

Network Virtualization: A Tutorial

George N. Rouskas

Department of Computer Science

North Carolina State University

<http://rouskas.csc.ncsu.edu/>

Outline

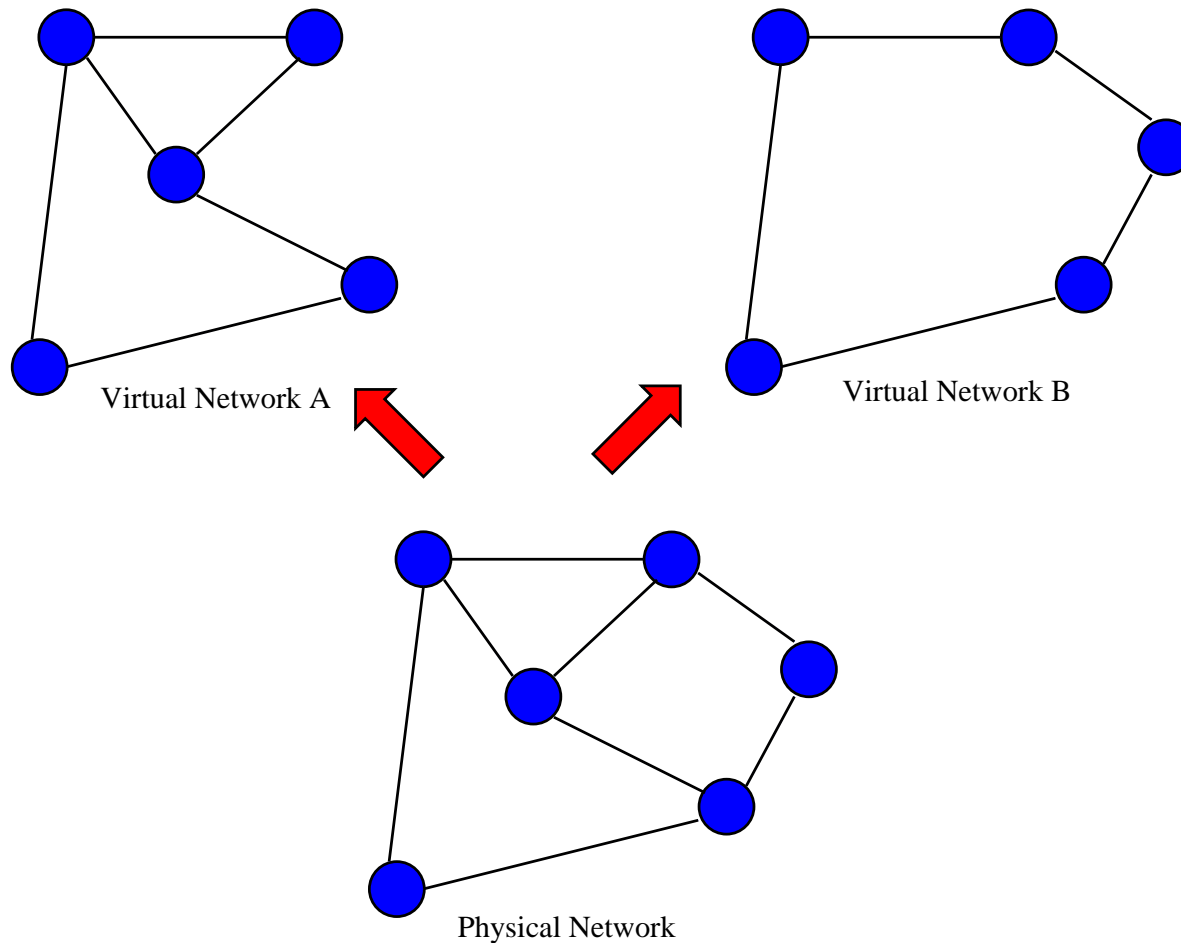
- Introduction and Historical Perspective
- Reference Model, Architectural Principles and Objectives
- Network Virtualization Projects
- Challenges and Future Directions

Virtualization As Resource Abstraction

- **Virtualization**: transparent **abstraction** of the physical computing platform and resources that supports **multiple** logical views of their properties
- Virtual anything
 - virtual memory
 - OS platform → virtual machines
 - storage → virtual SAN
 - computing → virtual data centers

