Virtual Network Embedding on Interconnection Networks

Rob Ballantyne, Soroush Haeri, and Ljiljana Trajkovic

ballanty@sfu.ca, soroosh.haeri@gmail.com, ljilja@sfu.ca

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

Overview

- Introduction
- Virtual network embedding
- Research question
- Data center and interconnection network topologies
- Simulations
- Conclusions
- References

Introduction: Virtualization

- Virtualization enables:
  - efficient use of resources
  - configuration and management of virtual resources separated from physical resources
- We consider combined virtualization of:
  - computing systems (servers)
  - network infrastructure (routers, switches, links)

Introduction: Applications

- To advance the Internet, early approaches considered virtualization of networks to experiment with network configurations and protocols:
  - operators were reluctant to experiment with deployed networks
  - virtualization enabled such experiments
- Cloud computing infrastructure providers offer a combined virtualization service through Infrastructure as a Service (IaaS)
  - examples: AWS, Azure, and Google Cloud Platform

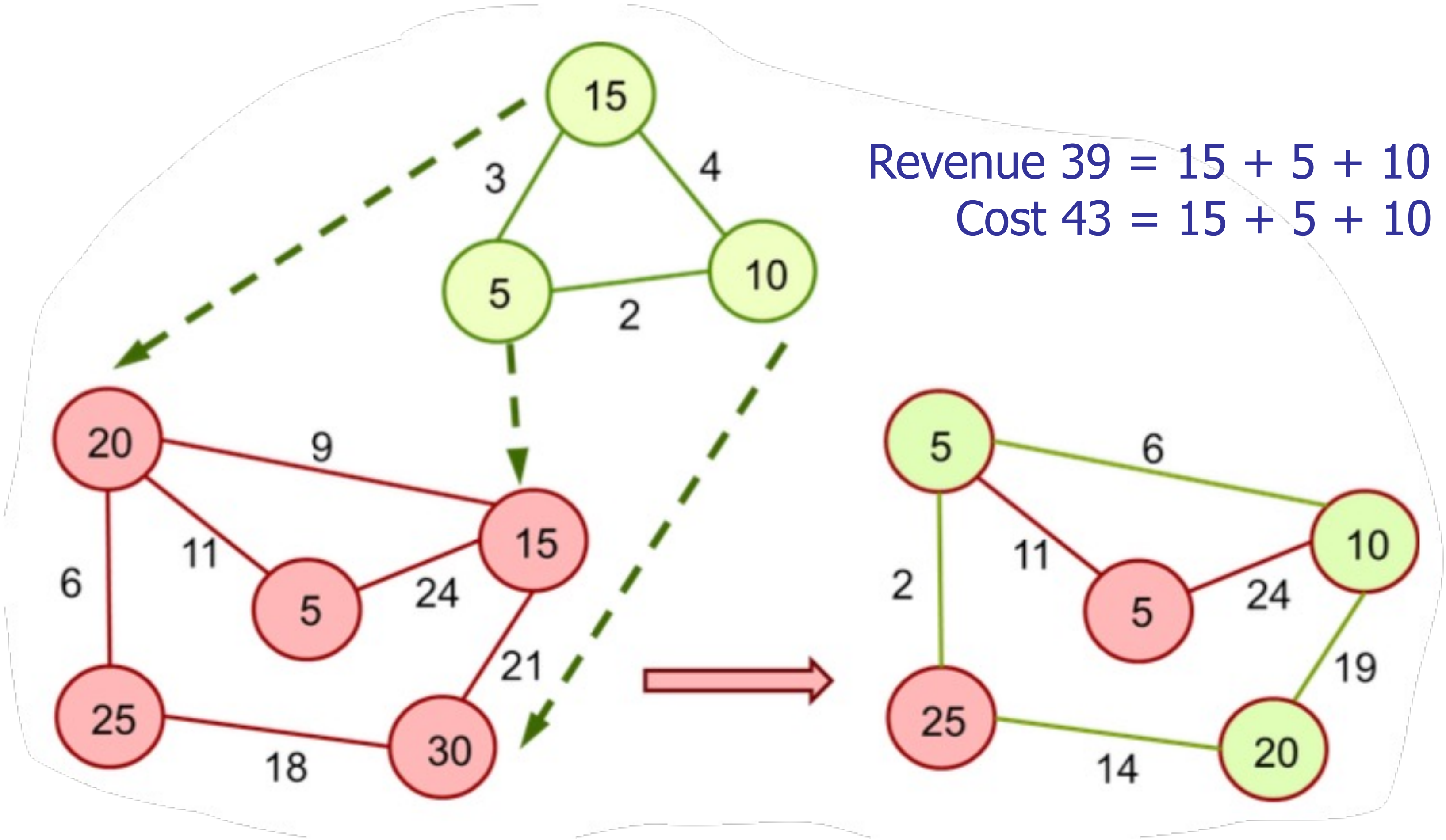
Virtual Network Embedding: Terminology

- Virtual nodes: virtual servers, network switches, and network routers
- Virtual network: virtual nodes and virtual links
- Substrate nodes: physical infrastructure servers, switches, and routers
- Substrate links: physical links
- Substrate network: substrate nodes and substrate links

Virtual Network Embedding

- Virtual Network Embedding (VNE) problem involves:
  - assigning virtual nodes to substrate nodes
  - assigning virtual links to substrate links
  - respecting resource constraints

VNE: Example



Revenue 39 = 15 + 5 + 10 + 3 + 2 + 4

Cost 43 = 15 + 5 + 10 + 3 + 2 + 4 + 4

Virtual Network Embedding: Algorithms

- VNE problem has been shown to be NP-hard
- Variety of heuristic and approximation algorithms have been proposed
- Proposed algorithms either:
  - solve virtual node mapping only and then find the virtual link mapping
  - coordinate both mappings

Research Question

- Prior work considered the substrate network to be predetermined:
  - initial interest in virtual networks was to map experimental networks into the Internet
  - later interest considered mapping virtual networks into data center substrate networks
  - this led to an emphasis on various algorithms and their performance
- We approach the problem from a design point of view and ask:
  - Are there network topologies that would offer improved embedding performance if used as a substrate network?

