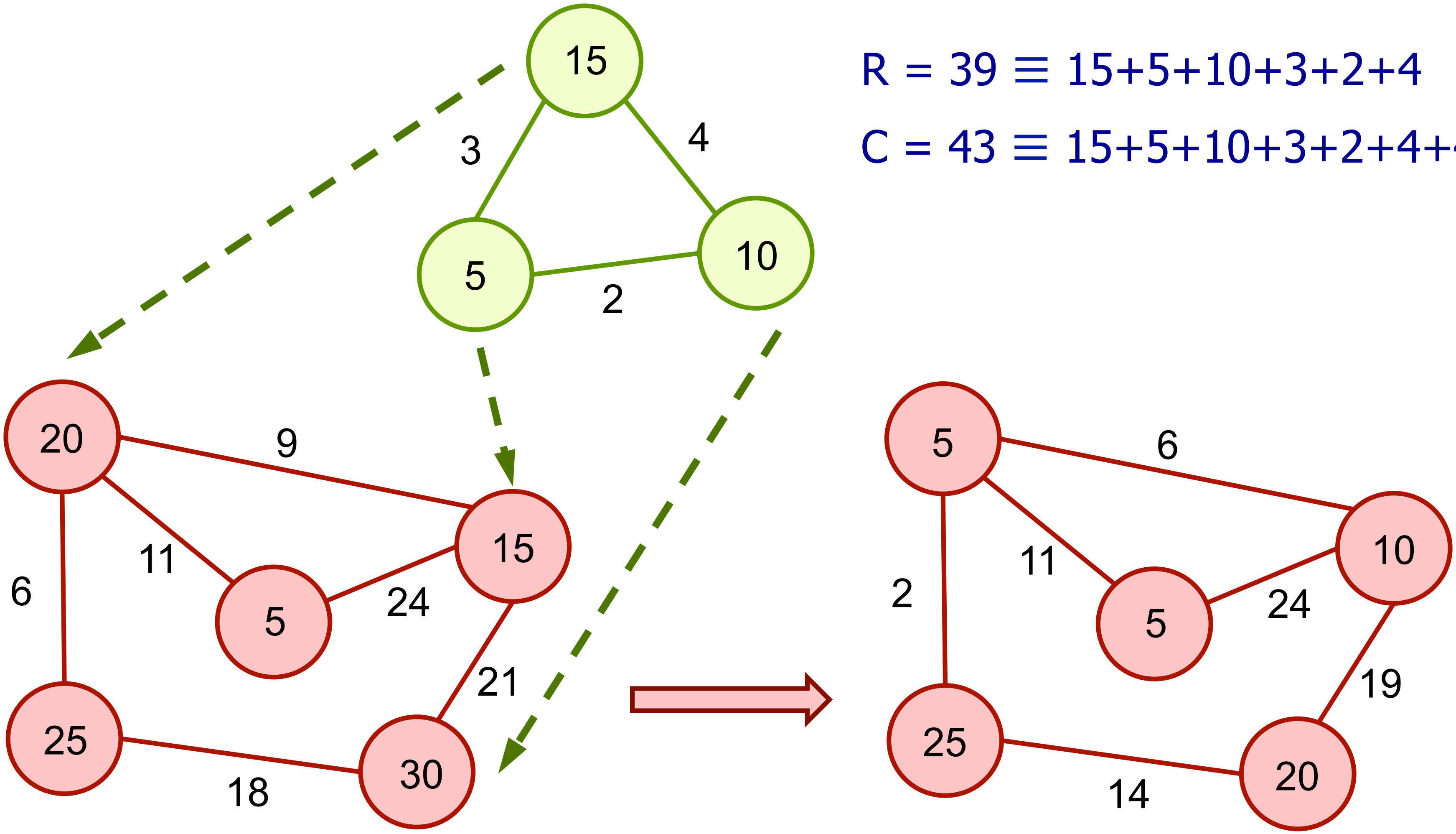


VNE formulation: constrains

- Virtual network graph: $G^{\Psi_i}(N^{\Psi_i}, E^{\Psi_i})$
- Resources:
  - virtual nodes: CPU capacity $\mathcal{C}(n^{\Psi_i})$
  - virtual links: bandwidth $\mathcal{B}(e^{\Psi_i})$



VNE: example



$R = 39 \equiv 15+5+10+3+2+4$

$C = 43 \equiv 15+5+10+3+2+4+4$

VNE objective

- Maximize the profit of InPs
- Contributing factors to the generated profit:
  - embedding revenue
  - embedding cost
  - acceptance ratio

- 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.
- 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.

VNE objective: revenue

- Maximize revenue:

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

  - w_c : weights for CPU requirements
  - w_b : weight for bandwidth requirements
  - general assumption: $w_c = w_b = 1$

VNE objective: revenue

- Generated revenue is not a function of the embedding configuration:
  - $\mathbf{R}(G^{\Psi_i})$ is constant regardless of the embedding configuration

VNE objective: cost

- Minimize the cost:

$$\mathbf{C}(G^{\Psi_i}) = \sum_{n^{\Psi_i} \in N^{\Psi_i}} \mathcal{C}(n^{\Psi_i}) + \sum_{e^{\Psi_i} \in E^{\Psi_i}} \sum_{e^s \in E^s} f_{e^s}^{e^{\Psi_i}}$$

  - $f_{e^s}^{e^{\Psi_i}}$: total allocated bandwidth of the substrate link e^s for virtual link e^{Ψ_i}
- $\mathbf{C}(G^{\Psi_i})$ depends on the embedding configuration

VNE objective: acceptance ratio

- Maximize acceptance ratio:

$$p_a^\tau = \frac{|\Psi^a(\tau)|}{|\Psi(\tau)|}$$

  - $|\Psi^a(\tau)|$: number of accepted Virtual Network Requests (VNRs) in a given time interval τ
  - $|\Psi(\tau)|$: number of all arrived VNRs in τ

VNE objective function

- Objective of embedding a VNR is to maximize:

$$\mathcal{F}(\Psi_i) = \begin{cases} \mathbf{R}(G^{\Psi_i}) - \mathbf{C}(G^{\Psi_i}) & \text{successful embeddings} \\ \Gamma & \text{otherwise} \end{cases}$$

  - Γ : large negative penalty for unsuccessful embedding
- The upper bound:

$$\mathcal{F}(\Psi_i) \leq 0$$

VNE algorithms: Global Resource Capacity Multicommodity Flow (GRC-M)

- GRC-M employs the Global Resource Capacity (GRC) for virtual node mapping while using the Multicommodity Flow algorithm for identifying the link mappings
- GRC algorithm is effective in calculating the embedding potential of substrate nodes
- MCF algorithm enables path splitting, which improves resources utilization

- 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 and Systems*, Montreal, Canada, May 2016, pp. 666–669.

