VNE-Sim: A Virtual Network Embedding Simulator

Soroush Haeri and Ljiljana Trajković

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

Roadmap

- Network virtualization
- Virtual network embedding
- VNE formulation
- VNE simulators
- VNE-Sim: architecture and components
- VNE: a discrete event system
- VNE algorithms
- Simulation scenarios
- Conclusions, future work, and references

Network virtualization

- Enables coexistence of multiple virtual networks on a physical infrastructure
- 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

Roadmap

- Network virtualization
- Virtual network embedding
- VNE formulation
- VNE simulators
- VNE-Sim: architecture and components
- VNE: a discrete event system
- VNE algorithms
- Simulation scenarios
- Conclusions, future work, and references

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 algorithm enables path splitting:
  - a flow may be divided into multiple flows with lower capacity
  - flows are routed through various paths

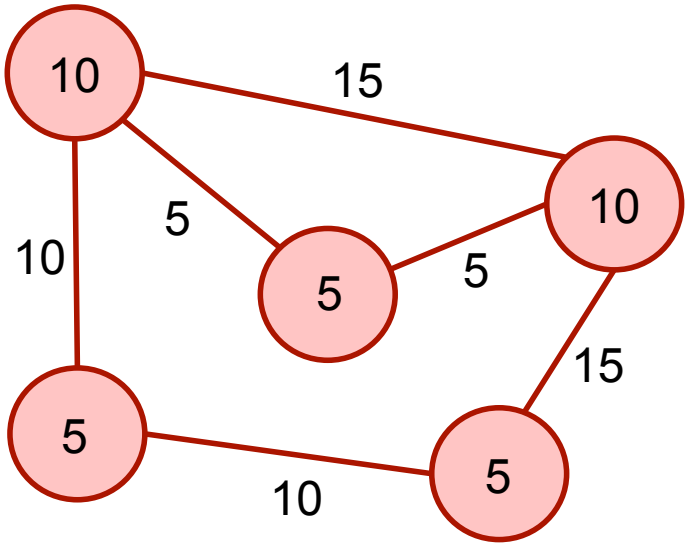
- D. G. Andersen, “Theoretical approaches to node assignment,” Dec. 2002, Unpublished Manuscript. [Online]. Available: http://repository.cmu.edu/compsci/86/.

Roadmap

- Network virtualization
- Virtual network embedding (VNE)
- VNE formulation
- VNE simulators
- VNE-Sim: architecture and components
- VNE: a discrete event system
- VNE algorithms
- Simulation scenarios
- Conclusions, future work, and references