Our Approach

- Compare interconnection network and data center network topologies
- Evaluate whether interconnection topologies offer improved performance of VNE solutions
- Consider:
  - Interconnection substrate network topologies:
    - Hypercube and Butterfly
  - Data center substrate network topologies:
    - Two-Tier and BCube
  - Assign various loads of virtual network requests (VNRs)

Research Question

- Metrics of the simulated VNE algorithm performance:
  - acceptance ratio
  - revenue to cost ratio
  - node utilization
  - link utilization

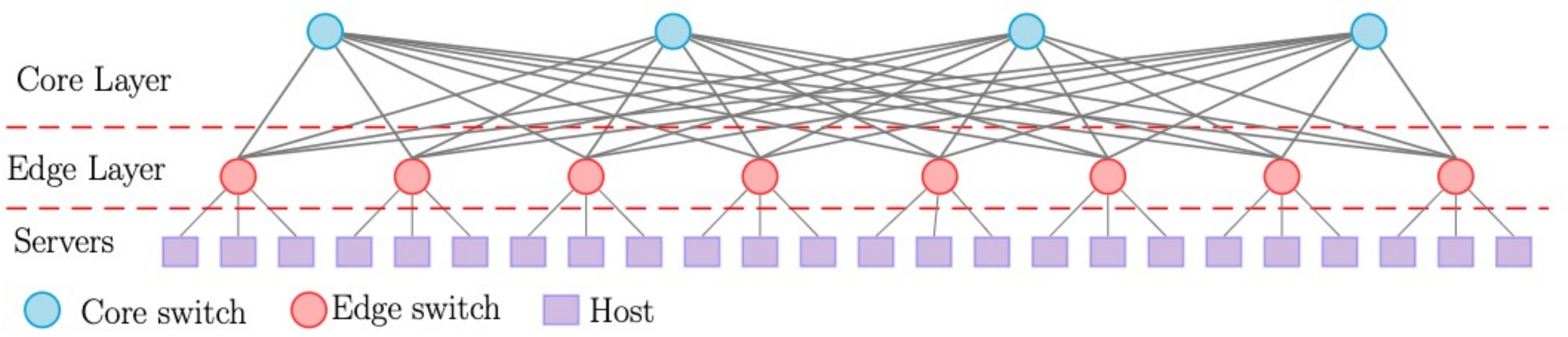
Topologies: Design Focus

Design focus:

- Data center networks:
  - high bandwidth, low latency, redundancy, reliability
- Interconnection networks:
  - very high bandwidth, low blocking probability, low latency
  - interconnection networks are used between tightly coupled systems:
    - cpu cores, cpu chips, physically close boards and servers

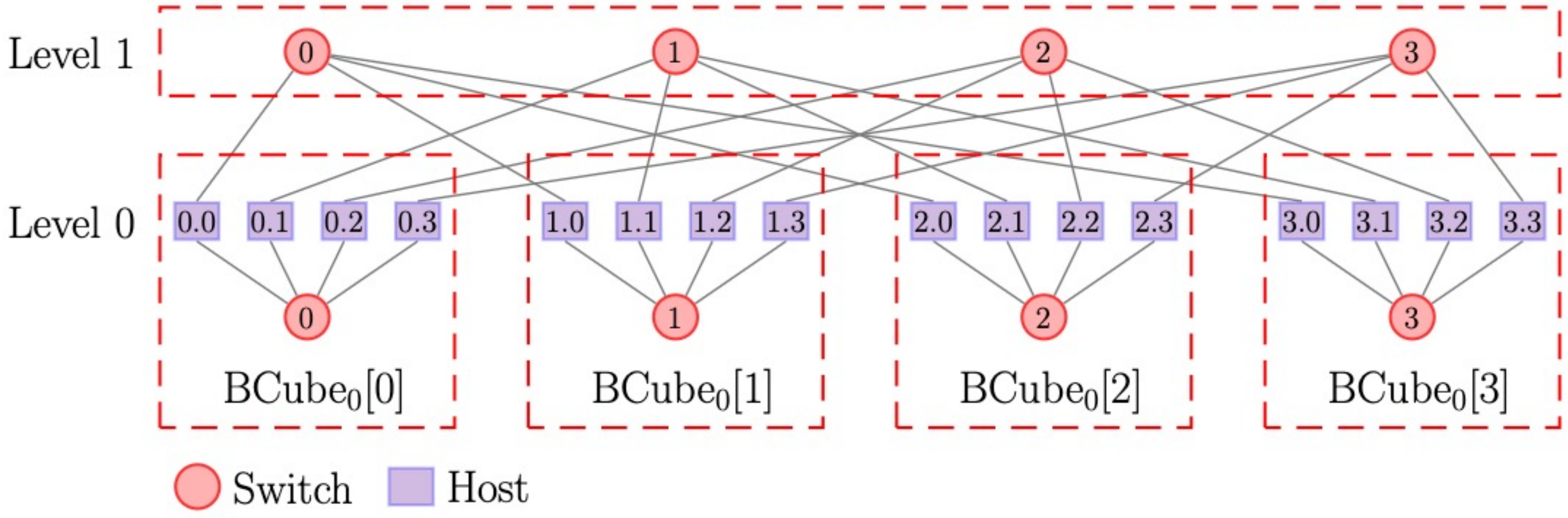
Two-Tier Data Center Topology

- Two-Tier Spine-Leaf switch-centric topology: a layer of core network switches and a layer of edge switches



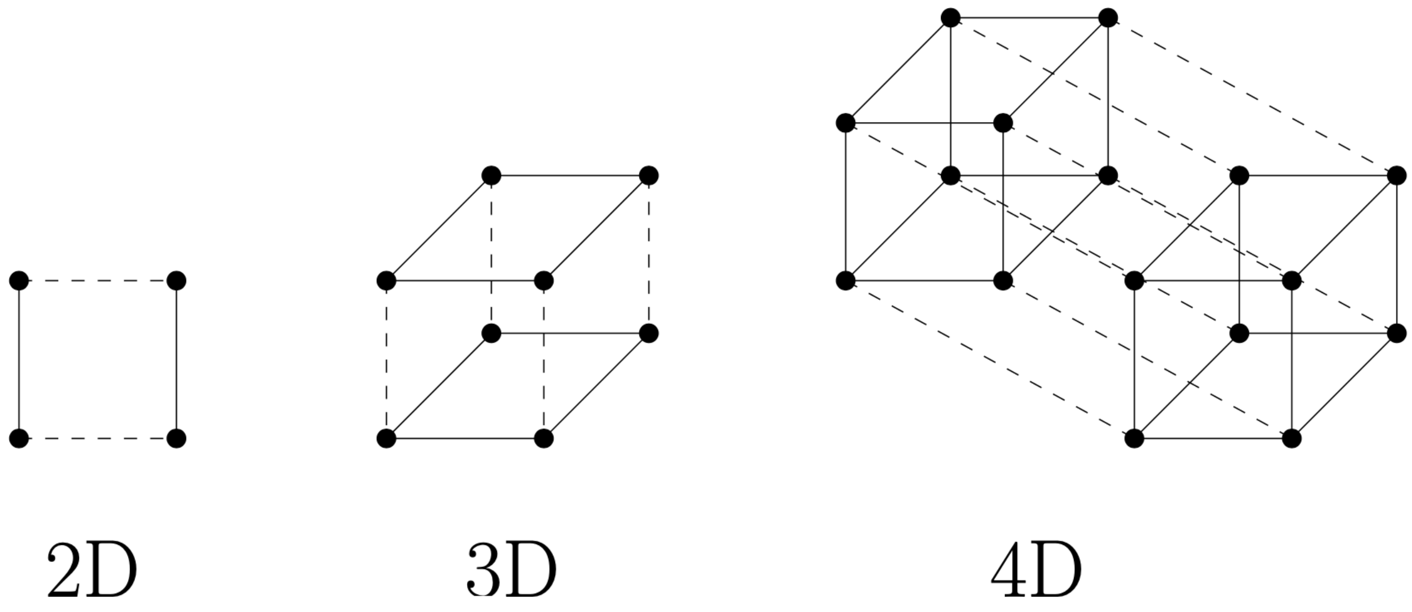
BCube Data Center Topology

- BCube network topology: a recursive design meant to easily scale and maintain topological properties
- BCube(1,4) network:



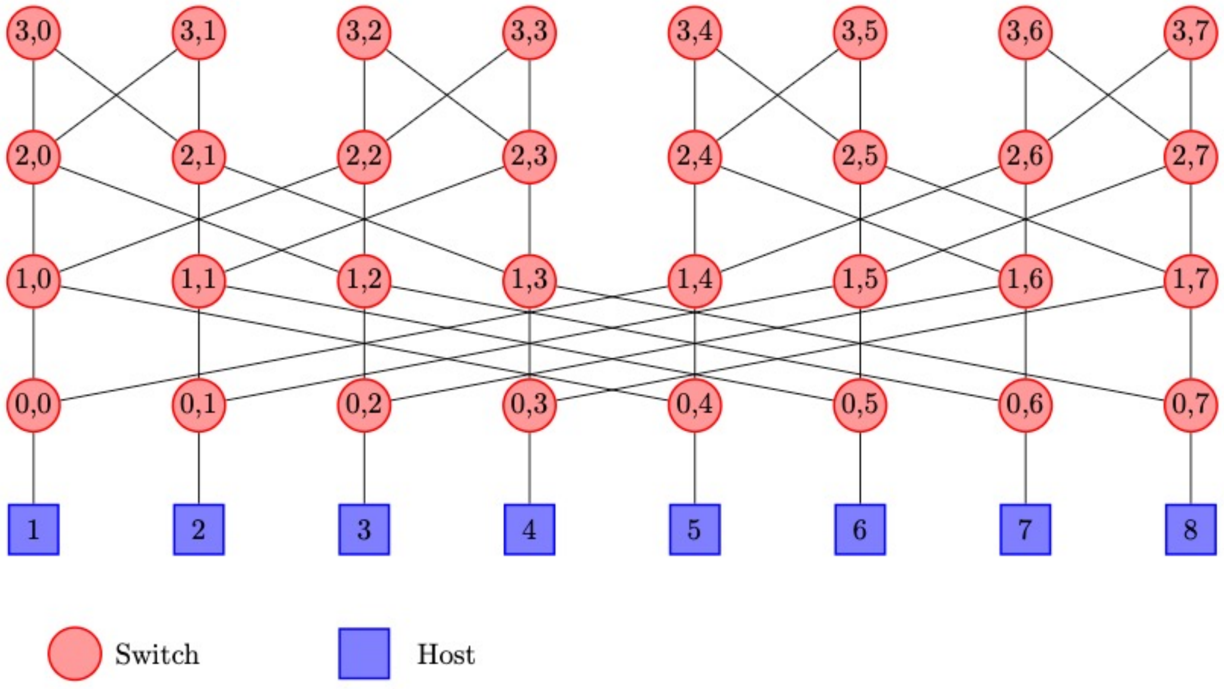
Hypercube Topology

- Hypercube networks of size = 2, 3, and 4:



Butterfly Topology

- Butterfly network with 8 hosts:



Network Topologies: Properties

Network topology	No. hosts	No. switches	No. links	Bisection bandwidth	Diameter
Two-Tier	$n_e n_s$	$n_e + n_c$	$n_e n_c + n_s$	$n_e n_c / 2$	4
BCube(k, n)	n^{k+1}	$(k+1)n^k$	$(k+1)n^{k+1}$	$\frac{n^{k+1}}{4(n-1)}$	$k+1$
Hypercube	2^k	0	$k2^{k-1}$	k	k
Butterfly	2^k	$(k+1)2^k$	$(k+1)2^{k+1}$	2^{k-1}	$k+1$

Simulations

- VNE-Sim: a discrete event simulator developed to implement and evaluate VNE algorithms
- Extended VNE-Sim platform to implement Hypercube and Butterfly topologies
- VNE-Sim generates:
  - substrate networks
  - time series of virtual network embedding requests (VNRs)
- VNE-Sim enables simulation of various VNE algorithms

Simulations: VNE Embedding Algorithms

Algorithm	Name	Description
Global Resource Capacity (GRC)	GRC	Heuristic + shortest path algorithm to solve link embedding
Global Resource Capacity	GRC-M	Heuristic + MCF to solve link embedding
Monte Carlo Tree Search (MCTS)	MaVEn-S	MCTS + shortest path algorithm to solve link embedding
Monte Carlo Tree Search	MaVEn-M	MCTS + MCF to solve link embedding
Mixed-Integer LP relaxation problem	D-ViNE	MILP + deterministic rounding to approximate solution
Mixed-Integer LP relaxation problem	R-ViNE	MILP + randomized rounding to approximate solution

Simulations: Parameters of Substrate Network

- Simulated networks have 64 servers to maintain predefined CPU resources
- Various topologies then determined the number of switches and links

Network Topology	Hosts	Switches	Links	Bisection Bandwidth	Diameter
Two-Tier	64	16	128	32	4
BCube(2,4)	64	48	192	5.33	7
Hypercube	64	0	192	6	6
Butterfly	64	448	896	32	7

Simulations: Virtual Network Requests

- Multiple series of random VNRs were generated as various request loads
- Request loads:
  10, 20, 30, 40, 50, 60, 70, and 80 Erlangs
- Loads were generated once and reused to simulate each topology and algorithm