VNE algorithms: Global Resource Capacity Multicommodity Flow (GRC-M)

- GRC algorithm calculates the embedding capacity $r(n_i^s)$ for a substrate node n_i^s :

$$\mathbf{r} = (1-d)\hat{\mathbf{c}} + d\mathbf{M}\mathbf{r}$$

- Objective of the MCF algorithm:

$$\text{minimize} \quad \sum_{e^s \in E^s} \frac{1}{\mathcal{B}(e^s) + \epsilon} \sum_{e^{\Psi_i} \in E^{\Psi_i}} f_{e^s}^{e^{\Psi_i}}$$

VNE algorithms: Global Resource Capacity Multicommodity Flow (GRC-M)

- Substrate network graph: $G^s(N^s, E^s)$
- Resources:
  - substrate nodes: CPU capacity $\mathcal{C}(n^s)$
  - substrate links: bandwidth $\mathcal{B}(e^s)$
- Virtual network graph: $G^{\Psi_i}(N^{\Psi_i}, E^{\Psi_i})$
- Resource requirements:
  - virtual nodes: CPU capacity $\mathcal{C}(n^{\Psi_i})$
  - virtual links: bandwidth $\mathcal{B}(e^{\Psi_i})$
- w_c : weight for CPU requirements (= 1)
- w_b : weight for bandwidth requirements (= 1)
- $f_{e^s}^{e^{\Psi_i}}$: total bandwidth of the substrate link e^s allocated to virtual link
- $|\Psi^a(\tau)|$: number of accepted Virtual Network Requests (VNRs) in a given time interval τ
- $|\Psi(\tau)|$: number of VNRs that arrived in τ

VNE algorithms: Global Resource Capacity (GRC)

- Node-ranking-based algorithm:
  - computes a score/rank for substrate and virtual nodes
  - employs a large-to-large and small-to-small mapping scheme to map the virtual nodes to substrate nodes
- Employs the Shortest-Path algorithm to solve VLiM
- Outperforms earlier similar proposals

- 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.

VNE algorithms: Global Resource Capacity (GRC)

- Calculates the embedding capacity $r(n_i^s)$ for a substrate node n_i^s:

$$r(n_i^s) = (1-d)\hat{\mathcal{C}}(n_i^s) + d \sum_{n_j^s \in \mathcal{N}(n_i^s)} \frac{\mathcal{B}\big(e^s(n_i^s, n_j^s)\big)}{\sum_{n_k^s \in \mathcal{N}(n_j^s)} \mathcal{B}\big(e^s(n_j^s, n_k^s)\big)}$$

  - $0 < d < 1$: damping factor
  - $e^s(n_i^s, n_j^s)$: substrate link connecting n_i^s and n_j^s
  - $\hat{\mathcal{C}}(n_i^s)$: normalized CPU resource of n_i^s

$$\hat{\mathcal{C}}(n_i^s) = \frac{\mathcal{C}(n_i^s)}{\sum_{n^s \in N^s} \mathcal{C}(n^s)}$$

VNE algorithms: Global Resource Capacity (GRC)

- Matrix form:

$$\mathbf{r} = (1-d)\hat{\mathbf{c}} + d\mathbf{M}\mathbf{r}$$

  - $\hat{\mathbf{c}} = (\hat{\mathcal{C}}(n_1^s), \hat{\mathcal{C}}(n_2^s), \ldots, \hat{\mathcal{C}}(n_j^s))^T$
  - $\mathbf{r} = (r(n_1^s), r(n_2^s), \ldots, r(n_k^s))^T$
  - $\mathbf{M}$ is a k-by-k matrix:

$$m_{ij} = \begin{cases} \dfrac{\mathcal{B}\big(e^s(n_i^s, n_j^s)\big)}{\sum\limits_{n_k^s \in \mathcal{N}(n_j^s)} \mathcal{B}\big(e^s(n_j^s, n_k^s)\big)} & e^s(n_i^s, n_j^s) \in E^s \\ 0 & \text{otherwise} \end{cases}$$

VNE algorithms: Global Resource Capacity (GRC)

- $\mathbf{r}$ is calculated iteratively:

$$\mathbf{r}_{k+1} = (1-d)\mathbf{c} + d\mathbf{M}\mathbf{r}_k$$

- Initially: $\mathbf{r}_0 = \hat{\mathbf{c}}$
- Stop condition: $|\mathbf{r}_{k+1} - \mathbf{r}_k| < \sigma,$
  - $0 < \sigma << 1$

VNE algorithms: R-Vine and D-Vine

- Formulate VNE problem as a Mixed Integer Program (MIP)
- Their objective is to minimize the cost of accommodating the VNRs
- Use a rounding-based approach to obtain a linear programming relaxation of the relevant MIP
- Use Multicommodity Flow algorithm for solving VLiM

- 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.

