Advanced Computer Networks

Network Virtualization, NFV, SDN

Credit to GN.Rouskas

Outline

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

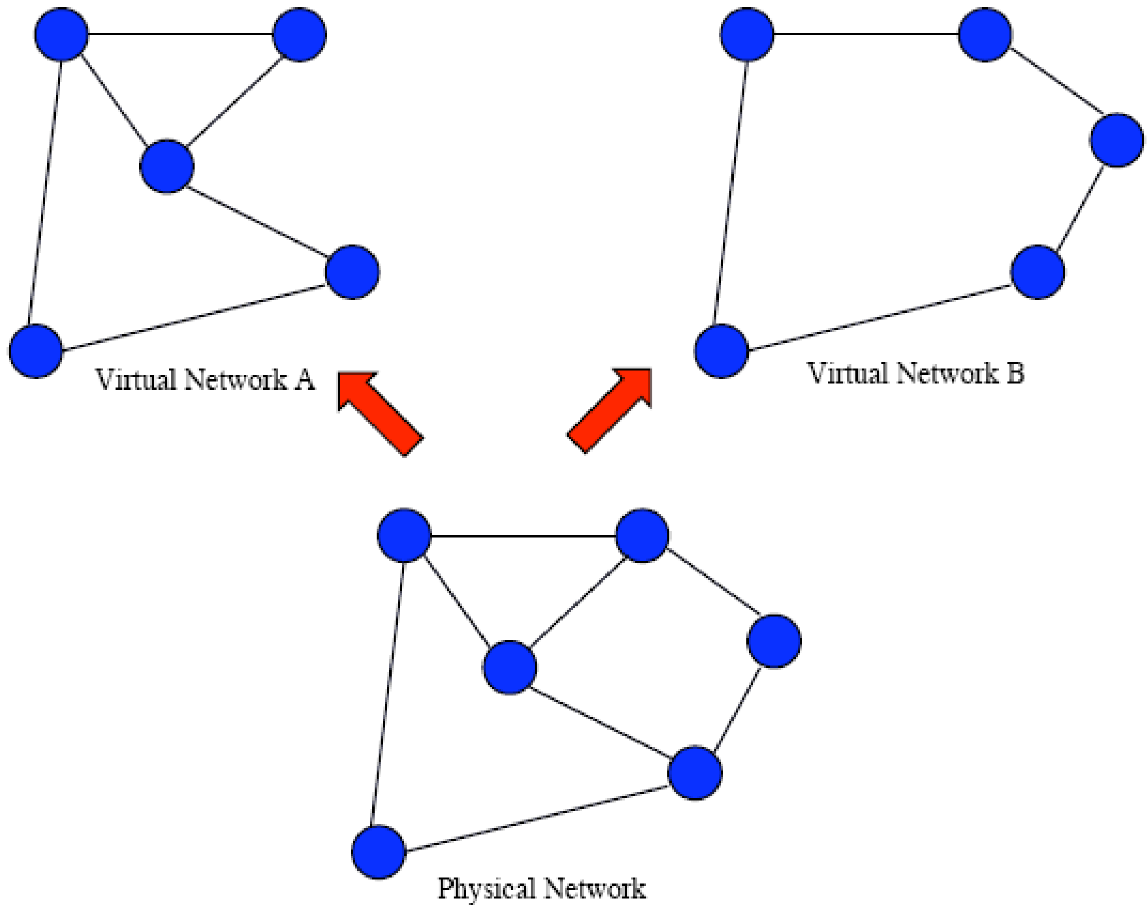
Virtualization As Resource Abstraction

- Virtualization: transparent abstraction of the physical computing platform and resources that supports multiple logical views of their properties

- Virtual anything
  - virtual memory/queue
  - OS platform → virtual machines
  - storage → virtual SAN
  - computing → virtual data centers

Network Virtualization

- Single physical network appears as multiple logical networks



Network Virtualization Environment

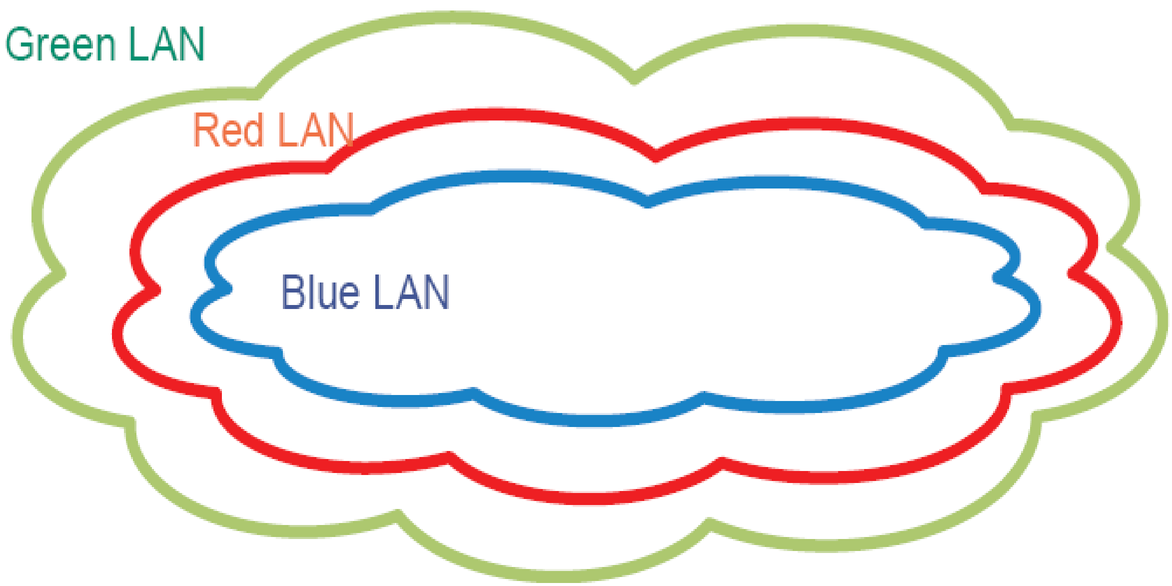
- Allows the creation of multiple virtual networks on same substrate
- Each virtual network:
  - is a collection of virtual nodes and virtual links
  - is a subset of underlying physical network resources
  - co-exists with, but is isolated from, other virtual networks

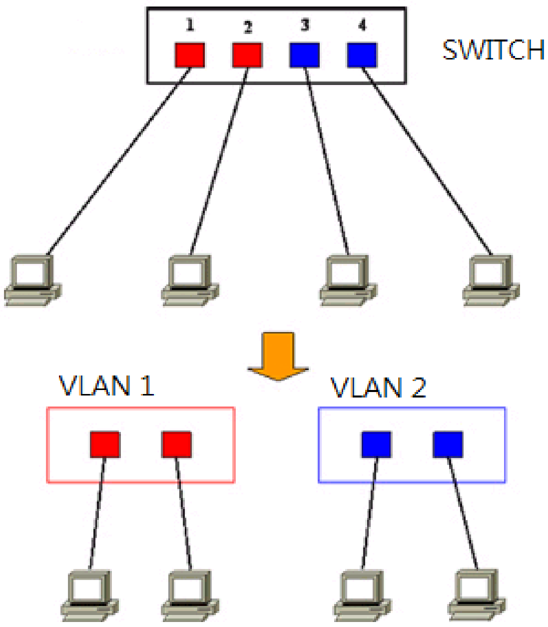
Historical Perspective

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

Virtual LANs

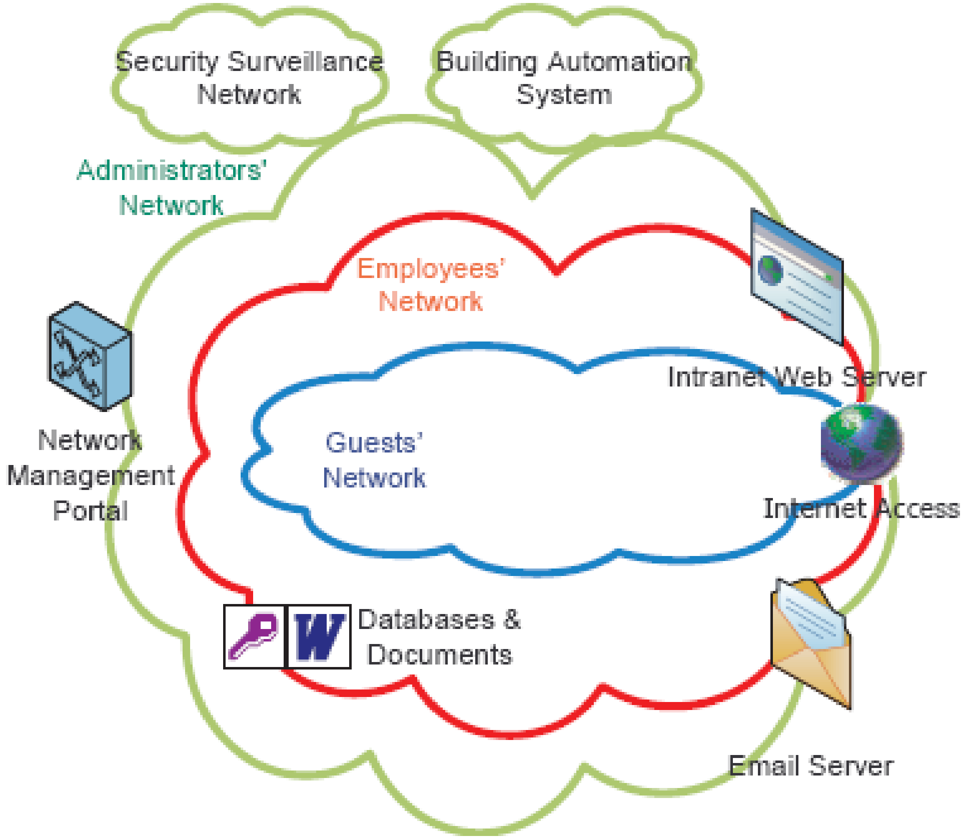
- L2 constructs, share same physical LAN infrastructure
- Groups of hosts with common interests, regardless of connectivity
- Segmented within boundary of broadcast domains → frame coloring
- High levels of trust, security, isolation, easy to configure





