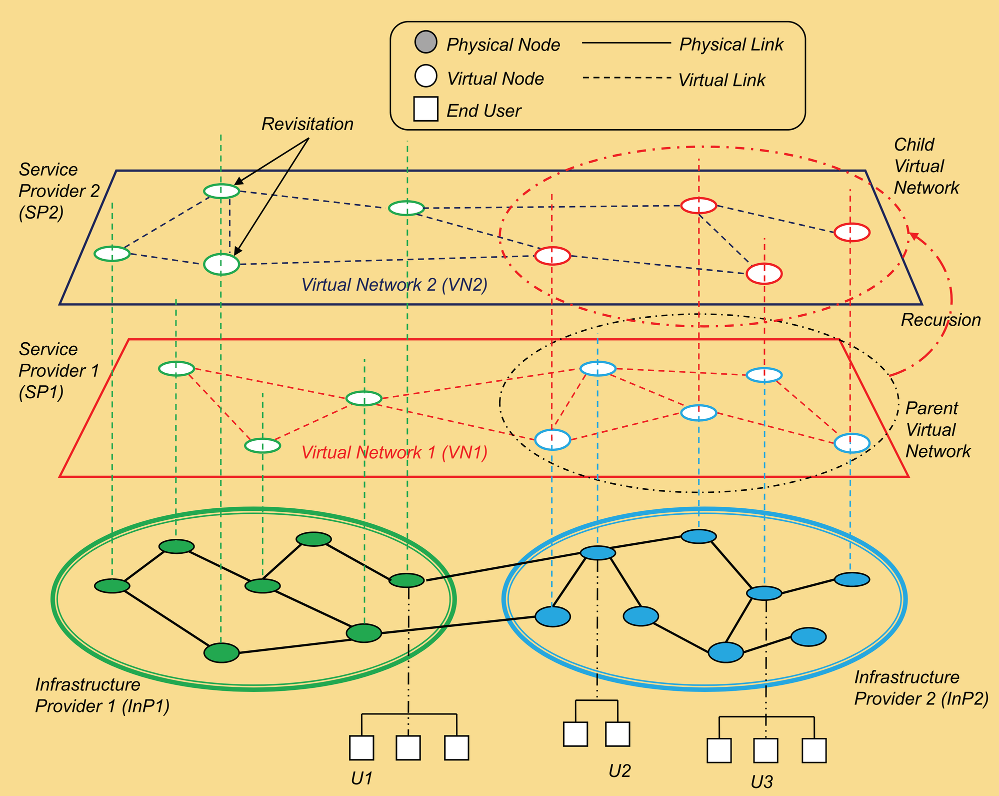
Identity Management and Resource Allocation in the Network Virtualization Environment

N.M. Mosharaf Kabir Chowdhury, Muntasir Raihan Rahman, and Fida-E Zaheer
David R. Cheriton School of Computer Science, University of Waterloo

Why Virtualize the Network?

Due to the existence of multiple stakeholders with conflicting goals and policies, alterations to the existing Internet are now limited to simple incremental updates. To fend off this ossification, *network virtualization* has been propounded as a diversifying attribute of the future inter-networking paradigm.

Instead of creating yet another *one-size-fits-all* architecture, network virtualization promotes an environment that allows coexistence of multiple heterogeneous network architectures on a shared underlying physical network by providing transparent abstraction of network resources.



Network virtualization environment

What is Network Virtualization Environment?

A network virtualization environment (NVE) is characterized by the decoupling of the roles of the traditional Internet Service Providers (ISPs) into two independent entities: *infrastructure providers (InPs)*, who manage the physical infrastructure; and *service providers (SPs)*, who create *virtual networks (VNs)* by aggregating resources from multiple InPs to offer heterogeneous end-to-end services. It allows:

- Proliferation of novel heterogeneous network architectures
- Instantiation of end-to-end VNs without global coordination
- Cost and energy efficiency through resource sharing
- Enhanced security and isolation between cohabiting VNs

References

1. N. M. Mosharaf Kabir Chowdhury, Raouf Boutaba, "Network Virtualization: The Past, The Present, and The Future", IEEE Communications Magazine, July, 2009.
2. N.M. Mosharaf Kabir Chowdhury, Fida-E Zaheer, Raouf Boutaba, "iMark: An Identity Management Framework for Network Virtualization Environment", IEEE/IFIP IM, June, 2009.
3. N. M. Mosharaf Kabir Chowdhury, Muntasir Raihan Rahman, Raouf Boutaba, "Virtual Network Embedding with Coordinated Node and Link Mapping", IEEE INFOCOM, April, 2009.

