Network Virtualization and Application Delivery Using Software Defined Networking

RAJ JAIN

Washington University in Saint Louis
Saint Louis, MO 63130
Jain@cse.wustl.edu

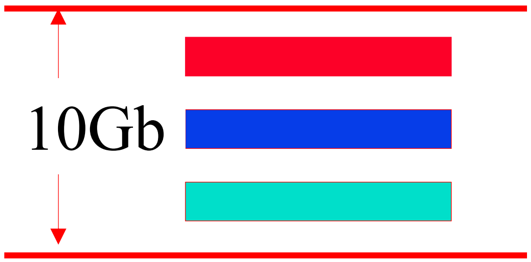
Invited Talk at ADCOM 2012
December 14, 2012, Bangalore, India

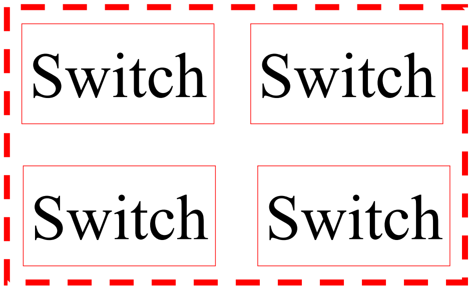
These slides and audio/video recordings are available at:
http://www.cse.wustl.edu/~jain/talks/adn_adc.htm

1. Virtualization: Why, How?
2. Recent Networking Virtualization Technologies
3. Our Research: Open Application Delivery
4. Software Defined Networking

Why Virtualize?

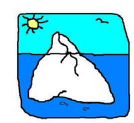
1. Sharing: Break up a large resource Large Capacity or high-speed
2. Isolation: Protection from other tenants
3. Aggregating: Combine many resources in to one
4. Dynamics: Fast allocation, Change/Mobility, load balancing
5. Ease of Management ⇒ Cost Savings
6. Mobility for fault tolerance





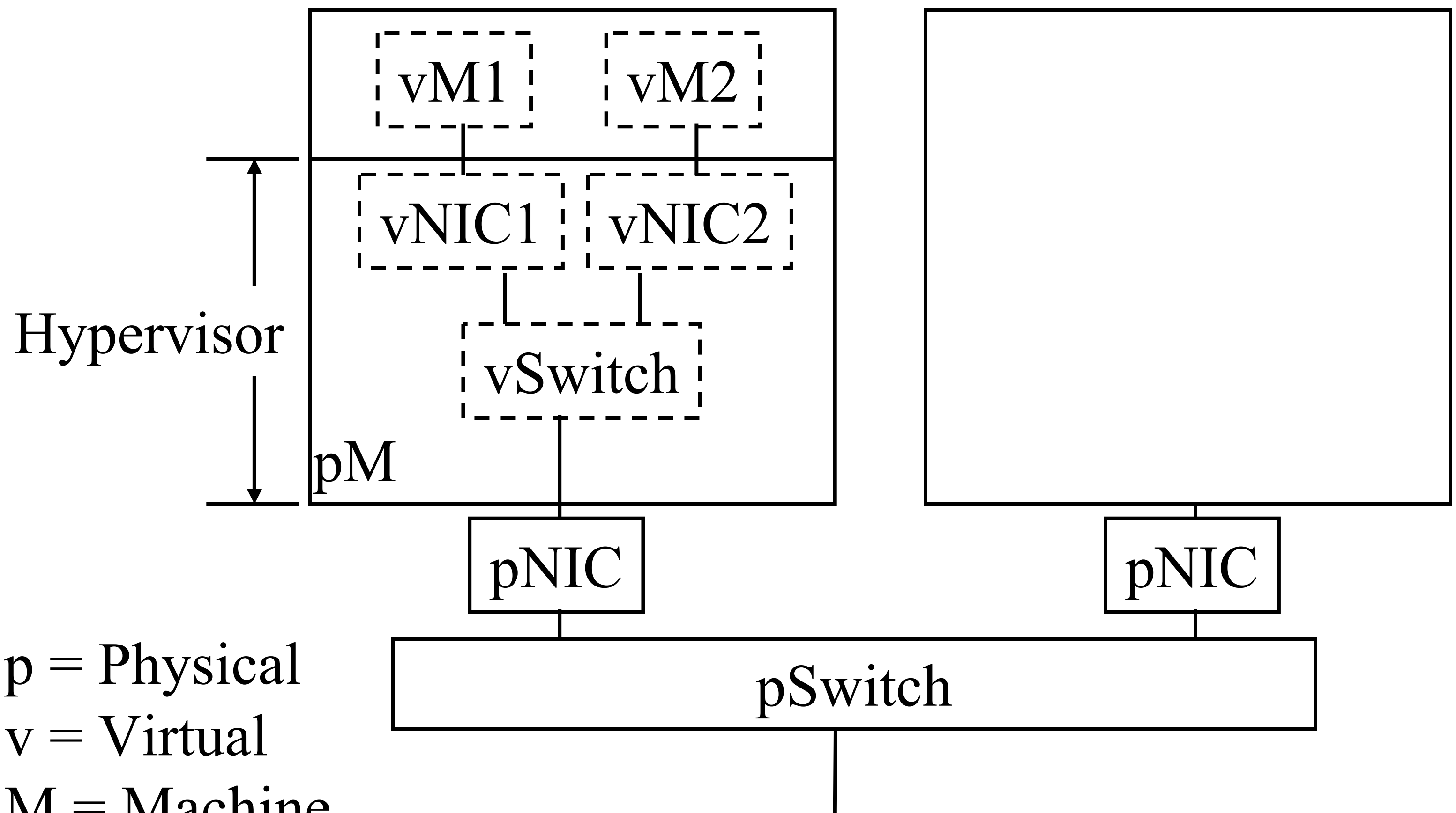
Virtualization in Computing

- **Storage**:
  - Virtual Memory ⇒ L1, L2, L3, ... ⇒ Recursive
  - Virtual CDs, Virtual Disks (RAID), Cloud storage
- **Computing**:
  - Virtual Desktop ⇒ Virtual Server ⇒ Virtual Datacenter
    Thin Client ⇒ VMs ⇒ Cloud
- **Networking**: Plumbing
  - Virtual Channels, Virtual LANs, Virtual Private Networks
  - Networks consist of: Hosts - L2 Links - L2 Bridges - L2 Networks - L3 Links - L3 Routers - L3 Networks – L4 Transports – L5 Applications
  - Each of these can be/need to be virtualized
  - Quick review of recent technologies for network virtualization



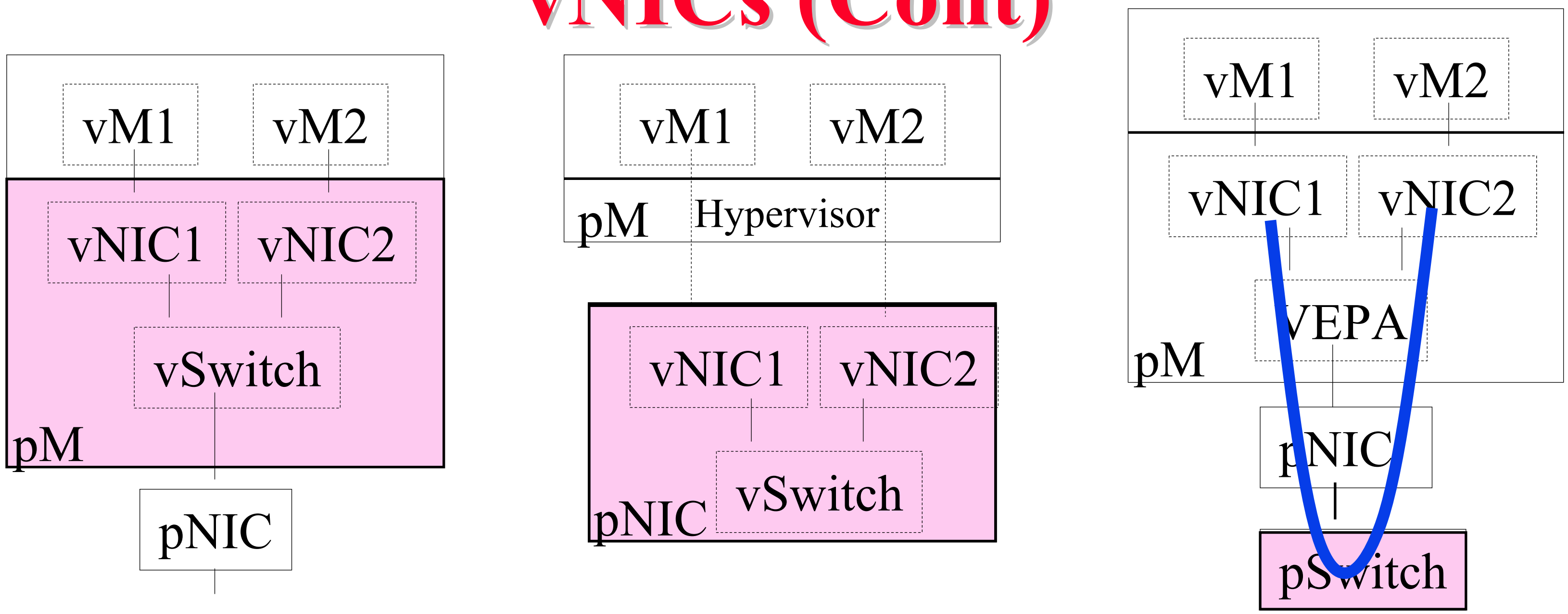
vNICs

- Each VM needs its own network interface card (NIC)



