UNIVERSITY OF Waterloo

Virtual Network Embedding with Coordinated Node and Link Mapping

N. M. Mosharaf Kabir Chowdhury
Muntasir Raihan Rahman and Raouf Boutaba
University of Waterloo

Motivation

- Network Virtualization
  - Coexistence of multiple virtual networks (VNs) over a shared substrate
  - Applications
    - Research experiments
    - Future Internet architecture (e.g., Clean-slate)

- Major challenge
  - Efficient assignment of substrate network resources to virtual network's requirements

VN Embedding Problem

- Given:
  - Single substrate network: $G^S = (N^S, E^S)$
  - Online VN requests: $G^V = (N^V, E^V)$
  - Requirements and Constraints of virtual nodes and virtual links

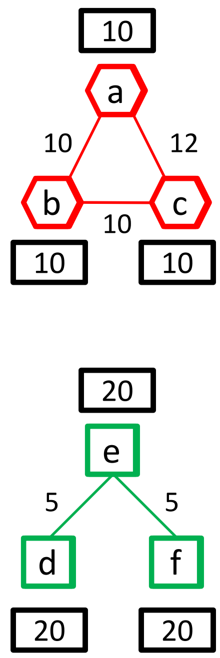
- Task:
  - Assign virtual nodes and links to substrate nodes and links
  - Allocate resources
    - CPU, bandwidth

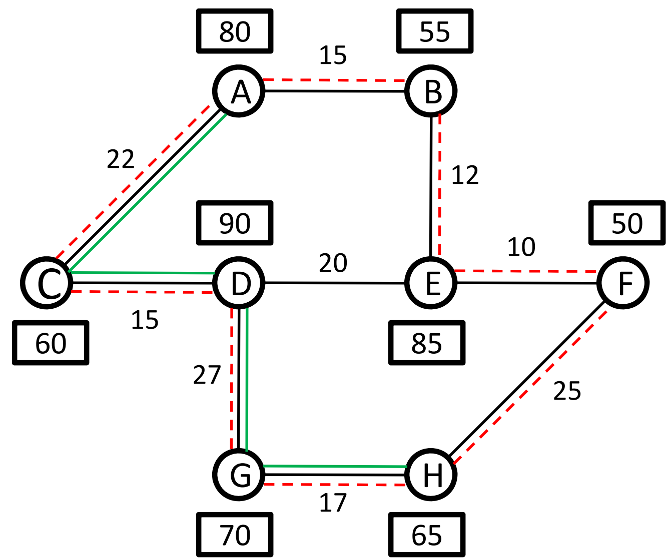
VN Embedding Objectives

- **Maximize**
  - Acceptance ratio
    - Percentage of request accepted
  - Revenue
    - Based on resources ***requested*** for a VN

- **Minimize**
  - Cost
    - Based on substrate network resources ***allocated*** for embedding VN requests