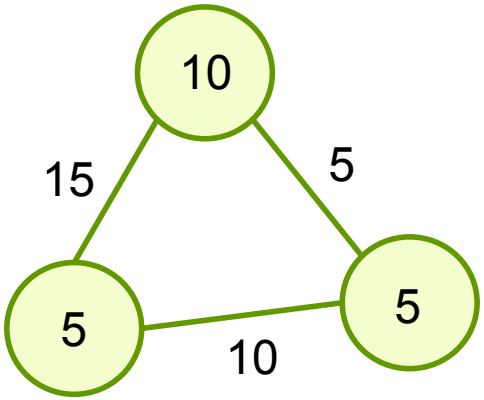
VNE formulation: constrains

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

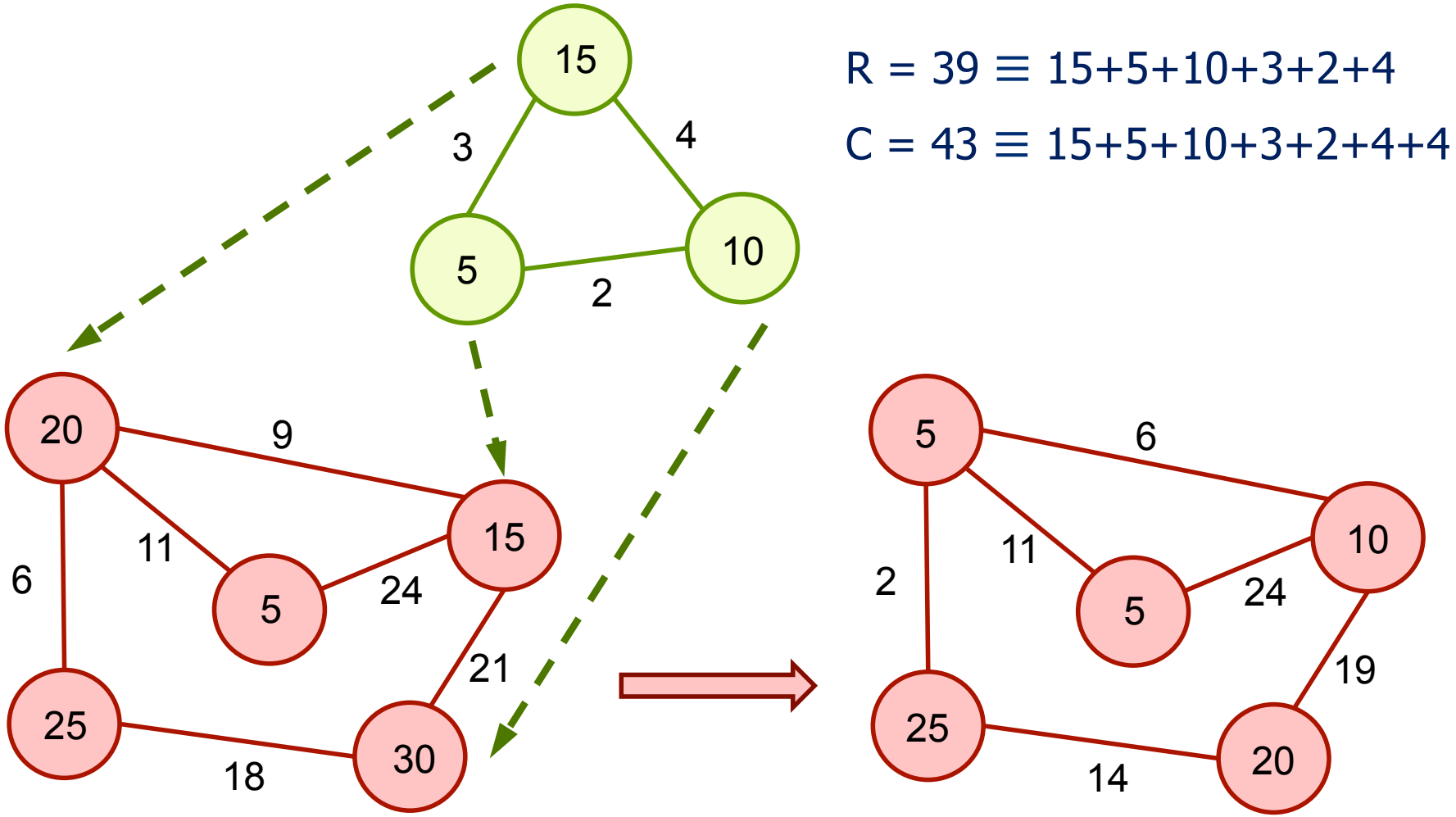


VNE formulation: constrains

- 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 ≡ 15+5+10+3+2+4

C = 43 ≡ 15+5+10+3+2+4+4

Roadmap

- Network virtualization
- Virtual network embedding
- VNE formulation
- VNE simulators
- VNE-Sim: architecture and components
- VNE: a discrete event system
- VNE algorithms
- Simulation scenarios
- Conclusions, future work, and references

Simulation of VNE algorithms

- Performance of VNE algorithms is only evaluated using discrete event simulations
- There is a need for a scalable, flexible, reliable, and modular VNE simulator

Existing VNE simulators

- Embed and ViNEYard:
  - both are early developed C-based VNE simulators
  - no longer maintained
  - lack documentation and do not provide interfaces for developing new VNE algorithms and performance metrics

- 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.
- 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.

Existing VNE simulators

- Alevin:
  - written in Java
  - modular design
  - provides a graphical user interface
  - design does not enable users to define network elements of their choice such as defining CPU, memory, and storage
  - writing scripts to perform batch simulations is rather cumbersome

- M. T. Beck, C. Linnhoff-Popien, A. Fischer, F. Kokot, and H. de Meer, “A simulation framework for virtual network embedding algorithms,” in *Proc. 16th Int. Telecommunications Netw. Strategy and Planning Symp.*, Funchal, Portugal, Sept. 2014, pp. 1–6.

Roadmap

- Network virtualization
- Virtual network embedding
- VNE formulation
- VNE simulators
- VNE-Sim: architecture and components
- VNE: a discrete event system
- VNE algorithms
- Simulation scenarios
- Conclusions, future work, and references

VNE-Sim

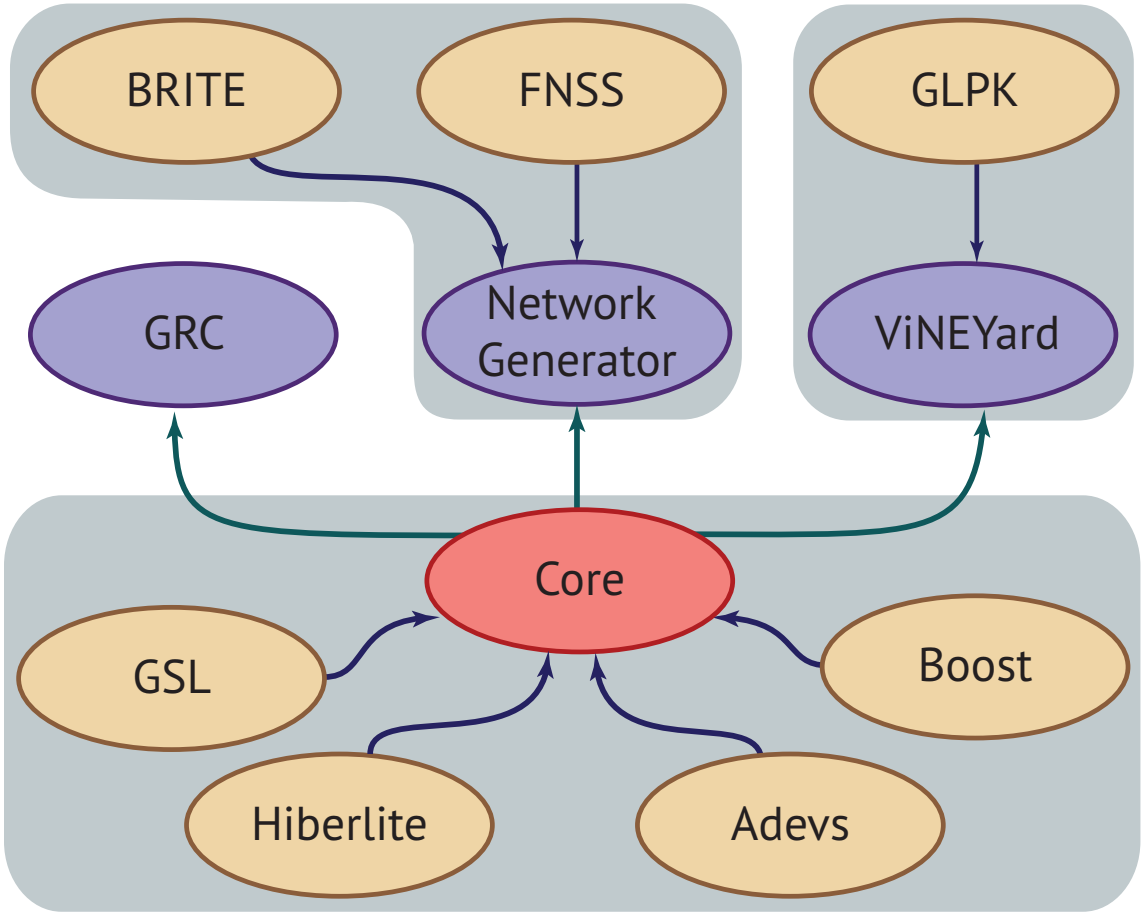
- A discrete event VNE simulator written in C++
- Used for performance evaluation of a number of VNE algorithms
- Publicly available under the terms of the MIT License
- Provides extensible interfaces for users to implement algorithms and customizable network elements
- Good memory management