Traffic

Fat-Tree:

- traffic forwarding is only performed by the switches

BCube:

- hosts are used to forward traffic
    - introduces additional traffic over the links that are connected to the hosts

Simulations: substrate networks

- Fat-Tree$_6$: 54 hosts, 45 switches, and 162 links
  - Switch to host ratio: 0.84
- BCube(2,4): 64 hosts, 48 switches, and 192 link
  - Switch to host ratio: 0.75
- CPU resources:
  - Hosts: 100 units
  - Switches: 0 units
- Bandwidth resources: 100 units per link

Simulations: virtual network graphs

- Waxman algorithm used to generate virtual network graphs:
  - $\alpha = 0.5$ and $\beta = 0.2$
  - number of nodes: uniformly distributed between 3 and 10
  - each virtual node: connected to a maximum of 3 virtual nodes

Simulations: virtual network graphs

- CPU requirements:
  - uniformly distributed between 2 and 20 units
- Bandwidth requirements:
  - uniformly distributed between 1 to 10 units
  - illustrates a substrate network with 10 Gbps links and virtual networks with 100 Mbps to 1 Gbps links

Simulations: other parameters

- Poisson distribution for arrivals with implies 1 to 8 units per 100 time units
- Exponentially distributed life-times with mean 1,000 time units
- Traffic loads: 10, 20, 30, 40, 50, 60, 70, and 80 Erlangs
- Total simulation time: 50,000 time units
- Performance metrics:
  - acceptance ratio, revenue to cost ratio, and substrate resource utilization

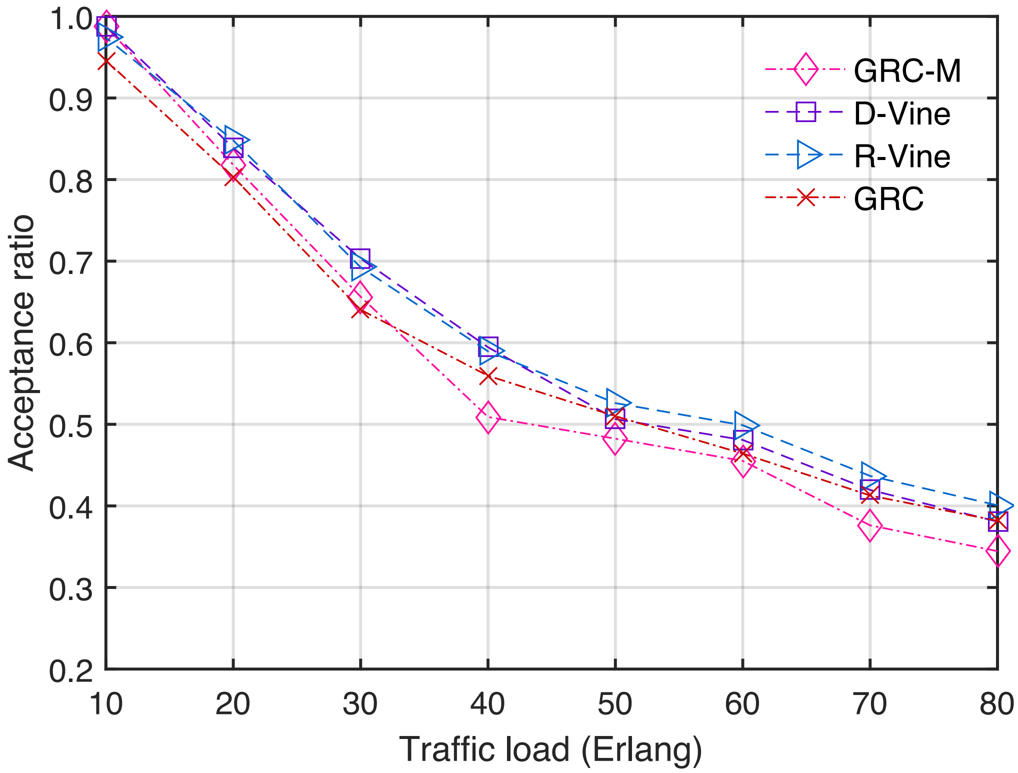
VNE-Sim

- A discrete event VNE simulator written in C++
- Based on the Discrete Event System Specification (DEVS) framework
- Employs the Adevs library

- A. M. Uhrmacher, "Dynamic structures in modeling and simulation: a reflective approach," *ACM Trans. Modeling and Computer Simulation*, vol. 11, no. 2, pp. 206–232, Apr. 2001.
- J. J. Nutaro, *Building Software for Simulation: Theory and Algorithms, with Applications in C++*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2010.
- S. Haeri and Lj. Trajković, "VNE-Sim: a virtual network embedding simulator," in *Proc. SIMUTOOLS*, Prague, Czech Republic, Aug. 2016.