p = Physical
v = Virtual
M = Machine

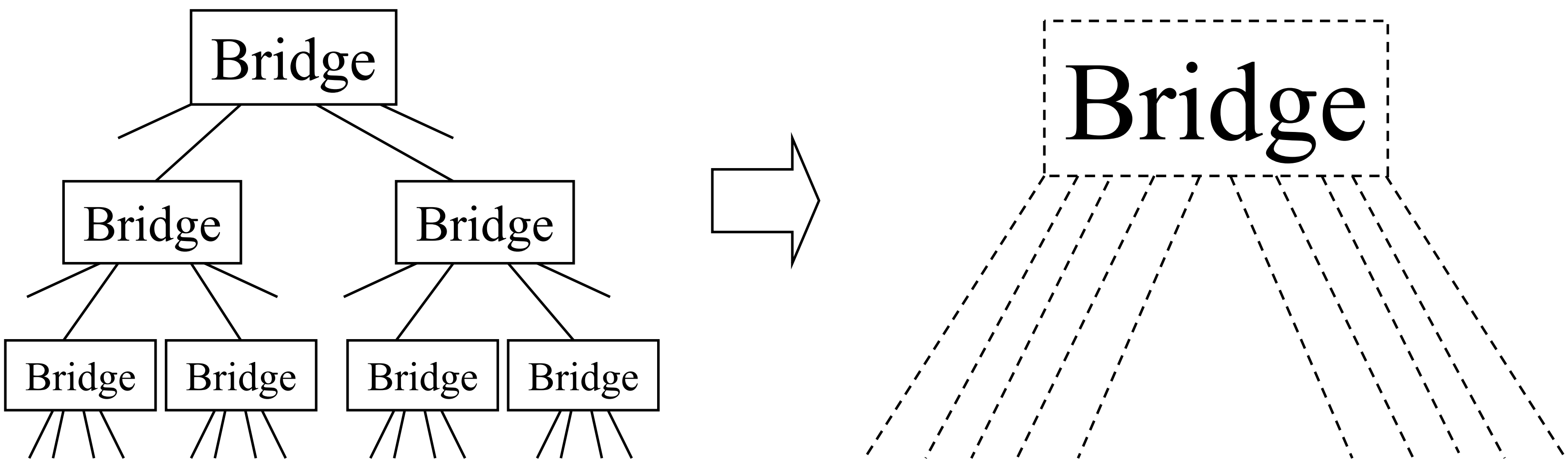
vNICs (Cont)



1. VM vendors: S/W NICs in Hypervisor w Virtual Ethernet Bridge (**VEB**)(overhead, not ext manageable, not all features)
2. NIC Vendors: NIC provides virtual ports using Single-Route I/O virtualization (**SR-IOV**) on PCI bus
3. Switch Vendors: Switch provides virtual channels for inter-VM Communications using virtual Ethernet port aggregator (**VEPA**): **802.1Qbg** (s/w upgrade), **802.1Qbh** (new switches)

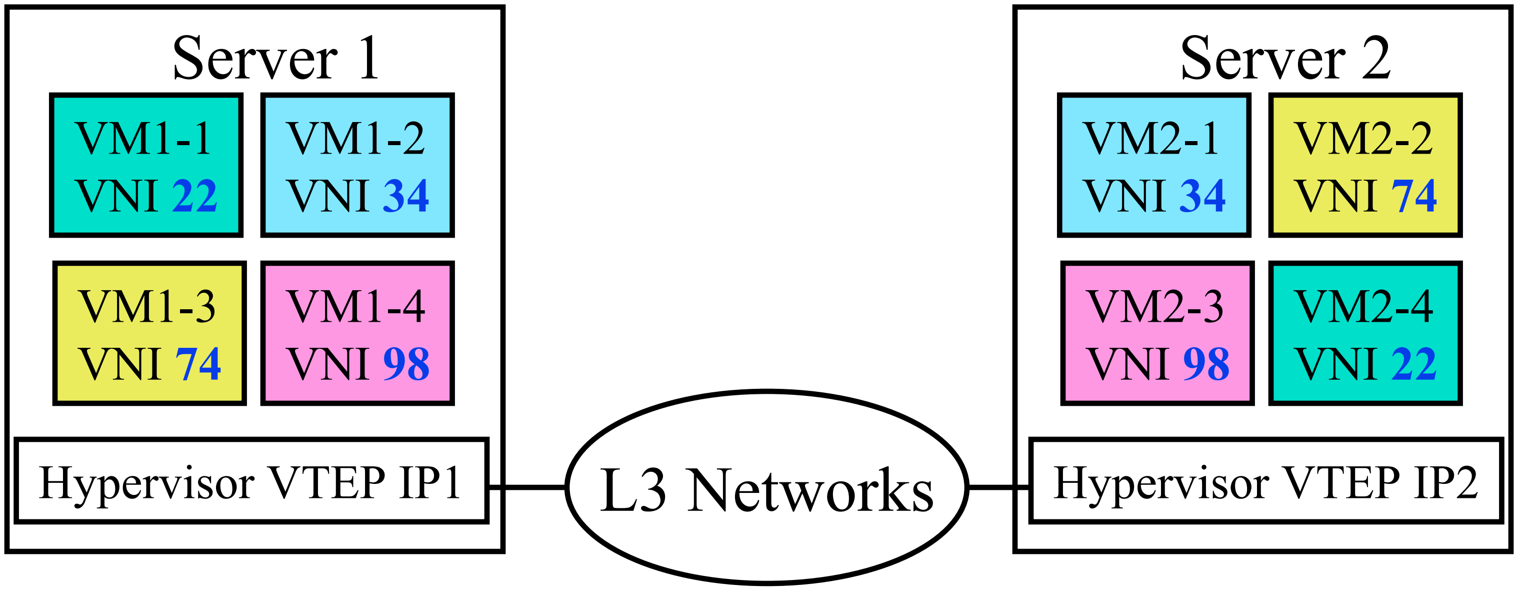
Bridge Port Extension

- Multiple physical bridges to make a single virtual bridge with a large number of ports
  ⇒ Easy to manage and configure
- **IEEE 802.1BR**



Multi-Tenants

- Each tenant needs its own networking domain with its VLAN IDs

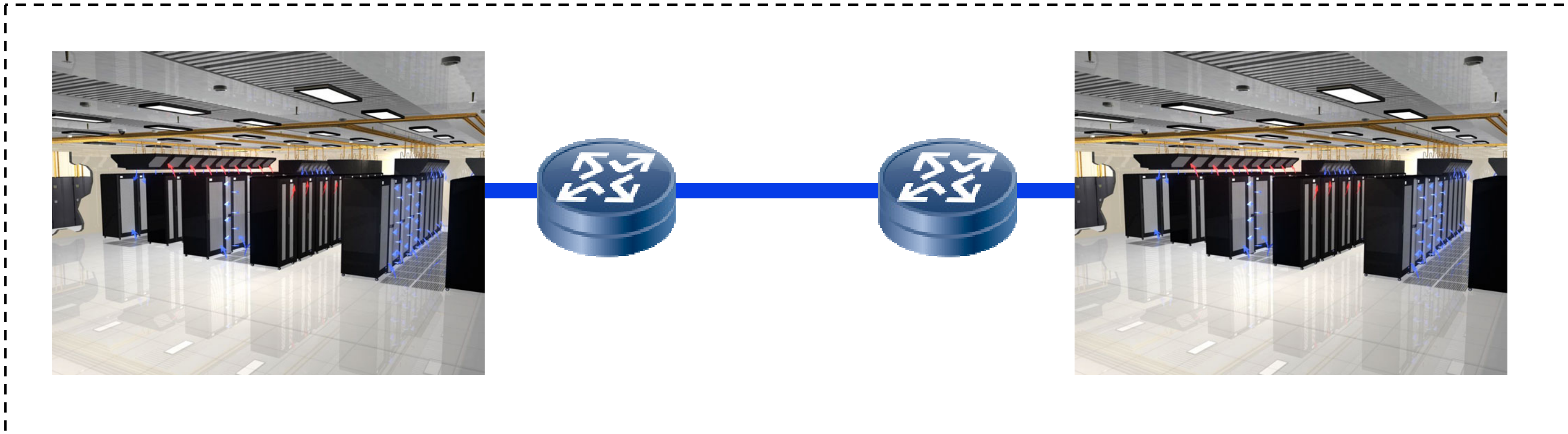


1. Virtual Extensible Local Area Networks (**VXLAN**)
2. Network Virtualization using Generic Routing Encapsulation (**NVGRE**)
3. Stateless Transport Tunneling Protocol (**STT**)

⇒ Network Virtualization over L3 (**NVO3**) group in IETF

Multi-Site

- Better to keep VM mobility in a LAN (IP address changes if subnet changes)



- Solution: IP encapsulation
- Transparent Interconnection of Lots of Links (**TRILL**)