- S. Haeri, Q. Ding, Z. Li, and Lj. Trajkovic, "Global resource capacity algorithm with path splitting for virtual network embedding," in *Proc. IEEE Int. Symp. Circuits and Systems*, Montreal, QC, Canada, May 2016, pp. 666–669.
- S. Haeri and Lj. Trajkovic, "Virtual network embeddings in data center networks," in *Proc. IEEE Int. Symp. Circuits and Systems*, Montreal, QC, Canada, May 2016, pp. 874–877.

VNE-Sim architecture

- Written using the 2011 C++ standard (C++11)
- The CMake build system is employed for code compilation
- VNE-Sim relies on several external tools:
  - Boost File System, Log, Thread, and Unit Test Framework libraries
  - GNU Scientific Library (GSL)
  - GNU Linear Programming Kit (GLPK)
  - SQLite3 library

VNE-Sim architecture



VNE-Sim components and their dependencies

VNE-Sim components

- Core package:
  - contains classes and interfaces for implementing various virtual network embedding algorithms
- Network-generator package:
  - generates various network topologies and Virtual Network Requests:
    - Boston University Representative Internet Topology Generator (BRITE) for random graphs
    - Fast Network Simulation Setup (FNSS) for well defined topologies
  - relies on configuration.xml file to define various parameters for simulation scenarios

VNE-Sim components

- GRC package:
  - contains the implementation of the Global Resource Capacity (GRC) algorithm
- ViNEYard package:
  - contains the implementation of R-ViNE and D-ViNE

- 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.

Roadmap

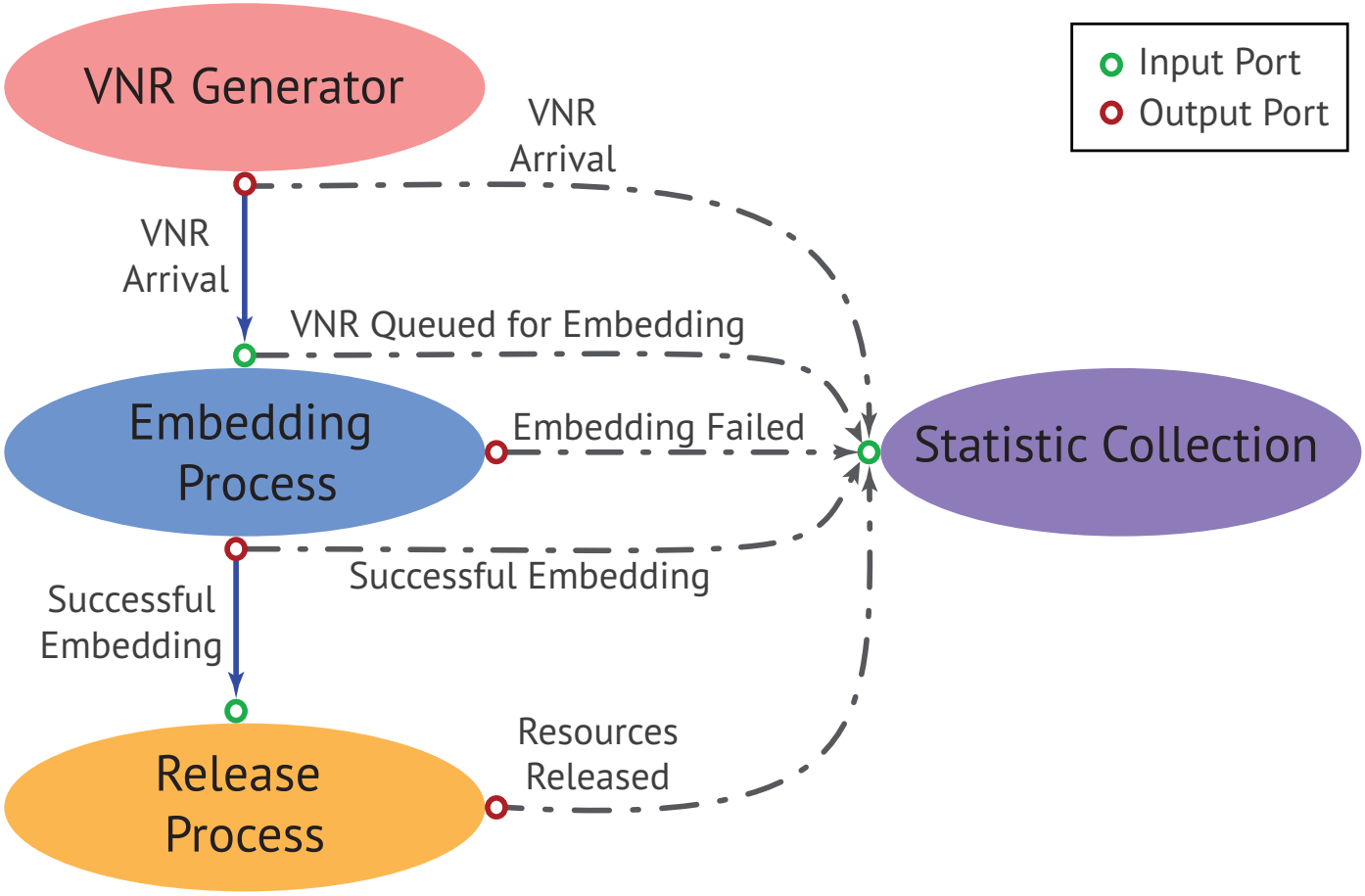
- Network virtualization
- Virtual network embedding
- VNE formulation
- VNE simulators
- VNE-Sim: architecture and components
- VNE: a discrete event system
- VNE algorithms
- Simulation scenarios
- Conclusions, future work, and references