Virtual Network Embedding





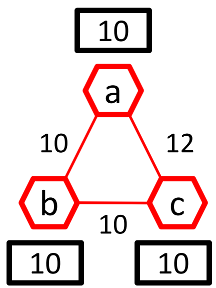
NP-hard

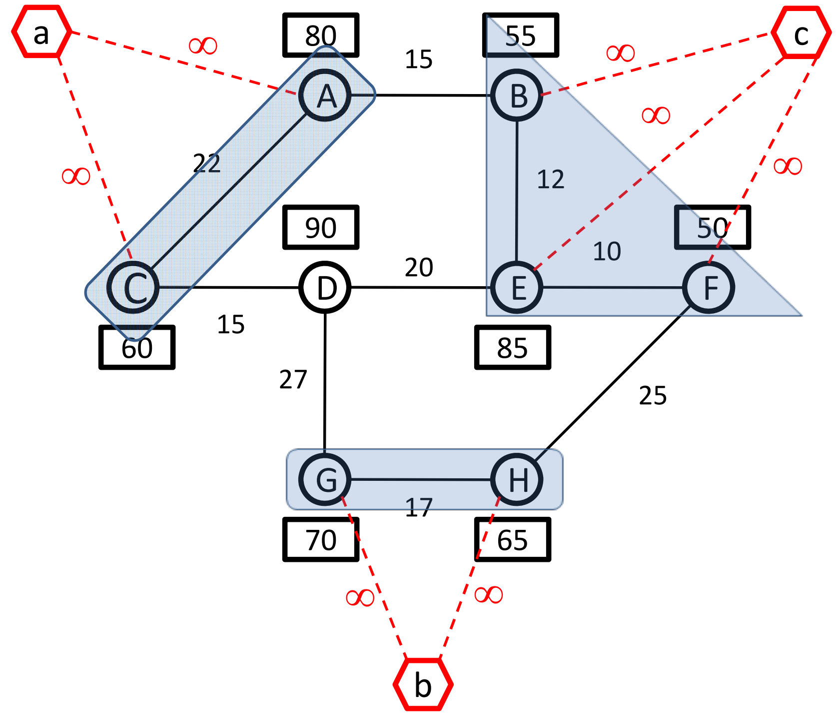
Observations

- Existing *heuristics*:
  1. Disjoint node and link mapping
  2. Ignore (completely) location constraints on virtual nodes

- Our approach
  - Node mapping influences link mapping
  - Location constraints take the front seat
    - Meta-VN request

Substrate Graph Augmentation





Mixed Integer Program

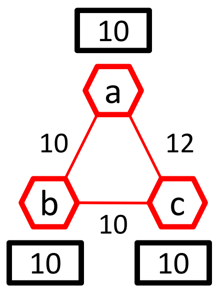
- Objective

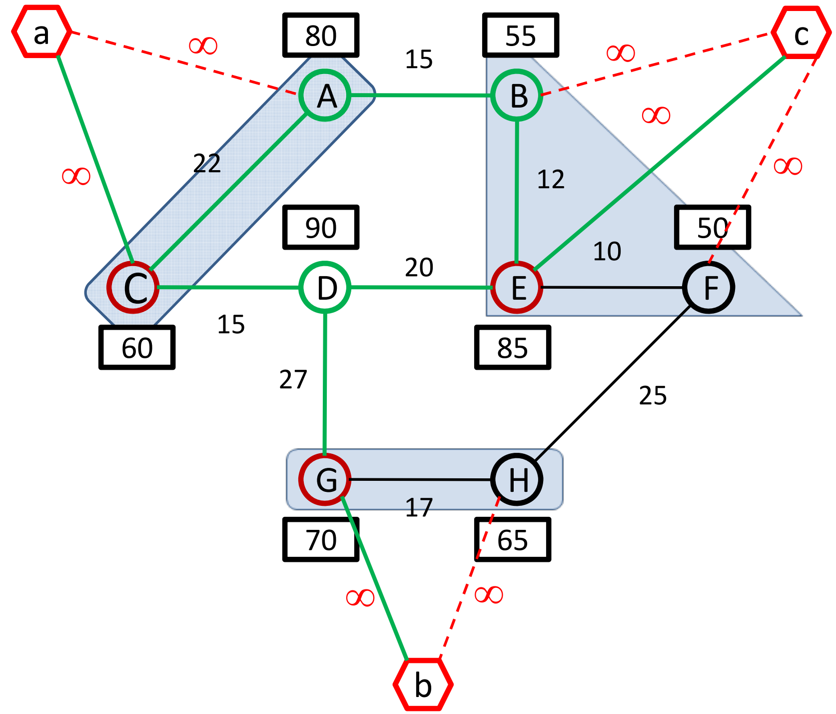
$$\text{minimize} \quad \sum_{uv \in E^S} \frac{\alpha_{uv}}{R_E(u,v)+\delta} \sum_i f^i_{uv} + \sum_{w \in N^S} \frac{\beta_w}{R_N(w)+\delta} \sum_{m \in N^{S'} \setminus N^S} x_{mw}\, c(m)$$

α, β: Tuning parameters
R: Remaining capacity

- Flow variables: $f^i_{uv} \geq 0$
  - Multi-commodity flow constraints
- Binary variables: $x_{uv} \in \{0, 1\}$
  - Exactly one substrate node is selected for each meta-node
  - At most one meta-node is mapped onto a substrate node