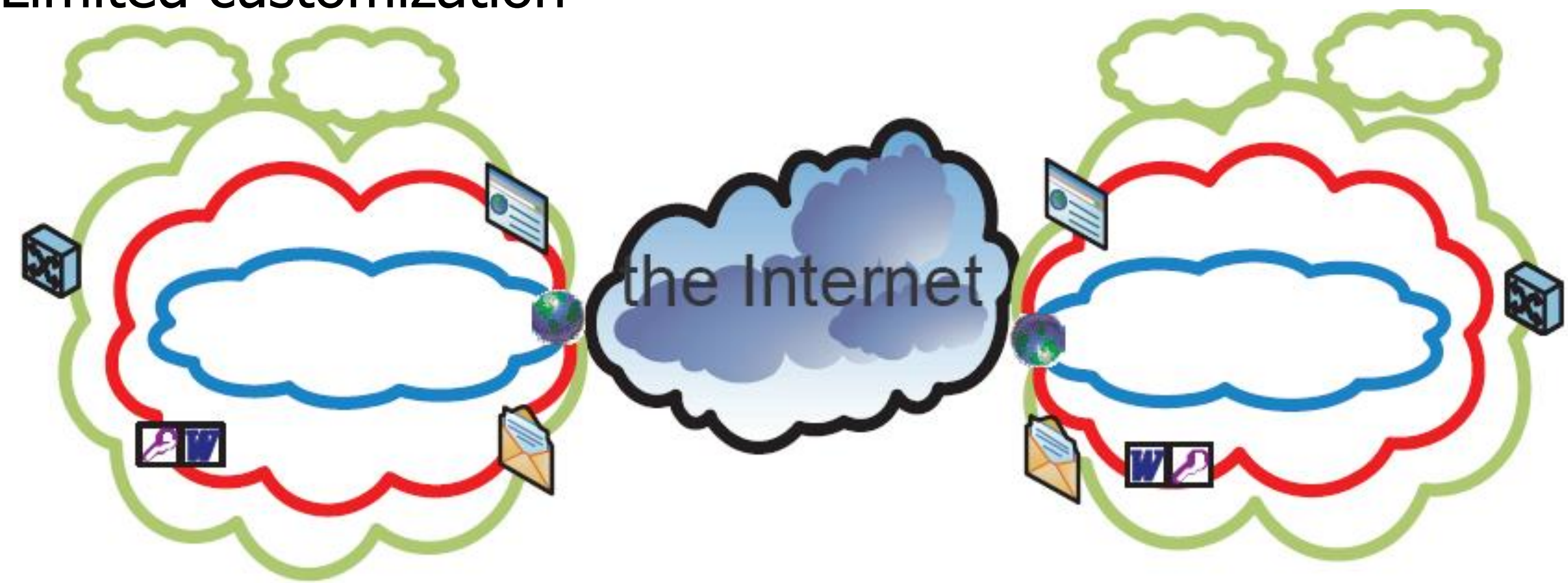
Virtual Service Networks

- 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

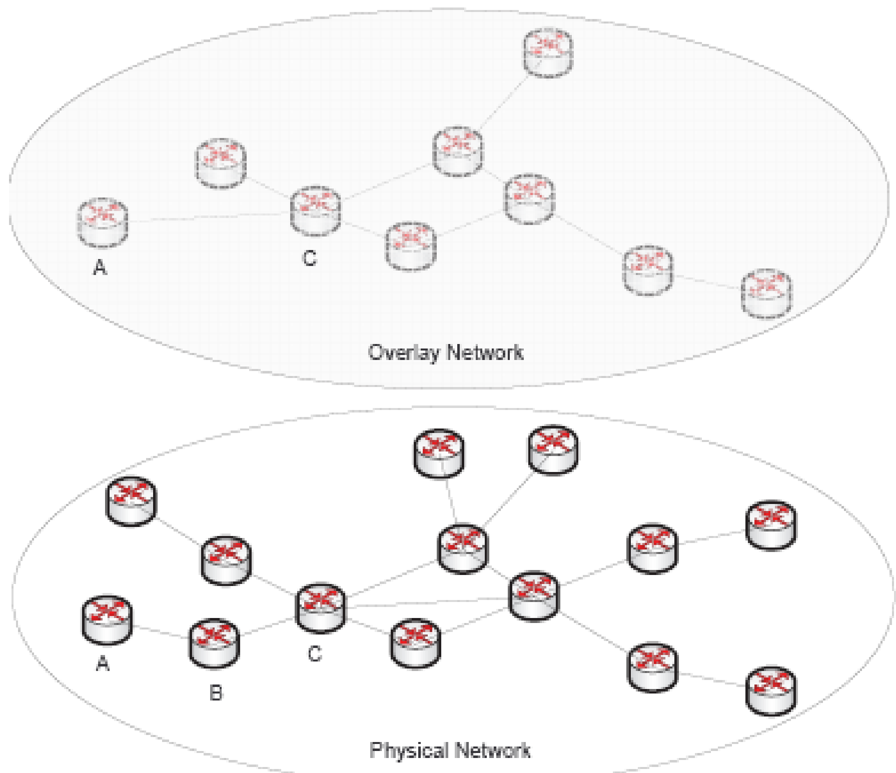
- Support co-existing networks through programmability
- Customized network functionality through
  - programmable interfaces to enable access to switches/routers
  - active code
    - call functions already installed
    - allow user to execute own code at switches/routers
- Operators must open networks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

Overlay Networks (1)

- A network built on top of one or more existing networks
  - without deploying entirely new networking equipments
- Adds an additional layer
- Changes properties in one or more layers of underlying network
- Many networks after PSTN have begun as overlay networks
  - the Internet also starts as an overlay network

Overlay Networks (2)

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



VN 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, . . .)

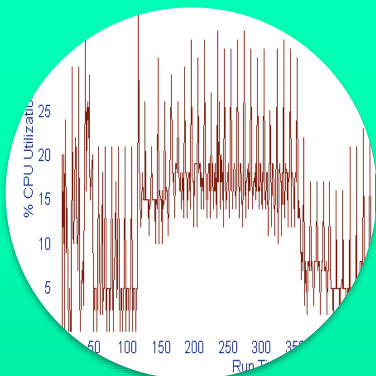
VN Motivation (2)

- 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

VN 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

Drivers Behind Virtualization



Hardware Resources Underutilized

- CPU utilizations ~ 10% - 25%
- One server – One Application
- Multi-core even more under-utilized

Data Centers are running out of space

- Last 10+ years of major server sprawl
- Exponential data growth
- Server consolidation projects just a start

Rising Energy Costs

- As much as 50% of the IT budget
- In the realm of the CFO and Facilities Mgr. now!