VNE: a discrete event system

- In VNE-Sim, VNE process is modeled as a discrete event system using the discrete event system specification (DEVS) framework
  - Adevs library is employed for the implementations
  - enables seamless implementation of various VNE approaches including single and batch processing

- 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, 2010.

VNE: a discrete event system



Directed acyclic graph used for modeling the VNE process in VNE-Sim

VNE: a discrete event system

- Virtual Network Requests (VNRs) are first generated by the VNR Generator and passed to the Embedding Process
- Successfully embedded VNRs are forwarded to the Release Process
- The Statistic Collection module is an observer that receives information from all input/output ports of other modules

Roadmap

- Network virtualization
- Virtual network embedding
- VNE formulation
- VNE simulators
- VNE-Sim: architecture and components
- VNE: a discrete event system
- VNE algorithms
- Simulation scenarios
- Conclusions, future work, and references

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: 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 (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:

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

  - $0 < d < 1$: 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:

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

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.

Roadmap

- Network virtualization
- Virtual network embedding
- VNE formulation
- VNE simulators
- VNE-Sim: architecture and components
- VNE: a discrete event system
- VNE algorithms
- Simulation scenarios
- Conclusions, future work, and references

Performance metrics

Name	Description
Acceptance ratio	Ratio of virtual networks that were successfully embedded into the substrate topology
Revenue to cost ratio	The virtualization overhead
Average revenue	Sum of CPU and bandwidth demands realized for the virtual networks
Average cost	Sum of CPU and bandwidth resources being used for the embedding
Node and link utilizations	CPU and bandwidth utilization of substrate nodes and links
Runtime	Average runtime of the algorithm