Mixed Integer Program Solution





NP-hard

LP Relaxation

- Relax the binary constraints on the x variable
- **Problem 1:** x values do not select single substrate nodes for every meta-node
  - Use rounding techniques (e.g., deterministic, randomized)
- **Problem 2:** x values are inconsistent
  - Use the product of x and f while rounding

ViNEYard (D-ViNE & R-ViNE)

For each VN request:

- Augment the substrate graph
- Solve the resulting LP

INITIALIZATION

- For each virtual node:
  - Calculate the probability for each meta-node to be selected for the corresponding virtual node
  - Selection:
    - D-ViNE: Select the meta-link with the highest probability
    - R-ViNE: Select a meta-link randomly with the calculated probability

NODE MAPPING

- Use MCF to map virtual edges

LINK MAPPING

- If the VN request is accepted
  - Update residual capacities of the substrate resources

FINALIZATION

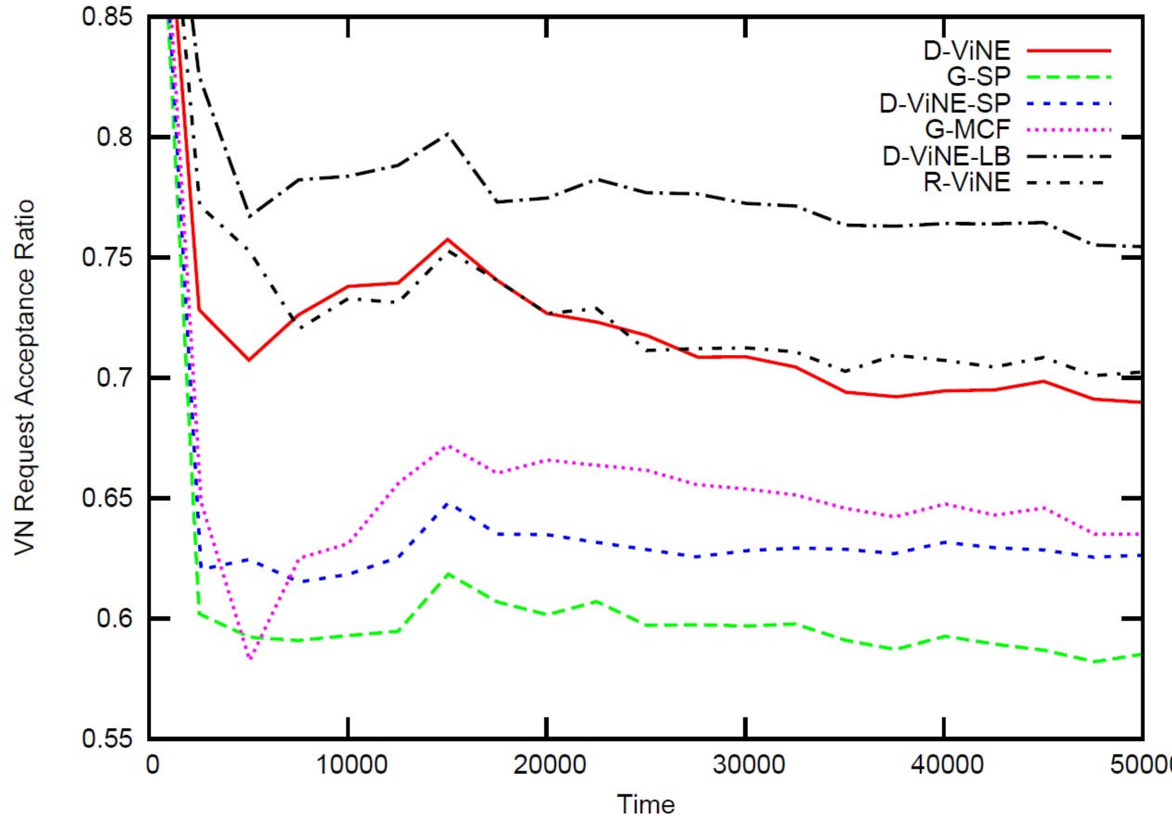
Performance Evaluation

Compared Algorithms	Description
D-ViNE	Deterministic Node Mapping with Splittable Link Mapping using MCF
R-ViNE	Randomized Node Mapping with Splittable Link Mapping using MCF
G-SP	Greedy Node Mapping with Shortest Path Based Link Mapping
G-MCF	Greedy Node Mapping with Splittable Link Mapping using MCF
D-ViNE-SP	Deterministic Node Mapping with Shortest Path Based Link Mapping
D-ViNE-LB	Deterministic Node Mapping with Splittable Link Mapping using MCF, where $\alpha_{uv} = \beta_w = 1$, for all $u, v, w \in N^S$