Network Virtualization

- Single physical network **appears as** multiple logical networks



Network Virtualization Environment

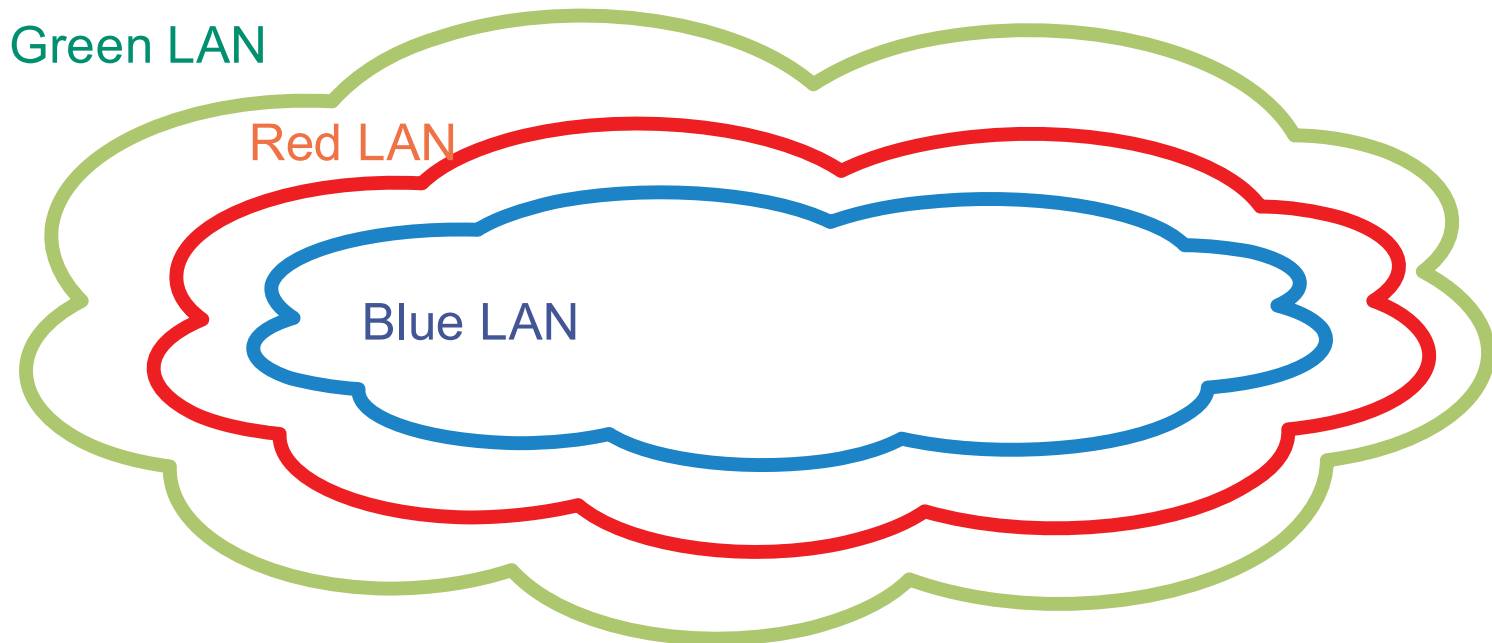
- Allows the creation of multiple virtual networks on same substrate
- Each virtual network:
 - is a collection of virtual nodes and virtual links
 - is a subset of underlying physical network resources
 - co-exists with, but is isolated from, other virtual networks

Historical Perspective

- Virtual LAN (VLAN)
- Virtual service network (VSN)
- Virtual private network (VPN)
- Active and programmable networks
- Overlay networks