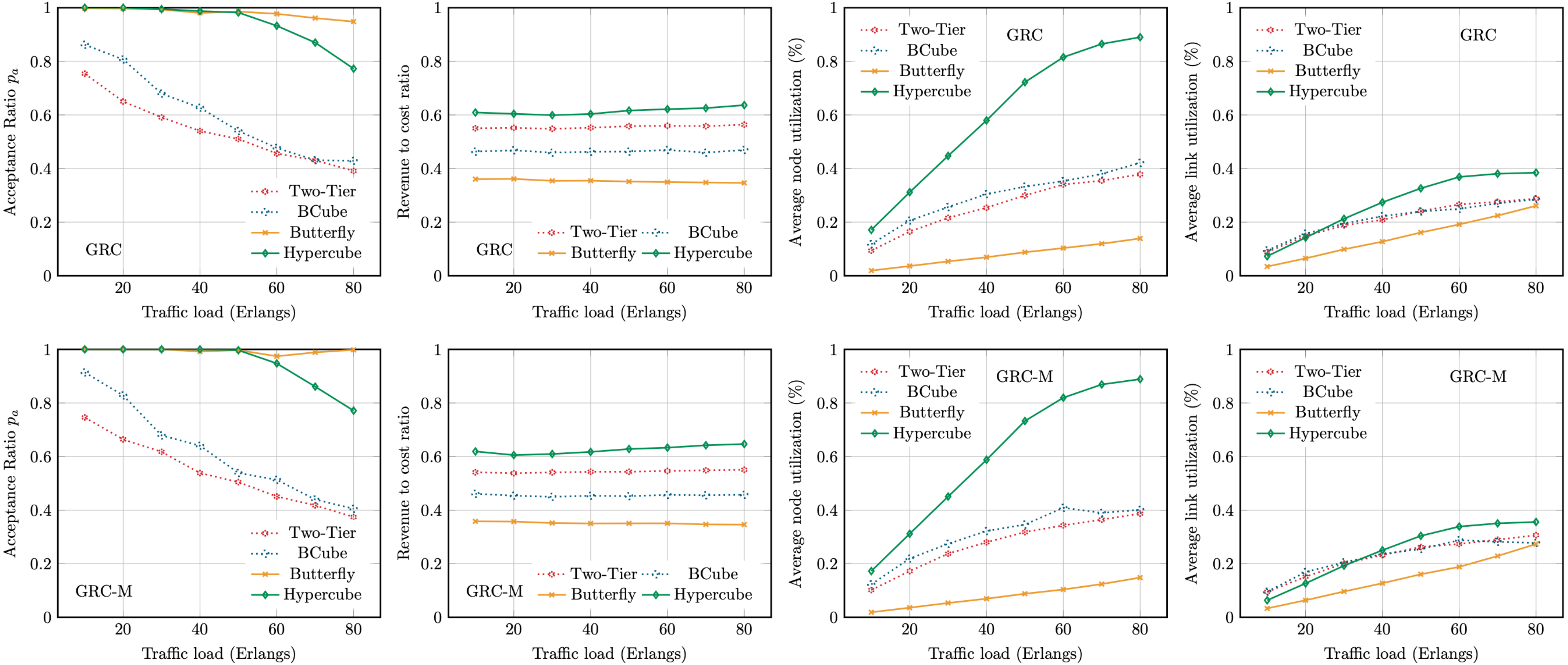
Results: Performance Metics

- Simulation results were aggregated to report:
  - Acceptance ratio:
    number of embedded VNRs / total number of VNRs
  - Revenue to cost ratio
  - Average node utilization
  - Average link utilization
  - Execution times of the embedding algorithms

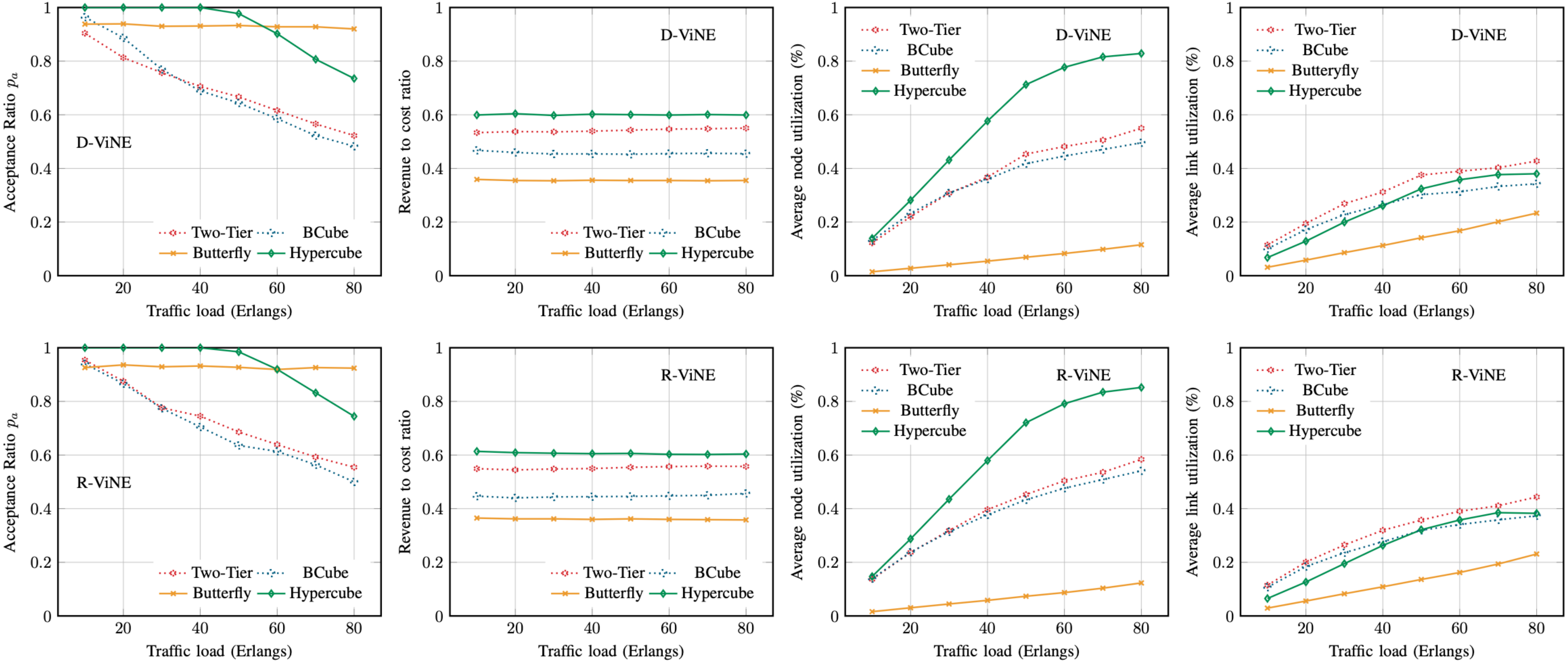
Results: Revenue to Cost Ratio

- Simple measures:
  - Revenue: sum of the resources in the VNR
  - Cost: sum of the resources consumed in the embedding
  - Revenue to cost ratio:
    - in the range [0, 1]: a single link may be mapped into multiple links