Identity Management Framework (iMark)

Local identifiers have little end-to-end significance in an NVE due to the heterogeneity of naming and addressing mechanisms in different VNs. An *identity management framework* provides necessary mechanisms, functionalities, and components to uniquely identify and locate (mobile) end hosts irrespective of their physical and logical locations through interoperability between heterogeneous identifier spaces.

Design Principles

iMark design principles aim for maximizing local autonomy with minimal global functionalities:

1. *Separation of Identity and Location:* iMark separates the identity of an entity from its location to provide inherent support for end host mobility.
2. *Local Autonomy:* No restriction imposed on individual physical or virtual networks.
3. *Global Identifier Space:* One globally agreed upon identifier space to connect all heterogeneous namespaces.

Architectural Overview

Components

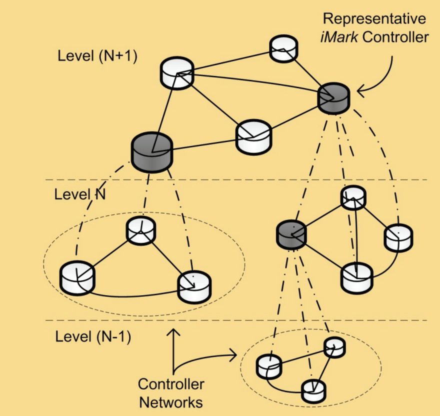
- *Controllers:* A controller is a logical entity in a VN that provides traditional control functionalities and additional network specific services. A common control space, known as the *controller network*, connects controllers of multiple VNs to support *federations*. Multiple federations and VNs are combined to create logical hierarchies.
- *Adapters:* Adapters are special entities that act as gateways between two adjoining VNs.

Concepts

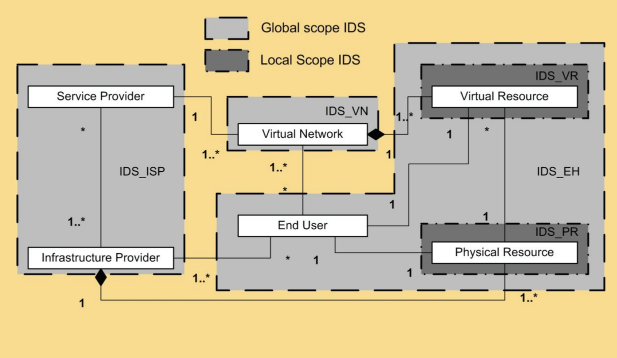
- *Entities:* iMark defines six major entities each with unique functionalities and requirements.
- *Identifier Spaces:* Identifier spaces (IDSes) define the relationships between the entities.
- *Mappings:* Finally, mappings between different IDSes (stored in Controllers) ensure locating and routing to end hosts' current locations based on their global identifiers.

IMark Operations

Primary iMark operations include *joining* of end hosts to a network that adds new mappings to the Controllers and explicit *lookups* for connection setup across heterogeneous identifier spaces.



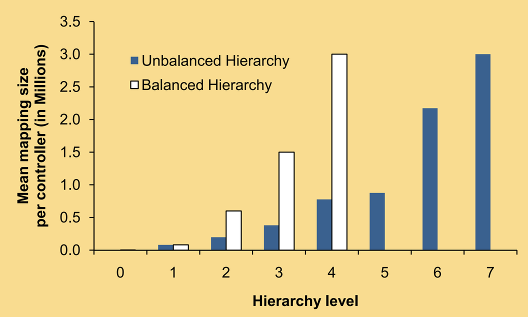
Federation and hierarchy in iMark



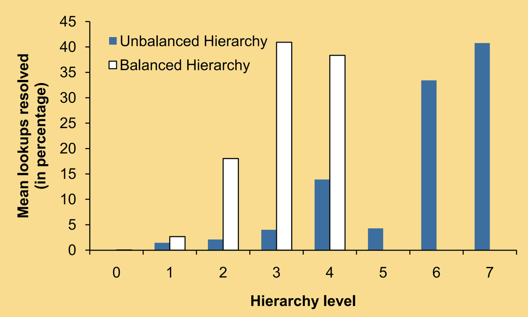
Entities and identifier spaces

Performance Evaluation

iMark faces performance challenges from two main sources: size of the mappings stored at different controllers, and lookup frequency at different levels of the controller.



