Identity Management and Resource Allocation in the Network Virtualization Environment

Mosharaf Chowdhury
School of Computer Science
University of Waterloo

NETWORK VIRTUALIZATION

Why Network Virtualization?

- Internet is *almost* ossified
  - Lots of band-aids and makeshift solutions (e.g., overlays)
  - A new architecture (aka clean-slate) is needed
- Hard to come up with a *one-size-fits-all* architecture
  - Almost impossible to predict what future might unleash
- Why not create an *all-sizes-fit-into-one* architecture instead!
  - Open and expandable
  - Coexistence of heterogeneous architectures

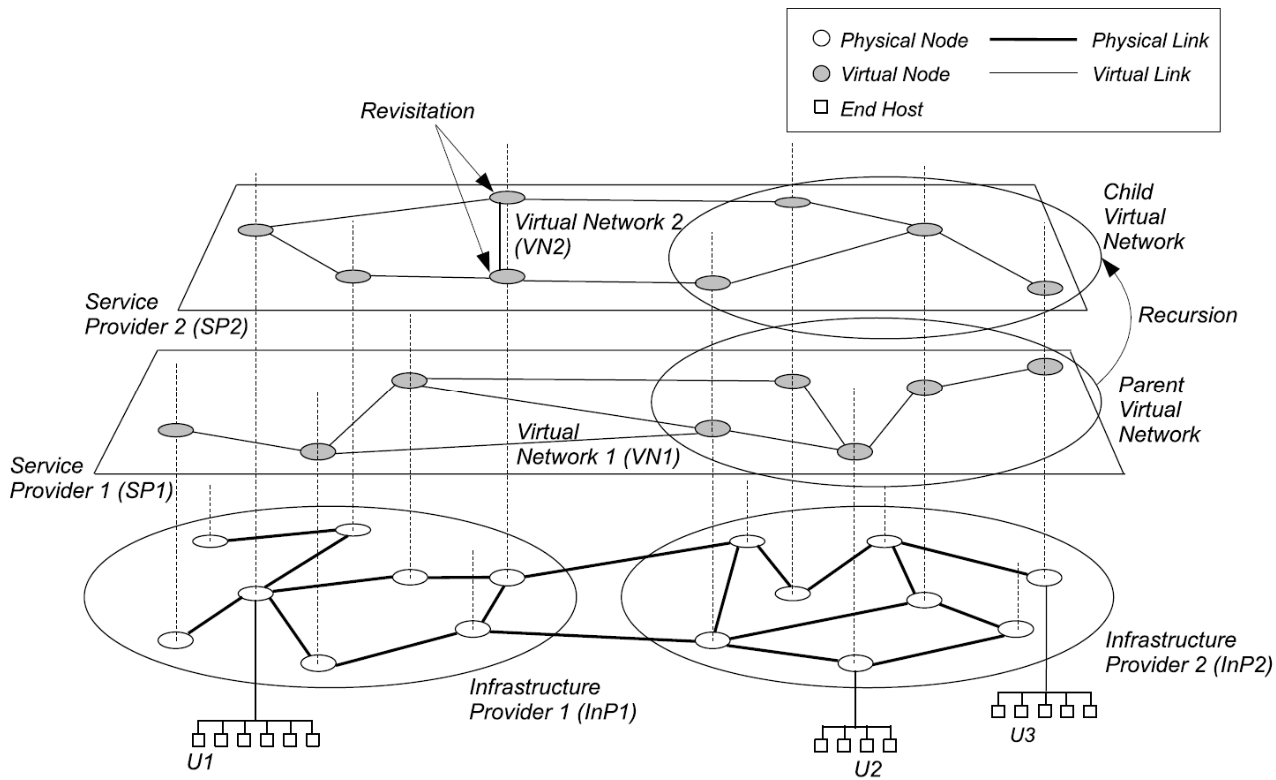
What is Network Virtualization?

- Transparent abstraction of networking platform and resources
  - Multiple logical interpretations of the physical characteristics
    - Multiple virtual networks (VNs)

- Additional level of indirection
  - Indirect access to network resources

- Resource partitioning and isolation
  - Physical and logical
  - Dynamic provisioning and configuration

Network Virtualization Environment



Challenges

Instantiation

- Concerned with issues related to successful creation of virtual networks

Operations

- Deals with operations of virtual networks and virtual components

Management

- Manages co-existing virtual networks

Virtual Network Embedding

Identity Management

Identity Management in the Network Virtualization Environment

IMARK

Motivation

- High level of dynamism
  - *Macro Level*: Merge/Separate VNs
  - *Micro Level*: Add/Join/Migration of end hosts and virtual routers
- Mobility
  - Geographical
  - Logical
- Überhoming
  - Simultaneously connect to multiple InPs and VNs

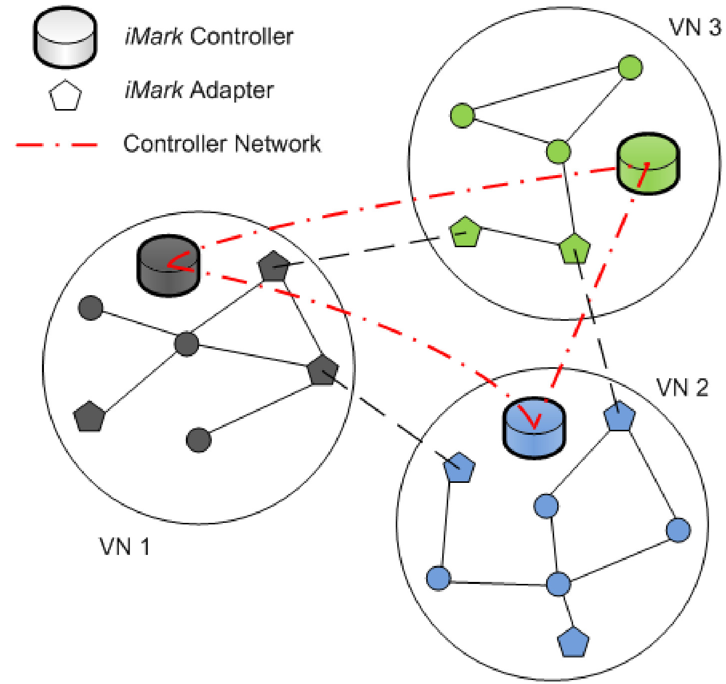
Design Principles

1. Separation of Identity and Location
   - Inherent support for mobility and Überhoming
2. Local Autonomy
   - Flexibility of naming and addressing in different VNs
   - Defined interfaces and mechanisms for cooperation
3. Global Identifier Space
   - Local identifiers have no end-to-end significance