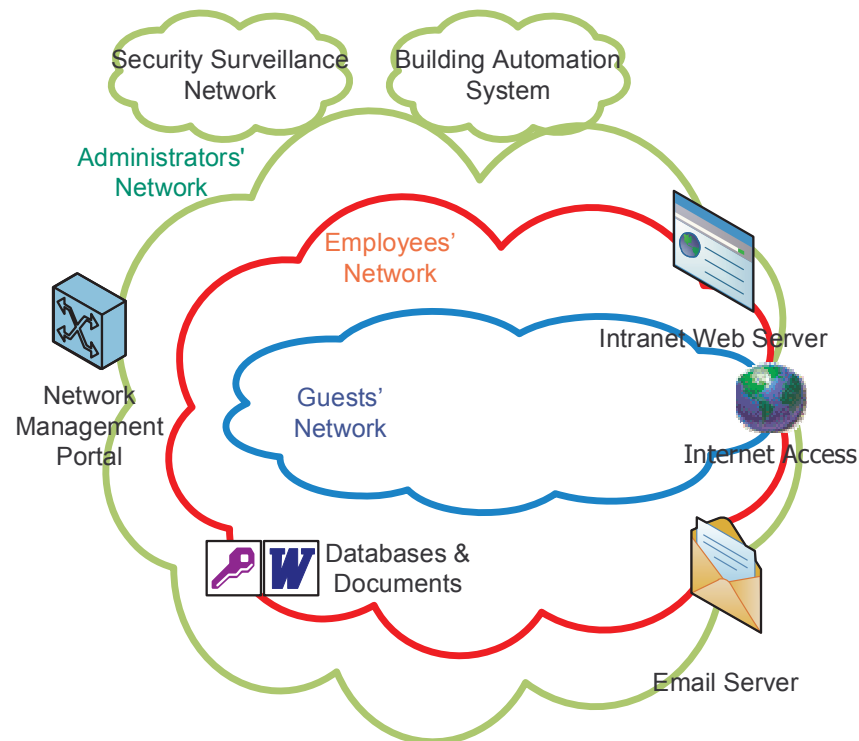
Virtual LANs

- L2 constructs, share same physical LAN infrastructure
- Groups of hosts with common interests, regardless of connectivity
- Segmented within boundary of broadcast domains → **frame coloring**
- High levels of trust, security, isolation, easy to configure



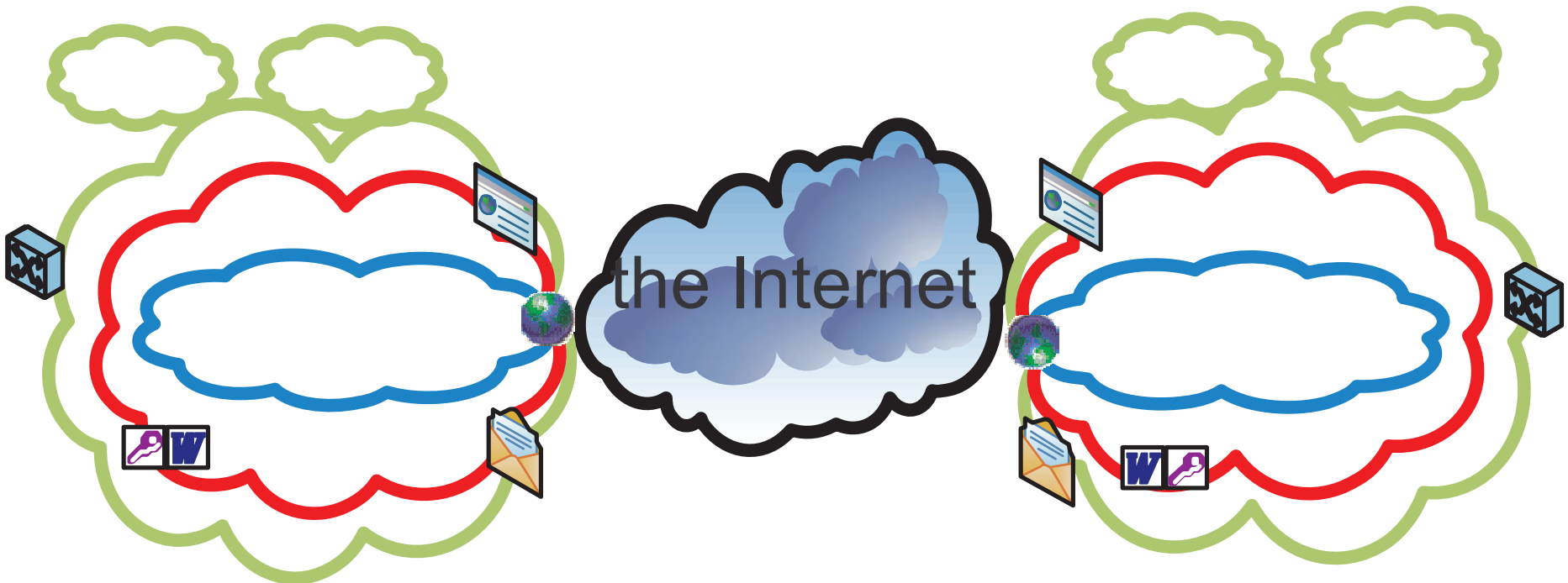
Virtual Service Networks

- Generalization of VLAN concept to broader network
- Define multiple network instances that
 - share the physical resources
 - are properly segmented (functionality, access permissions, etc.)