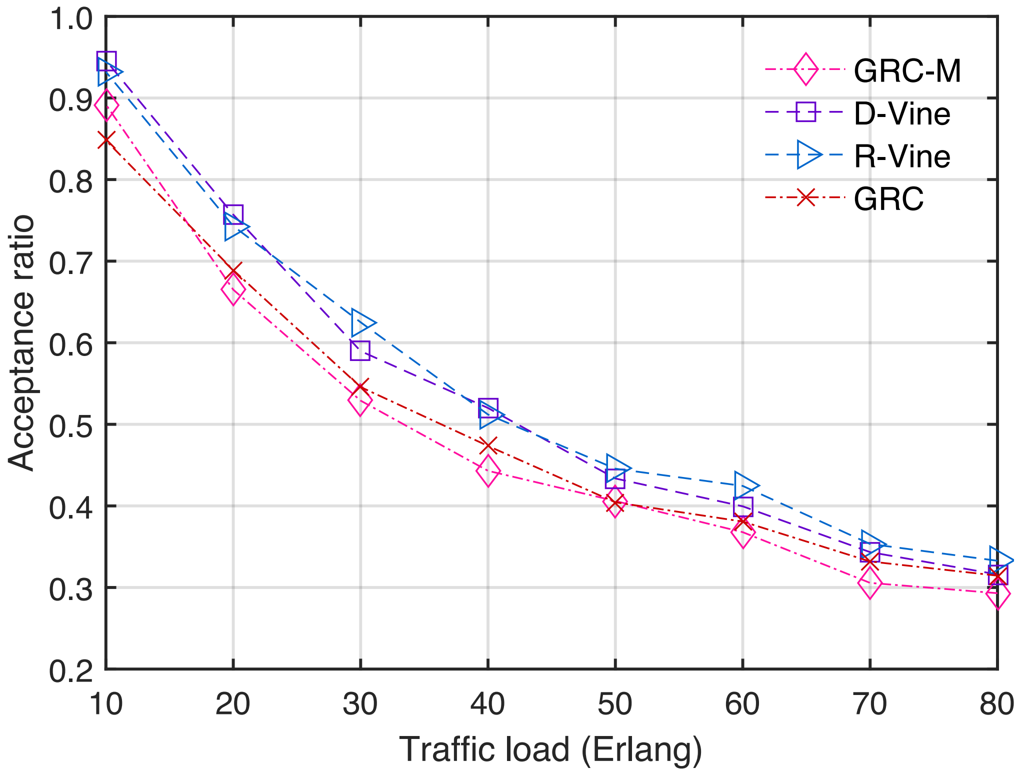
Acceptance ratio

Conventional



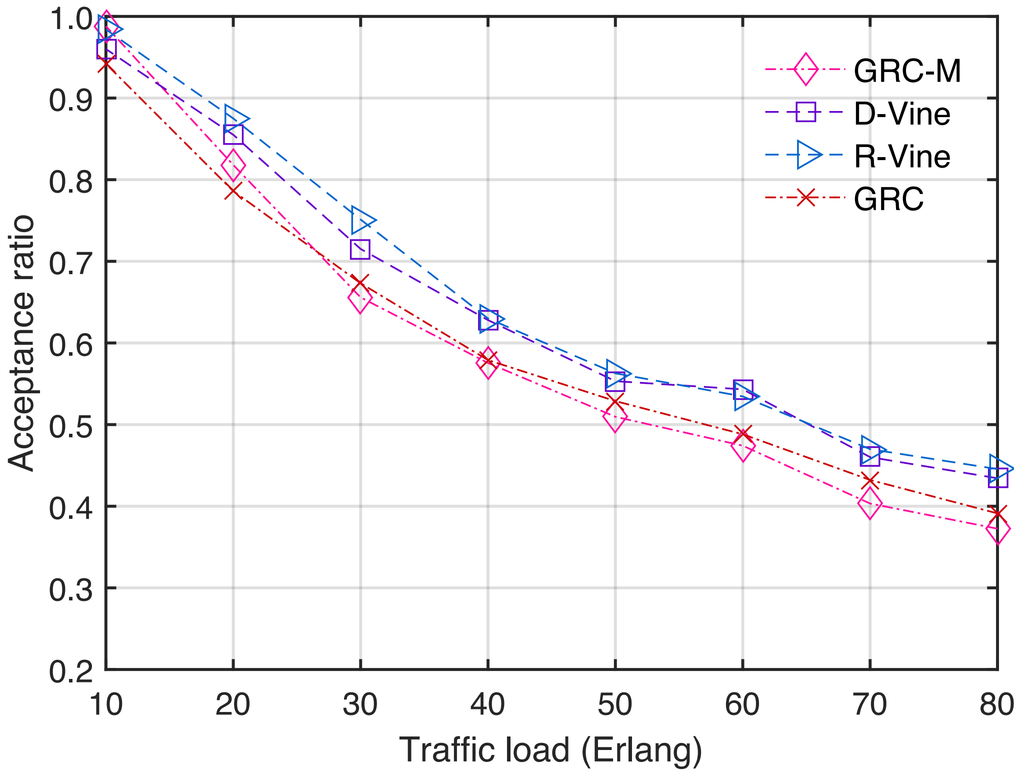
Acceptance ratio

Diamond



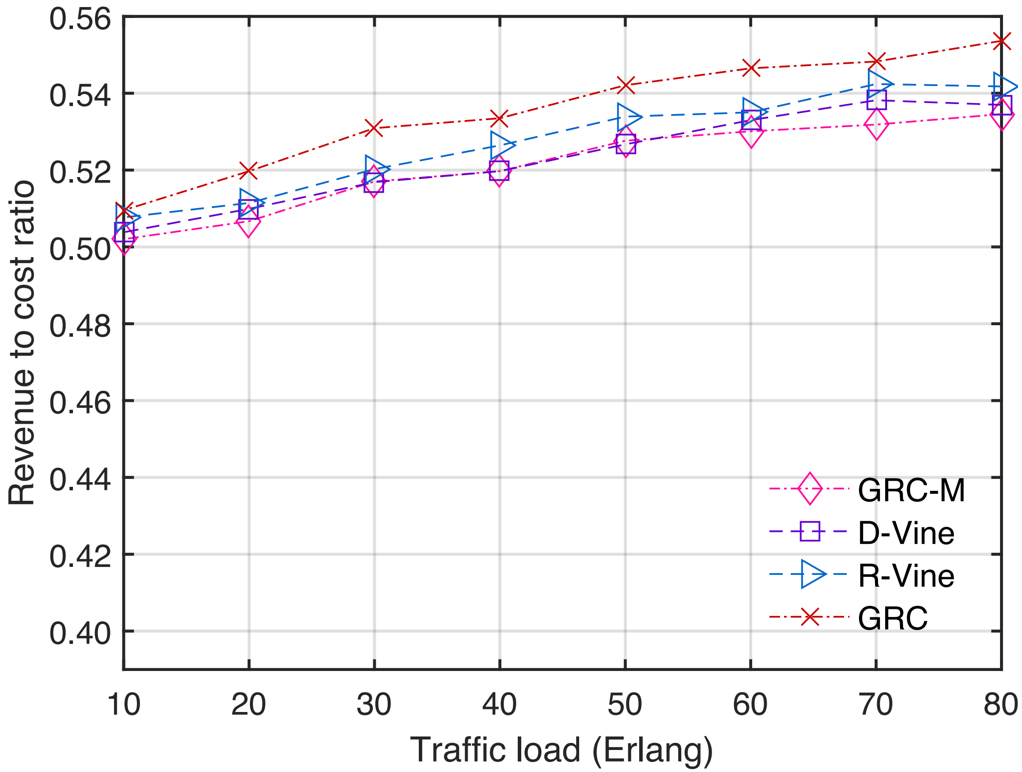
Acceptance ratio

F²Tree



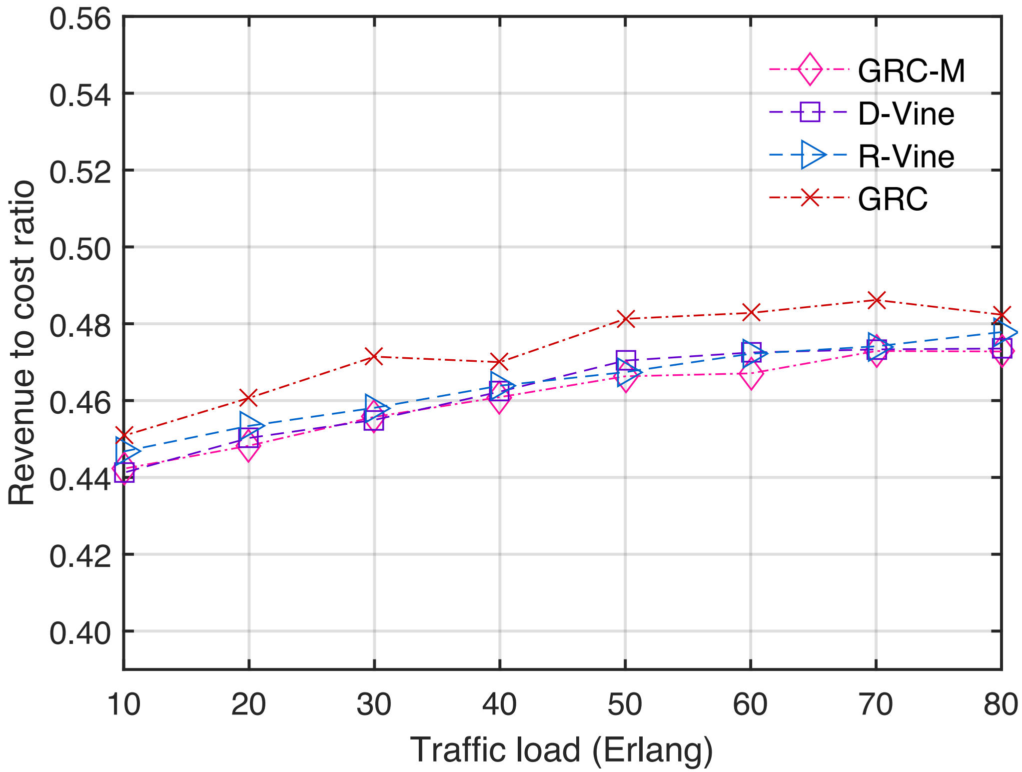
Revenue to cost ratio

Conventional



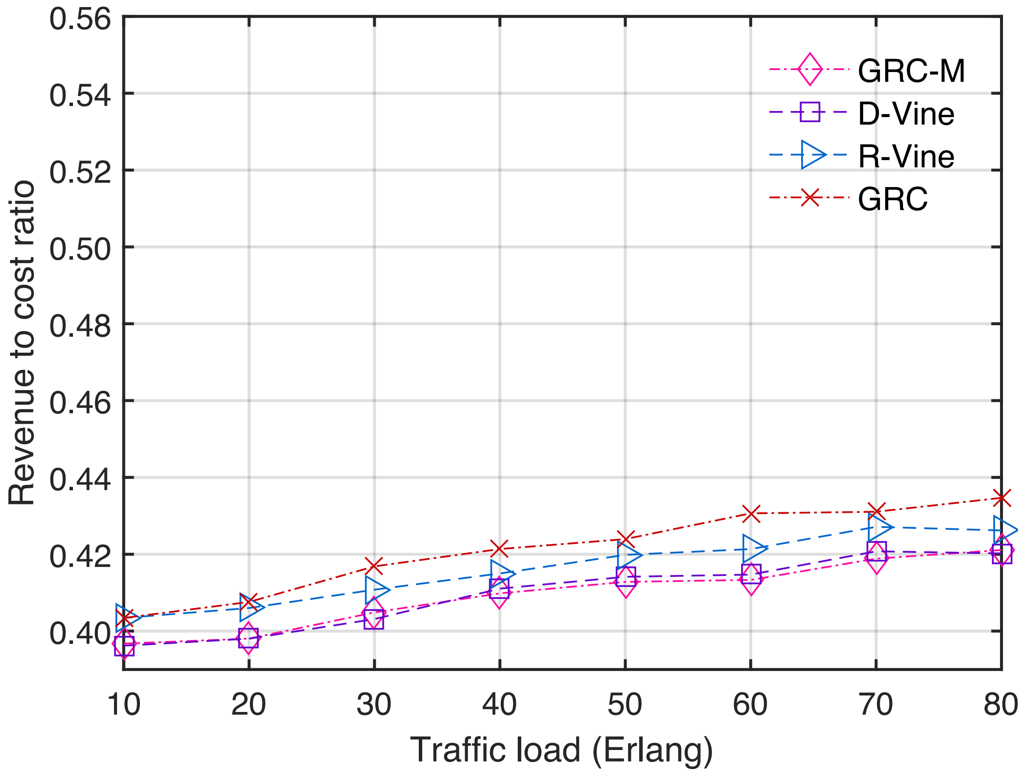
Revenue to cost ratio

Diamond



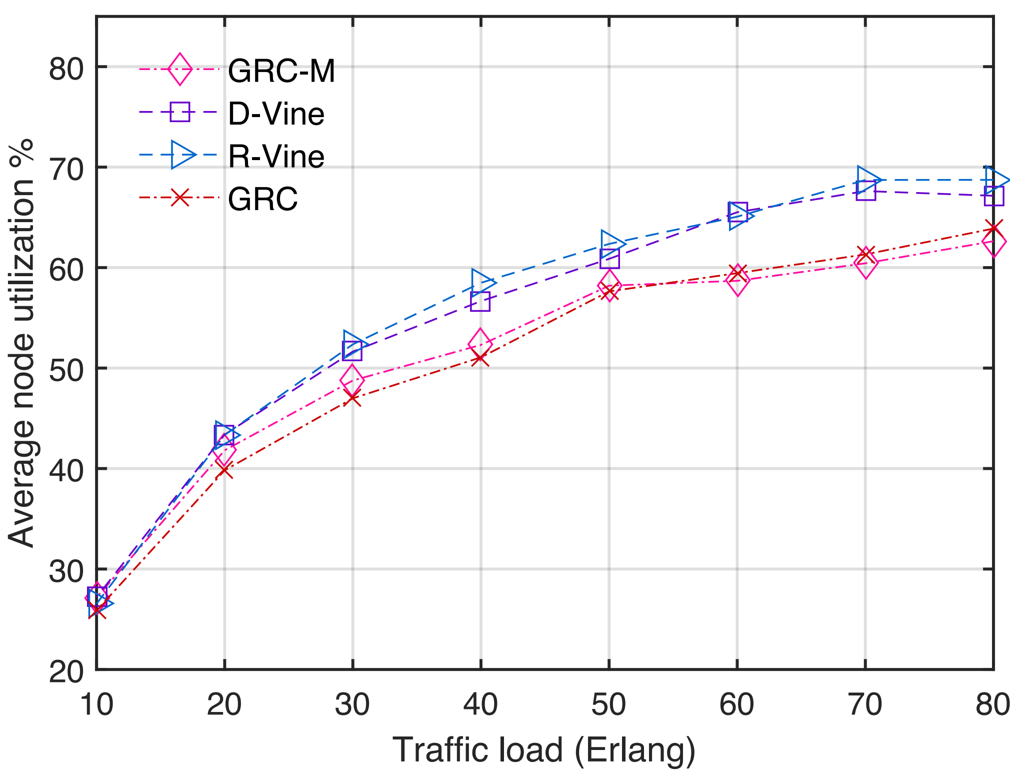
Revenue to cost ratio

F²Tree



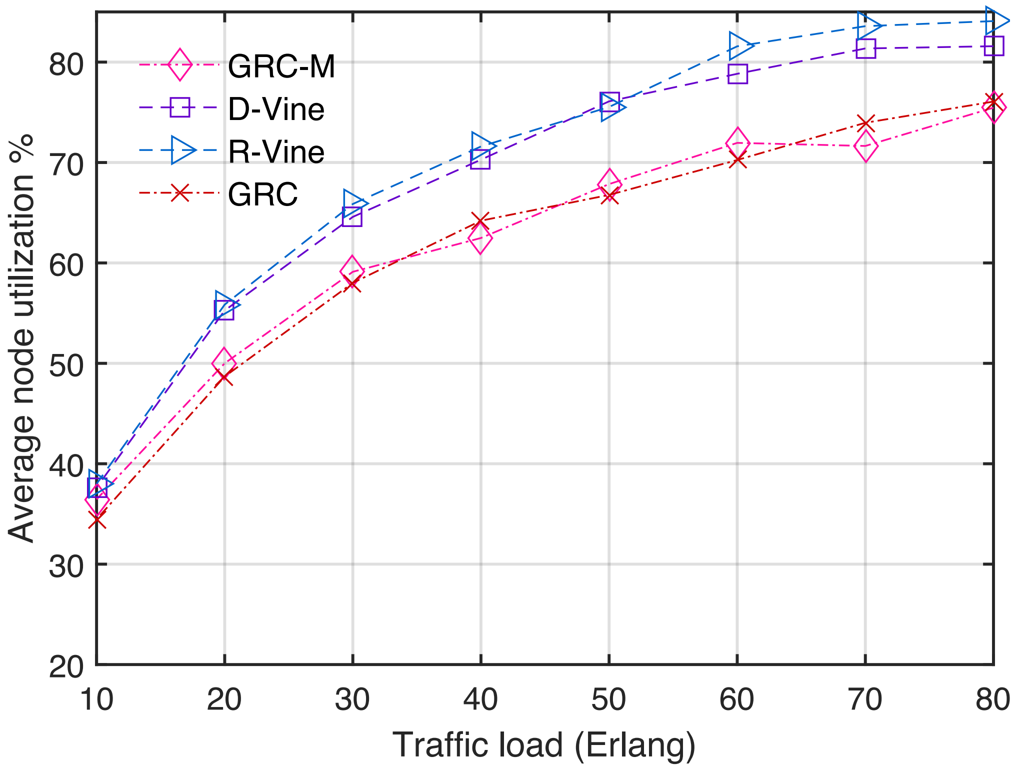
Average node utilization

Conventional



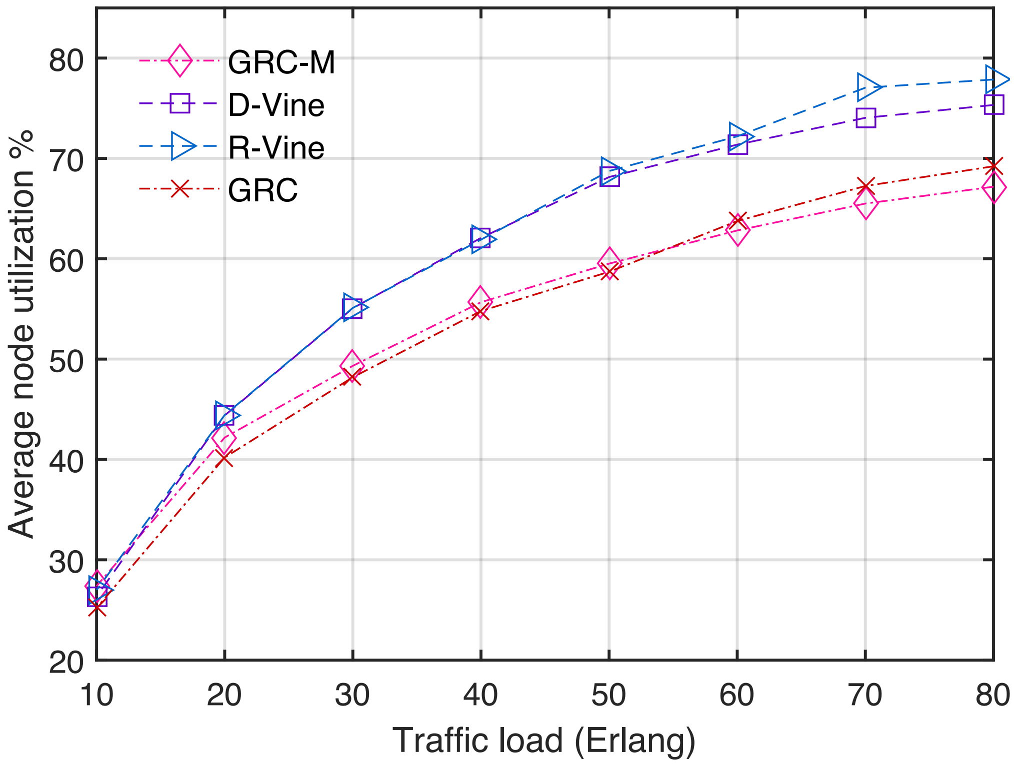
Average node utilization

Diamond



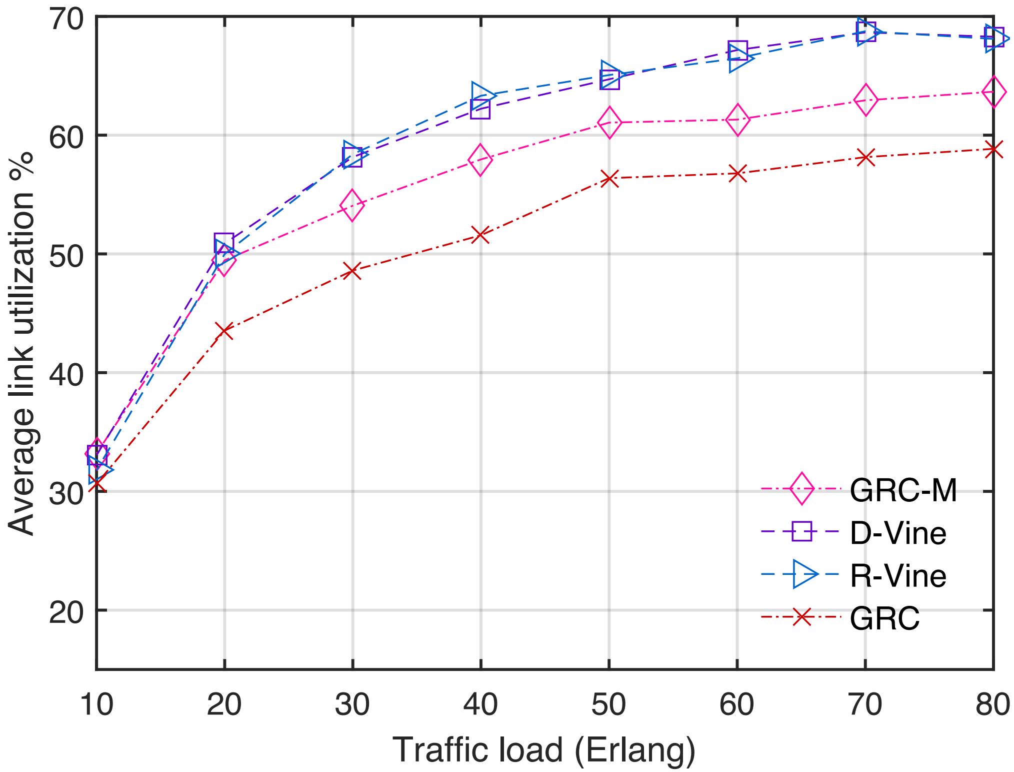
Average node utilization

F^{2}Tree



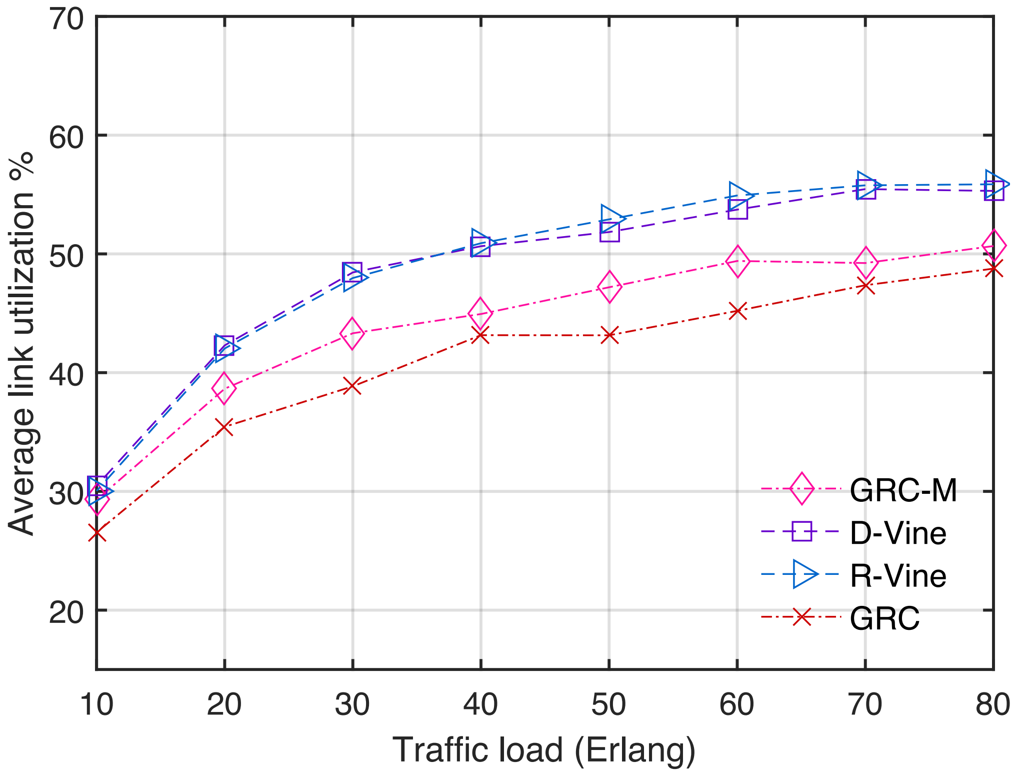
Average link utilization

Conventional



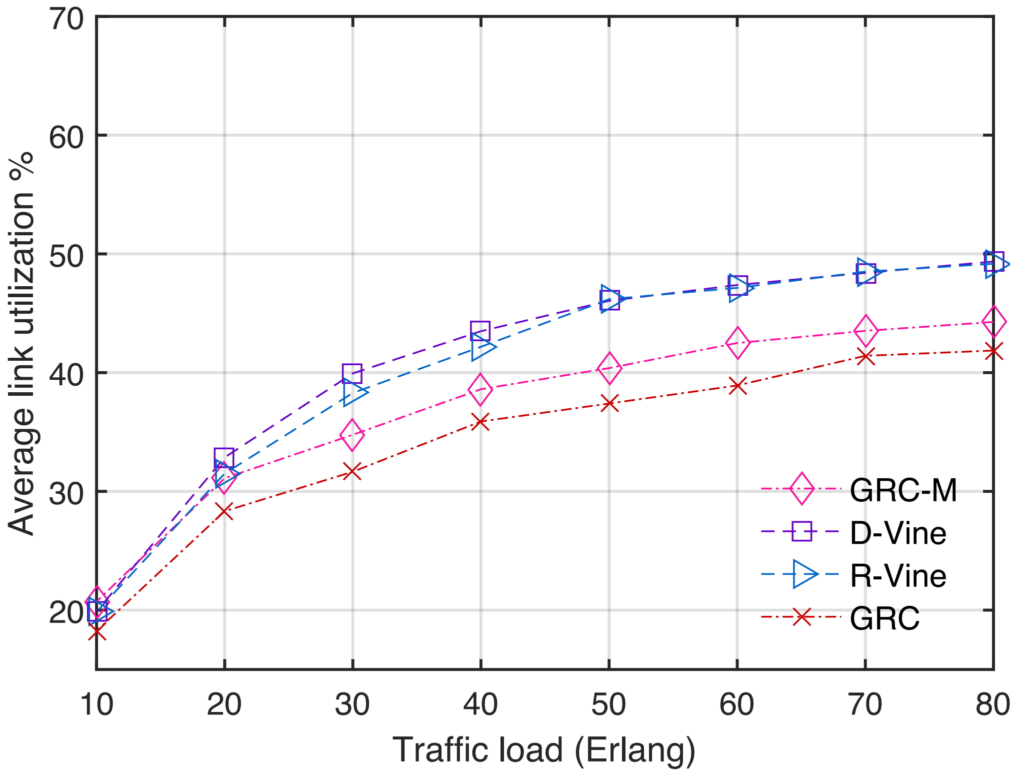
Average link utilization

Diamond



Average link utilization

F^{2}Tree



Simulation results: Conventional, F^2Tree, Diamond

- F^2Tree topology:
  - exhibits higher acceptance ratio compared to conventional two-tier and Diamond topologies
  - may accept additional VNRs since it provides multiple paths between hosts