iMark Overview

- Concepts
  1. Identifier Spaces
  2. Mappings

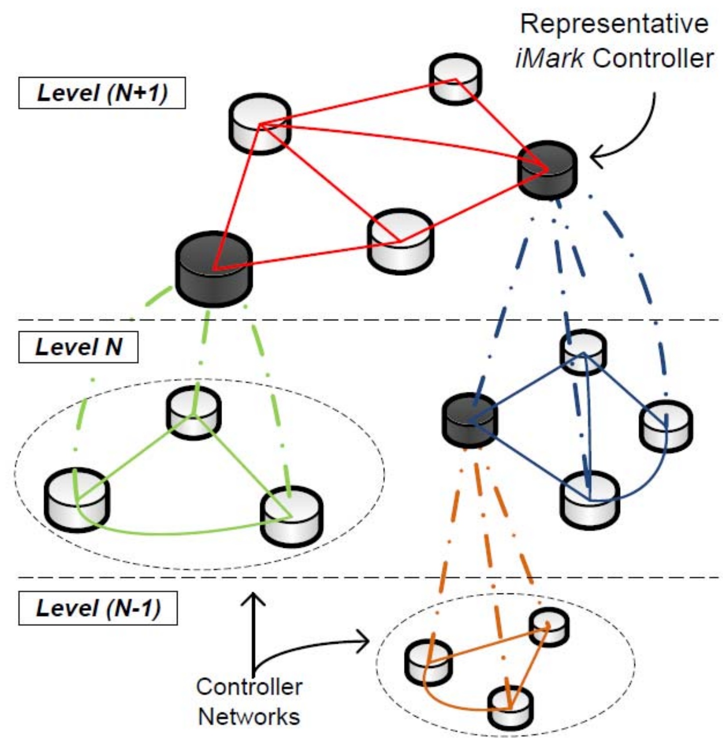
- Components
  1. Controllers
  2. Adapters



Operations: Macro Level

- Federation
  - Multiple VNs create common administrative domain
  - Controller network

- Hierarchy
  - Aggregation of mappings in representative controllers
  - Balanced and unbalanced

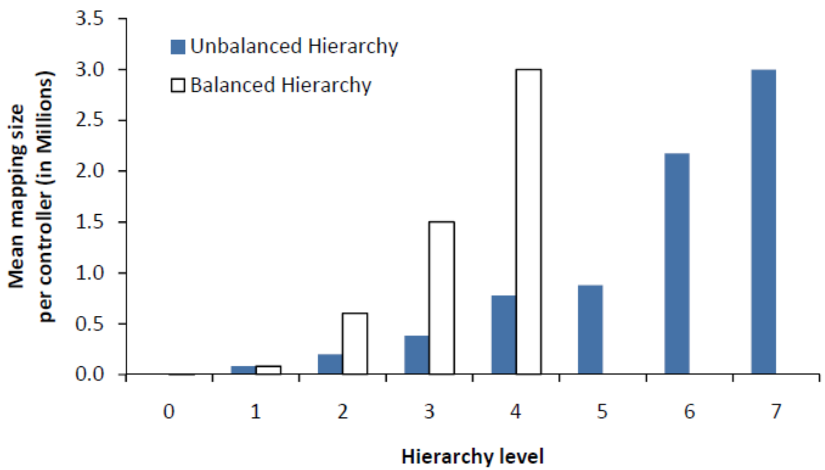


Operations: Micro Level

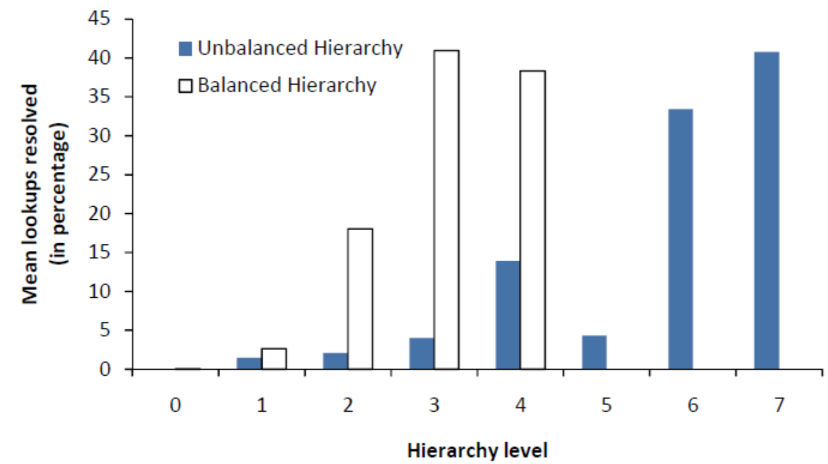
- Join
  - Add mappings
- Lookup and Connection Setup
  - State setup in the network
- Leave
  - Remove mappings
- Mobility
  - Soft handoff

Evaluation

Mean Mapping Size Per Controller



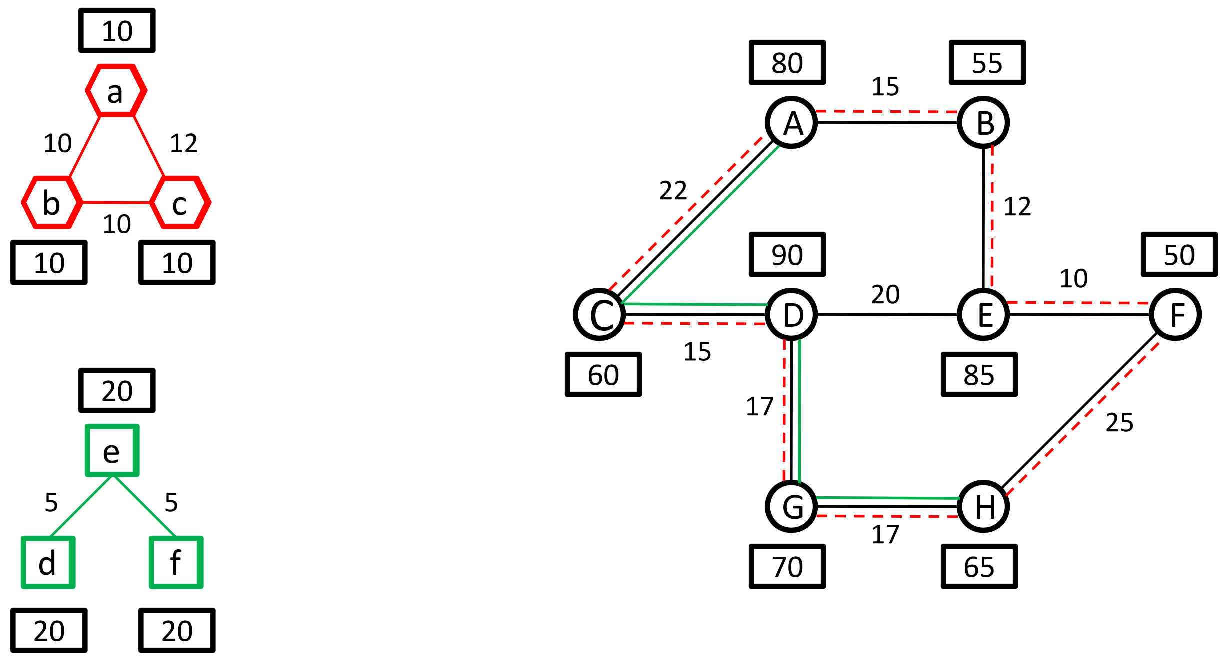
Mean Lookups Resolved



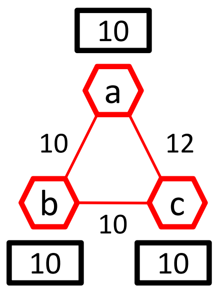
Intra-domain Resource Allocation through Virtual Network Embedding