Mean mapping size per Controller



Mean lookups resolved at different levels of iMark hierarchy

Virtual Network Embedding

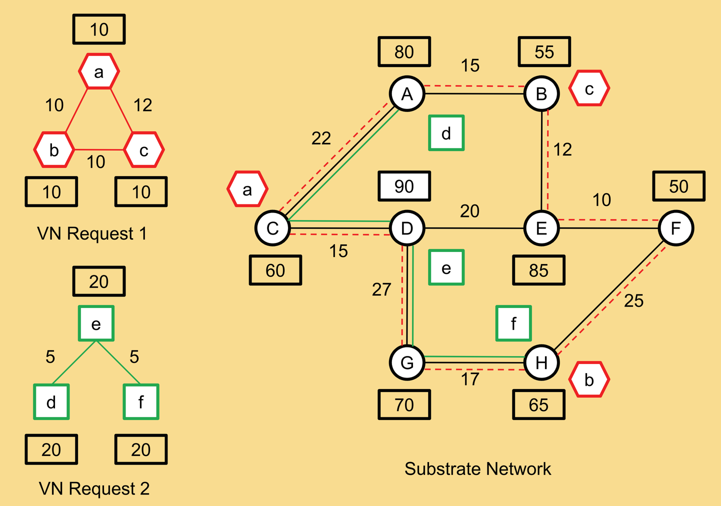
Problem Definition

Given a weighted substrate network $G^S = (N^S, E^S)$ and online VN requests $G^V = (N^V, E^V)$ with requirements and constraints on virtual nodes and virtual links, the VN embedding problem deals with assigning virtual nodes and links onto substrate nodes and paths respectively, by allocating substrate network resources (e.g., CPU, bandwidth).

Objectives

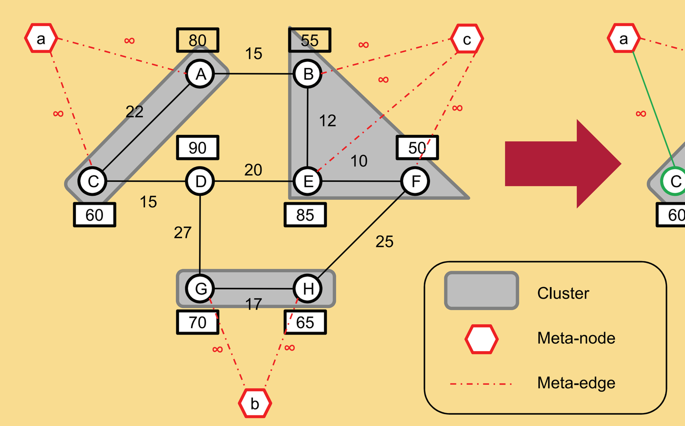
Over time,

- *Maximize*
  - Acceptance Ratio: Percentage of accepted VN requests
  - Average Revenue: Calculated based on resources *requested* for a VN request
- *Minimize*
  - Average Cost: Calculated based on substrate network resources *allocated* for embedding a VN request

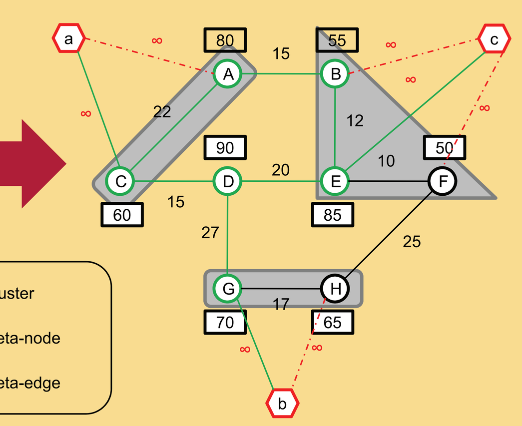


Substrate Graph Augmentation

For each VN request, the substrate graph is augmented by considering each virtual node as a *meta-node* and connecting each meta-node to a *cluster* of nodes using *meta-links*. Each meta-link has infinite capacity, and a cluster is selected based on the location constraint of a virtual node.



Augmented substrate graph for VN request 1



Combined node and link mapping in the MIP solution

MIP Formulation and LP Relaxation

By considering each virtual link as a commodity and by assigning appropriate *binary* constraints on meta-links, the VN embedding problem can now be represented as an extended mixed-integer multi-commodity flow (MCF) problem.

Binary Constraints on Meta-links

- Exactly one substrate node is selected for each meta-node
- At most one meta-node is mapped onto a substrate node

Solving the mixed-integer program (MIP) will simultaneously embed virtual nodes and virtual links. However, solving an MIP is known to be **NP-hard**. A linear program (LP) is generated by relaxing the binary constraints on the meta-links.

ViNEYard Algorithms

Deterministic and randomized rounding techniques are employed to select meta-links resulting in two algorithms (D-ViNE and R-ViNE respectively), known together as ViNEYard algorithms.

Pseudocode

1. *Augment the substrate graph to form the MIP*
2. *Solve the relaxed LP*
3. *For each virtual node, calculate the probability for each meta-link to be selected for a particular virtual/meta-node*
   - ***D-ViNE**: Select the meta-link with the highest probability*
   - ***R-ViNE**: Select a meta-link randomly with some probability*
4. *Use MCF to embed the virtual links*
5. *Update residual capacities of substrate nodes and links*

Performance Evaluation