Virtual Private Networks

- Connect multiple sites using **tunnels** over public network
- L3/L2/L1 technologies
- Intranet vs. extranet
- Limited customization

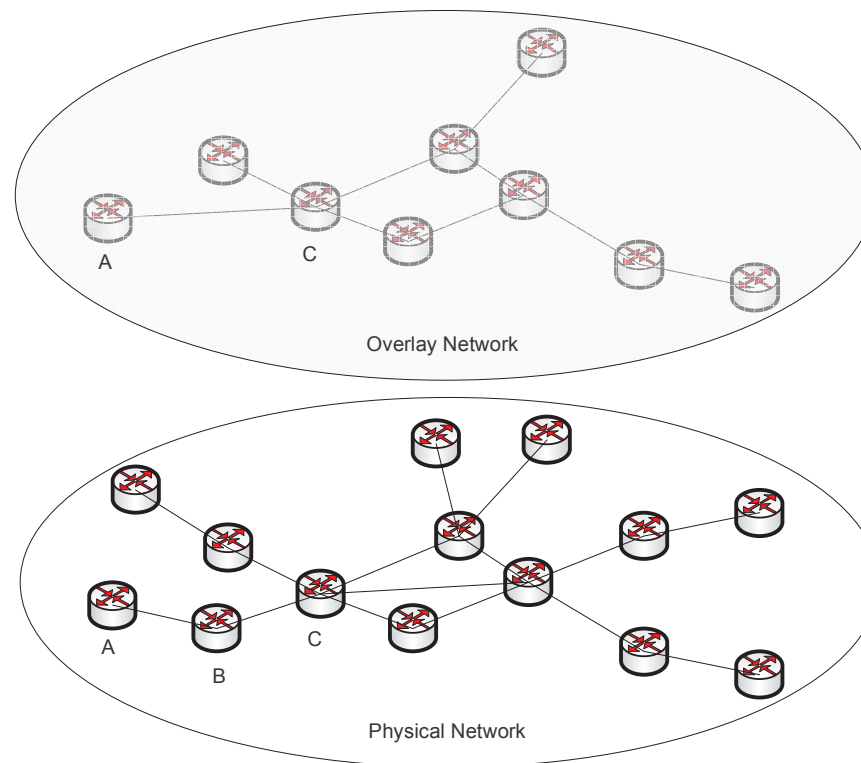


Active and Programmable Networks

- Support co-existing networks through programmability
- Customized network functionality through
 - programmable interfaces to enable access to switches/routers
 - active code
 - call functions already installed
 - allow user to execute own code at switches/routers
- Operators must open networks to third parties → not used

Overlay Networks

- Application layer virtual networks
- Built upon existing network using tunneling/encapsulation
- Implement new services without changes in infrastructure
- Application-specific, not flexible enough



Motivation (1)

- The Internet is **ossified**:
 - significant innovation at application and link layers
 - only incremental changes to core protocols
 - half-layer solutions (MPLS, IPSec)
 - adapt to new contexts (TCP over wireless)
 - application overlays (CDN, ALM, OverQoS, SOS, P2P, . . .)

Motivation (2)

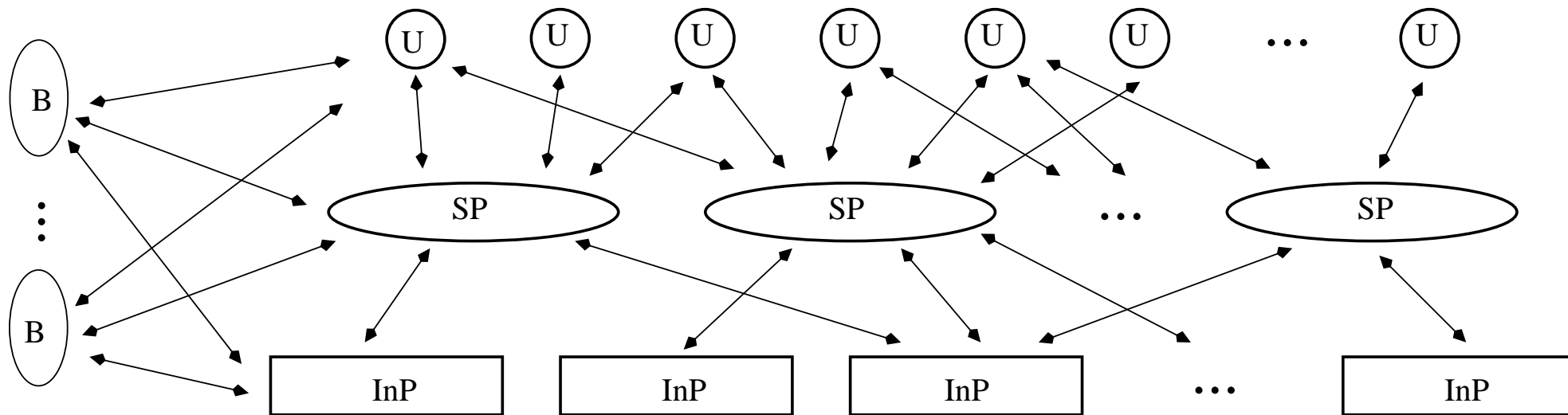
- **Clean-slate** architecture impossible to deploy
 - multiple stakeholders
 - conflicting goals/policies → difficult to reach agreement
 - almost no economic incentives to introduce changes
 - defining **one-size-fits-all** architecture a challenge
 - depends on focus and context → 4 distinct FIA projects
 - future needs unknown, difficult to predict
 - must accommodate installed base