Administration Costs are Increasing

- Number of operators going up
- Number of Management Applications going up

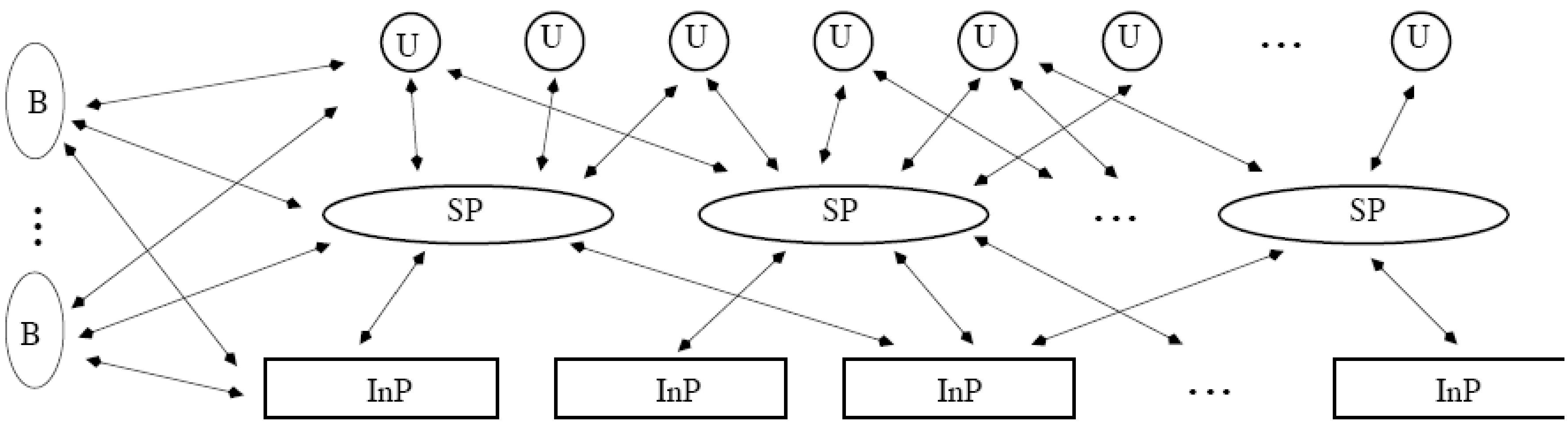
Operational Flexibility

Other Significant Virtualization Benefits

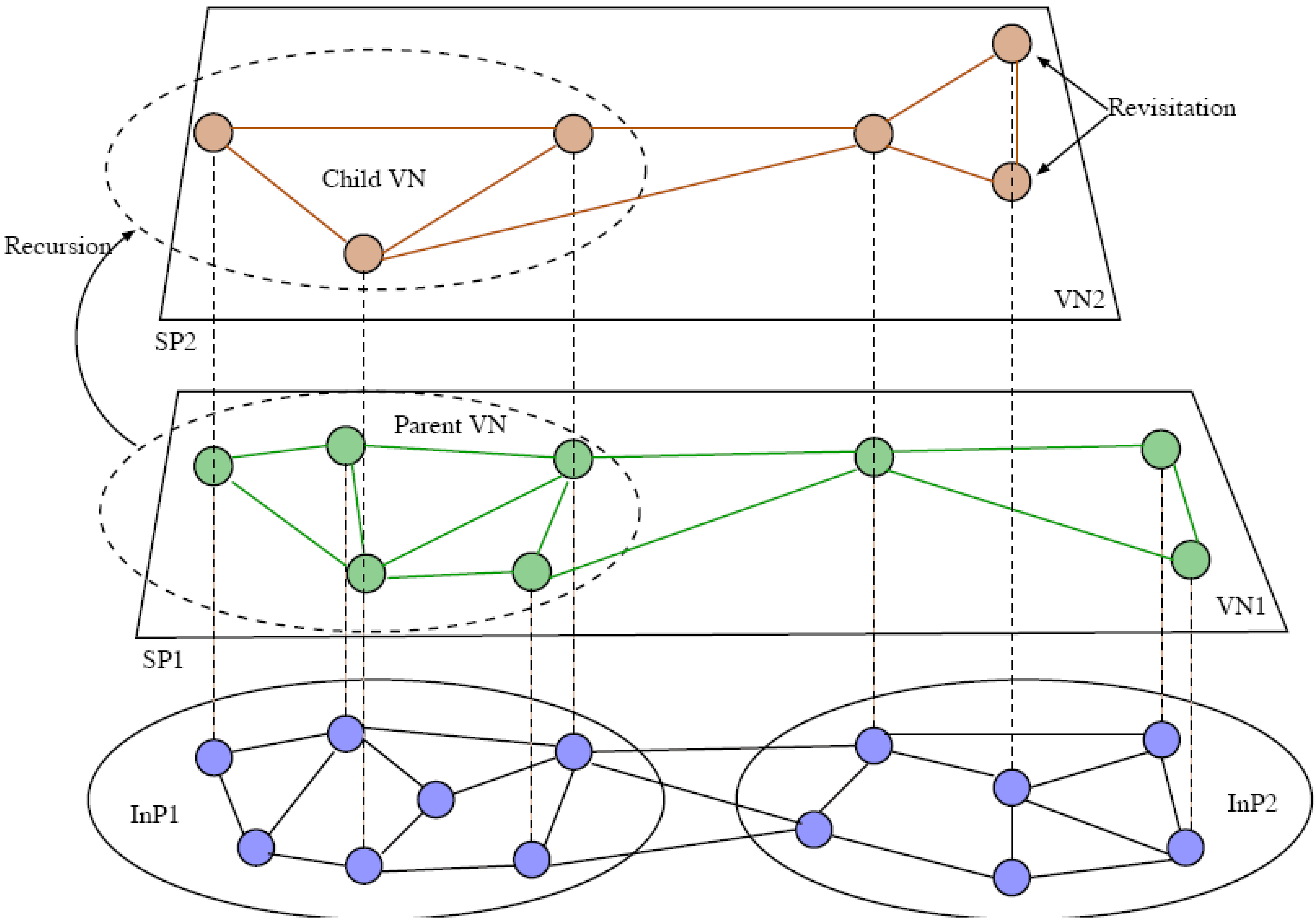
- Some key benefits:
  - Ability to quickly spawn test and development environments
  - Provides failover capabilities to applications that can't do it natively
  - Maximizes utilization of resources (compute & I/O capacity)
  - Server portability (migrate a server from one host to the other)
- Virtualization is not limited to servers and OS
  - Network virtualization
  - Storage virtualization
  - Application virtualization
  - Desktop virtualization

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 functionality without 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 ←→ policy
  - “know what happened → 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

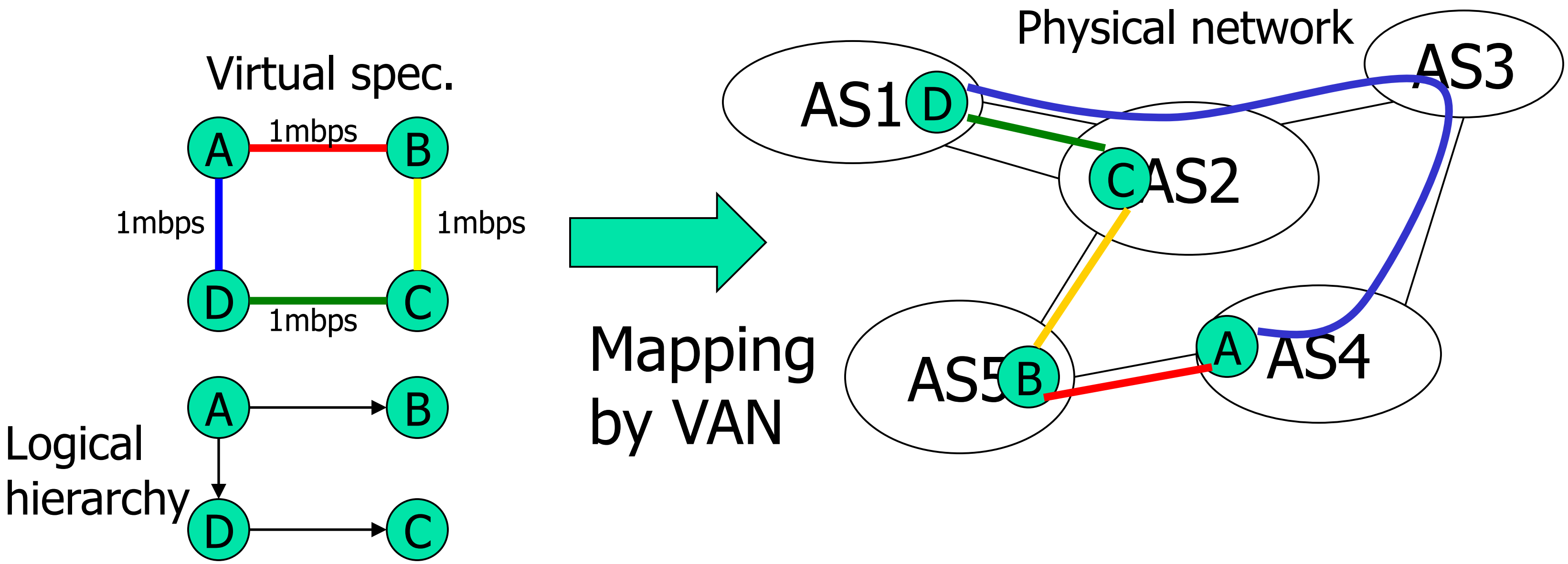
- Targeted technology
  - wired, wireless, IP, heterogeneous
- Layer (in network stack) where virtualization is introduced
  - Physical → application
- Application domain
  - specific problem domain addressed!wide range
- Virtualization granularity
  - node, link, full