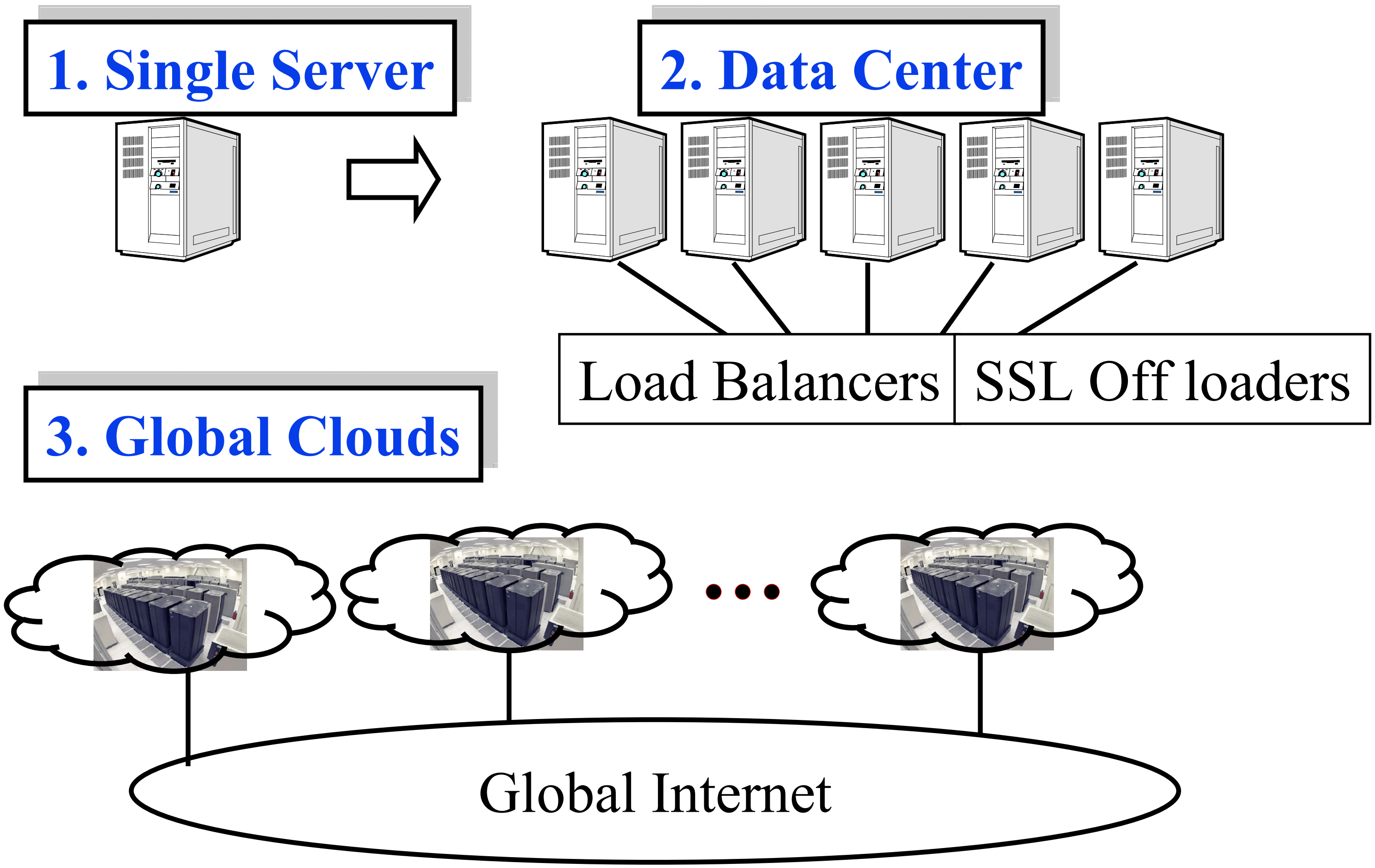
Clouds and Mobile Apps

- August 25, 2006: Amazon announced EC2
  ⇒ Birth of Cloud Computing in reality
  (Prior theoretical concepts of computing as a utility)
- *Web Services To Drive Future Growth For Amazon* ($2B in 2012, $7B in 2019)
  - Forbes, Aug 12, 2012

- June 29, 2007: Apple announced iPhone
  ⇒ Birth of Mobile Internet, Mobile Apps
  - Almost all services are now mobile apps: Google, Facebook, Bank of America, …
  - Almost all services need to be global (World is flat)
  - Almost all services use cloud computing

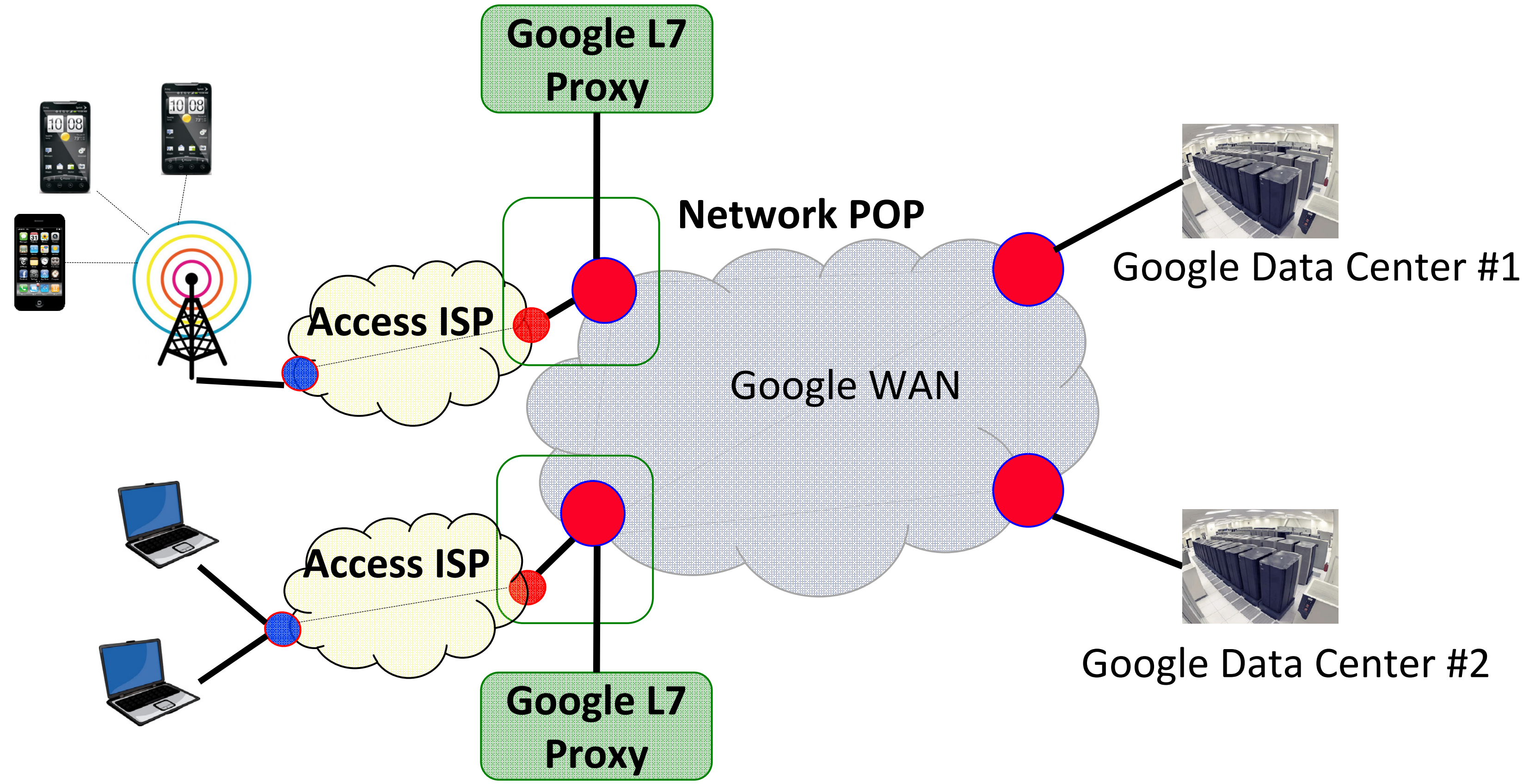
Networks need to support efficient service setup and delivery

Service Center Evolution



Need to make the global Internet look like a data center

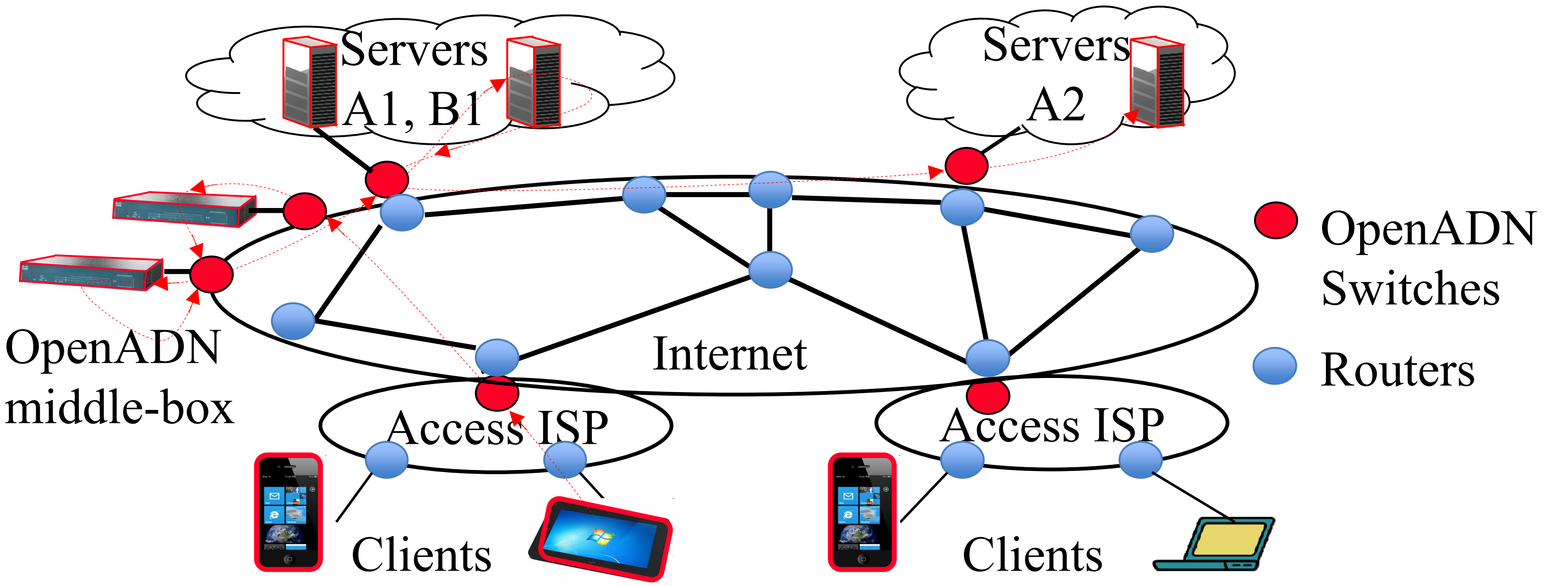
Google WAN



- Google appliances in Tier 3 ISPs
- Details of Google WAN are not public
- ISPs can not use it: L7 proxies require app msg reassembly

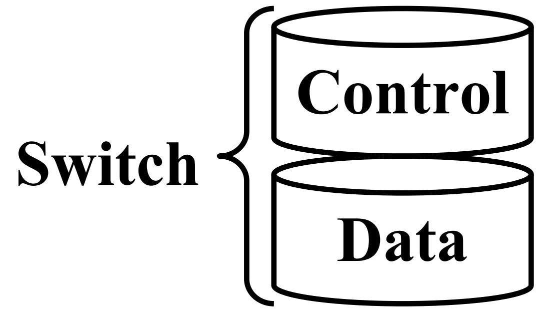
Our Solution: OpenADN

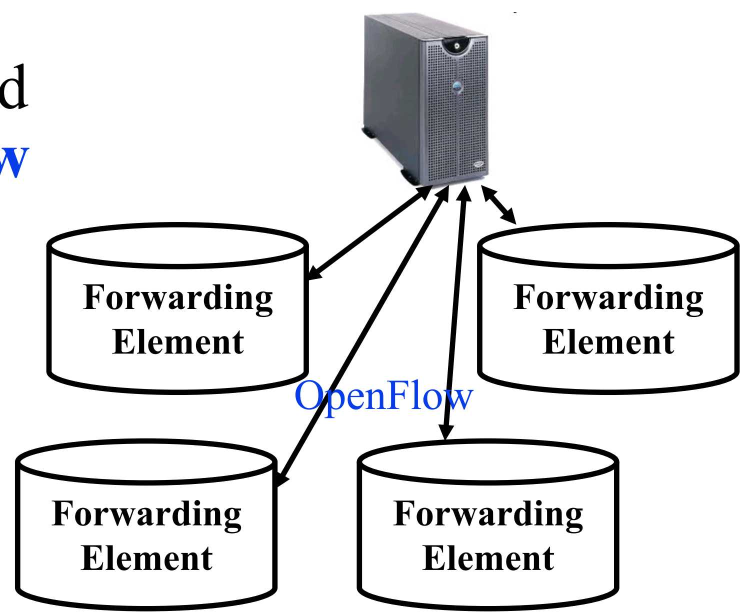
- Open Application Delivery Networking Platform
  Platform = OpenADN aware clients, servers, switches, and middle-boxes
- Allows Application Service Providers (ASPs) to quickly setup services on Internet using cloud computing⇒ Global datacenter