Motivation (3)

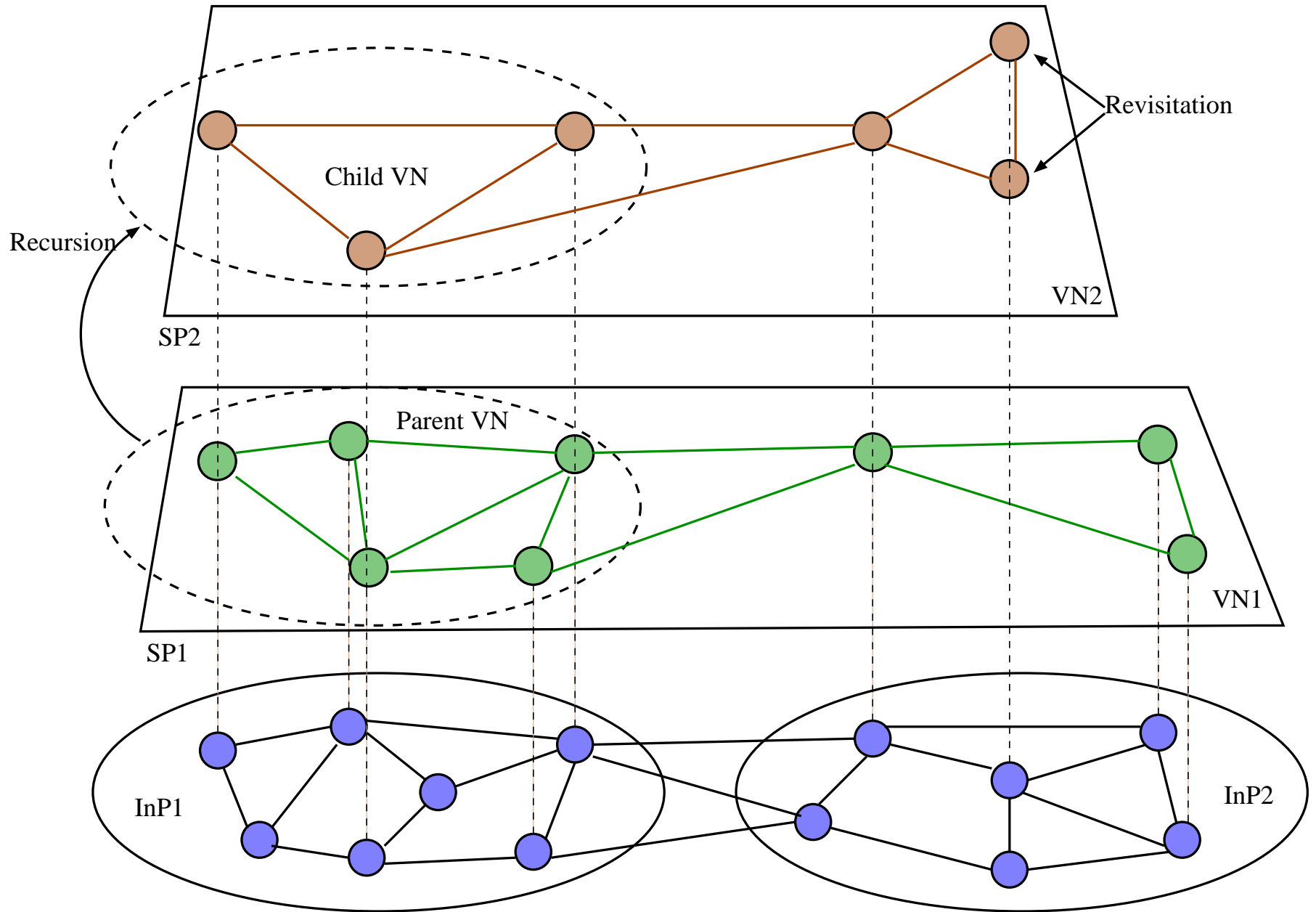
- Network virtualization offers **all-sizes-fit-into-one** solution
- Open and expandable model
 - multiple heterogeneous architectures on shared physical substrate
 - promotes innovation and customized services/applications
- Testbed for deployment/evaluation of new network architectures and protocols