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

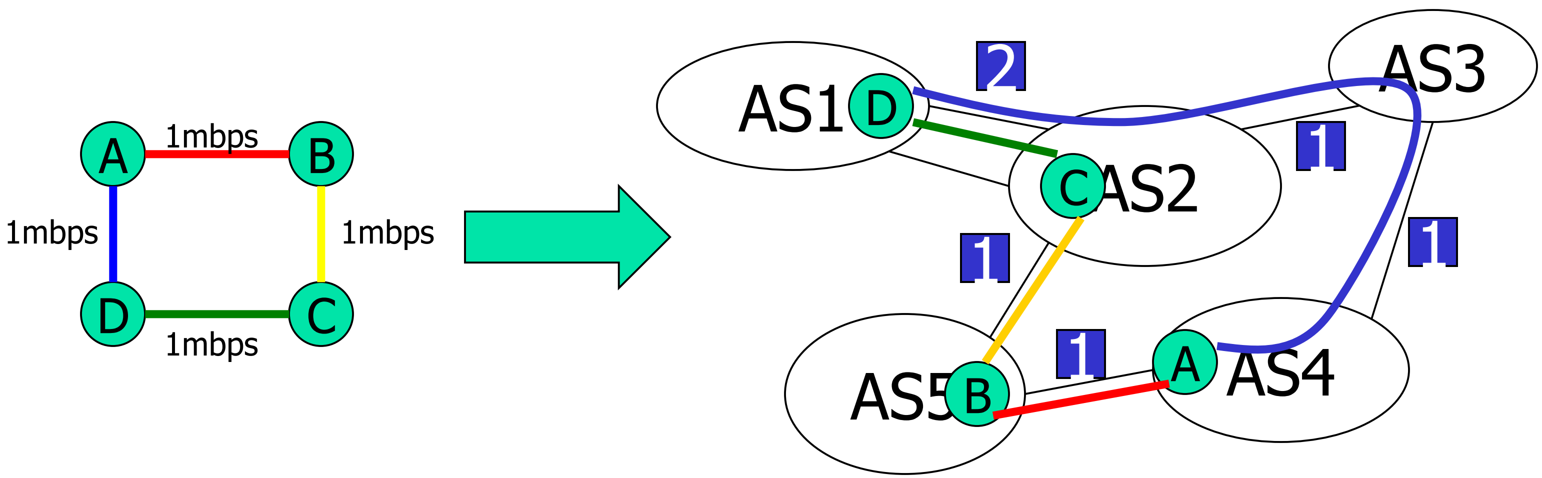
A Driving Example

- A web caching application needs…
  - Coverage for certain network area
  - Connectivity among caching service components
  - Resources to move cached contents
- Solution: requests a VN that provides
  - Coverage: spans a ring between AS1, AS2, AS4, and AS5
  - Resources: provides at least 1mbps for all connections
  - Reliability: prohibits more than 2 virtual links from traversing the same physical link



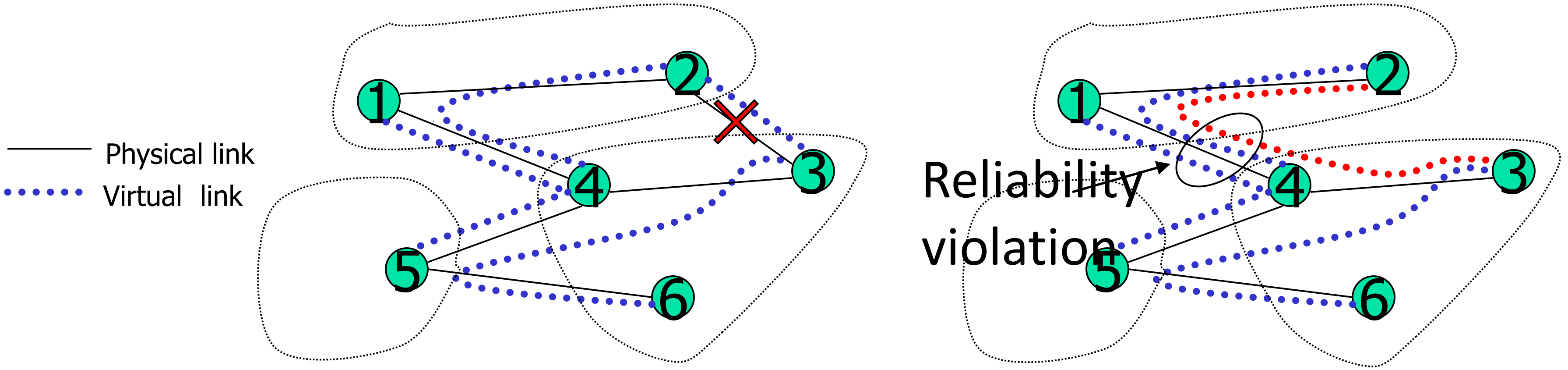
Specification Mapping

- **Heuristic mapping algorithm**
  - Sort VNs and PNs by degree; map VN to PN by degree-order
  - Mark all PL without enough bandwidth for the VLs as infeasible
  - Each PL has a “mapped-onto” counter, initially 0
  - pick a VL and map it to a physical path with lowest maximum counter among all PLs traversed
  - After each VL is mapped, increment counter and subtract available bandwidth for each PL; mark a PL infeasible as appropriate
  - Repeat until all VLs are mapped



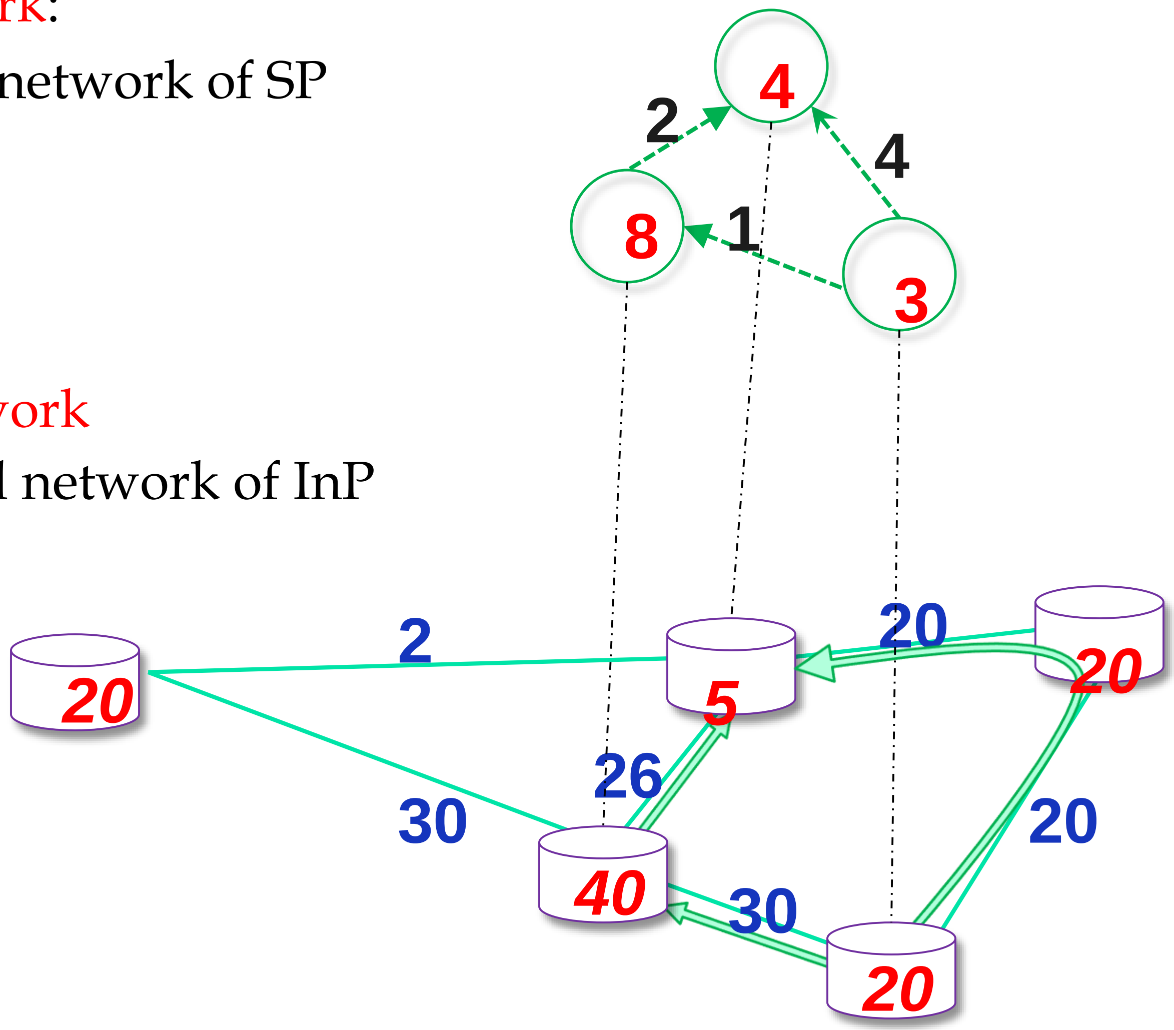
Failure Recovery

- When a physical link fails, the VLs it carries must be restored
- First try Local repair
  - Find an alternative path with adequate resources between the two disconnected AS'es
  - Fast, and preserve original topology
  - But
    - May violate reliability constraint
    - Alternative path may not exist
- Example
  - Physical link between 2 and 3 goes down
  - Alternative path goes through 2, 1, 4, and 3