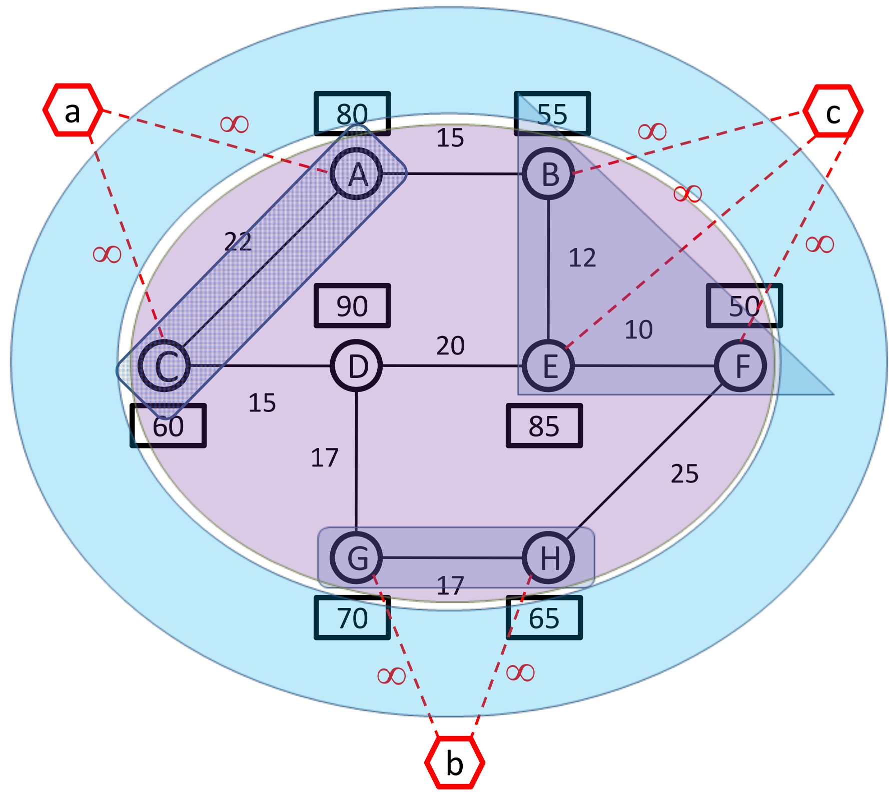
VINEYARD

Virtual Network Embedding



Substrate Graph Augmentation





D-ViNE and 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-node with the highest probability
    – R-ViNE: Select a meta-node randomly with the calculated probability