Reference Model

- **Infrastructure providers:** manage physical networks
- **Service providers:**
 - create and manage virtual networks
 - design and deploy end-to-end services
- **Users:** select from competing services offered by providers
- **Brokers:** aggregate offers, match requirements to services/resources



Architectural Principles (1)



Architectural Principles (2)

- **Concurrency:**
 - multiple service providers compete
 - multiple virtual networks coexist → diversity
- **Nesting:** virtual network hierarchy
 - child VNs derived from parent VNs
- **Inheritance:** child VN inherits architectural attributes
 - creation of value-added services
 - VN economics
- **Revisitation:** single physical node hosts multiple virtual nodes
 - deploy diverse functionalities
 - simplify operation and management

Design Objectives (1)

● Isolation

- in terms of logical view, resources, operation
- attacks, failures, bugs, misconfigurations do not affect other VNs.

● Flexibility

- service providers may assemble VNs with customized
 - network topology
 - control and data plane functionalitywithout the need to coordinate with each other

● Scalability

- support multiple VNs
- utilize resources efficiently

Design Objectives (2)

● Programmability

- deploy customized protocols/services in network elements
- at packet level or below (e.g., optical devices?)
- how to expose to service providers/users
- opens up vulnerabilities (active networks redux?)

● Heterogeneity

- support a diverse set of
 - physical network technologies (wireless, sensor, optical, . . .)
 - virtual network capabilities

● Manageability

- mechanism \leftrightarrow policy
- “know what happened” \rightarrow SPs and InPs held accountable

Design Objectives (3)

● Dual use

- experimental **and** deployment facility
- design, evaluate, and deploy new services
- plausible pathway for adopting radically new network architectures
- PlanetLab, GENI, VINI

● Legacy support

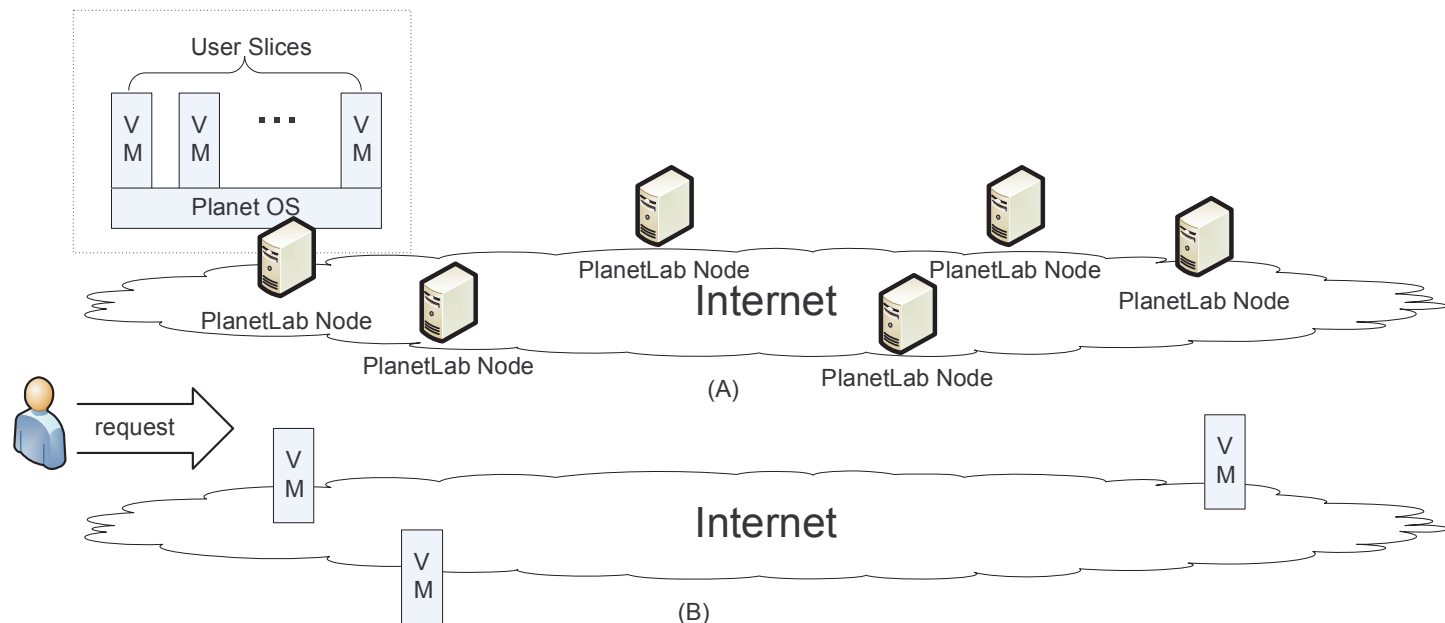
- existing Internet: another instance of a VN
- deploy IPv6 as VN → compete with IPv4