Step 1: Centralization of Control Plane

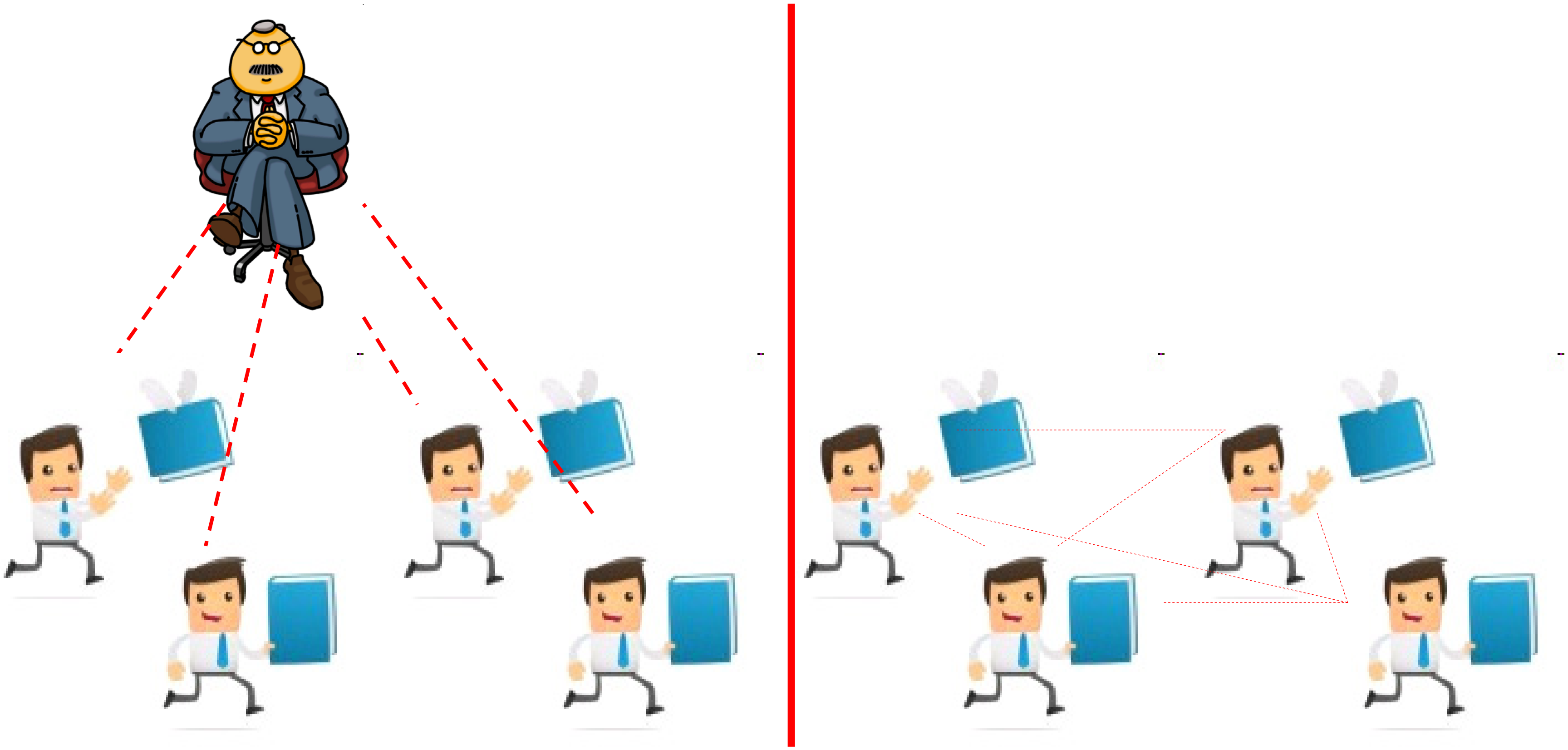
- Control = Prepare forwarding table
- Data Plane: Forward using the table
- Forwarding table is prepared by a central controller
- Protocol between the controller and the forwarding element: **OpenFlow**
- Centralized control of policies
- Switches are simple.
  Controller can be complex
  Can use powerful CPUs
- Lots of cheap switches
  = Good for large datacenters





Ref: [MCK08] ``OpenFlow: Enabling Innovation in Campus Networks," OpenFlow Whitepaper, March 2008
http://www.openflow.org/documents/openflow-wp-latest.pdf

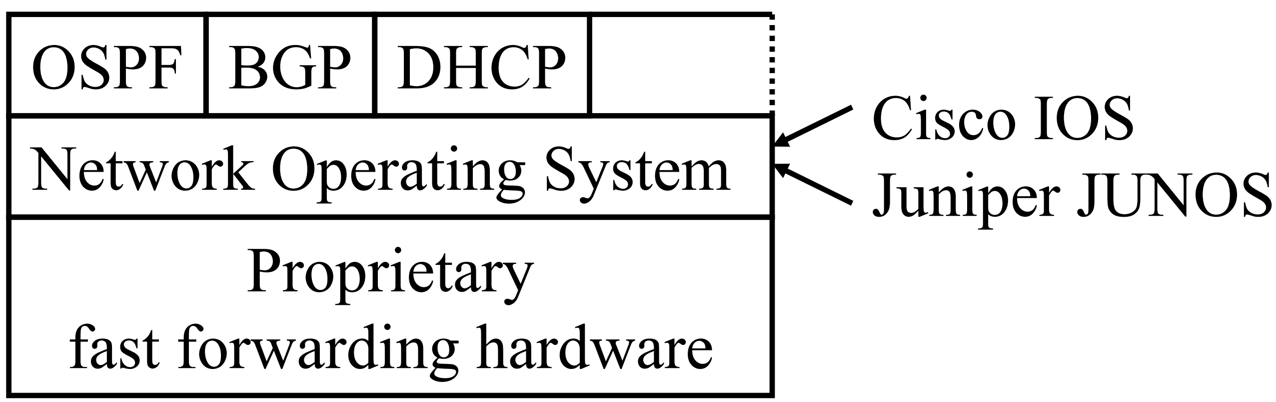
Centralized vs. Distributed



- Fully centralized is not scalable.
  Fully distributed is not manageable.
  ⇒ Hierarchy

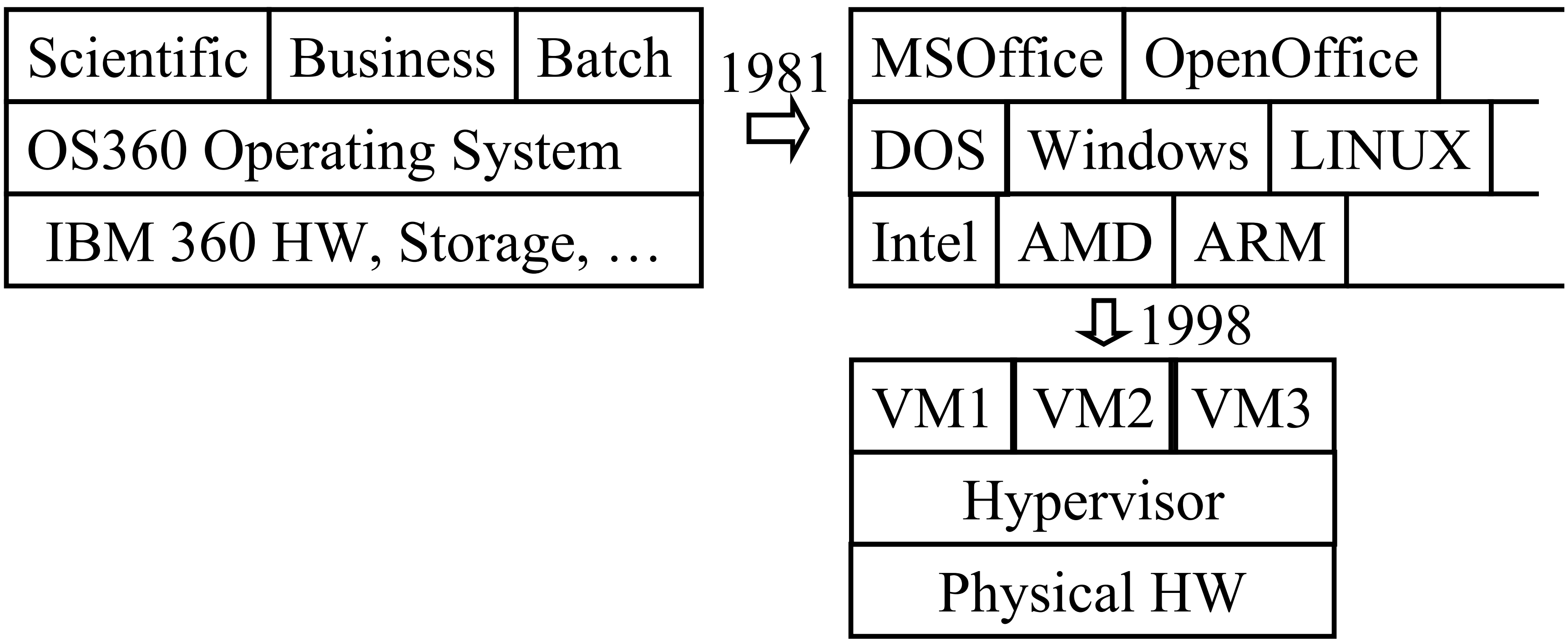
Step 2: Standardized Abstractions

- The routers are expensive because there is no standard implementation.
- Every vendor has its own hardware, operating/ management system, and proprietary protocol implementations.
- Similar to Mainframe era computers.
  No cross platform operating systems (e.g., Windows) or cross platform applications (java programs).