- Conventional two-tier and F^2Tree topologies have the highest and the lowest revenue to cost ratios and link utilizations, respectively
- Conventional two-tier and Diamond topologies show the lowest and highest node utilizations, respectively
- F^2Tree topology has higher wiring density and number of nodes and, hence, the embedding cost is higher while the link and node utilizations are lower

Simulation results:
Conventional, F^2Tree, Diamond

- While Diamond topology has smaller number of nodes than F^2Tree, it still exhibits higher node utilization and has comparable performance
- The simulated data center topologies are much smaller than the deployed networks due to high memory requirements and long simulation times
- However, large data centers possess similar structures as those simulated in this study

Simulation results: Fat-Tree vs. BCube

- Fat-Tree topology offers up to:
  - 10% higher acceptance ratio
  - 20% higher node utilization
  - 10% higher link utilization
- The revenue to cost ratio of the Fat-Tree topology is slightly lower than the BCube topology
- Desirable:
  - high acceptance ratio, high substrate resource utilization, and high revenue to cost ratio

Conclusions

- Links that are connected to the hosts are important for the virtual network embeddings:
  - especially for embedding virtual nodes that require multiple connections to other nodes
- Performing traffic forwarding using only the core switches instead of the hosts may lead to higher VNR acceptance ratio

Conclusions

Simulated Fat-Tree topology:

- Has higher switch to host ratio (0.84) compared to the BCube topology (0.75)
- Additional paths between the hosts enable:
  - higher acceptance ratio
  - higher resource utilization
- Tradeoff:
  - lower revenue to cost ratio

References

Data Center Networks:

- T. Chen, X. Gao, and G. Chen, “The features, hardware, and architectures of data center networks: a survey,” *J. Parallel Distrib. Comput.*, vol. 96, pp. 45–774, Oct. 2016.
- G. Chen, Y. Zhao, D. Pei, and D. Li, “Rewiring 2 links is enough: Accelerating failure recovery in production data center networks,” in *Proc. IEEE ICDCS*, Columbus, Ohio, USA, June 2015, pp. 569–578.
- Y. Sun, J. Chen, Q. Lu, and W. Fang, “Diamond: an improved fat-tree architecture for large-scale data centers,” *J. Commun.*, vol. 9, no. 1, pp. 91–98, Jan. 2014.