NODE MAPPING

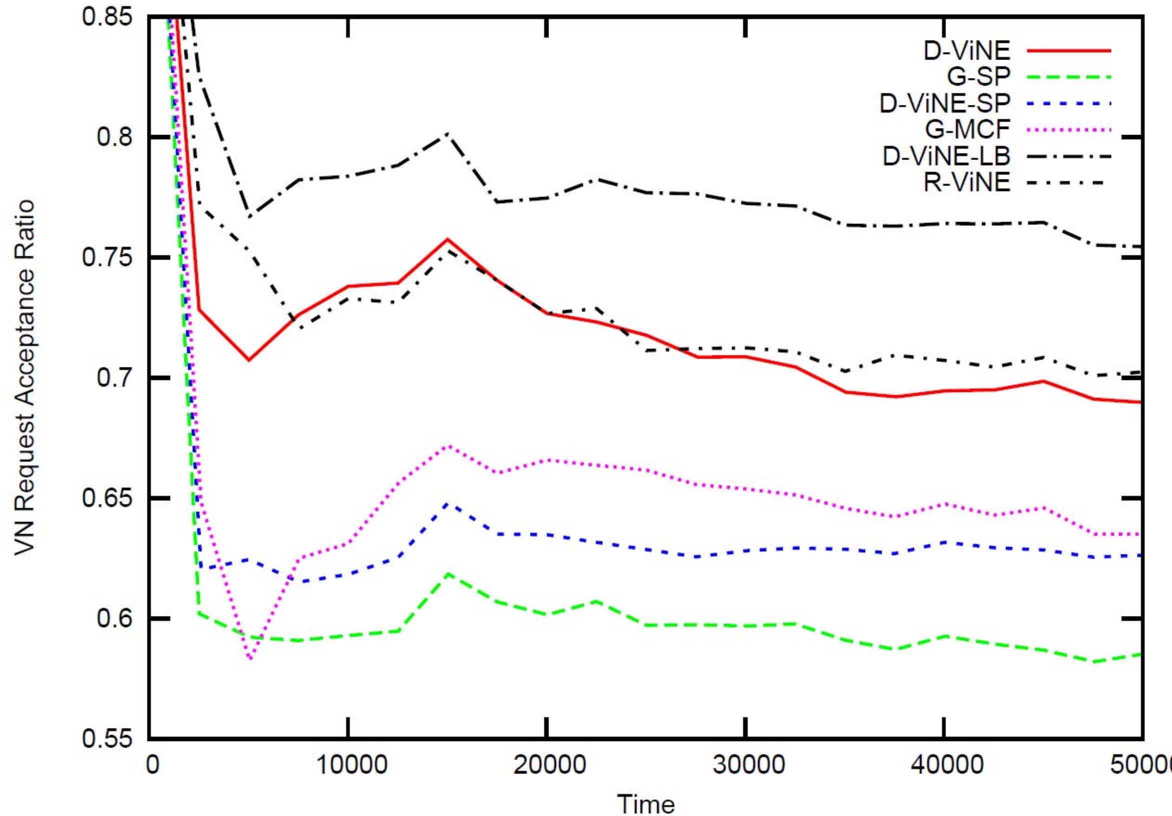
– Use MCF to map virtual edges

LINK MAPPING

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

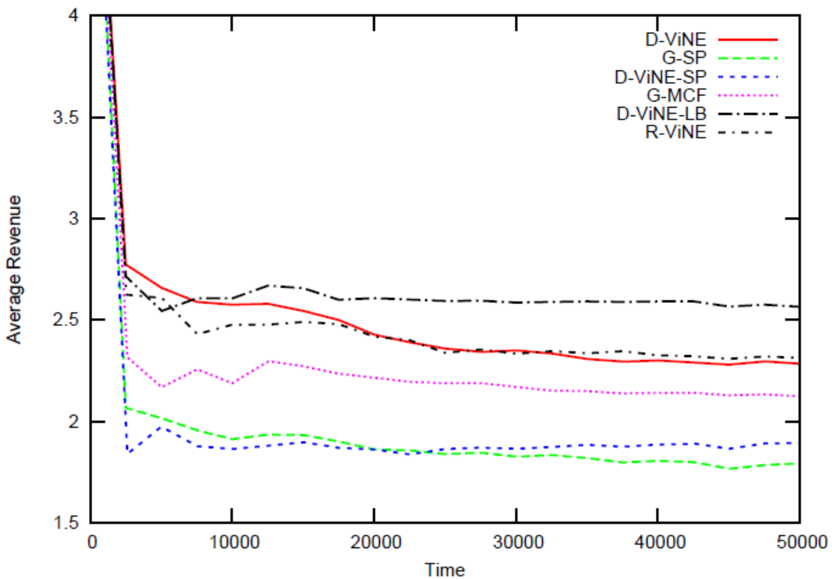
FINALIZATION

Acceptance Ratio

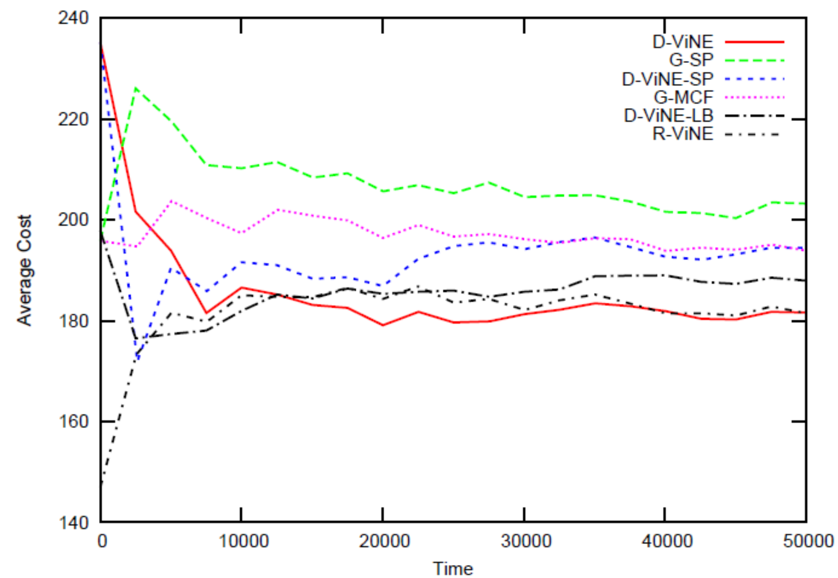


Revenue Vs Cost

Revenue

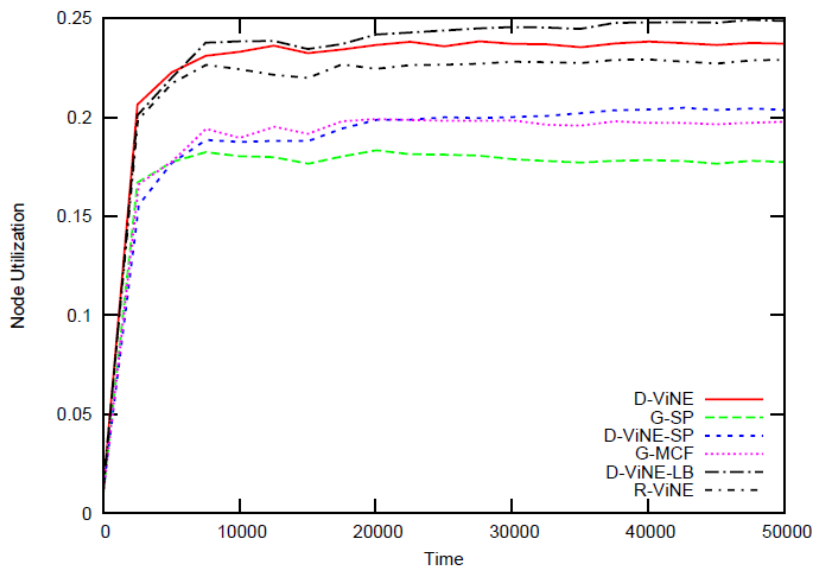


Cost

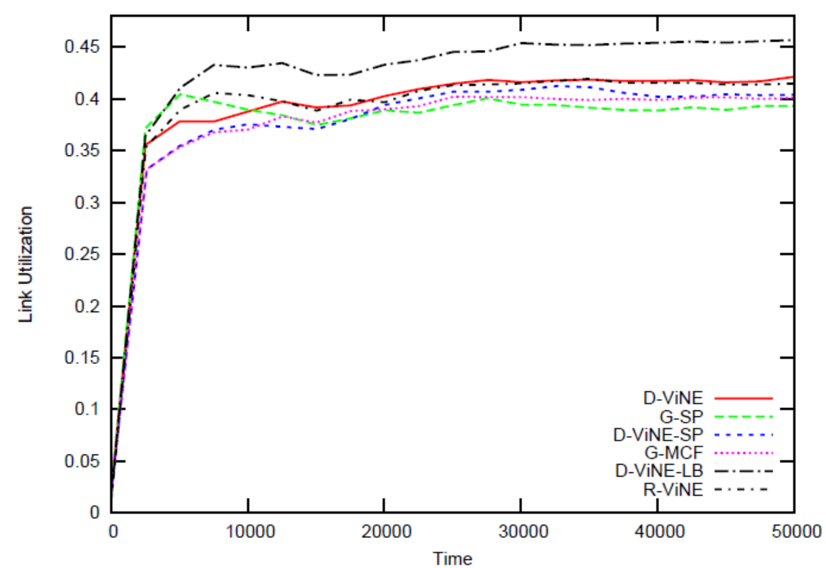


Resource Utilization

Node Utilization



Link Utilization



What have we done? What will we do?

SUMMARY

Contributions

- Survey of Network Virtualization (Submitted + TechReport)
  - Historical perspective
  - Categorization of existing projects
  - Enumeration of open problems
- Identity Management Framework (IM'09)
  - Interoperability with flexibility to support mobility and Überhoming
- Virtual Network Embedding Algorithms (INFOCOM'09)
  - Better embedding quality
  - Mathematical foundation

Future Work

- iMark Prototype Development
  - Further evaluation
- Theoretical Analysis of D-ViNE and R-ViNE
  - Approximation factors
  - Economic models
- Inter-domain VN embedding

Collaborators

- Fida-E Zaheer (iMark)
- Muntasir Raihan Rahman (ViNEYard)
- Network Virtualization Project Members

Questions?

Mosharaf Chowdhury
http:// www.mosharaf.com/

BACKUP SLIDES

Related Concepts

1. Virtual Local Area Networks (VLAN)
2. Virtual Private Networks (VPN)
3. Active and Programmable Networks
4. Overlay Networks

Downsides of Overlay Networks

- Largely used as narrow fixes for specific problems
  - No holistic view
- Most overlays are designed in the application layer
  - Cannot support radically different concepts

Anderson et al.

What is a Virtual Network (VN)?