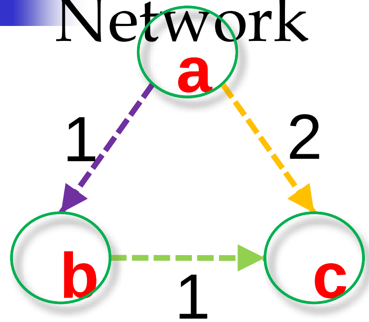
Virtual Network Embedding

- **Virtual Network**:
  - the logical network of SP
- **Substrate Network**
  - the physical network of InP

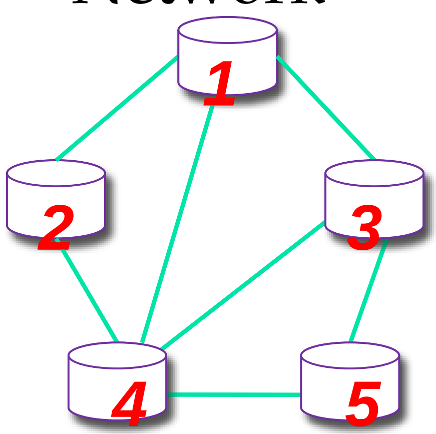


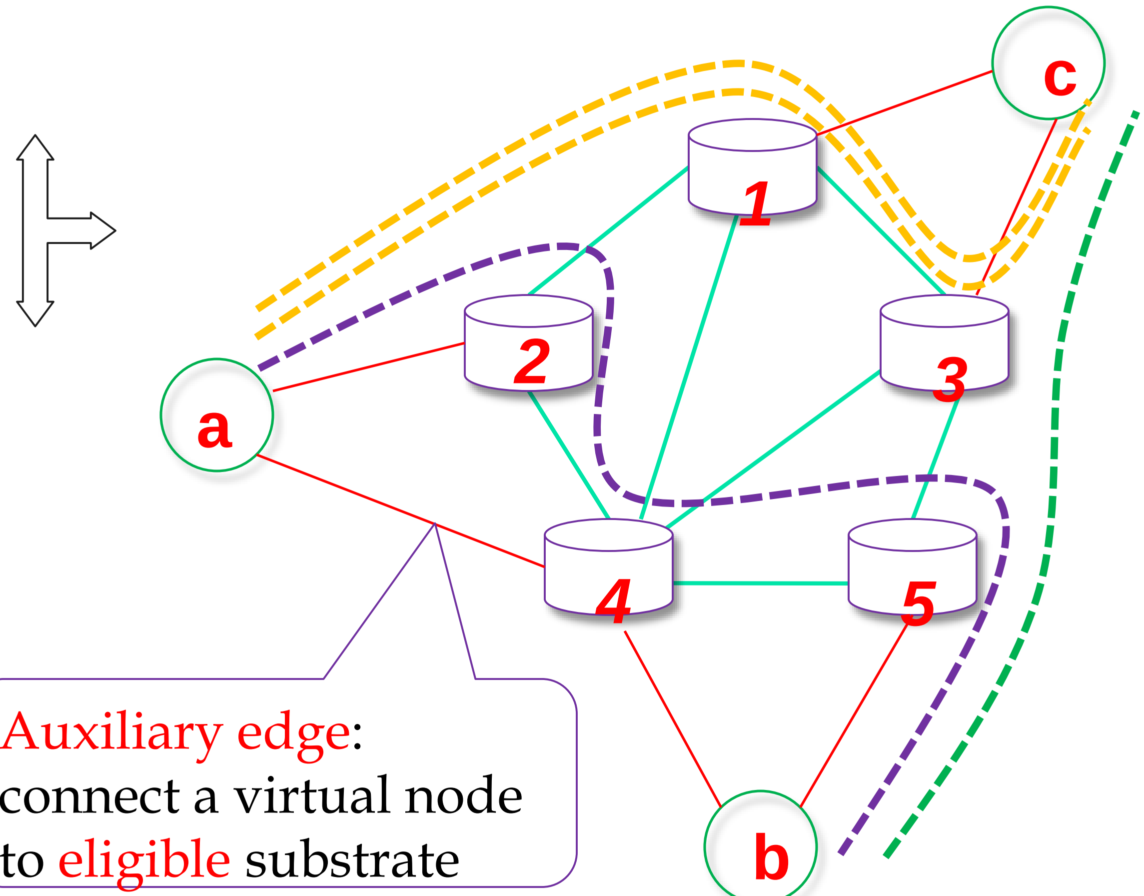
Network Flow Construction

- Virtual Network



- Substrate Network





Auxiliary edge: connect a virtual node to eligible substrate nodes

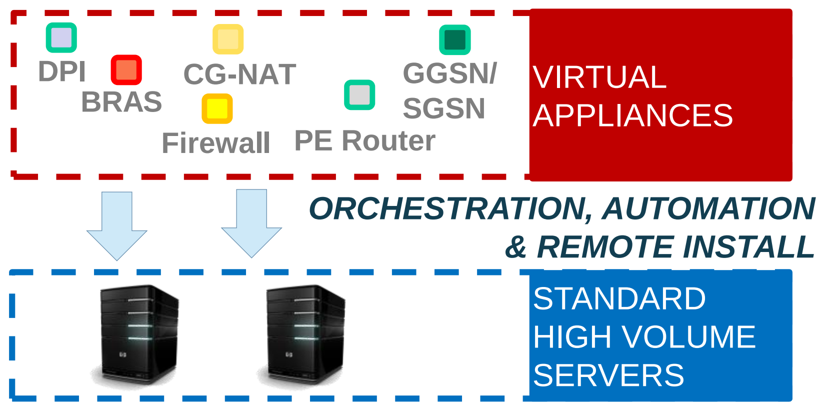
The NFV Concept

A means to make the network more flexible and simple by minimising dependence on HW constraints

Traditional Network Model: APPLIANCE APPROACH

- Network Functions are **based on specific HW&SW**
- **One physical node per role**

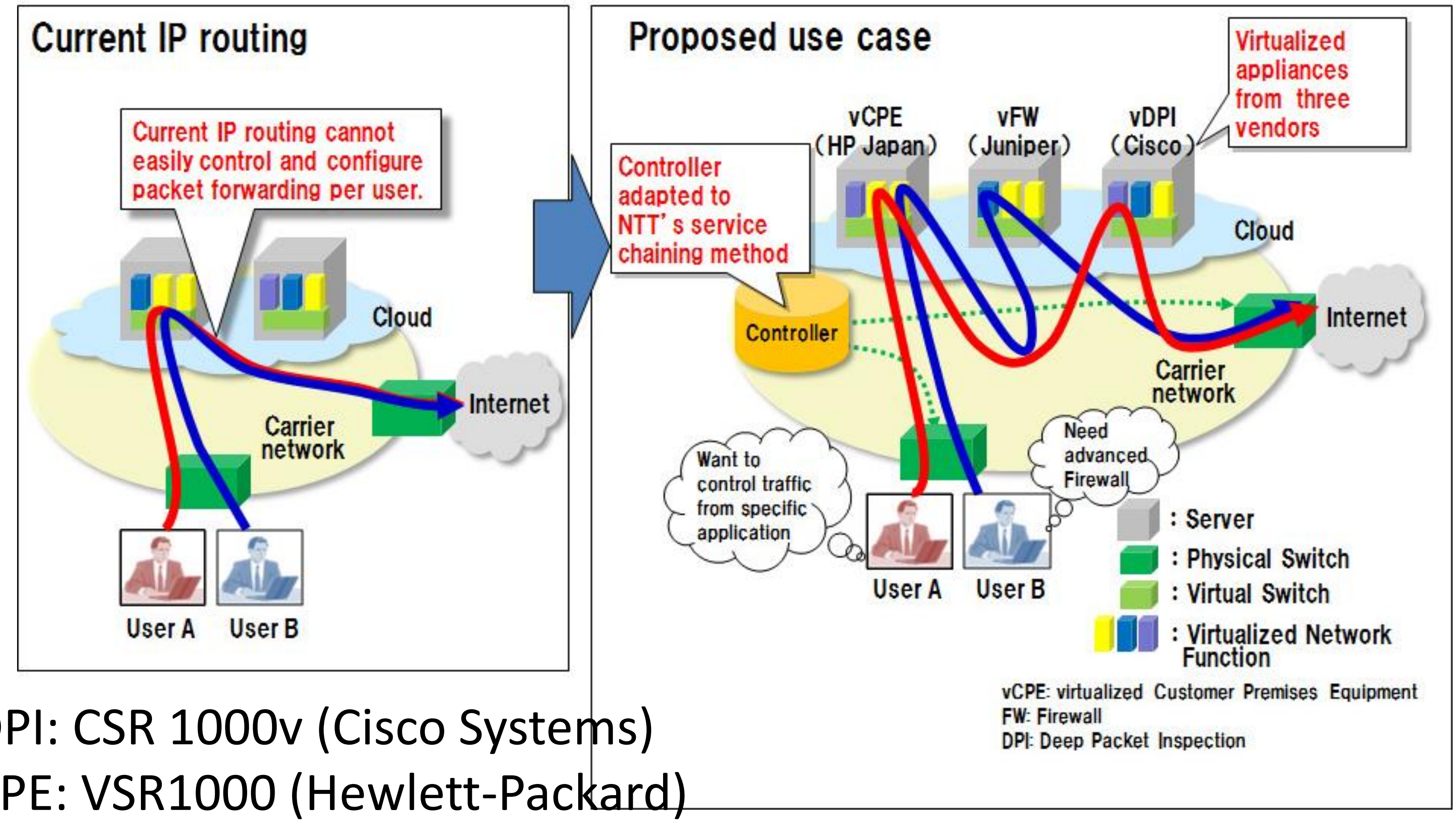
Virtualised Network Model: VIRTUAL APPLIANCE APPROACH