- H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron, “Towards predictable datacenter networks,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 242–253, Oct. 2011.

References

- C. Guo, G. Lu, H. J. Wang, S. Yang, C. Kong, P. Sun, W. Wu, and Y. Zhang, "SecondNet: a data center network virtualization architecture with bandwidth guarantees," in *Proc. ACM CoNEXT 2010*, Philadelphia, PA, USA, Dec. 2010, p. 15.
- 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.
- M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 63–74, Oct. 2008.
- C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, "DCell: a scalable and fault-tolerant network structure for data centers," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 75–86, Oct. 2008.
- C. E. Leiserson, "Fat-Trees: universal networks for hardware-efficient supercomputing," *IEEE Trans. Comput.*, vol. 30, no. 10, pp. 892–901, Oct. 1985.

References

Network Virtualization:

- M. Chowdhury and R. Boutaba, “Network virtualization: state of the art and research challenges,” *IEEE Commun. Mag.*, vol. 47, no. 7, pp. 20–26, July 2009.
- 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.

References

Virtual Network Embedding Algorithms:

- 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.
- 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.
- S. Zhang, Y. Qian, J. Wu, and S. Lu, "An opportunistic resource sharing and topology-aware mapping framework for virtual networks," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, Mar. 2012, pp. 2408–2416.
- 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.
- M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: substrate support for path splitting and migration," *SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 19–29, Mar. 2008.

References

Virtual Network Embedding Algorithms:

- 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 and Systems*, Montreal, Canada, May 2016, pp. 666–669.
- S. Haeri and Lj. Trajković, "Virtual network embeddings in data center networks," in *Proc. IEEE Int. Symp. Circuits and Systems*, Montreal, Canada, May 2016, pp. 874–877.
- S. Haeri and Lj. Trajković, "VNE-Sim: a virtual network embedding simulator," in *Proc. SIMUTOOLS*, Prague, Czech Republic, Aug. 2016.