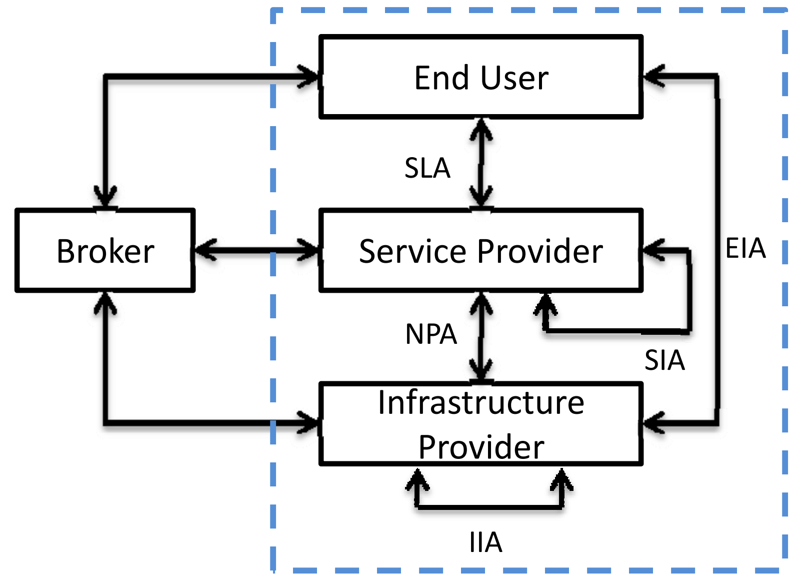
- A collection of virtual nodes and virtual links forming a virtual topology
  - Subset of physical topology
  - Basic entity of the NVE
- A virtual node is hosted on a particular physical node
  - Multiple virtual nodes can coexist
- A virtual link spans over a physical path
  - Includes a portion of the underlying physical resources

Business Model

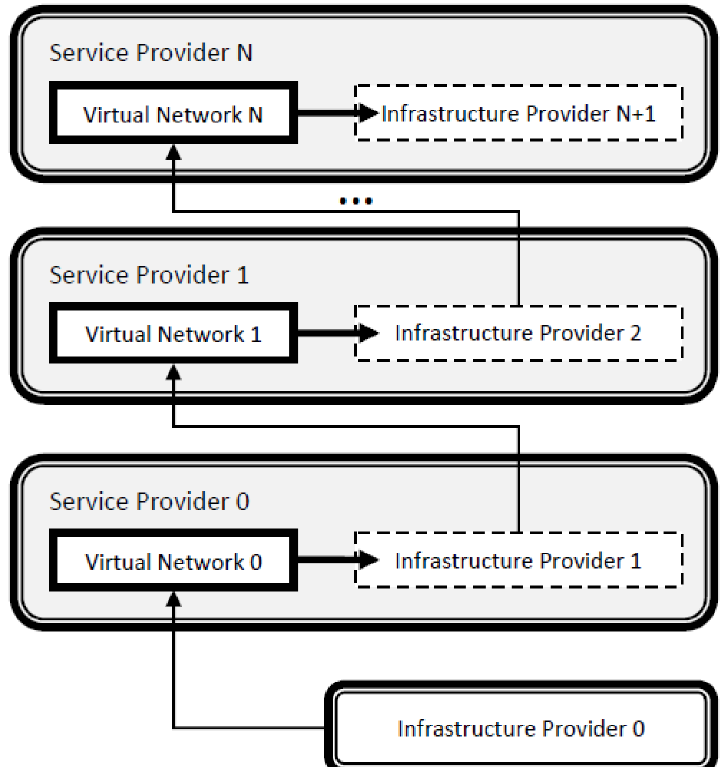
Players

- Infrastructure Providers (*InP*)
  - Manage underlying physical networks
- Service Providers (*SP*)
  - Create and manage virtual networks
  - Deploy customized end-to-end services
- End Users
  - Buy and use services from different service providers
- Brokers
  - Mediators/Arbiters

Relationships



Hierarchy of Roles



Basic Concepts

Principles

- Concurrence
- Recursion
- Inheritance
- Revisitation

Design Goals

- Flexibility
- Manageability
- Scalability
- Isolation
- Stability and Convergence
- Programmability
- Heterogeneity
- Experimental and Deployment Facility
- Legacy Support

What is Network Virtualization? (Revisited)

Network virtualization is a networking environment that allows multiple service providers to dynamically compose multiple heterogeneous virtual networks that coexist together in isolation from each other, and to deploy customized end-to-end services on-the-fly as well as manage them on those virtual networks for the end-users by effectively sharing and utilizing underlying network resources leased from multiple infrastructure providers.

Classification

- *Networking technology*
  - Targeted technology for virtualization
- *Layer of virtualization*
  - Particular layer in the network stack where virtualization is introduced
- *Architectural domain*
  - Specific problem domain that virtualization addresses
- *Level of virtualization*
  - Granularity at which virtualization is realized

Existing Projects

Project	Architectural Domain	Networking Technology	Layer of Virtualization	Level of Virtualization
VNRMS	Virtual network management	ATM/IP		Node/Link
Tempest	Enabling alternate control architectures	ATM	Link	
NetScript	Dynamic composition of services	IP	Network	Node
Genesis	Spawning virtual network architectures		Network	Node/Link

Existing Projects (Cont.)

Project	Architectural Domain	Networking Technology	Layer of Virtualization	Level of Virtualization
VNET	Virtual machine Grid computing		Link	Node
VIOLIN	Deploying on-demand value-added services on IP overlays	IP	Application	Node
X-Bone	Automating deployment of IP overlays	IP	Application	Node/Link
PlanetLab	Deploy and manage overlay-based testbeds	IP	Application	Node
UCLP	Dynamic provisioning and reconfiguration of lightpaths	SONET	Physical	Link

Existing Projects (Cont.)

Project	Architectural Domain	Networking Technology	Layer of Virtualization	Level of Virtualization
AGAVE	End-to-end QoS-aware service provisioning	IP	Network	
GENI	Creating customized virtual network testbeds	Heterogeneous		
VINI	Evaluating protocols and services in a realistic environment		Link	
CABO	Deploying value-added end-to-end services on shared infrastructure	Heterogeneous		Full

Major Ongoing Projects

Project	Originated In	Link
4WARD	Europe	http://www.4ward-project.eu/
AKARI	Japan	http://akari-project.nict.go.jp/
CABO	USA	http://www.cs.princeton.edu/~jrex/virtual.html
Clean Slate	USA	http://cleanslate.stanford.edu/
GENI	USA	http://www.geni.net/
NouVeau	Canada	http://netlab.cs.uwaterloo.ca/virtual/
PlanetLab	USA	http://www.planet-lab.org/
Trilogy	Europe	http://www.trilogy-project.org/
UCLP	Canada	http://www.uclp.ca/
VINI	USA	http://www.vini-veritas.net/