- Network Functions are **SW-based over well-known HW**
- **Multiple roles** over **same HW**

Source: Adapted from D. Lopez Telefonica I+D, NFV

Service Chaining for NW Function Selection in Carrier Networks



vDPI: CSR 1000v (Cisco Systems)

vCPE: VSR1000 (Hewlett-Packard)

vFW: FireFly (Juniper Networks)

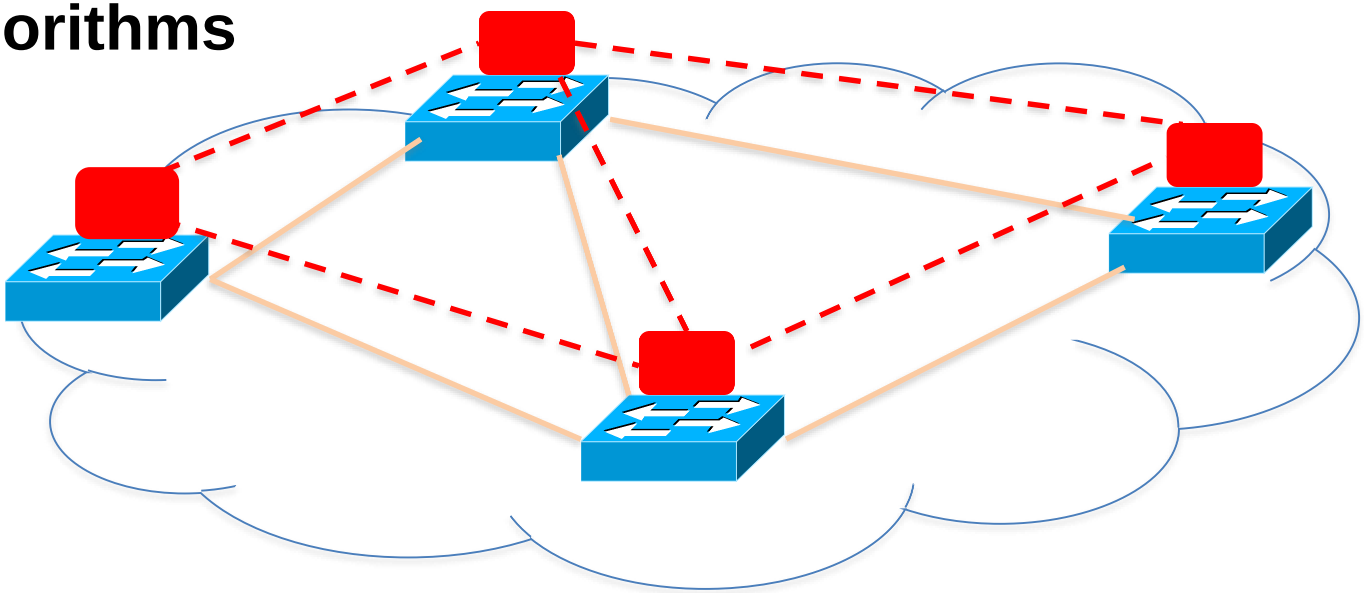
VIM (NW Controller): Service

Source: ETSI Ongoing

Rethinking the “Division of Labor”

Traditional Computer Networks

Control plane:
Distributed algorithms

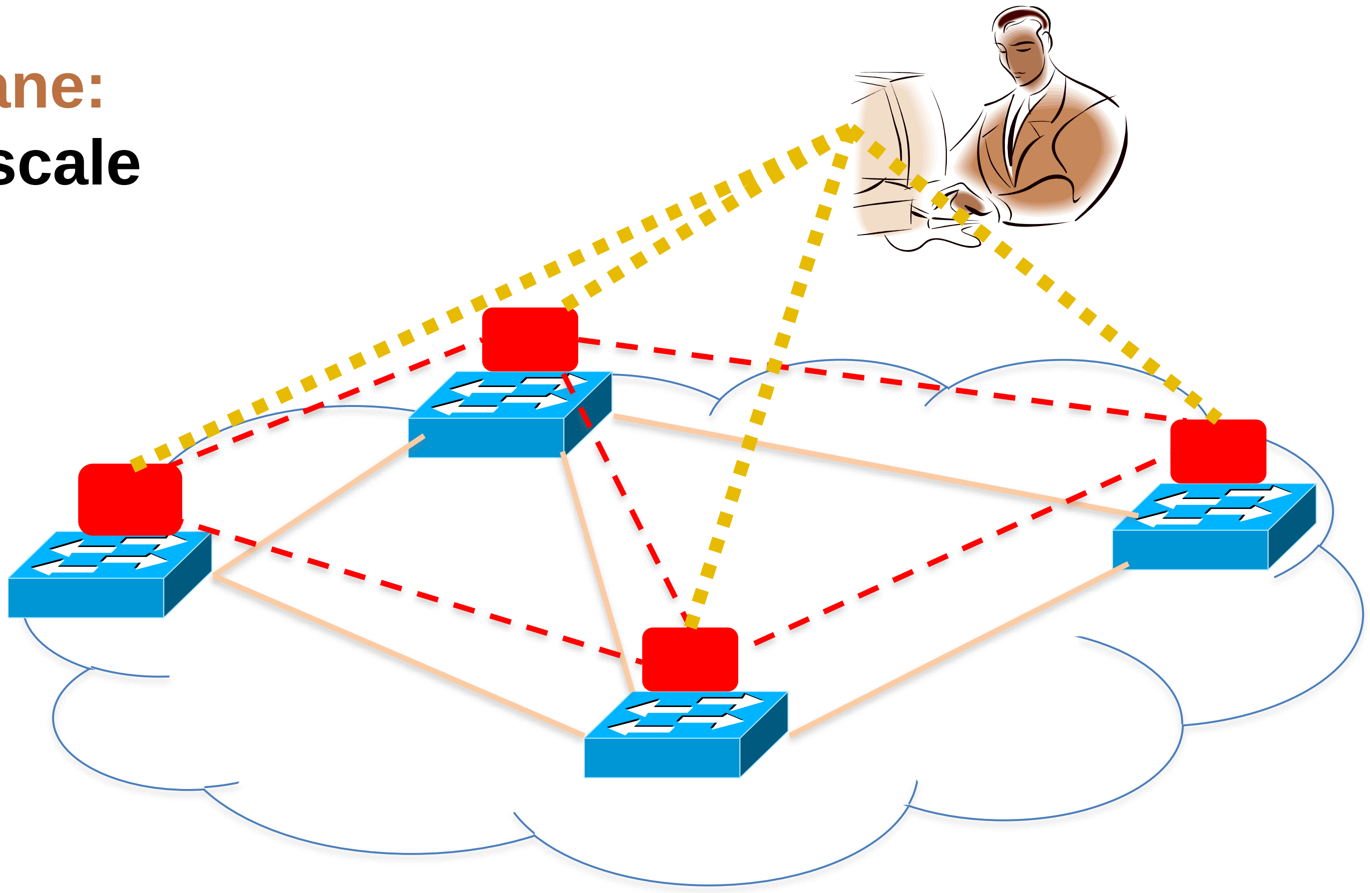


Track topology changes, compute routes, install forwarding rules

Traditional Computer Networks

Management plane:

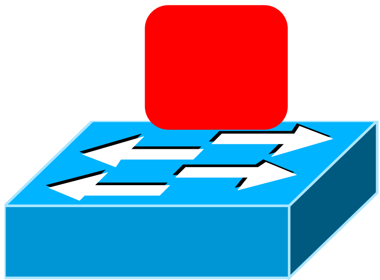
Human time scale



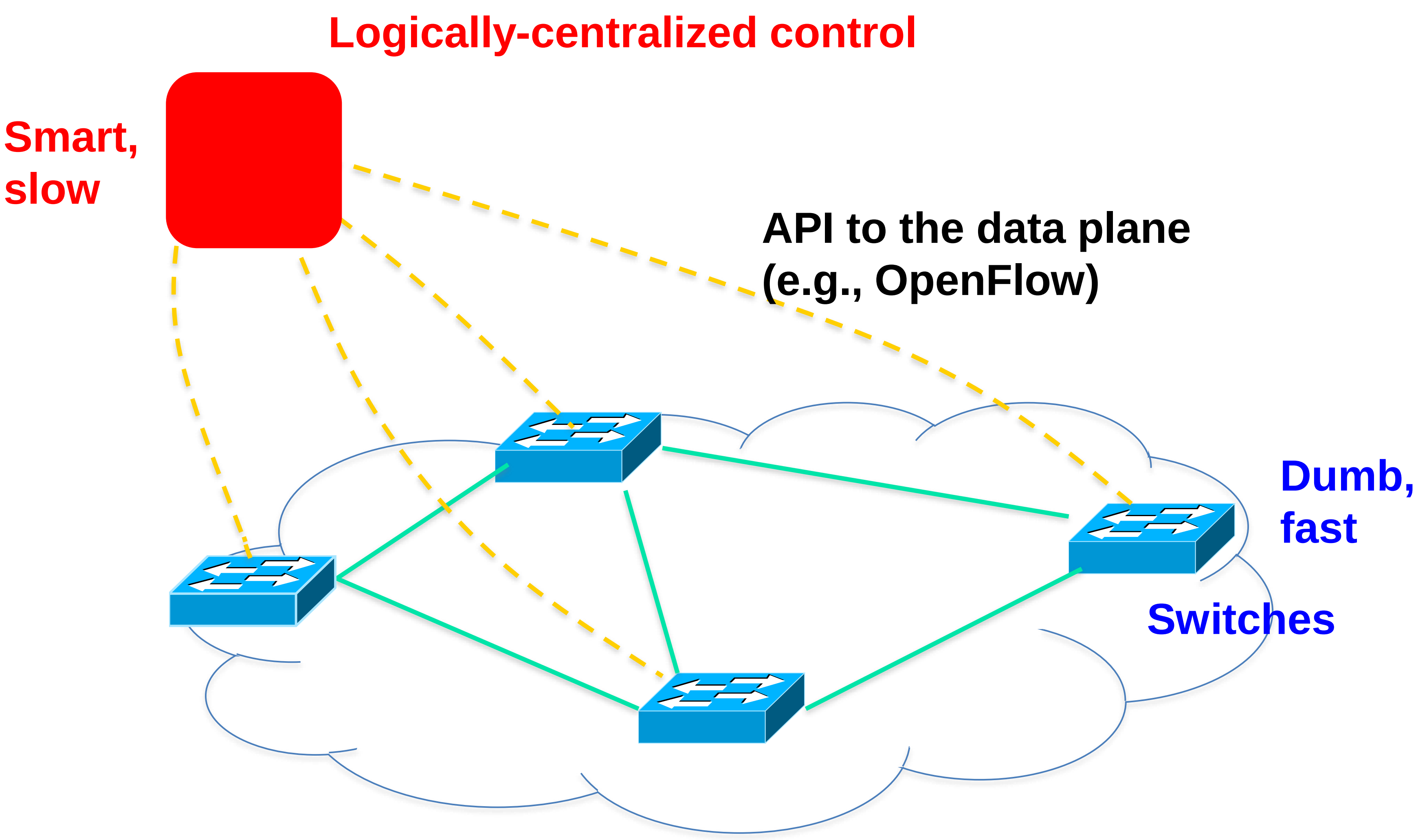
Collect measurements and configure the equipment

Death to the Control Plane!

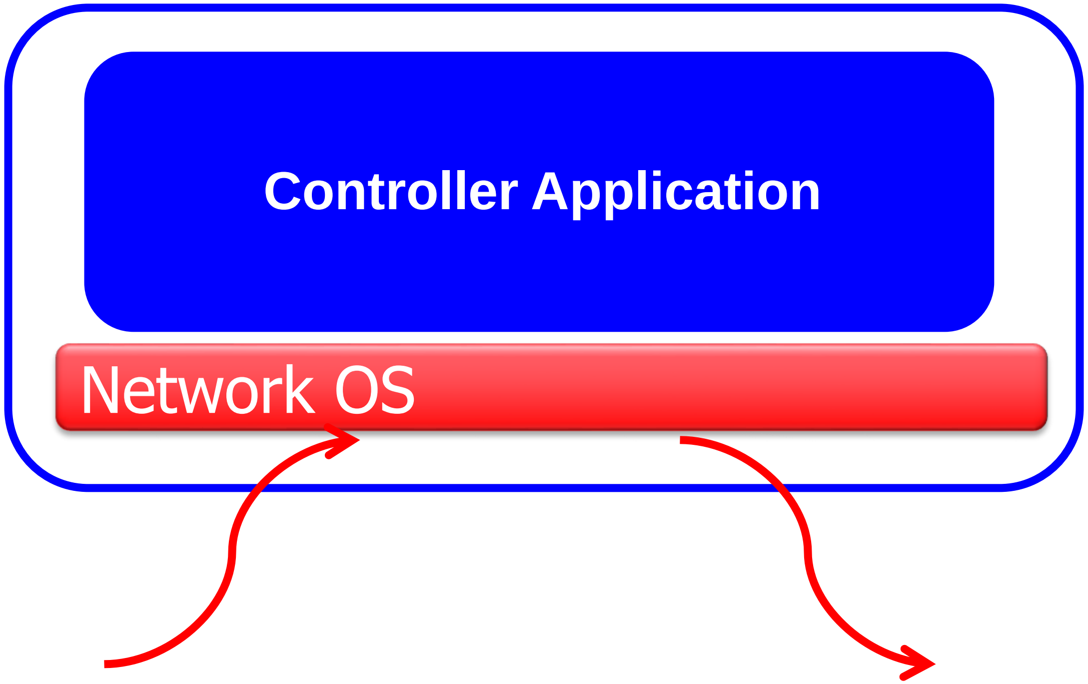
- Simpler management
  - No need to “invert” control-plane operations
- Faster pace of innovation
  - Less dependence on vendors and standards
- Easier interoperability
  - Compatibility only in “wire” protocols
- Simpler, cheaper equipment
  - Minimal software



Software Defined Networking (SDN)



Controller: Programmability



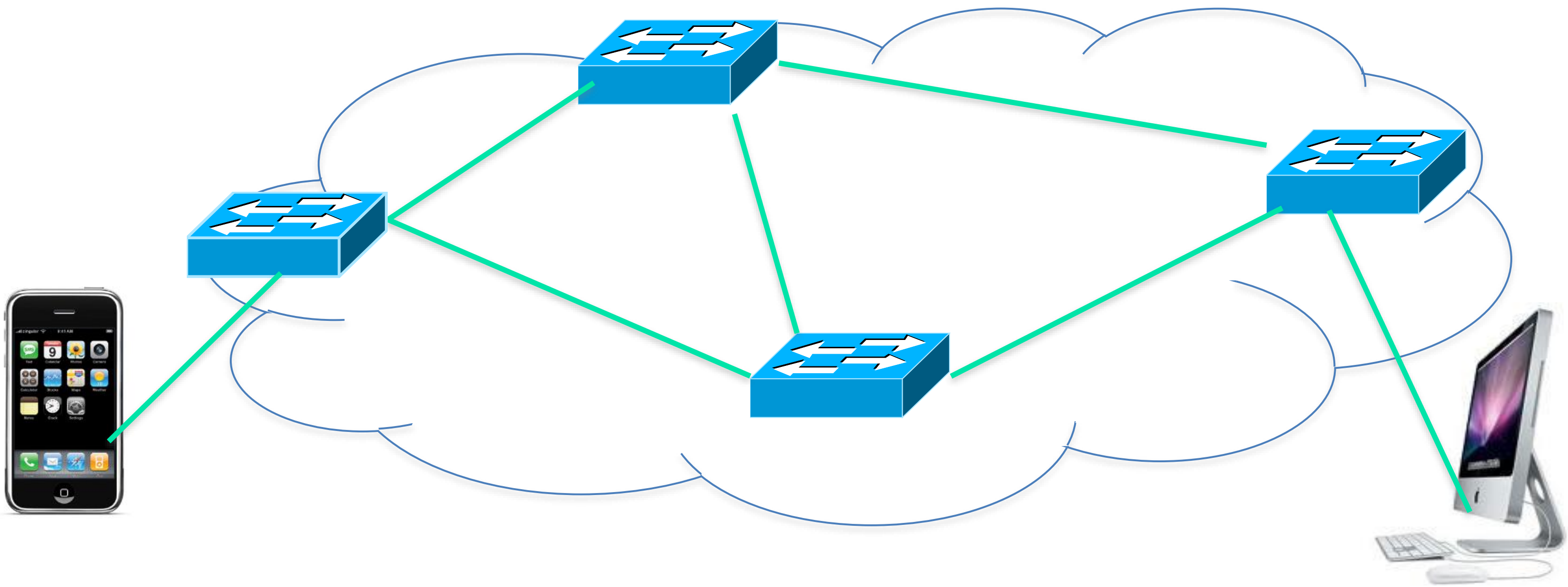
Events from switches
Topology changes,
Traffic statistics,
Arriving packets