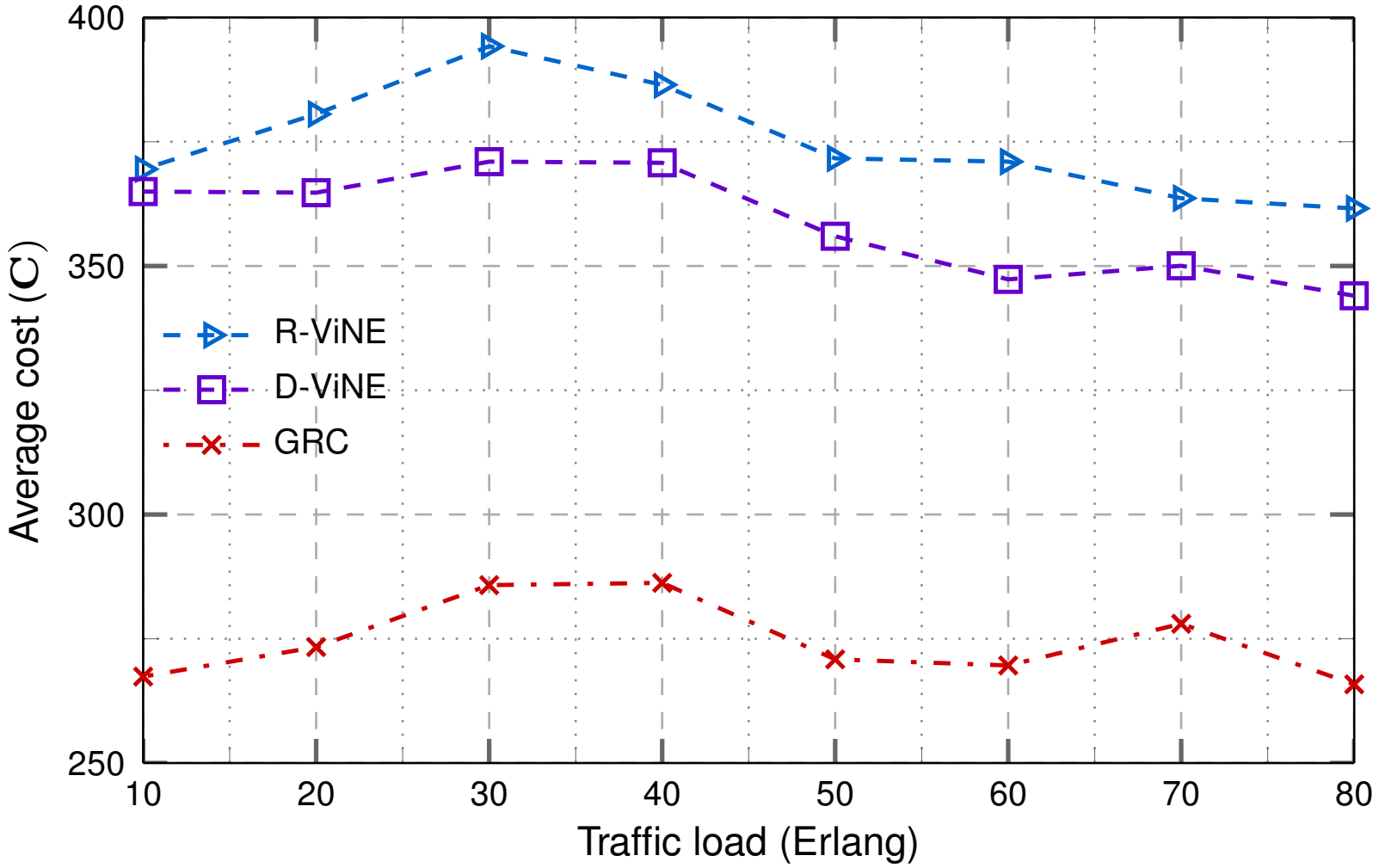
Simulation scenarios

- Substrate network:
  - 50 nodes and 221 edges
- Each node of the substrate network is randomly placed on a 25 × 25 grid
- The CPU capacity of substrate nodes and the bandwidth of substrate links are uniformly distributed between 50 and 100 units

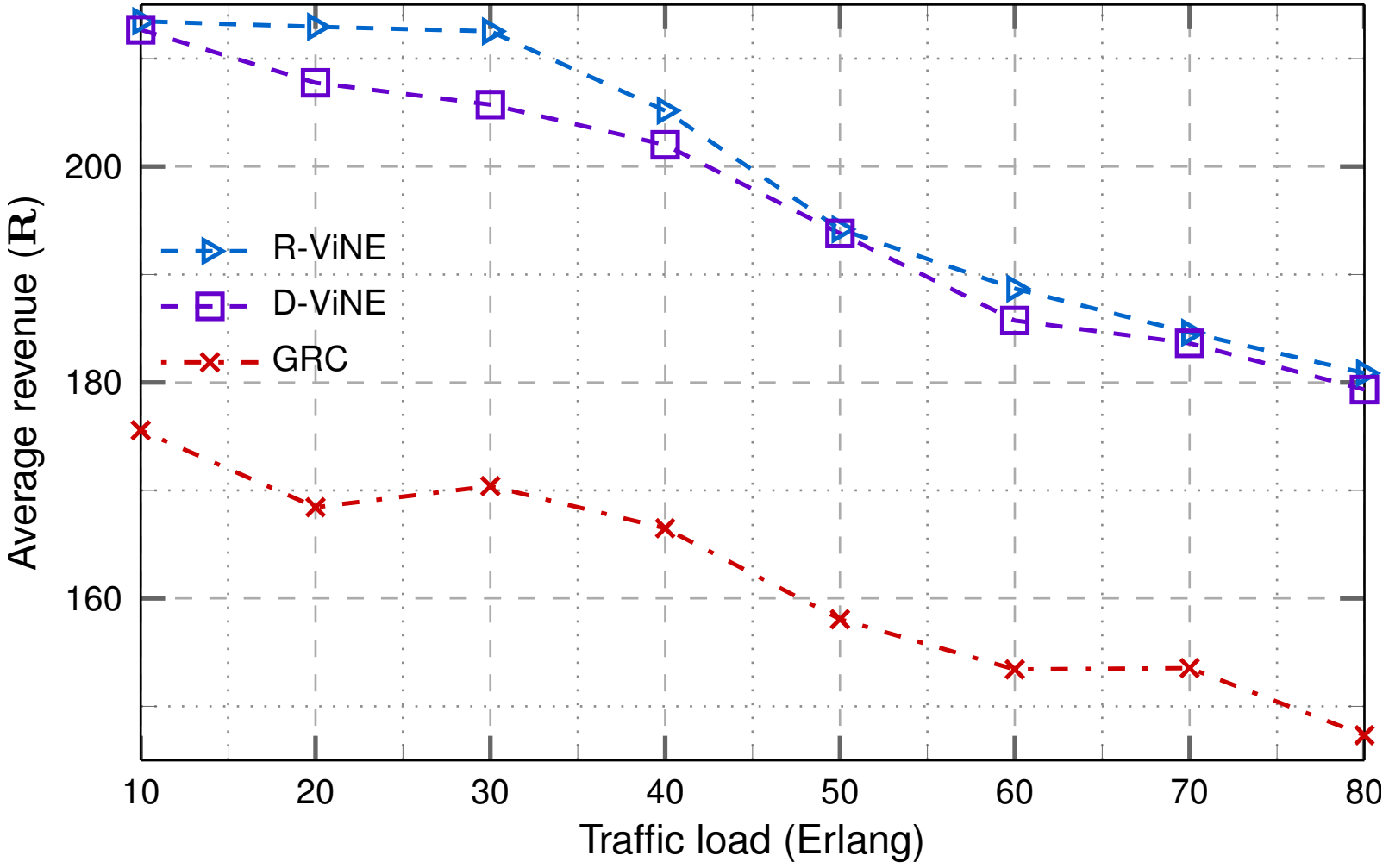
Simulation scenarios

- Virtual network requests:
  - number of nodes uniformly distributed between 3 and 10
- CPU requirements:
  - uniformly distributed between 2 and 20 units
- Bandwidth requirements:
  - uniformly distributed between 0 and 50 units