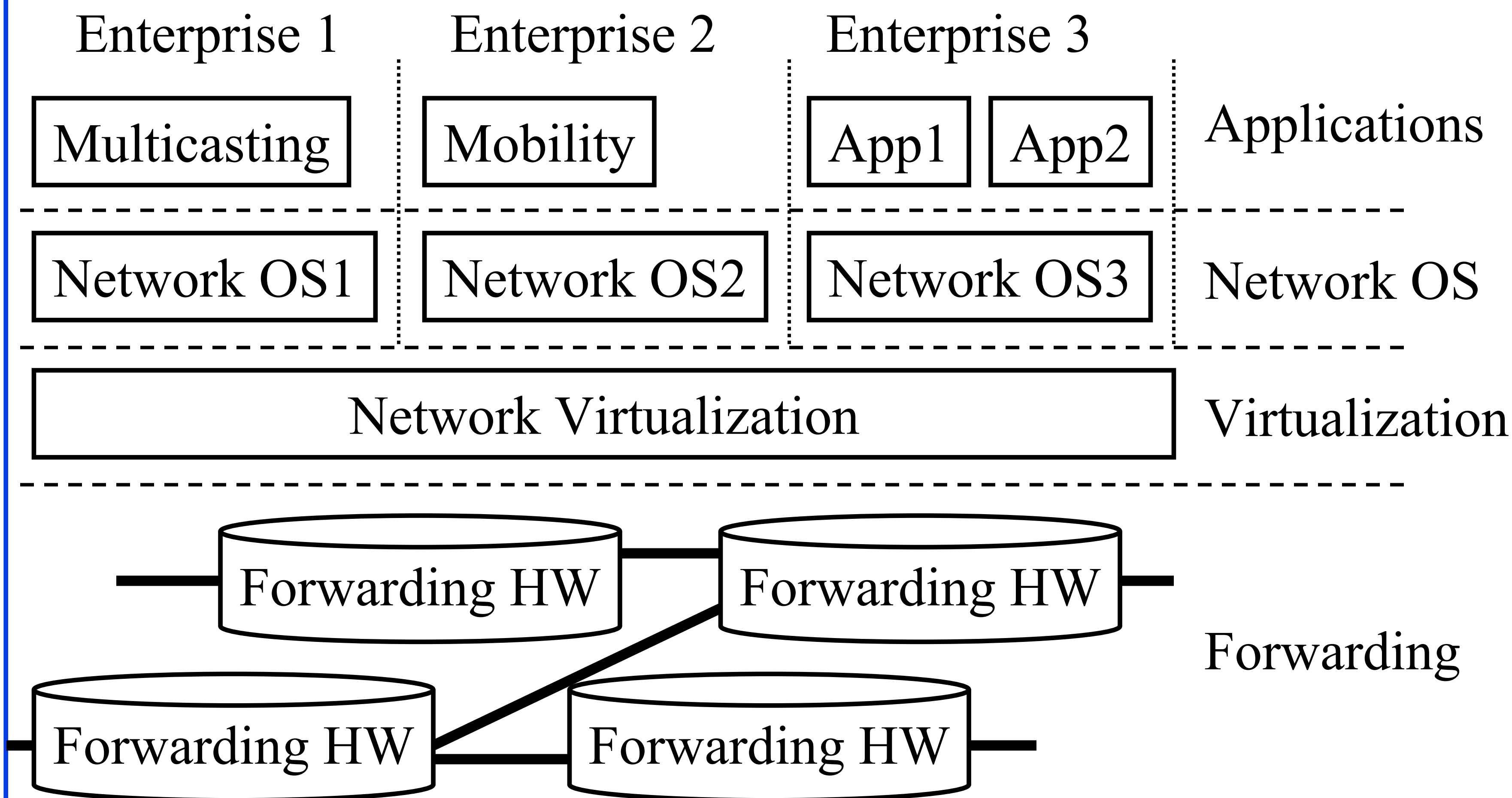
Example: PC Paradigm Shift

- Computing became cheaper because of clear division of hardware, operating system, and application boundaries with well defined APIs between them
- Virtualization ⇒ simple management + multi-tenant isolation



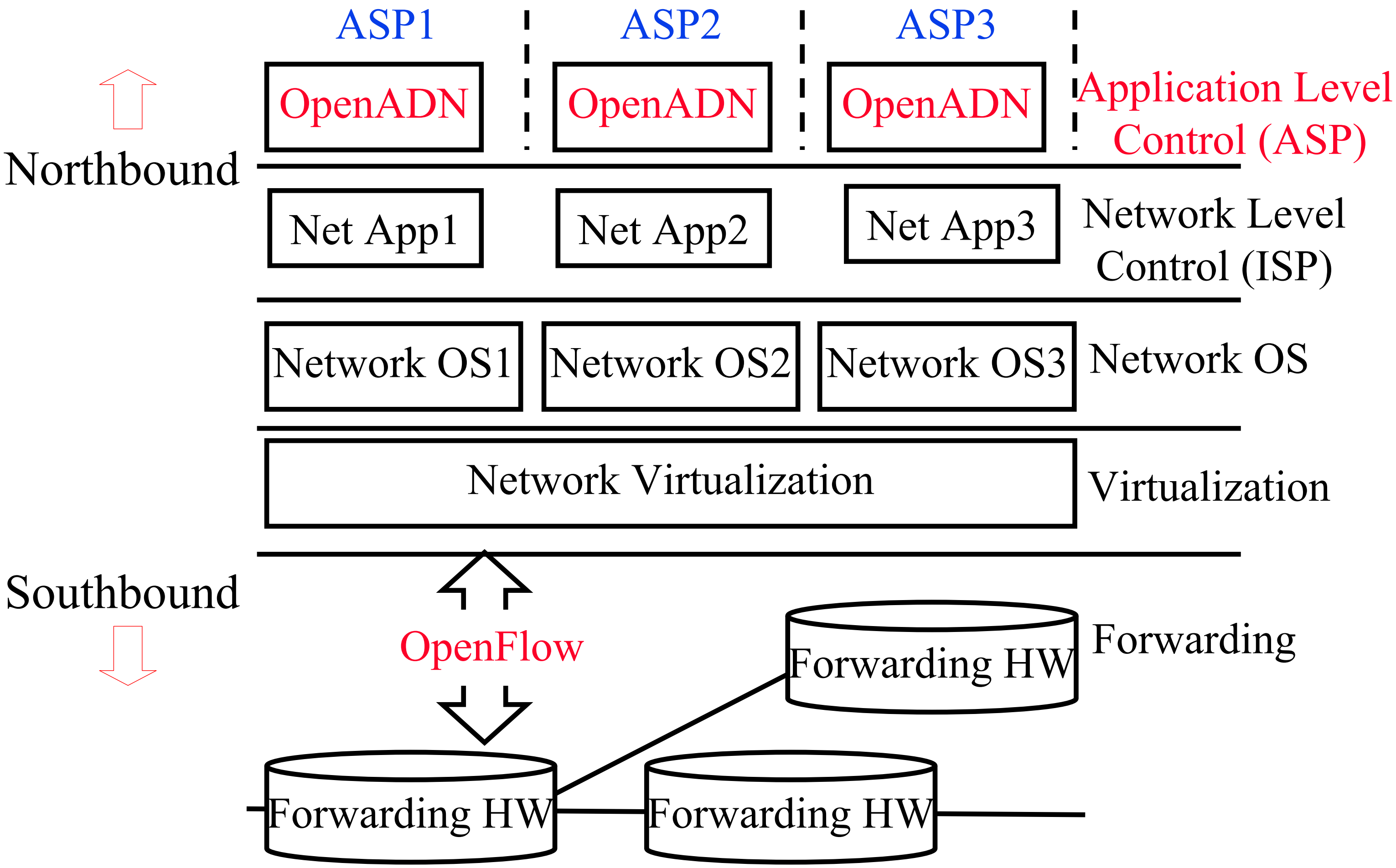
Software Defined Networking

- Layered abstractions with standardized APIs



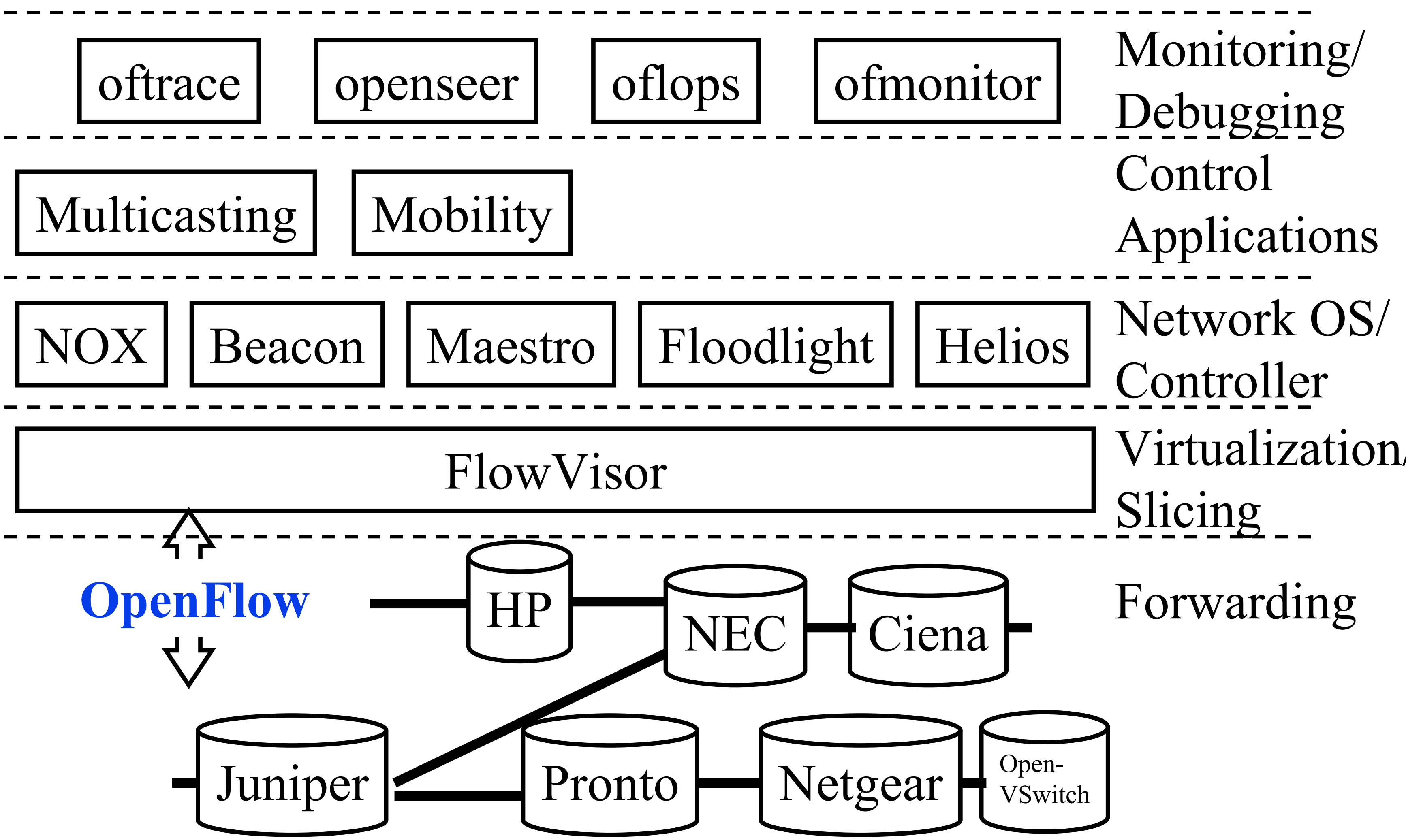
Ref: http://www.itc23.com/.../K1_McKeown-ITC_Keynote_Sept_2011.pdf