Simulation Setup

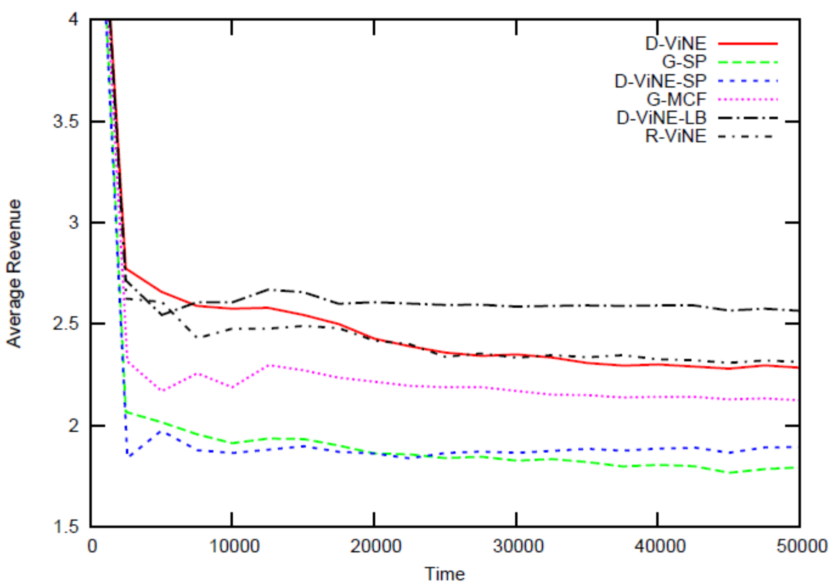
- Substrate network
  - 50 nodes in a 25x25 grid with 0.5 link probability
  - CPU/BW uniformly distributed in the range: 50-100 units
- VN requests
  - Poisson arrival rates from 4 VN requests 100 time units
  - Exponentially distributed lifetime of 1000 time units
  - 2-10 nodes with 0.5 link probability
- *Tools:* GT-ITM (Georgia Tech Internet Topology Models), GLPK (GNU Linear Programming Toolkit)