Entities and Identifier Spaces

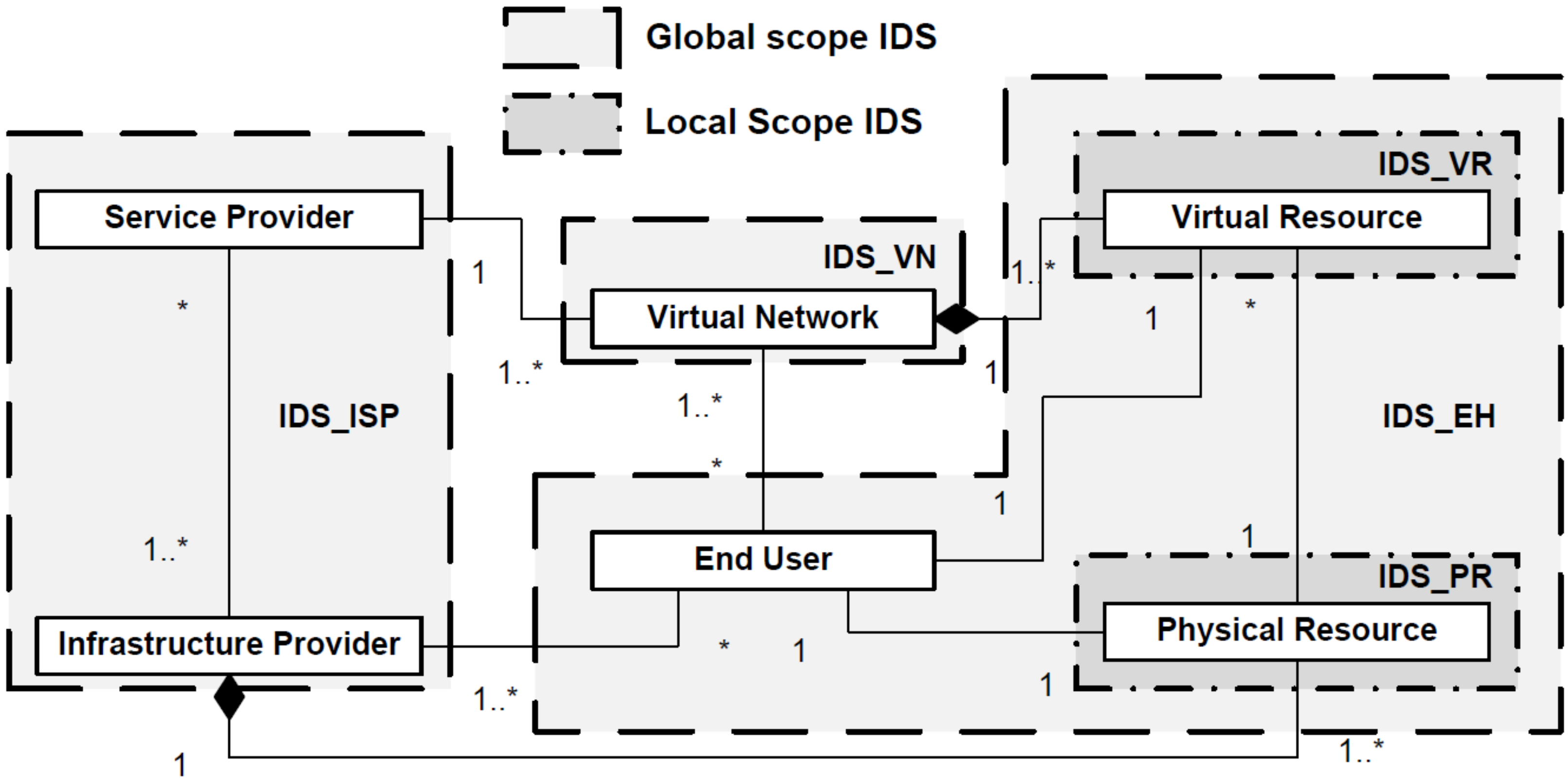
Entities

1. Service Provider
2. Virtual Network
3. Virtual Resource
4. Infrastructure Provider / Physical Network
5. Physical Resource
6. End User

Identifier Spaces

1. IDS_ISP
2. IDS_VN
3. IDS_VR
4. IDS_PR
5. IDS_EH

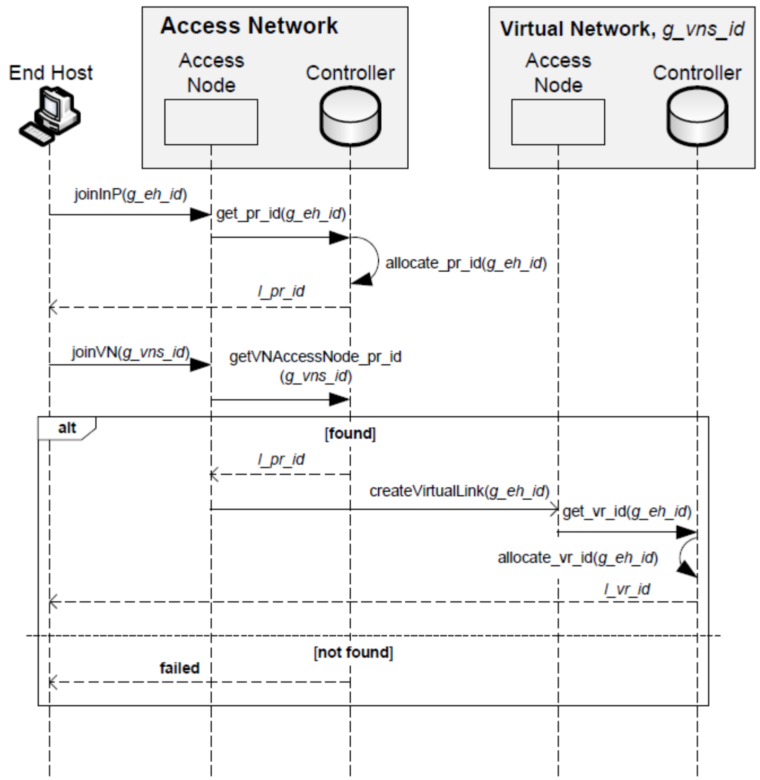
Relationships between Entities



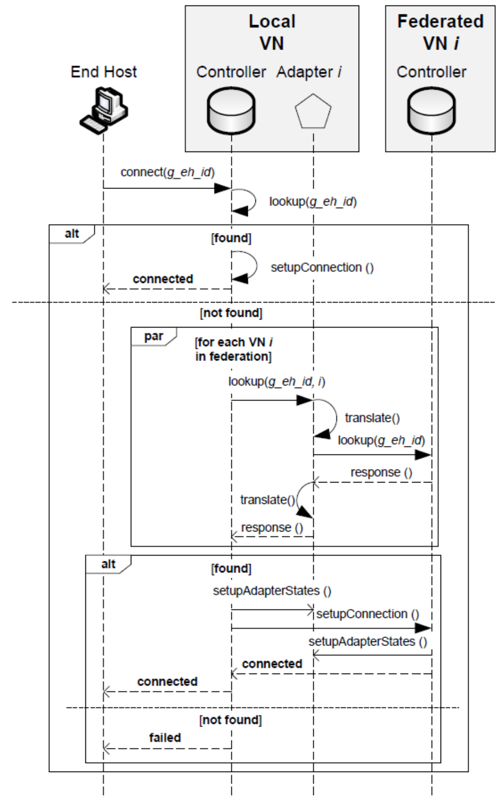
Mappings between Different Identifiers

$(\langle from\_id \leftrightarrow to\_id \rangle)$	Purpose
$\langle g\_eh\_id \leftrightarrow l\_vr\_id \rangle$	Identifies any resource within a virtual network and vice versa.
$\langle g\_eh\_id \leftrightarrow l\_pr\_id \rangle$	Identifies any resource within a physical network and vice versa.
$\langle g\_eh\_id \rightarrow g\_vn\_id \rangle$	Stores the virtual network an end host is connected to.
$\langle l\_vr\_id \rightarrow l\_pr\_id \rangle$	Finds the local identifier of the physical host of a virtual resource within a physical network.
$\langle g\_vn\_id \rightarrow \{l\_pr\_id\} \rangle$	Gets the local identifiers of the access nodes of a virtual network inside a physical network.
$\langle g\_vn\_id \rightarrow g\_isp\_id \rangle$	Finds the owner SP of a virtual network.
$\langle g\_vn\_id \rightarrow \{g\_isp\_id\} \rangle$	Obtains the set of InPs that host the virtual network in the underlying network.