SDN's Layered Abstraction



- SDN provides standardized mechanisms for distribution of control information

SDN Architecture Component Examples

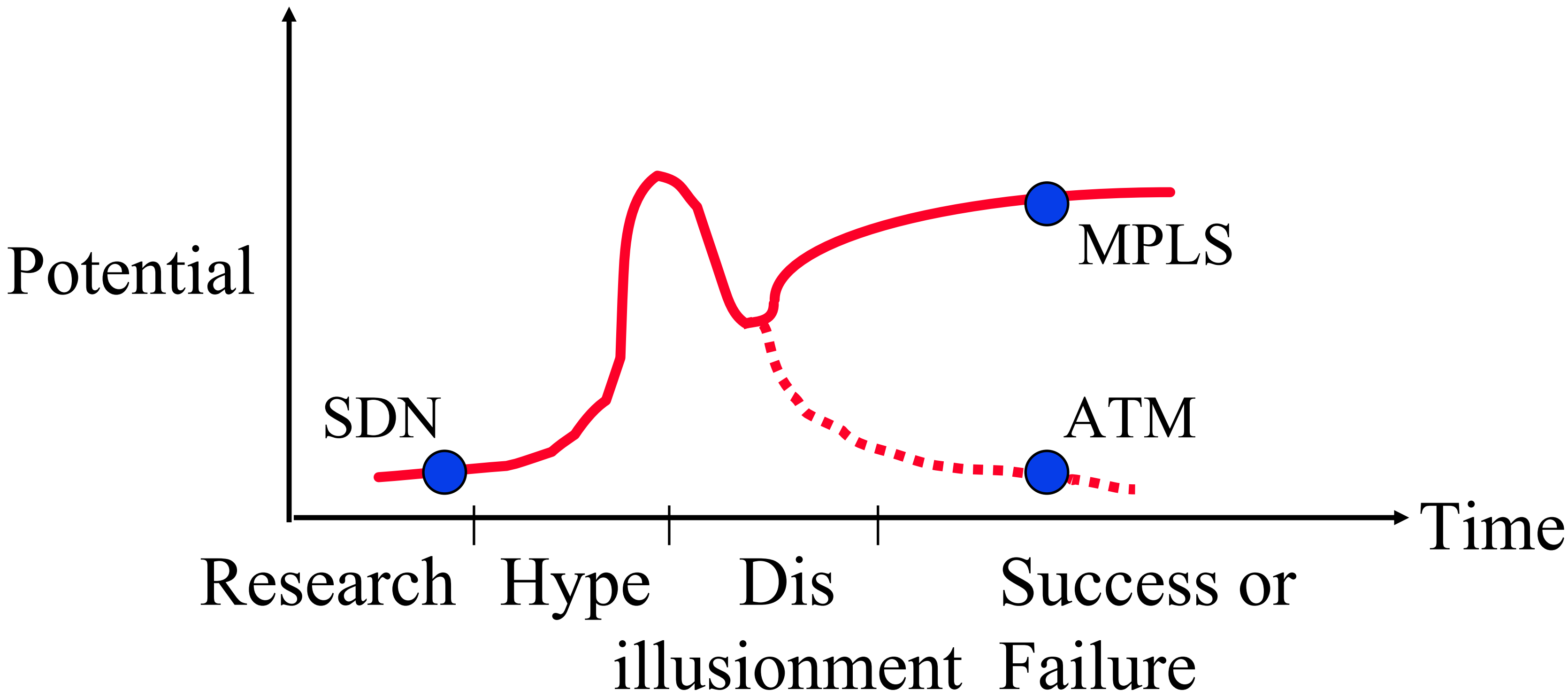


Ref: https://courses.soe.ucsc.edu/courses/cmpe259/Fall11/01/pages/lectures/srini-sdn.pdf
http://www.cse.wustl.edu/~jain/talks/adn_adc.htm

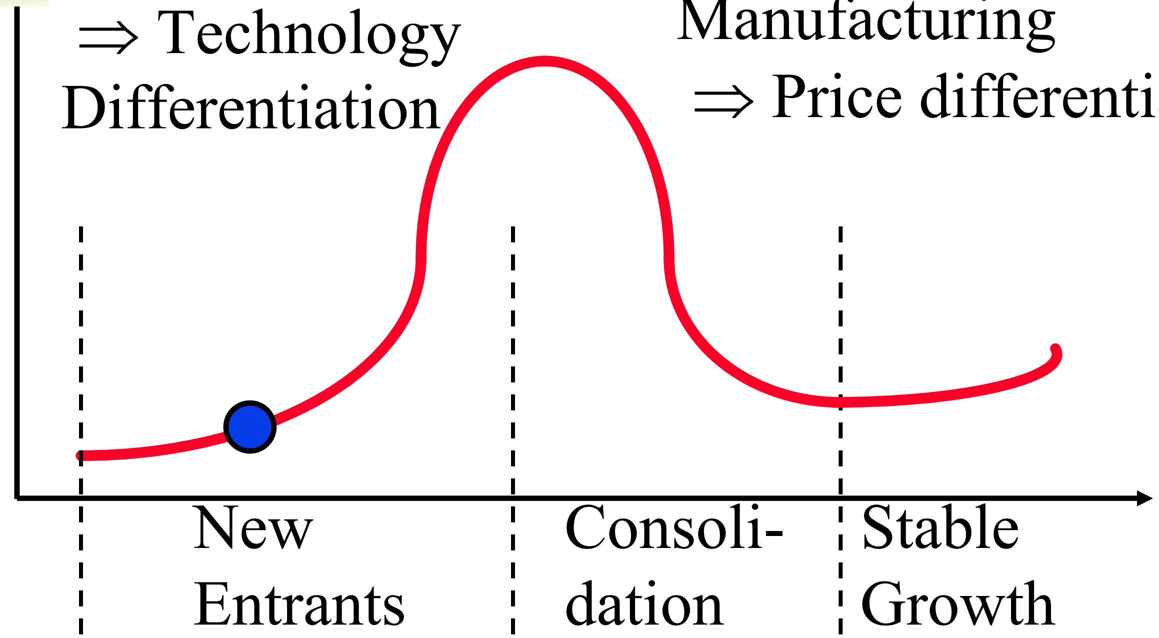
SDN Impact

- Why so much industry interest?
  - Commodity hardware
    ⇒ Lots of cheap forwarding engines ⇒ Low cost
  - Programmability ⇒ Customization
  - Those who buy routers, e.g., Google, Amazon, Docomo, DT will benefit significantly
- Tsunami of software defined devices:
  - Software defined wireless base stations
  - Software defined optical switches
  - Software defined routers