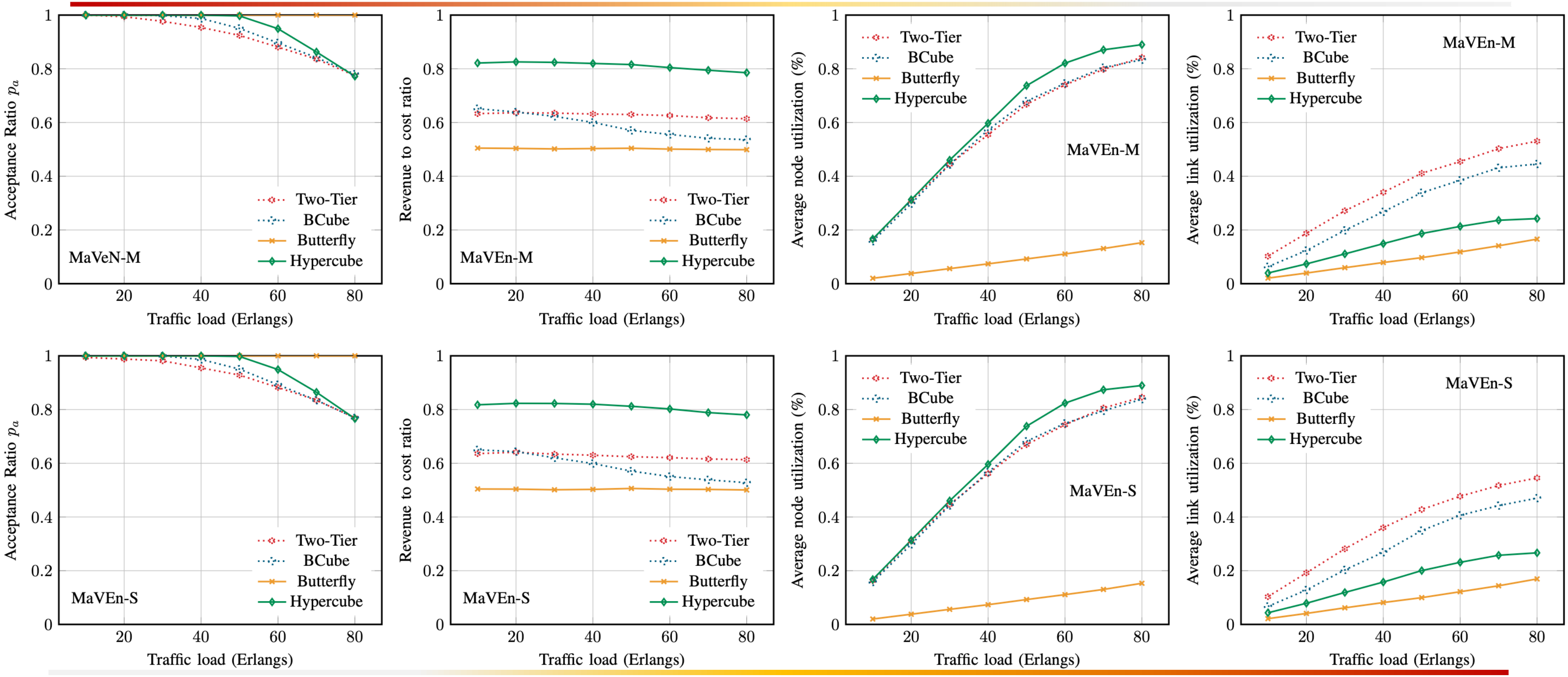
Performance Results: GRC, GRC-M



Performance Results: D-ViNE, R-ViNE

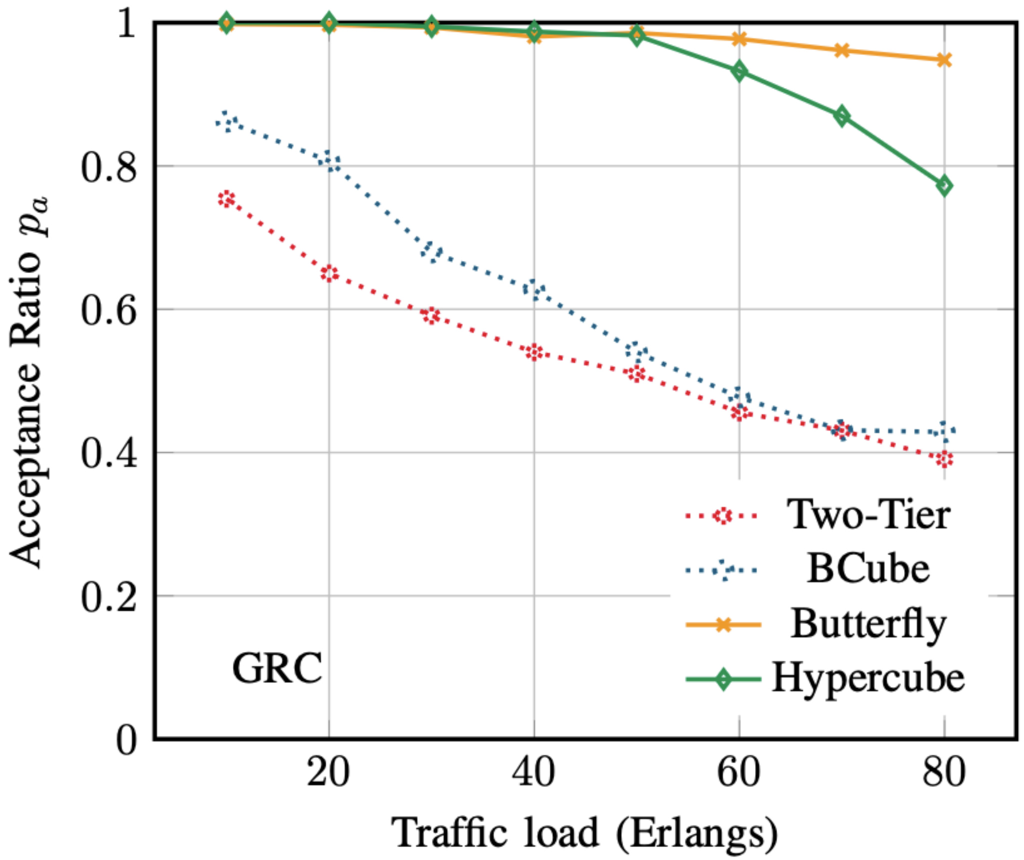


Performance Results: MaVNe-M, MaVNe-S

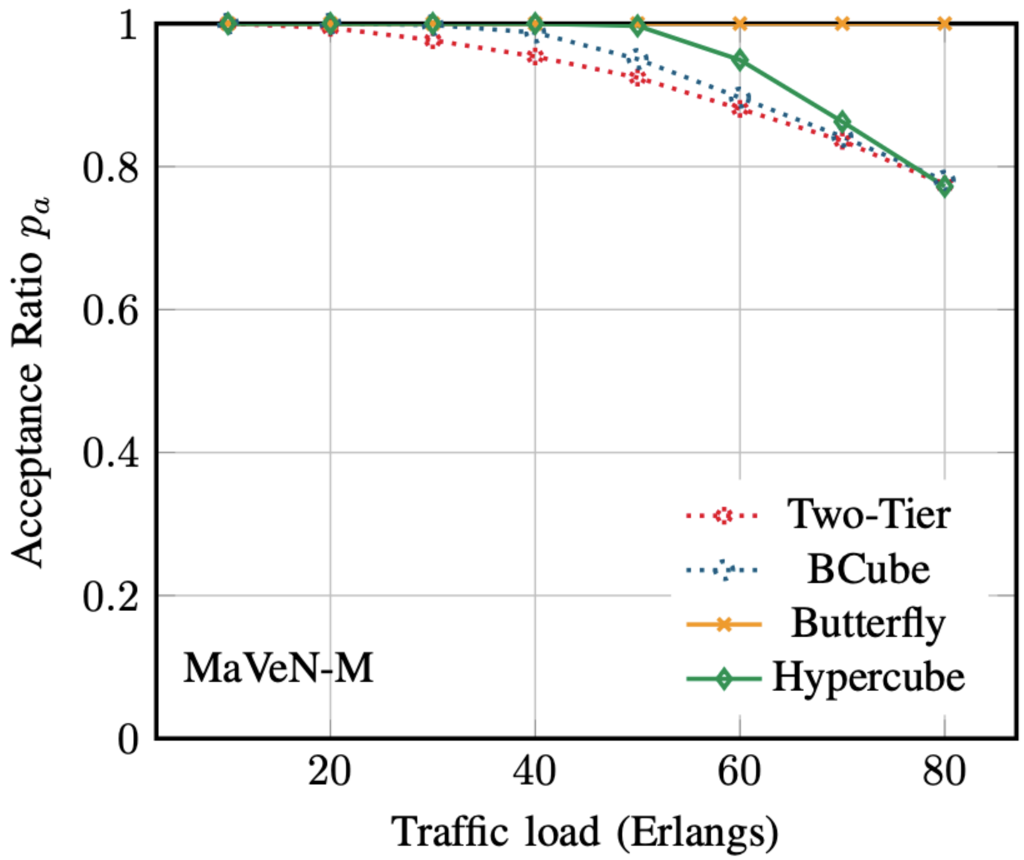


Performance Results: Acceptance Ratio

GRC

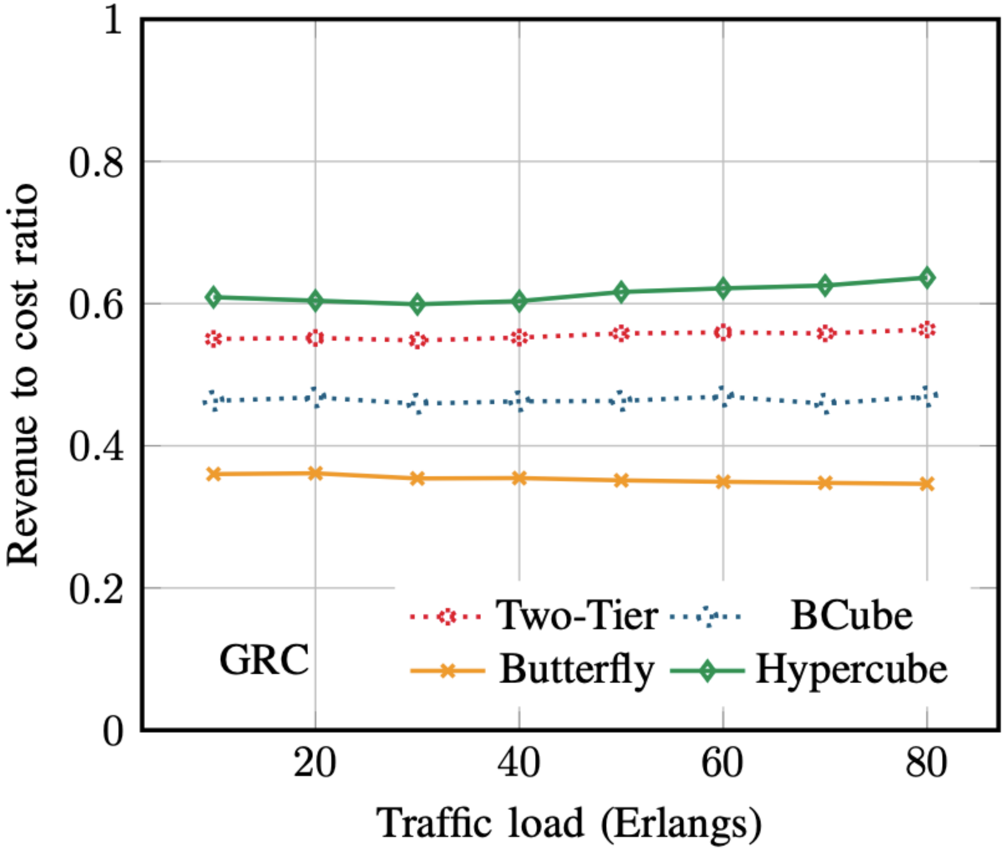


MaVEn-M

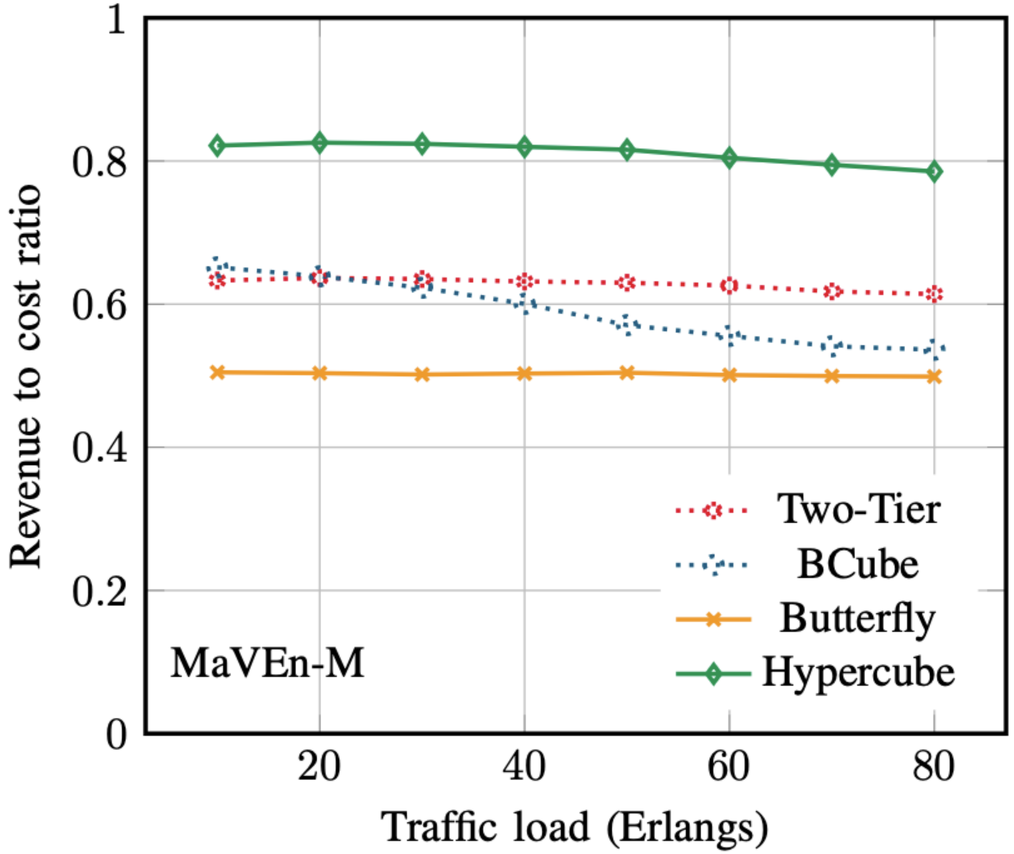


Performance Results: Revenue to Cost Ratio

GRC

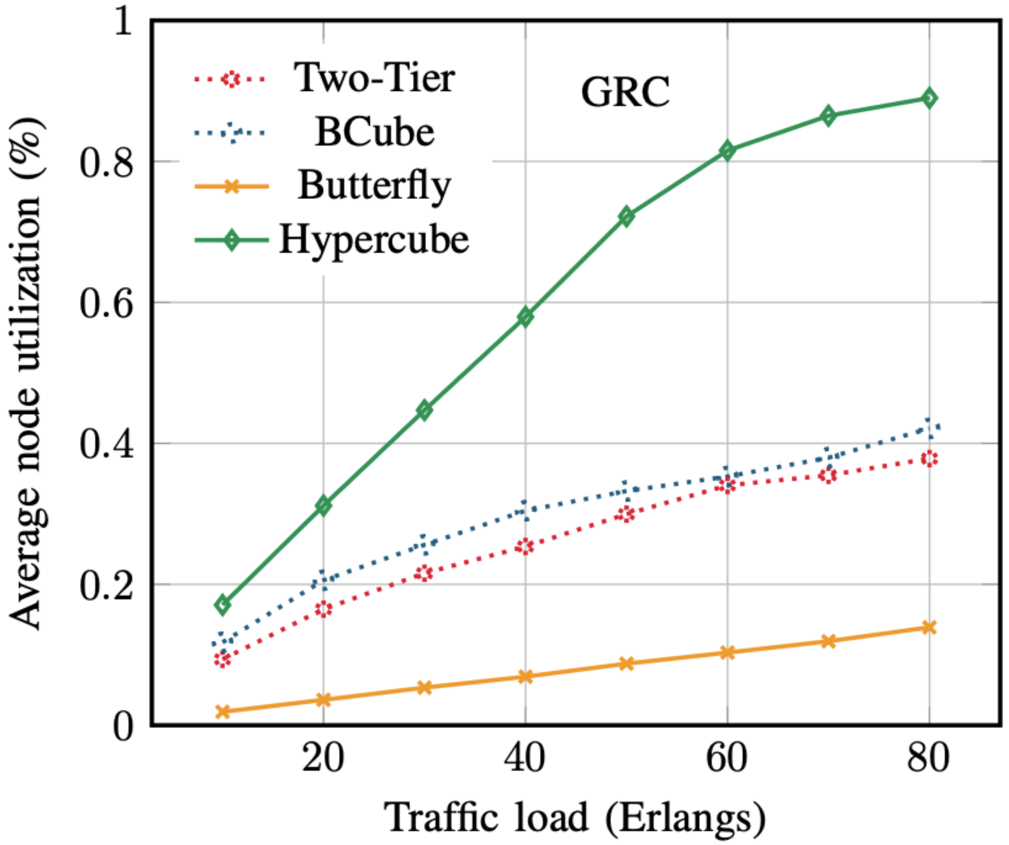


MaVEn-M

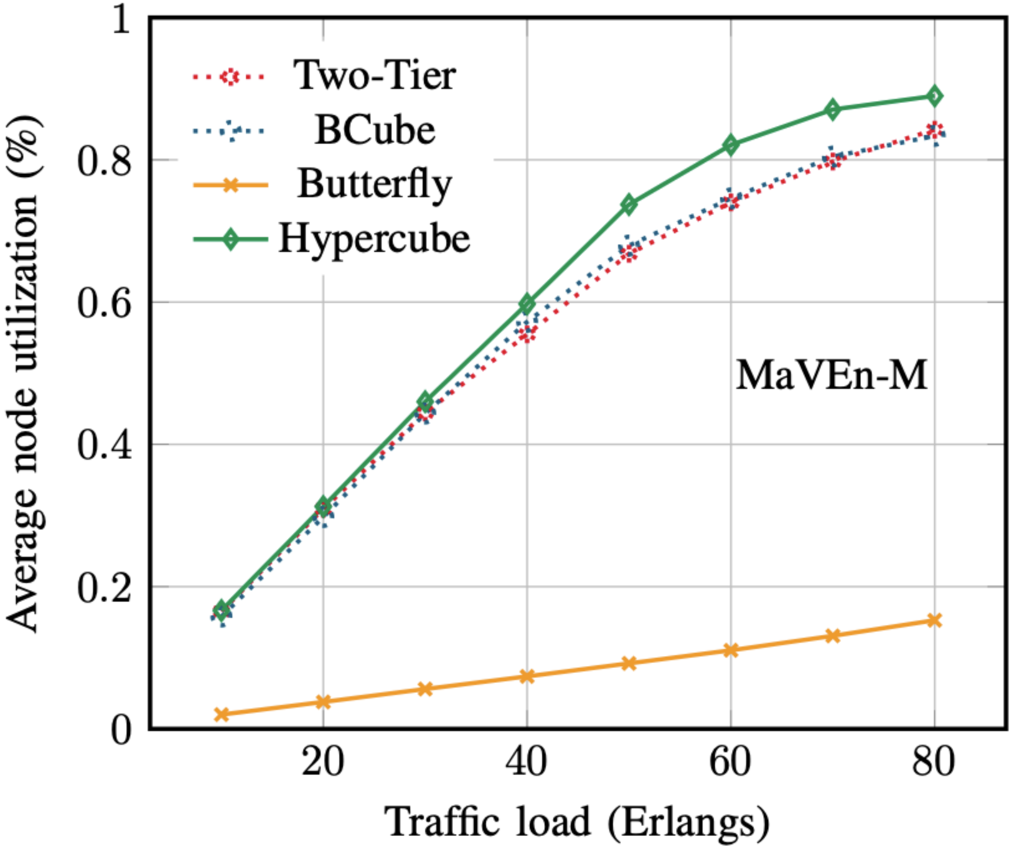


Performance Results: Node Utilization

GRC

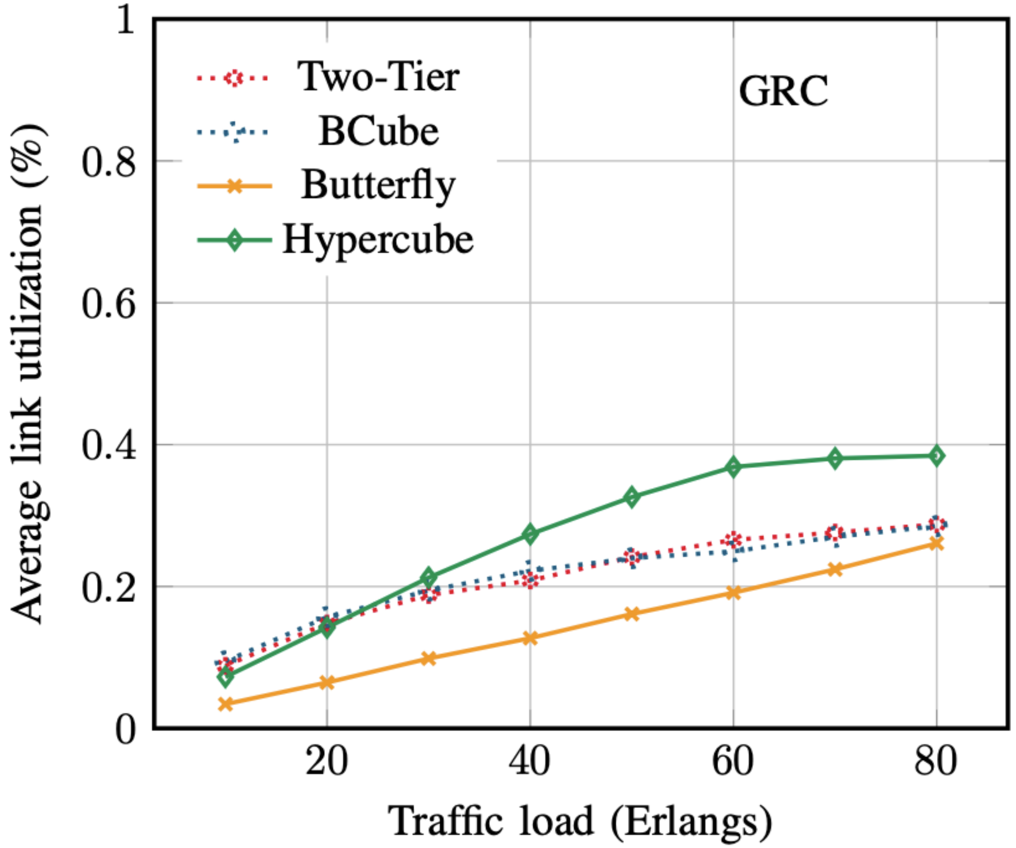


MaVEn-M

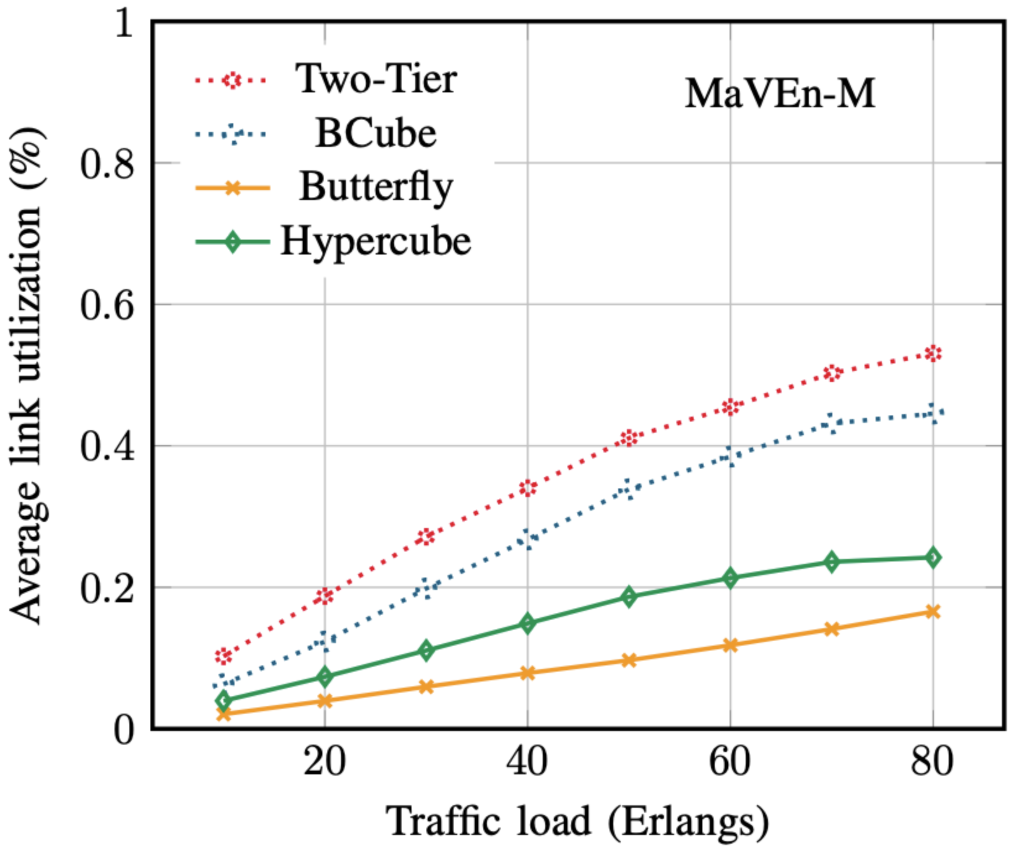