Sequence Diagram: Join



Sequence Diagram: Lookup



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:

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

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

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

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

- Domain Constraints:

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

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

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. $1 \le \alpha_{uv} \le R_E(uv)$ and $1 \le \beta_w \le R_N(w)$ 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

Algorithm 5.3 D-ViNE: Deterministic Rounding-based Virtual Network Embedding Algorithm

procedure D-VINE($G^V = (N^V, E^V)$)
    Create augmented substrate graph $G^{S'} = \left(N^{S'}, E^{S'}\right)$
    Solve Program 5.2
    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 $z_{max} = \arg\max_{z \in \Omega(n)} \{p_z | \varphi(z) = 0\}$ ▷ break ties arbitrarily
        set $\mathcal{M}_N(n) \leftarrow z_{max}$ ▷ Map n to z_{max}
        $\varphi(z_{max}) \leftarrow 1$
    end for
    Solve MCF to map virtual edges.
    Update residual capacities of the network resources.
end procedure

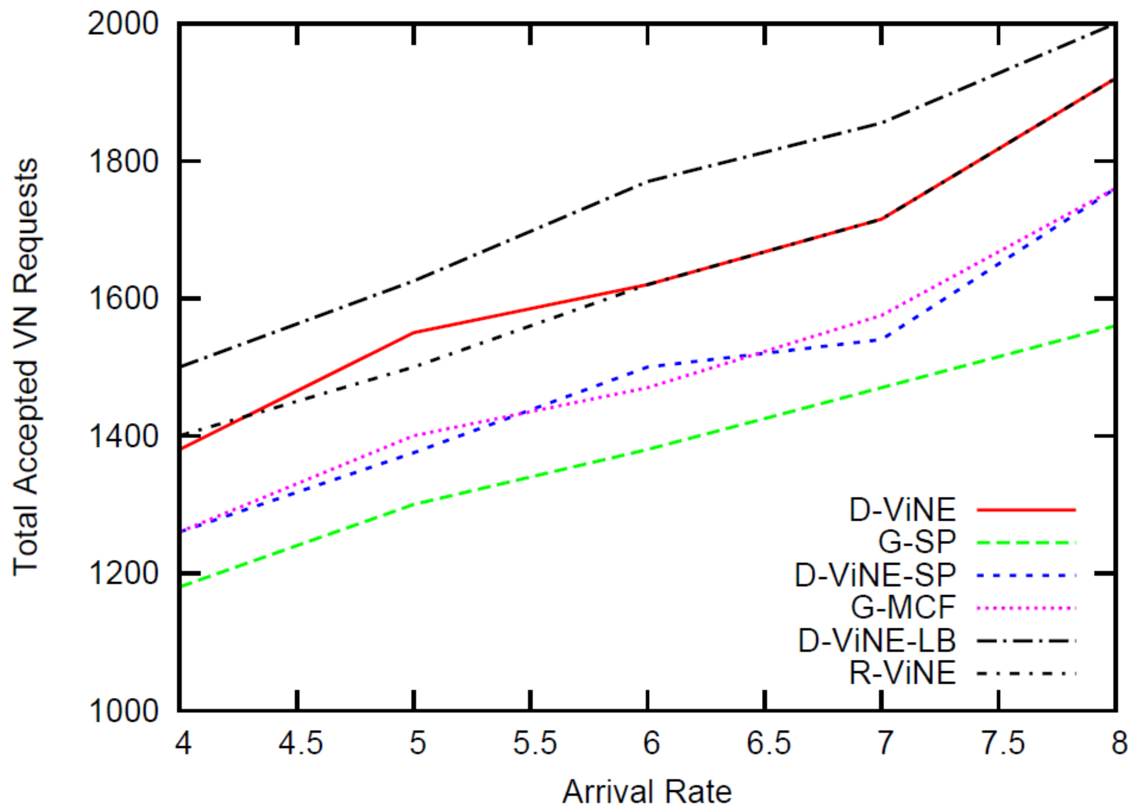
R-ViNE

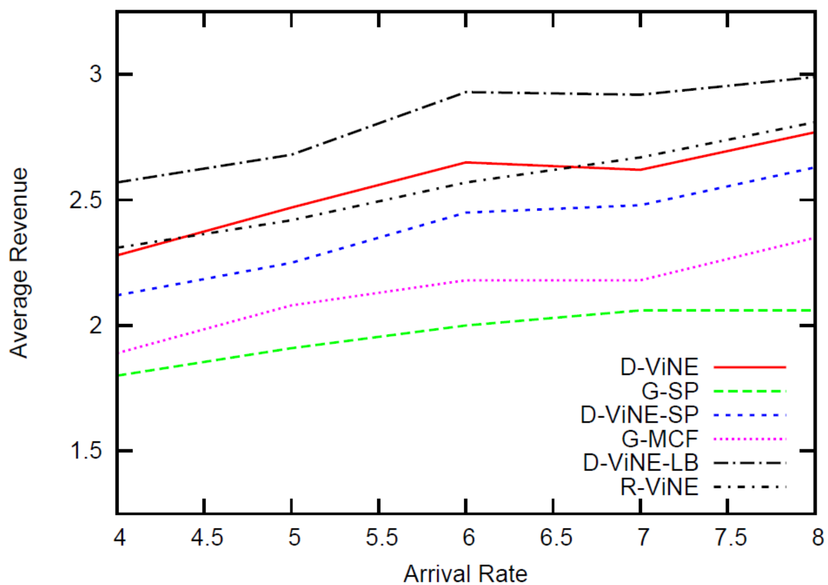
Algorithm 5.4 R-ViNE: Randomized Rounding-based Virtual Network Embedding Algorithm