Life Cycles of Technologies

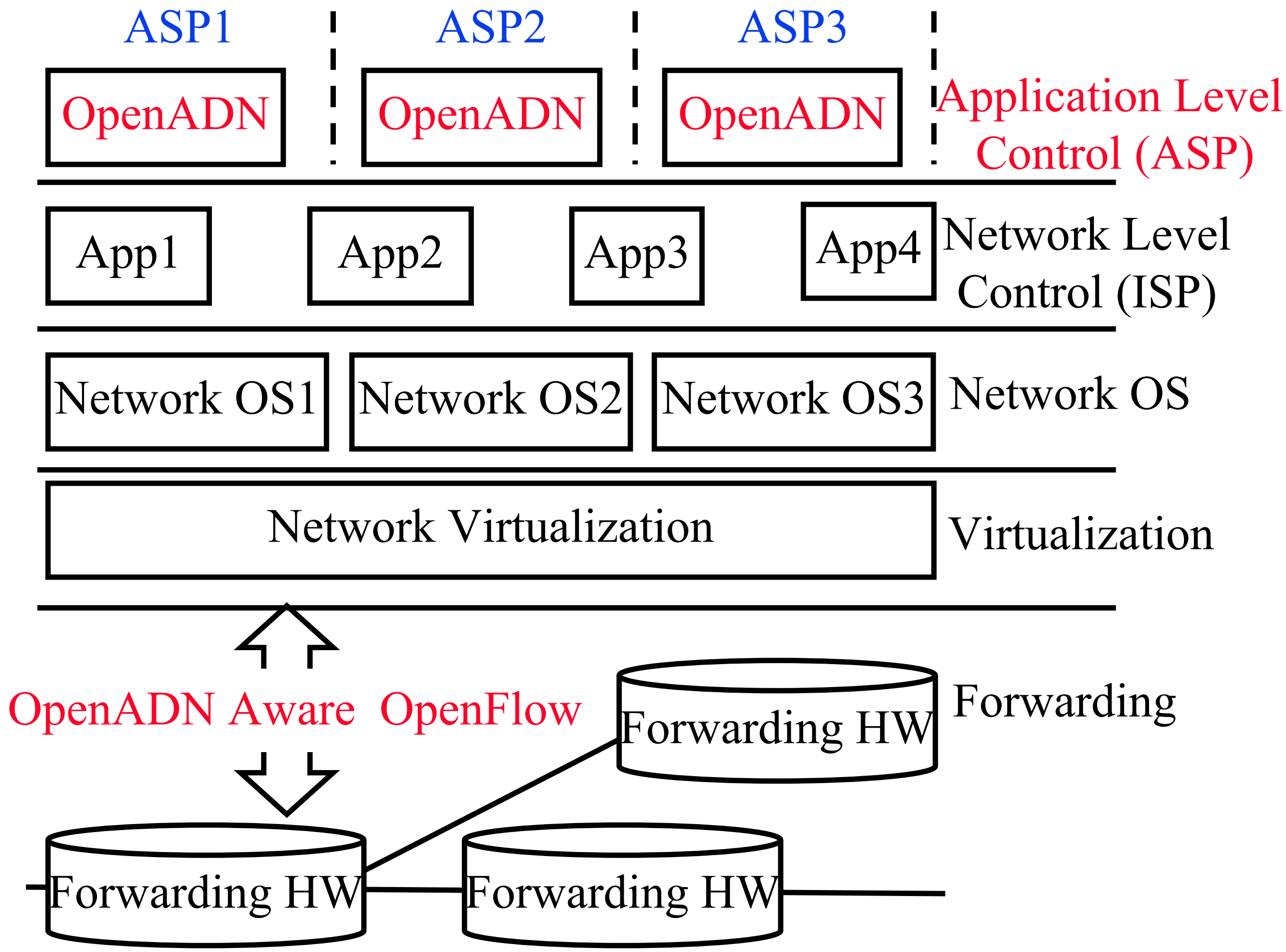


Industry Growth: Formula for Success



- Paradigm Shifts ⇒ Leadership Shift
- Old market leaders stick to old paradigm and loose
- Mini Computers→PC, Phone→Smart Phone, PC→Smart Phone

OpenADN in SDN's Layered Abstraction

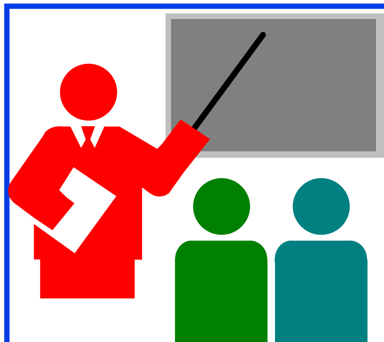


- SDN provides standardized mechanisms for distribution of control information

Key Features of OpenADN

1. Edge devices only.
   Core network can be current TCP/IP based, OpenFlow or future SDN based
2. Coexistence (Backward compatibility):
   Old on New. New on Old
3. Incremental Deployment
4. Economic Incentive for first adopters
5. Resource owners (ISPs) keep complete control over their resources

Most versions of Ethernet followed these principles. Many versions of IP did not.



Summary

1. Cloud computing ⇒ Virtualization of computing, storage, and networking
   ⇒ Numerous recent standards related to networking virtualization both in IEEE and IETF
2. Recent Networking Architecture Trends:
   1. Centralization of Control plane
   2. Standardization of networking abstractions
      ⇒ Software Defined Networking (SDN)
   3. Most networking devices will be software defined
3. OpenADN enables delivery of applications using North-bound SDN API

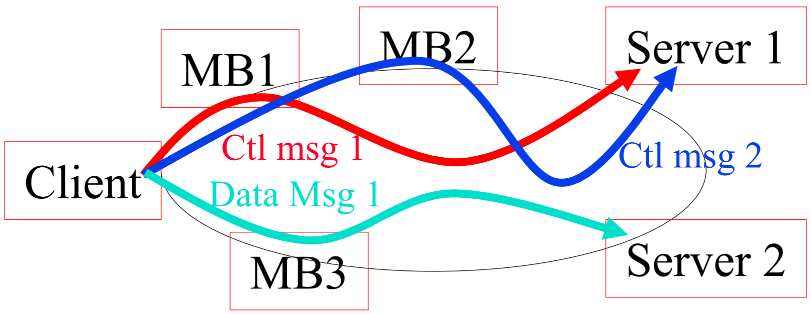
Thank You!

धन्यवाद

ಧನ್ಯವಾದ നന്ദി நன்றி

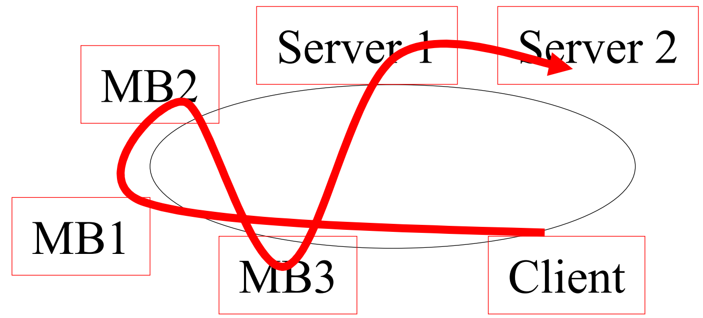
OpenADN vs. Serval

1. Message-Level Granularity



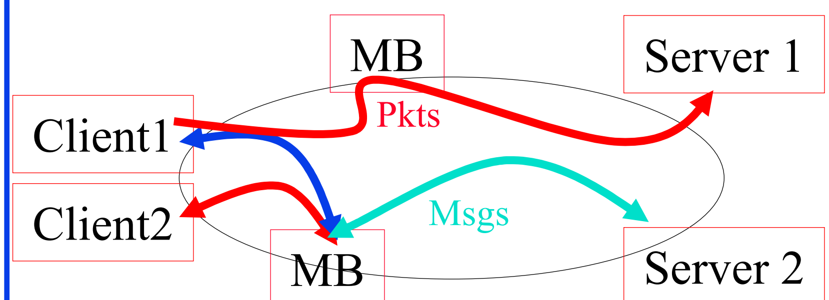
MB = Middle Box

2. Sequence of Middle and End entities



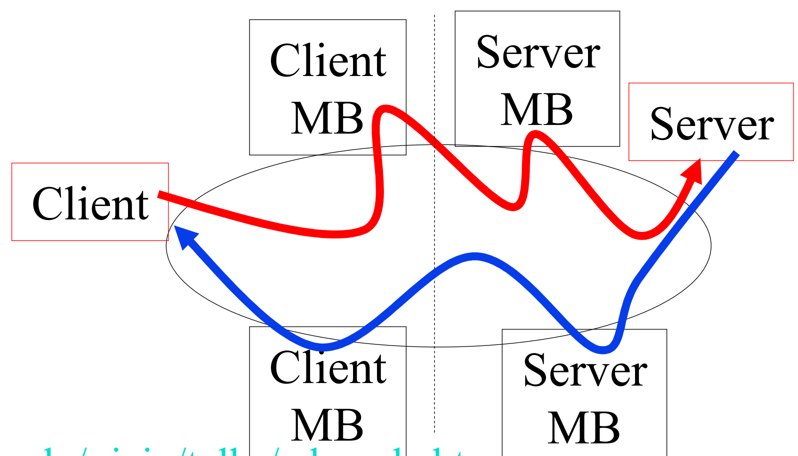
Client →MB1 → MB2 → MB3 → Server1 → Server2

3. Packet & Message-Level MBs



TCP Splicing

4. Receiver & Sender Policies



http://www.cse.wustl.edu/~jain/talks/adn_adc.htm

Extension 5: Cross-Layer Communication

- Application puts a "label" in "Application Label Switching (APLS) layer "3.5" (between IP and TCP header)
- Like MPLS which is layer "2.5"

L2 Header	[L2.5 Header]	L3 Header	**APLS Header**	L4 Header	L5 Hdr+Data

Encrypted
If required

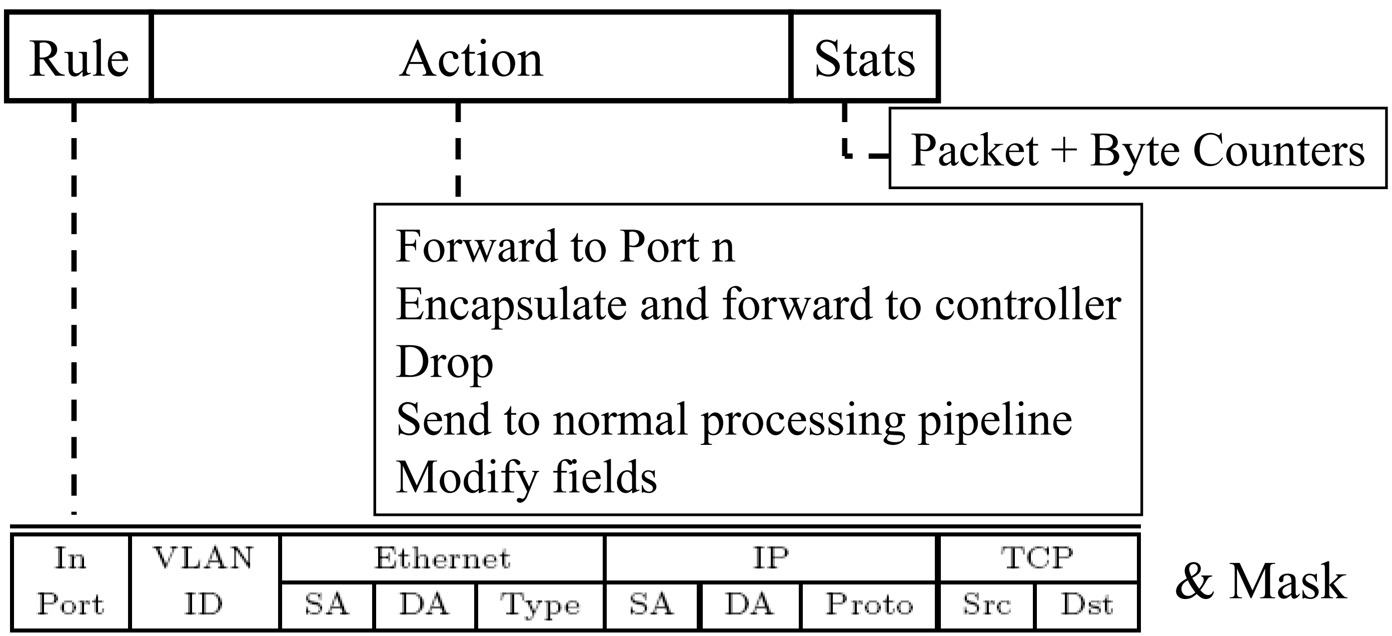
- Legacy routers forward based on L3 or L2.5 header
- Only Applications (user and server) and openADN appliances and middle boxes read/write APLS labels
- L3 protocol type field indicates the presence of APLS header
- APLS header protocol type field indicates L4 protocol: could be TCP, UDP, SCTP, … ⇒Works with all L4 Protocols,
  - Works with IP, MPLS, …