Performance Results: Link Utilization

GRC



MaVEn-M



Results: Average VNR Processing Times

VNE Algorithm	Two-Tier		BCube(2,4)		Hypercube		Butterfly	
	Min (s)	Max (s)	Min (s)	Max (s)	Min (s)	Max (s)	Min (s)	Max (s)
GRC	0.019	0.021	0.046	0.049	0.015	0.018	0.801	0.882
GRC-M	0.377	0.430	0.765	0.959	0.300	0.349	23.388	28.510
MaVEn-M	0.683	1.211	1.316	1.719	0.788	0.922	8.489	10.915
MaVEn-S	1.060	1.542	1.948	2.945	0.921	1.405	34.057	41.713
D-ViNE	1.187	1.615	2.397	3.247	1.008	1.914	35.298	42.698
R-ViNE	1.700	1.770	2.290	3.518	0.962	1.766	36.648	50.526

Conclusions

- Interconnection topologies offered higher acceptance ratios
- Disadvantage of data center networks may be mitigated by employing advanced algorithms (MaVEn-M and MaVEn-S) that require additional processing time
- Butterfly topology offered the best acceptance ratio but had low:
  - revenue to cost ratio, node and link utilizations
  - due to the high number of network elements required to implement the network topology

References

- 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.
- A. Fischer, J. F. Botero, M. T. Beck, H. de Meer, and X. Hesselbach, "Virtual network embedding: a survey," *IEEE Commun. Surveys Tuts*., vol. 15, no. 4, pp. 1888–1906, 2013.
- S. Haeri and Lj. Trajkovic, "VNE-Sim: a virtual network embedding simulator," in *Proc. 9th EAI Int. Conf. Simulation Tools Techniques*, Prague, Czech Republic, Aug. 2016, pp. 112–117.
- S. Haeri and Lj. Trajkovic, "Virtual network embedding via Monte Carlo tree search," *IEEE Trans. Cybern.,* vol. 48, no. 2, pp. 510–521, Feb. 2018.
- M. Rost and S. Schmid, "On the hardness and inapproximability of virtual network embeddings," *IEEE/ACM Trans. Netw*., vol. 28, no. 2, pp. 791–803, 2020.
- A. Satpathy, M. N. Sahoo, C. Swain, P. Bellavista, M. Guizani, K. Muhammad, and S. Bakshi, "Virtual network embedding: Literature assessment, recent advancements, opportunities, and challenges," *IEEE Commun. Surveys Tuts.*, pp. 1–1, 2025.

Questions?

Email: ballanty@sfu.ca