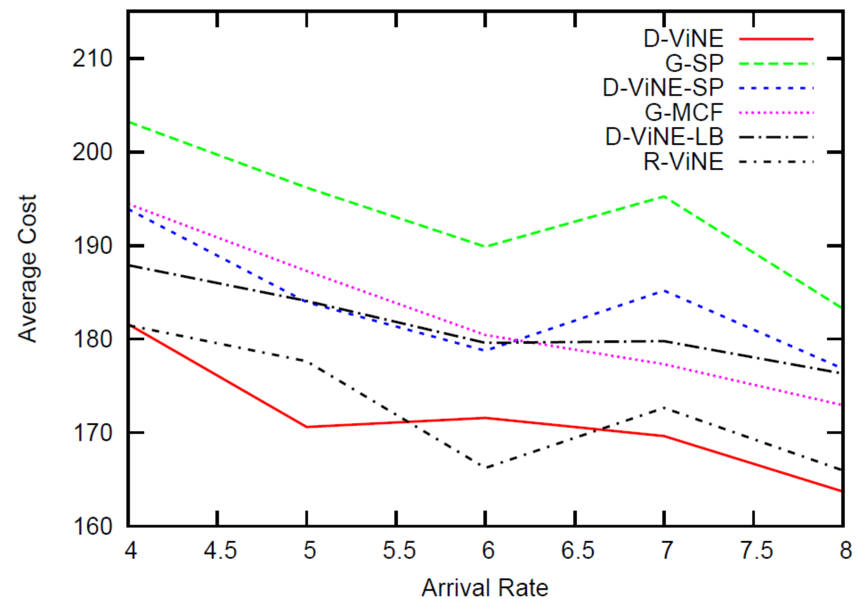
procedure R-VINE($G^V = (N^V, E^V)$)
  Create augmented substrate graph $G^{S'} = \left(N^{S'}, E^{S'}\right)$
  Solve Program 5.2
  for all $n^S \in N^S$ **do**
    $\varphi(n^S) \leftarrow 0$
  end for
  for all $n \in N^V$ **do**
    if $\Omega(n) \cap \{n^S \in N^S | \varphi(n^S) = 1\} = \emptyset$ **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
    $p_{sum} \leftarrow \sum_{z \in \Omega(n)} p_z$
    for all $z \in \Omega(n)$ **do**
      $p_z \leftarrow p_z / p_{sum}$
    end for
    set $\mathcal{M}_N(n) \leftarrow z$ with probability p_z ▷ $\sum_{z \in \Omega(n)} p_z = 1$
    $\varphi(z) \leftarrow 1$ with probability p_z
  end for
  solve MCF to map virtual edges.
  Update residual capacities of the network resources.
end procedure

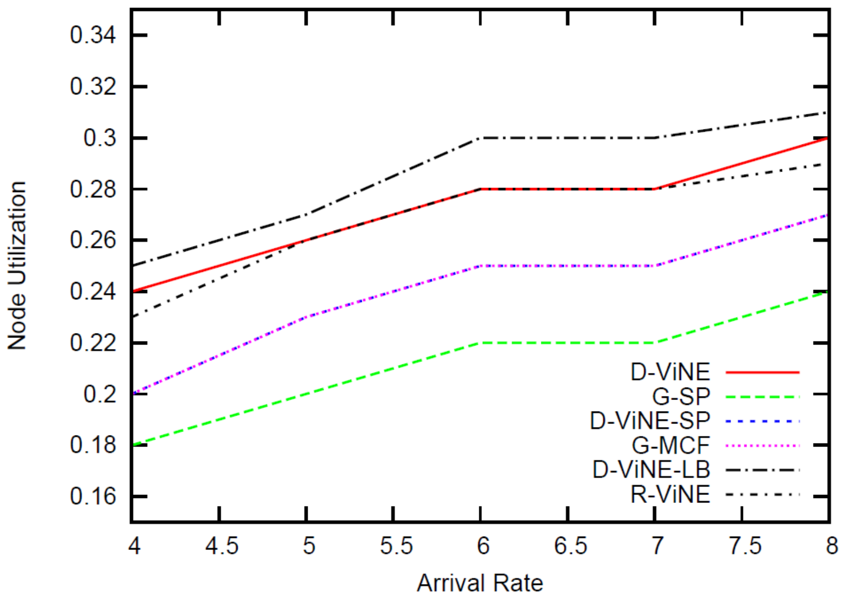
Summary of Compared Algorithms

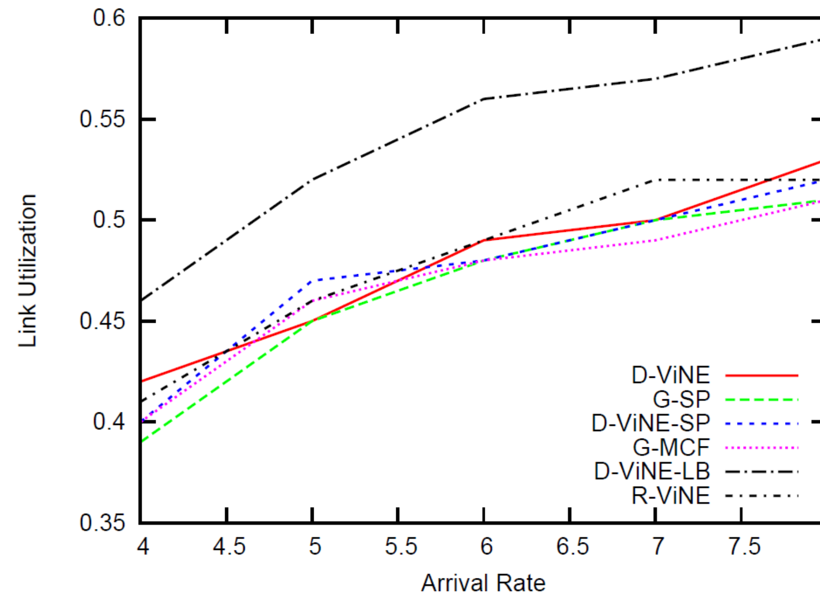
Notation	Algorithm Description
D-ViNE	Deterministic Node Mapping with Splittable Link Mapping using MCF
R-ViNE	Randomized Node Mapping with Splittable Link Mapping using MCF
G-SP [119]	Greedy Node Mapping with Shortest Path Based Link Mapping
G-MCF [117]	Greedy Node Mapping with Splittable Link Mapping 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 $\alpha_{uv} = \beta_w = 1, \forall u, v, w \in N^S$


2000
1800
1600
1400
1200
1000
4
4.5
5
5.5
6
6.5
7
7.5
8
Total Accepted VN Requests
Arrival Rate
D-ViNE
G-SP
D-ViNE-SP
G-MCF
D-ViNE-LB
R-ViNE


Average Revenue
Arrival Rate
D-ViNE
G-SP
D-ViNE-SP
G-MCF
D-ViNE-LB
R-ViNE
1.5
2
2.5
3
4
4.5
5
5.5
6
6.5
7
7.5
8


Average Cost
Arrival Rate
D-ViNE
G-SP
D-ViNE-SP
G-MCF
D-ViNE-LB
R-ViNE
160
170
180
190
200
210
4
4.5
5
5.5
6
6.5
7
7.5
8


Node Utilization
0.34
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0.16
4
4.5
5
5.5
6
6.5
7
7.5
8
Arrival Rate
D-ViNE
G-SP
D-ViNE-SP
G-MCF
D-ViNE-LB
R-ViNE


Link Utilization
0.6
0.55
0.5
0.45
0.4
0.35
4
4.5
5
5.5
6
6.5
7
7.5
8
Arrival Rate
D-ViNE
G-SP
D-ViNE-SP
G-MCF
D-ViNE-LB
R-ViNE