Network Virtualization Projects: Classification

- Targeted technology
 - wired, wireless, IP, heterogeneous
- Layer (in network stack) where virtualization is introduced
 - physical → application
- Application domain
 - specific problem domain addressed → wide range
- Virtualization granularity
 - node, link, full

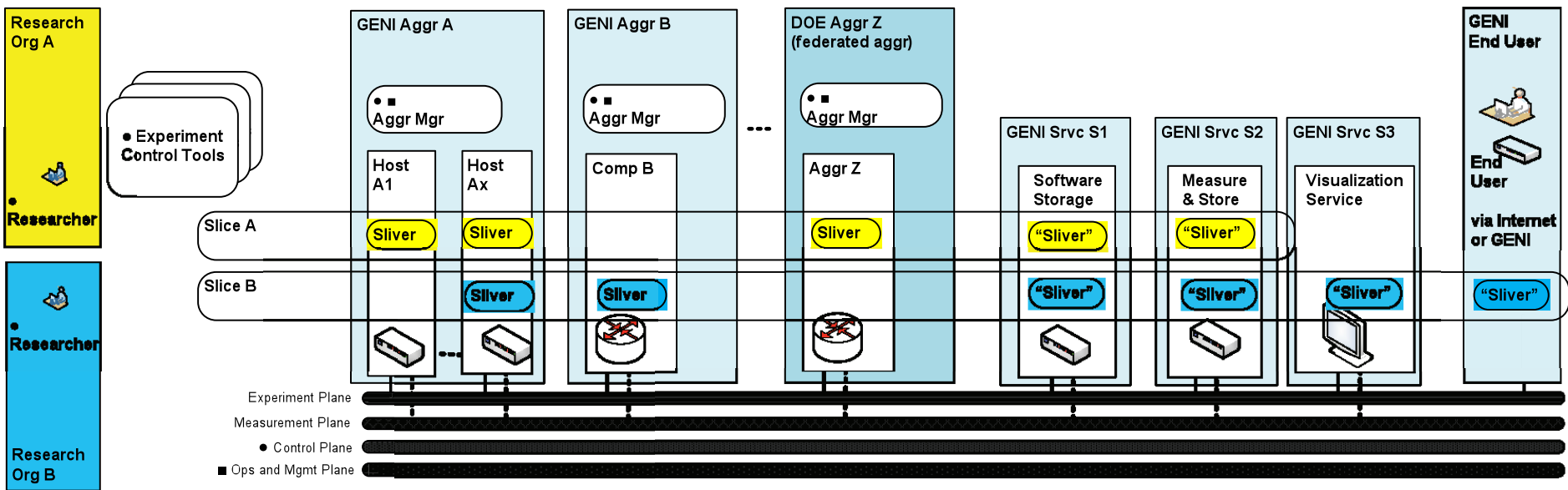
PlanetLab

- Overlay testbed for geographically distributed network services
- **Service-oriented** network architecture:
 - sliceability → applications acquire and run on a slice
 - decentralized control → each node subject to local policies
 - management sub-services running on own slices
 - existing, widely adopted API that does not change