Acceptance Ratio

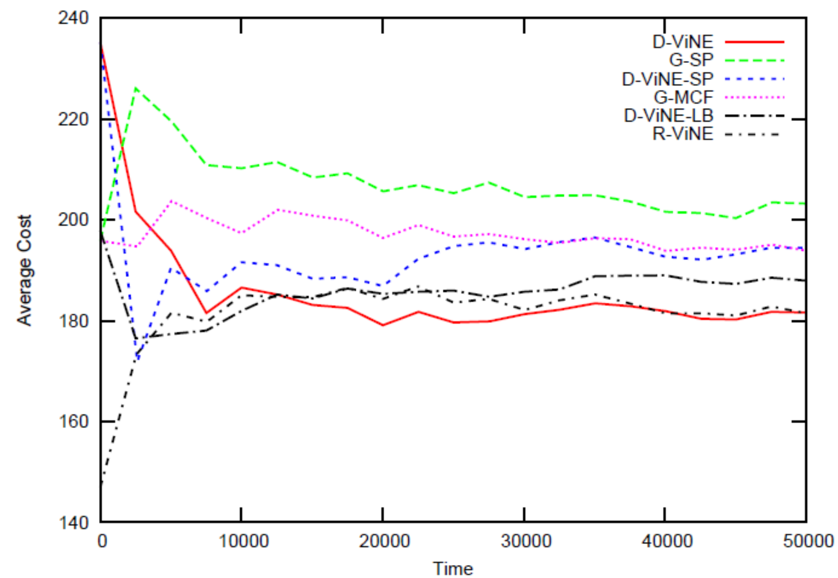


Revenue Vs Cost

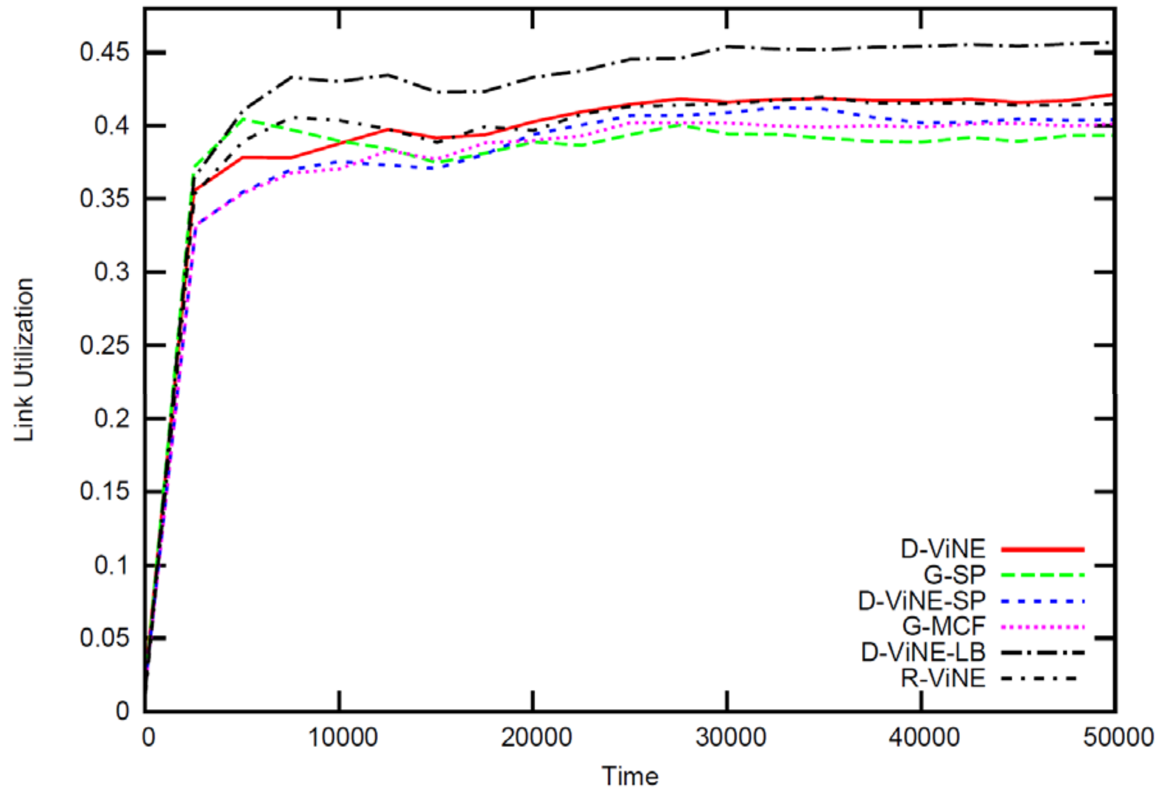
Revenue



Cost



Link Utilization



Conclusions

- ViNEYard Algorithms
  - Improved correlation between the node mapping and the link mapping phases
  - Increased acceptance ratio and revenue with decreased cost
- Ongoing and Future Work
  - Window-based extension to D-ViNE and R-ViNE a.k.a. **WiNE**
  - Extension to inter-domain scenario a.k.a. **PolyViNE**
  - Approximation factors for D-ViNE and R-ViNE

Thank You!

Questions?

http://www.mosharaf.com/

BACKUP SLIDES

Program 5.1 (Mixed Integer Program for Virtual Network Embedding)

Variables:

- f^i_{uv}: A flow variable denoting the total amount of flow in the $u \rightarrow v$ direction on the substrate edge (u, v) for the i'th virtual edge.
- x_{uv}: A binary variable, which has the value '1' if $\sum_i \left(f^i_{uv} + f^i_{vu}\right) > 0$; otherwise, it is set to '0'.