Cross-Layer Communication (Cont)

- APLS header allows:
  - Session Affinity: All packets go to the same server
  - Sender policy: send this through video translator
  - Receiver Policy: Load balancing
  - Network Policy: QoS
  - Forwarding through appropriate set of middle boxes

OpenFlow (Cont)

- Three Components:
  - Flow table: How to identify and process a flow
  - Secure Channel: Between controller and the switch
  - Open Flow Protocol: Standard way for a controller to communicate with a switch



OpenFlow (Cont)

- Controller forwards the packets correctly as the mobile clients move
- Reference designs for Linux, Access points (OpenWRT), and NetFPGA (hardware)
- Allows both proactive (flow tables loaded before hand) and reactive (Flow entries loaded on demand)
- Allows wild card entries for aggregated flows
- Multiple controllers to avoid single point of failure: Rule Partitioning, Authority Partitioning
- Open Networking Foundation announced Open Switch Specification V1.2 on Jan 29, 2012: Includes IPv6 and experimenter extensions.

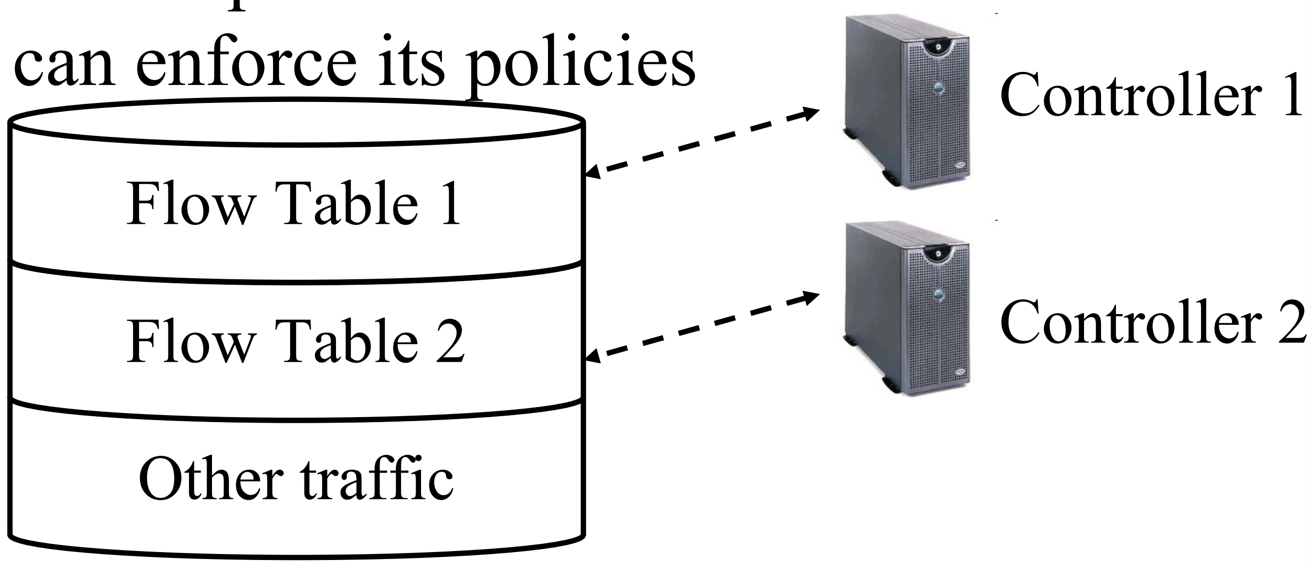
Ref: [MCK08], OpenFlow.org, OpenNetworking.org

Why worry about Future Internet?

Billion dollar question!

Step 2: Multi-Tenants Clouds

- Problem: Multiple tenants in the datacenter
- Solution: Use multiple controllers.
  Each tenant can enforce its policies



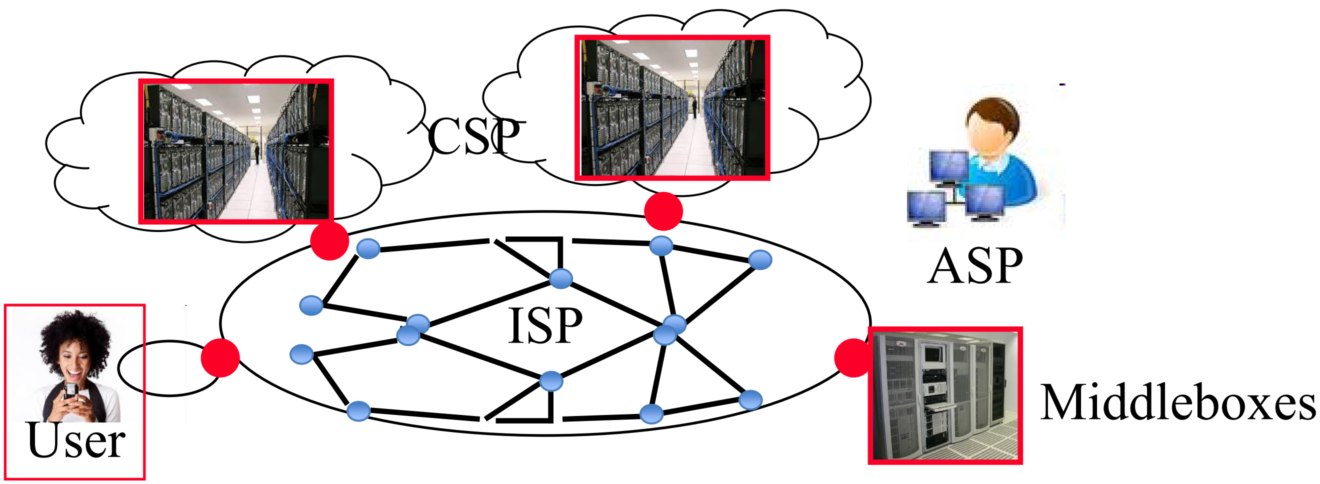
- Significant industry interest ⇒ Open Networking Foundation, https://www.opennetworking.org/

Resource Control

- ASPs keep complete control of their data.
  ISP does not have to look at the application headers or data to enforce application level policies
- ISPs keep complete control of their equipment.
  ASPs communicate their policies to ISP's control plane
- Middle boxes can be located anywhere on the global Internet (Of course, performance is best when they are close by)
- ISPs own OpenADN switches and offer them as a service
- ASPs or ISPs can own OpenADN middle boxes
- No changes to the core Internet

Beneficiaries of This Technology

- Equipment/Software vendors: OpenADN-aware appliances
- ASPs: Deploy servers anywhere and move them anytime
- ISPs: Offer new application delivery/middlebox services
- Cloud Service Providers (CSPs): Freedom to move VMs, Less impact of downtime
- CDNs, e.g., Akamai, can extend into application delivery



OpenADN Innovations

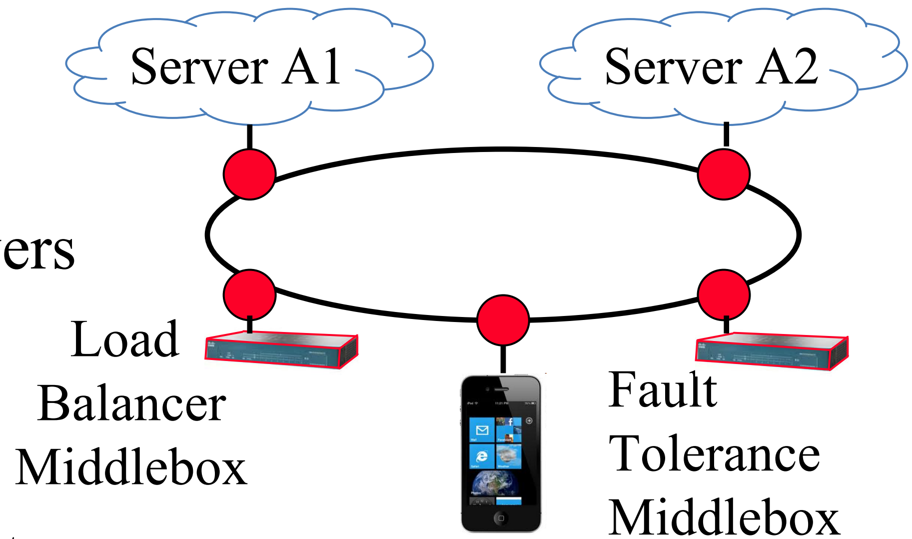
1. Cross-Layer Communication
2. MPLS like Labels
3. Extended OpenFlow flow-based handling, centralized policy control
4. Software Defined Networking: Standardized abstractions, Multi-Tenants, Control Plane programming for data plane
5. ID/Locator Split
6. Layer 7 Proxies without layer 7 visibility

Networking: Failures vs Successes

- 1986: MAP/TOP (vs Ethernet)
- 1988: OSI (vs TCP/IP)
- 1991: DQDB
- 1994: CMIP (vs SNMP)
- 1995: FDDI (vs Ethernet)
- 1996: 100BASE-VG or AnyLan (vs Ethernet)
- 1997: ATM to Desktop (vs Ethernet)
- 1998: ATM Switches (vs IP routers)
- 1998: MPOA (vs MPLS)
- 1999: Token Rings (vs Ethernet)
- 2003: HomeRF (vs WiFi)
- 2007: Resilient Packet Ring (vs Carrier Ethernet)
- IntServ, DiffServ, …

Technology alone does not mean success.

OpenADN Features



Message level:

- Server selection
- Load balancing between servers
- Fault tolerance
- Server mobility
- User Mobility
- Secure L5-L7 headers and data
- Middlebox services: Intrusion detection, Content based routers, application firewalls, …
  - Control plane and data plane MBs
- Middlebox traversal sequence
- Message level policies
- TCP Splicing