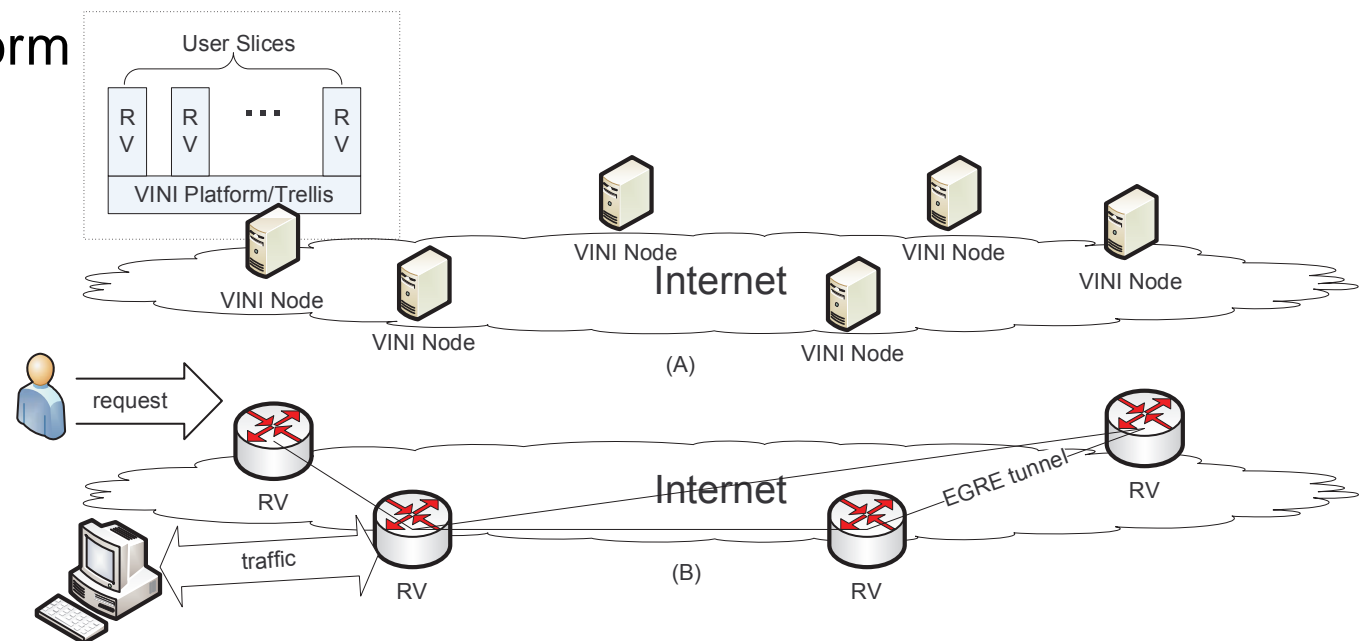
GENI

- Open large-scale experimental facility
 - for researchers: deploy and evaluate new network architectures
 - for users: carry real traffic
 - connects to Internet to reach external sites
- Supports slices of resources partitioned in space and time



VINI and Trellis

- Virtual network infrastructure: infrastructure + support for VNs
- Extends PlanetLab:
 - each virtual node is a router → better routing control
 - tools to create tunnels between virtual nodes
 - realistic simulations by directing traffic to/from outside world
- PL-VINI PlanetLab implementation → own physical testbed w/ Trellis software platform



VIOLIN

- Application level virtual network architecture
- Virtual networks created in software
 - vRouters, vLANs, vHosts → VMs on physical overlay hosts
- Complete isolation between VNs using VM technologies
- Secure, confined, dedicated environment for
 - performing risky experiments
 - deploy untrusted applications
- VIOLIN software implemented within Linux UML container
 - may run in PlanetLab slide
 - independent of PlanetLab, may run on other testbeds

Network Management

- X-Bone: a tool to deploy overlay networks across the Internet
 - automated configuration and management of overlay networks
 - support for resource discovery/allocation, monitoring, isolation
 - → VI: IP net of tunneled links among virtual hosts/routers
- UCLP: set of software tools in the form of Web services
 - end users dynamically provision/reconfigure optical networks (L1)
 - concatenate/divide lightpaths within domain or across domains
 - create customized logical IP networks

Full Virtualization

- CABO: concurrent architectures are better than one
 - exploits virtualization to separate InPs from SPs
 - supports programmable routers
 - users cannot program the network
 - SPs customize their network to provide end-to-end services
 - virtual router migration, accountability, multi-layer routing
- 4WARD: early-stage effort, little implementation
 - promises to bring NV to end users, not just experimenters
 - carrier-grade virtualization of network resources
 - trusted commercial setting for instantiation/inter-operation of VNs
 - InPs, VNPs (similar to SPs), VNOs (brokers, connect customers to services)

The Holy Grail of Network Virtualization

- A **network environment** in which multiple SPs:
 - **lease** underlying physical resources from multiple InPs
 - dynamically **compose** heterogeneous VNs
 - co-exist in isolation within same physical infrastructure
 - **compete** with each other by
 - deploying customized E2E services on-the-fly
 - managing the services on the VNs for the end users
 - effectively sharing and utilizing the physical resources

How to Get There: Future Directions (1)

- **Instantiation:** creating VNs
 - resource discovery (within/across InP domains, discovery plane?)
 - VN mapping: online/offline → NP-hard problem
 - signaling VN requests, bootstrapping VNs
 - API (request format, programmability of NEs)
- **Logistics:** operating VNs
 - virtual routers: performance, scalability, migration
 - virtual links: link scheduling, cross-InP links
 - resource (CPU, disk, link b/w) scheduling → max VN instances
 - naming and addressing
 - failure handling: isolation, no cascade effects

How to Get There: Future Directions (2)

● Management:

- mobility of end users, virtual routers, logical mobility of users across VNs
- configuration and monitoring, down to low level NEs
- responsibilities of InPs ↔ SPs
- support for multiple management paradigms

● Participant interactions:

- technology agnostic: E2E VNs across wireless, sensor, optical, . . . , resources
- inter-VN sharing of information
- economics: resource trading, market creation