Objective:

$$\text{minimize} \quad \sum_{uv \in E^S} \frac{\alpha_{uv}}{R_E(uv) + \delta} \sum_i f^i_{uv} + \sum_{w \in N^S} \frac{\beta_w}{R_N(w) + \delta} \sum_{m \in N^{S'} \setminus N^S} x_{mw}\, c(m) \tag{5.7}$$

Constraints:

- Capacity Constraints:

$$\sum_i (f^i_{uv} + f^i_{vu}) \leq R_E(uv)\, x_{u,v}, \forall u, v \in N^{S'} \tag{5.8}$$

$$R_N(w) \geq x_{mw}\, c(m), \forall m \in N^{S'} \setminus N^S, \forall w \in N^S \tag{5.9}$$

- Flow Related Constraints:

$$\sum_{w \in N^{S'}} f^i_{uw} - \sum_{w \in N^{S'}} f^i_{wu} = 0, \forall i, \forall u \in N^{S'} \setminus \{s_i, t_i\} \tag{5.10}$$

$$\sum_{w \in N^{S'}} f^i_{s_i w} - \sum_{w \in N^{S'}} f^i_{w s_i} = b(e^V_i), \forall i \tag{5.11}$$

$$\sum_{w \in N^{S'}} f^i_{t_i w} - \sum_{w \in N^{S'}} f^i_{w t_i} = -b(e^V_i), \forall i \tag{5.12}$$

- Meta and Binary Constraints:

$$\sum_{w \in \Omega(m)} x_{mw} = 1, \forall m \in N^{S'} \setminus N^S \tag{5.13}$$

$$\sum_{m \in N^{S'} \setminus N^S} x_{mw} \leq 1, \forall w \in N^S \tag{5.14}$$

$$x_{uv} \leq R_E(uv), \forall u, v \in N^{S'} \quad (5.15)$$

$$x_{uv} = x_{vu}, \forall u, v \in N^{S'} \quad (5.16)$$

- Domain Constraints:

$$f_{uv}^i \geq 0, \forall u, v \in N^{S'} \quad (5.17)$$

$$x_{uv} \in \{0, 1\}, \forall u, v \in N^{S'} \quad (5.18)$$

Remarks:

- The objective function (5.7) of the MIP tries to minimize the cost of embedding the VN request as well as balance the load. By dividing the cost with the residual capacity, it is ensured that the resources with more residual capacities are preferred over the resources with less residual capacities. $1 \leq \alpha_{uv} \leq R_E(uv)$ and $1 \leq \beta_w \leq R_N(w)$ are parameters to control the importance of load balancing while embedding a request. $\delta \to 0$ is a small positive constant to avoid dividing by zero in computing the objective function.
- Constraint set (5.8) and (5.9) contains the node and edge capacity bounds. Summing up f_{uv}^i and f_{vu}^i in (5.8) ensures that the summation of flows on both directions of the undirected edge (uv) remains within its available bandwidth.
- Constraint sets (5.13) and (5.14) are related to the augmented portion of the substrate graph. Constraint set (5.13) makes sure that only one substrate node is selected for each meta-node, whereas constraint set (5.14) ensures that no more than one meta-node is placed on a substrate node.
- Constraint sets (5.15) and (5.16) together with (5.4) ensure that x_{uv} is set whenever there is any flow in either direction of the substrate edge uv.