Commands to switches
(Un)install rules,
Query statistics,
Send packets

E.g.: Network Virtualization

Controller #1 **Controller #2** **Controller #3**

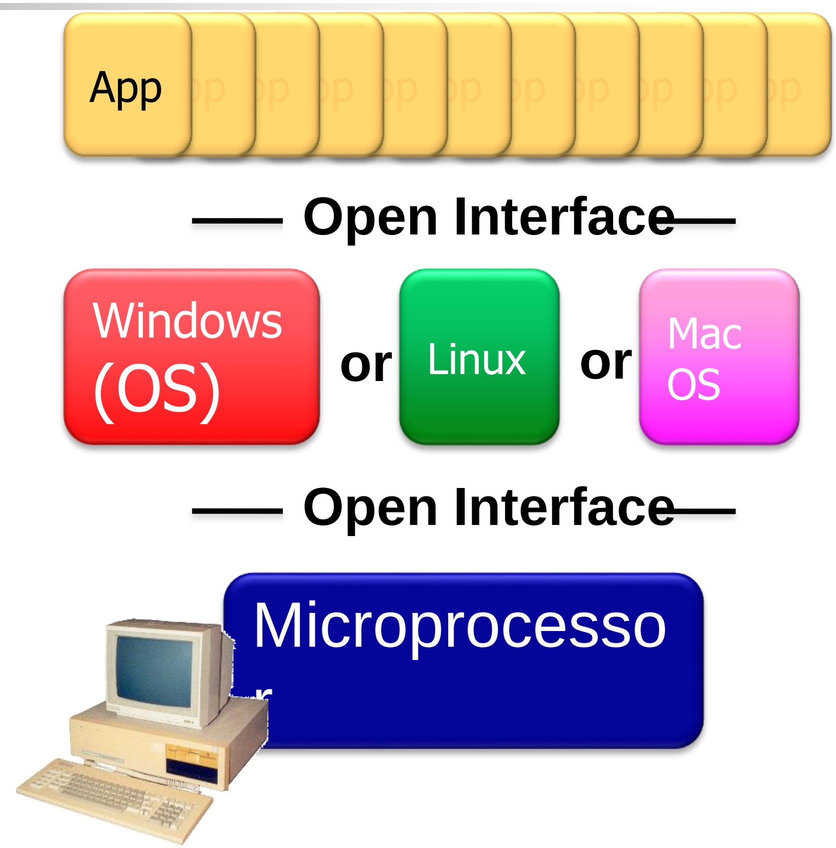
Partition the space of packet headers



A Helpful Analogy

From Nick McKeown's talk "Making SDN Work" at the Open Networking Summit, April 2012

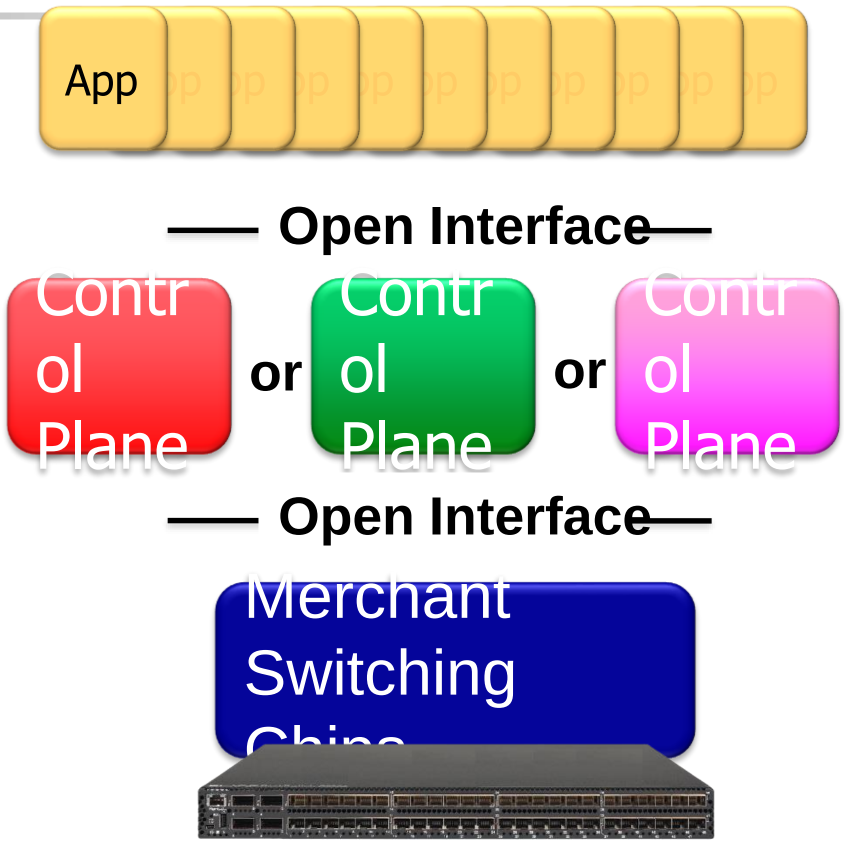
Mainframes



Vertically integrated
Closed, proprietary
Slow innovation
Small industry

Horizontal
Open interfaces
Rapid innovation
Huge industry

Routers/Switches

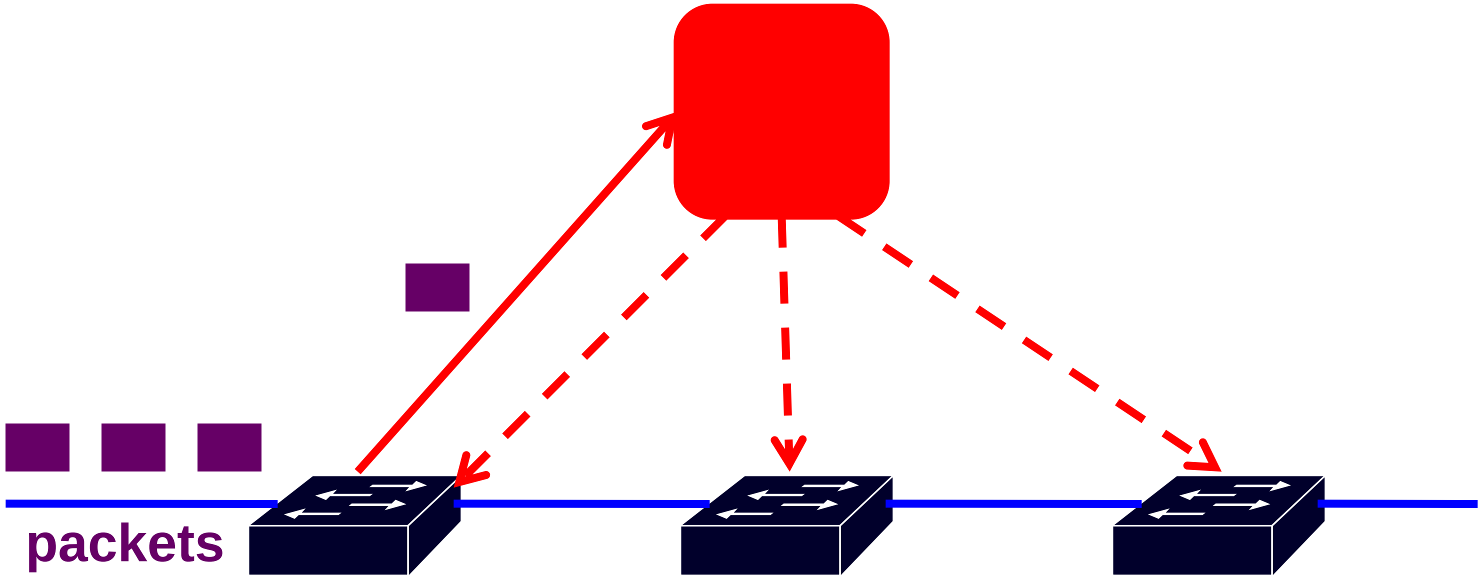


Vertically integrated
Closed, proprietary
Slow innovation

Horizontal
Open interfaces
Rapid innovation

Challenges:Controller Delay and Overhead

- Controller is much slower the the switch
- Processing packets leads to delay and overhead
- Need to keep most packets in the “fast path”



Challenges: Distributed Controller