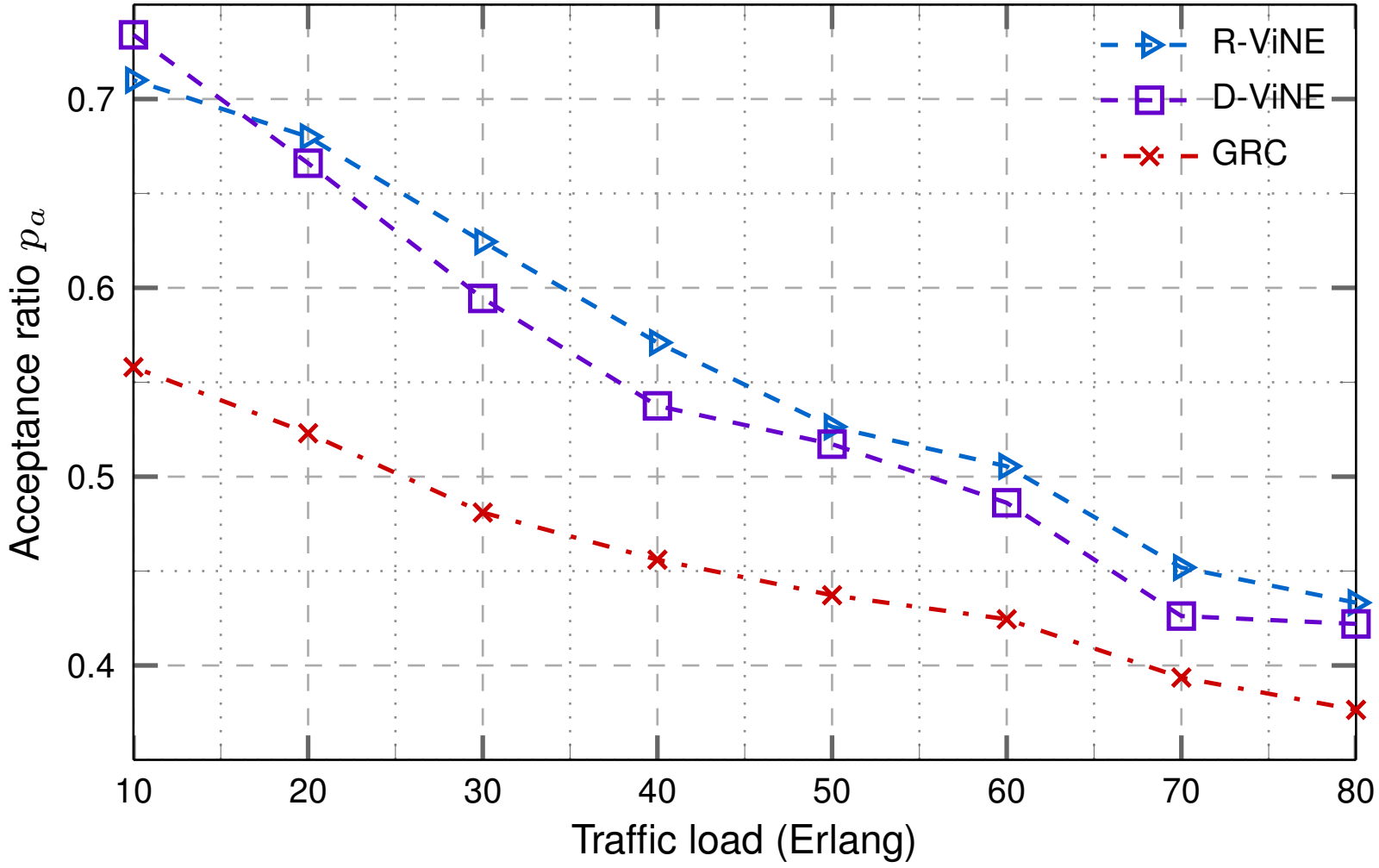
Simulation results: average cost



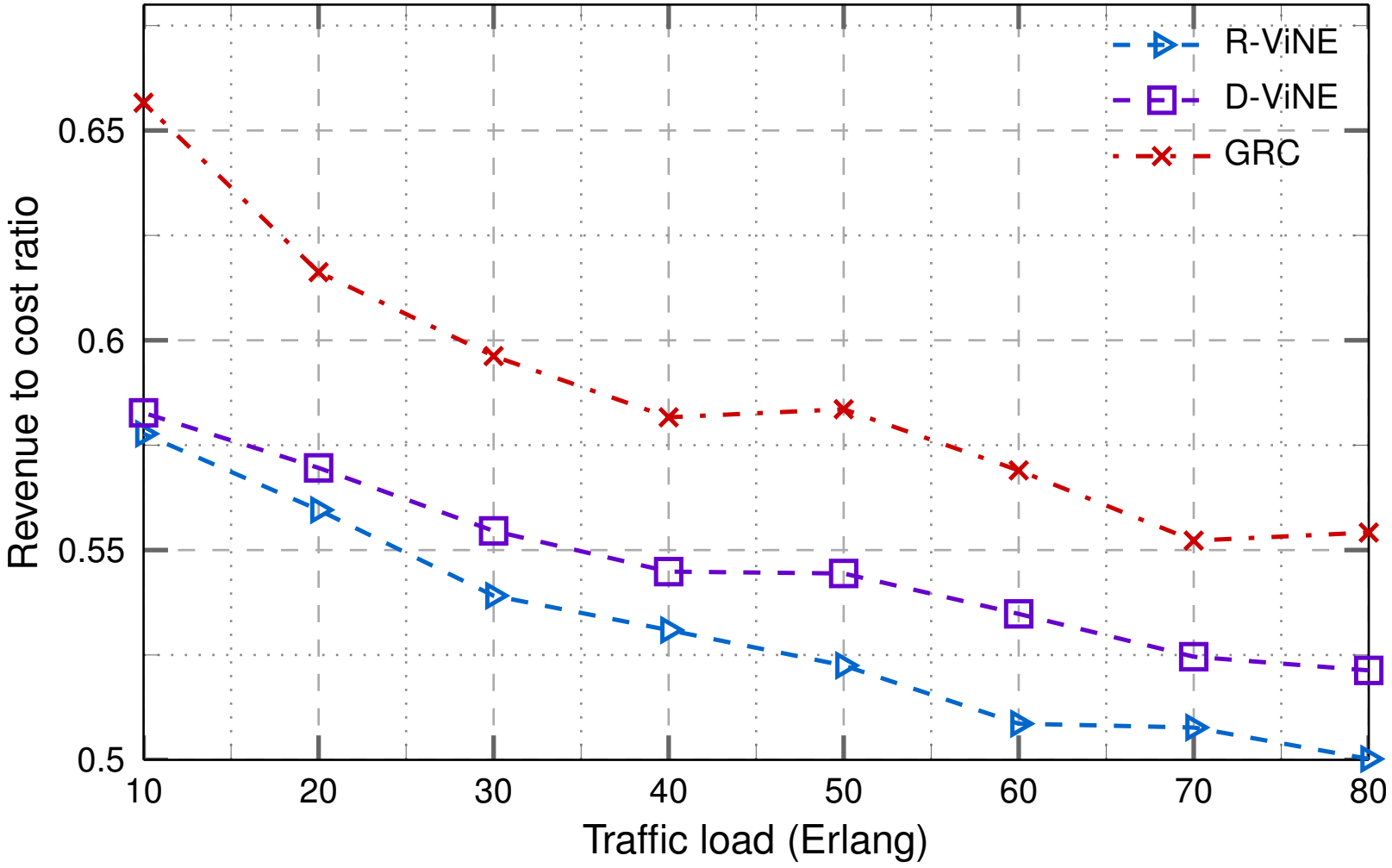
Simulation results: average revenue



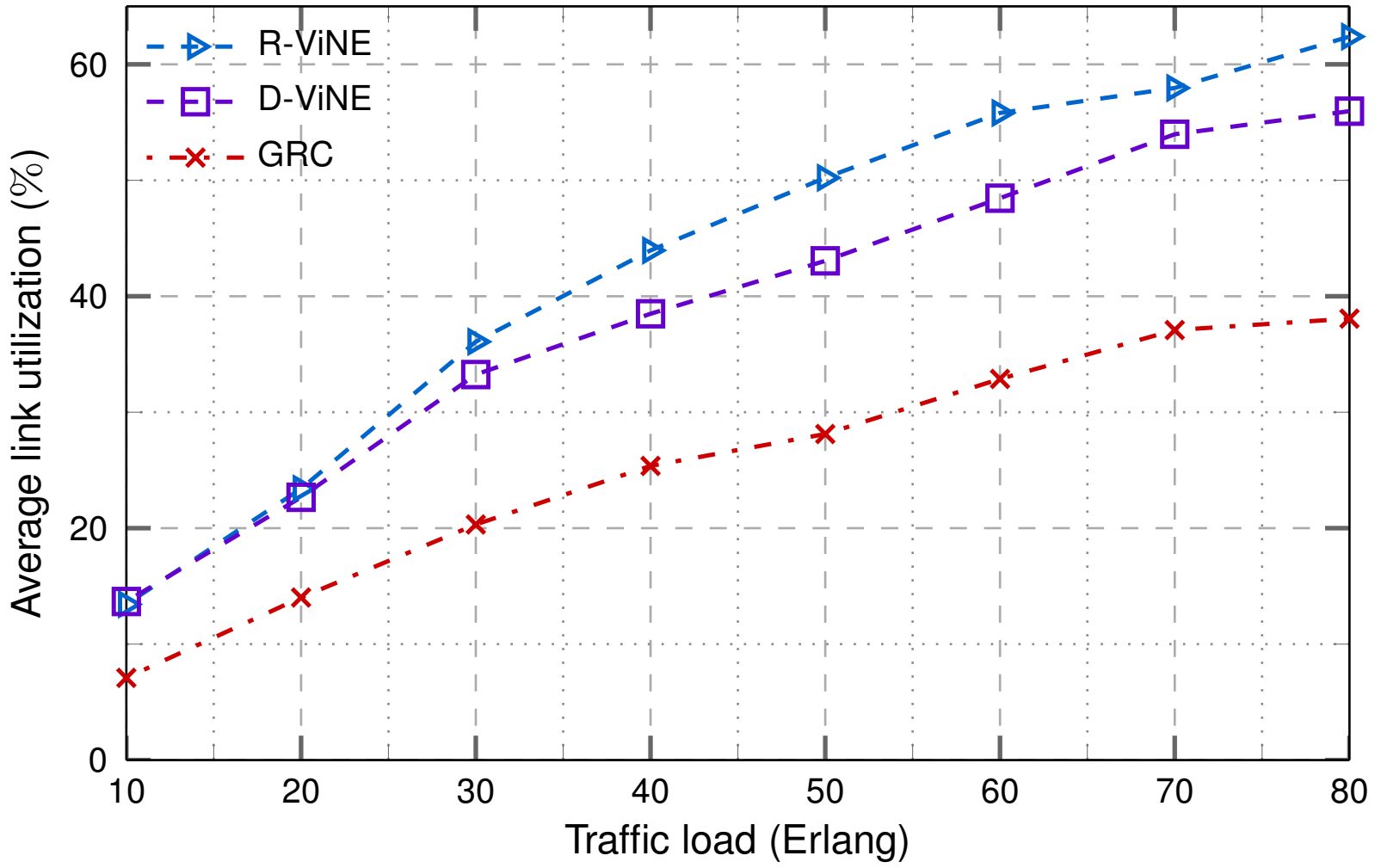
Simulation results: acceptance ratio



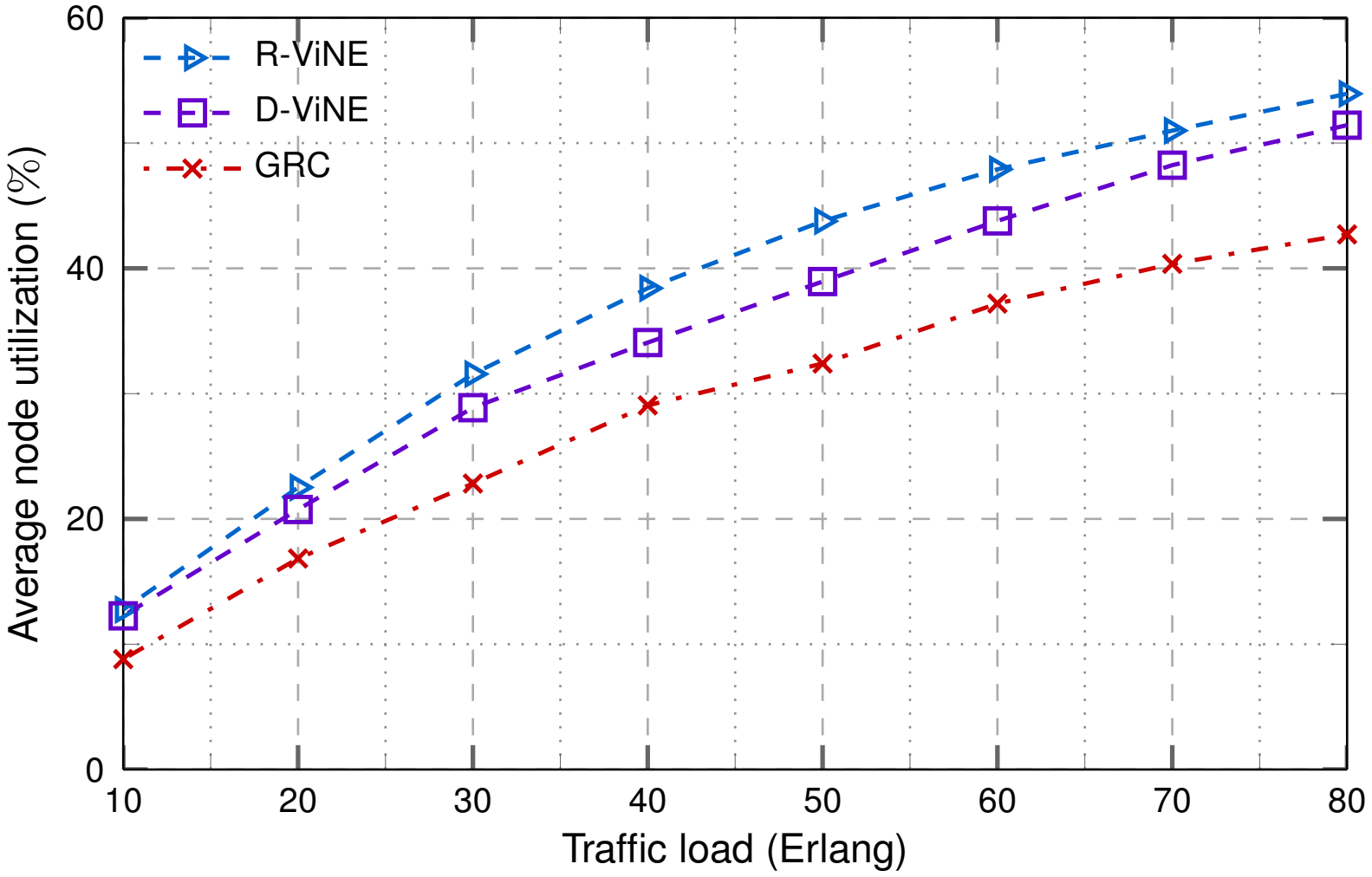
Simulation results: revenue to cost ratio



Simulation results: link utilization



Simulation results: node utilization



Roadmap

- Network virtualization
- Virtual network embedding
- VNE formulation
- VNE simulators
- VNE-Sim: architecture and components
- VNE: a discrete event system
- VNE algorithms
- Simulation scenarios
- Conclusions, future work, and references

Conclusions and future work

VNE-Sim: a virtual network embedding simulator

- formalizes the VNE process as a discrete event system based on the DEVS framework
- modular and scalable design
  - enables researchers to seamlessly implement new VNE algorithms

Future work includes implementing:

- other VNE algorithms
- a source code documentation system
- a scripting infrastructure for visualization of results

References

- S. Haeri, Q. Ding, Z. Li, and Lj. Trajkovic, "Global resource capacity algorithm with path splitting for virtual network embedding," in *Proc. IEEE Int. Symp. Circuits and Systems*, Montreal, QC, Canada, May 2016, pp. 666–669.
- S. Haeri and Lj. Trajkovic, "Virtual network embeddings in data center networks," in *Proc. IEEE Int. Symp. Circuits and Systems*, Montreal, QC, Canada, May 2016, pp. 874–877.

References

Discrete Event System Simulation:

- 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, 2010.

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.