D-ViNE

UNIVERSITY OF Waterloo

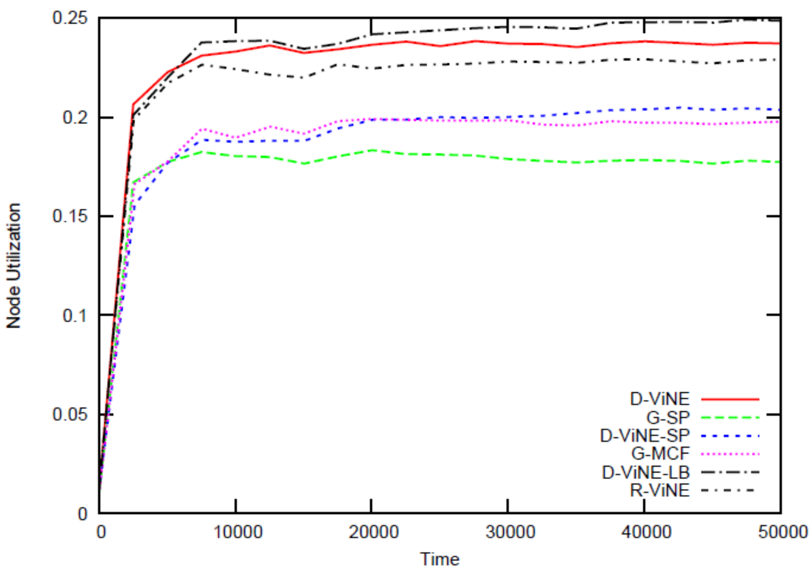
procedure D-VINE($G^V = (N^V, E^V)$)
  Create augmented substrate graph $G^{S'} = \left(N^{S'}, E^{S'}\right)$
  Solve **VNE_LP_RELAX**
  for all $n^S \in N^S$ **do**
    $\varphi(n^S) \leftarrow 0$
  end for
  for all $n \in N^V$ **do**
    if $\Omega(n) \cap \{n^S \in N^S | \varphi(n^S) = 1\} = \emptyset$ **then**
      VN request cannot be satisfied
      return
    end if
    for all $z \in \Omega(n)$ **do**
      $p_z \leftarrow (\sum_i f^i_{\mu(n)z} + f^i_{z\mu(n)}) x_{\mu(n)z}$
    end for
    Let $z_{max} = \arg\max_{z \in \Omega(n)} \{p_z | \varphi(z) = 0\}$
    set $\mathcal{M}_N(n) \leftarrow z_{max}$
    $\varphi(z_{max}) \leftarrow 1$
  end for
  Solve MCF to map virtual edges.
  Update residual capacities of the network resources.
end procedure

R-ViNE

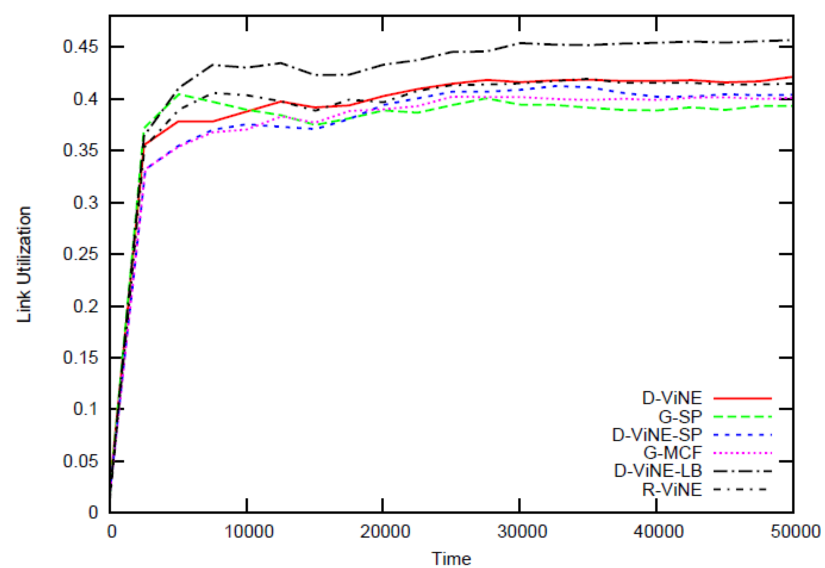
```
procedure R-VINE(G^V = (N^V, E^V))
    Create augmented substrate graph G^{S'} = (N^{S'}, E^{S'})
    Solve VNE_LP_RELAX
    for all n^S ∈ N^S do
        φ(n^S) ← 0
    end for
    for all n ∈ N^V do
        if Ω(n) ∩ {n^S ∈ N^S | φ(n^S) = 1} = ∅ then
            VN request cannot be satisfied
            return
        end if
        for all z ∈ Ω(n) do
            p_z ← (Σ_i f^i_{μ(n)z} + f^i_{zμ(n)}) x_{μ(n)z}
        end for
        p_sum ← Σ_{z∈Ω(n)} p_z
        for all z ∈ Ω(n) do
            p_z ← p_z / p_sum
        end for
        set M_N(n) ← z with probability p_z
        φ(z) ← 1 with probability p_z
    end for
    solve MCF to map virtual edges.
    Update residual capacities of the network resources.
end procedure
```

Resource Utilization

Node Utilization



Link Utilization



Effect of Increasing Load