OpenADN: Mobile Apps on Global Clouds Using OpenFlow and SDN

Raj Jain

Project Leader: Subharthi Paul
Washington University in Saint Louis
Saint Louis, MO 63130
Jain@cse.wustl.edu

Microsoft Corporation Seminar, Bellevue, WA

April 24, 2012

Overview

1. Networking Application Trends
2. OpenFlow and SDN
3. OpenADN Vision and Extensions
4. Experimental Results
5. Key Features

Trend: Explosion of Mobile Apps



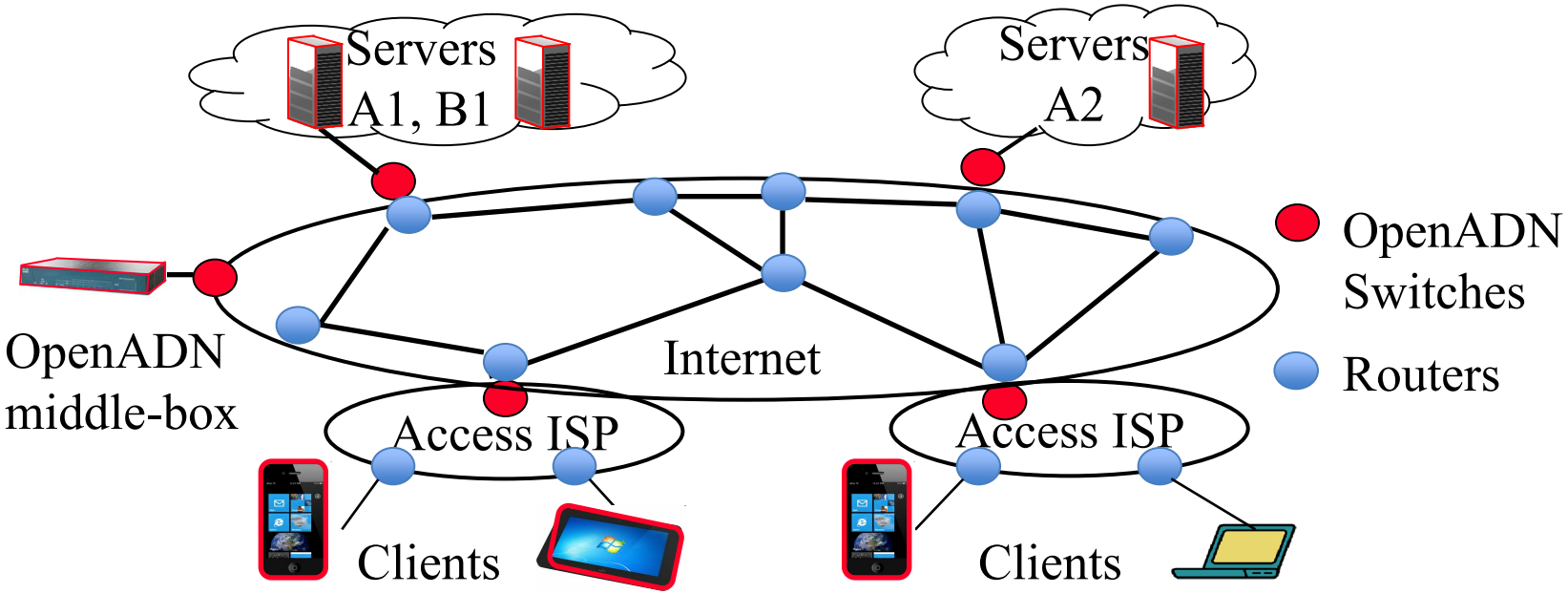

- All top 50 Internet sites are services [Alexa]
- Almost all services are now mobile apps: Google, Facebook, Bank of America, …
- Almost all services need to be global (World is flat)
- Almost all services use cloud computing (Easy management)

Networks need to support efficient service setup and delivery

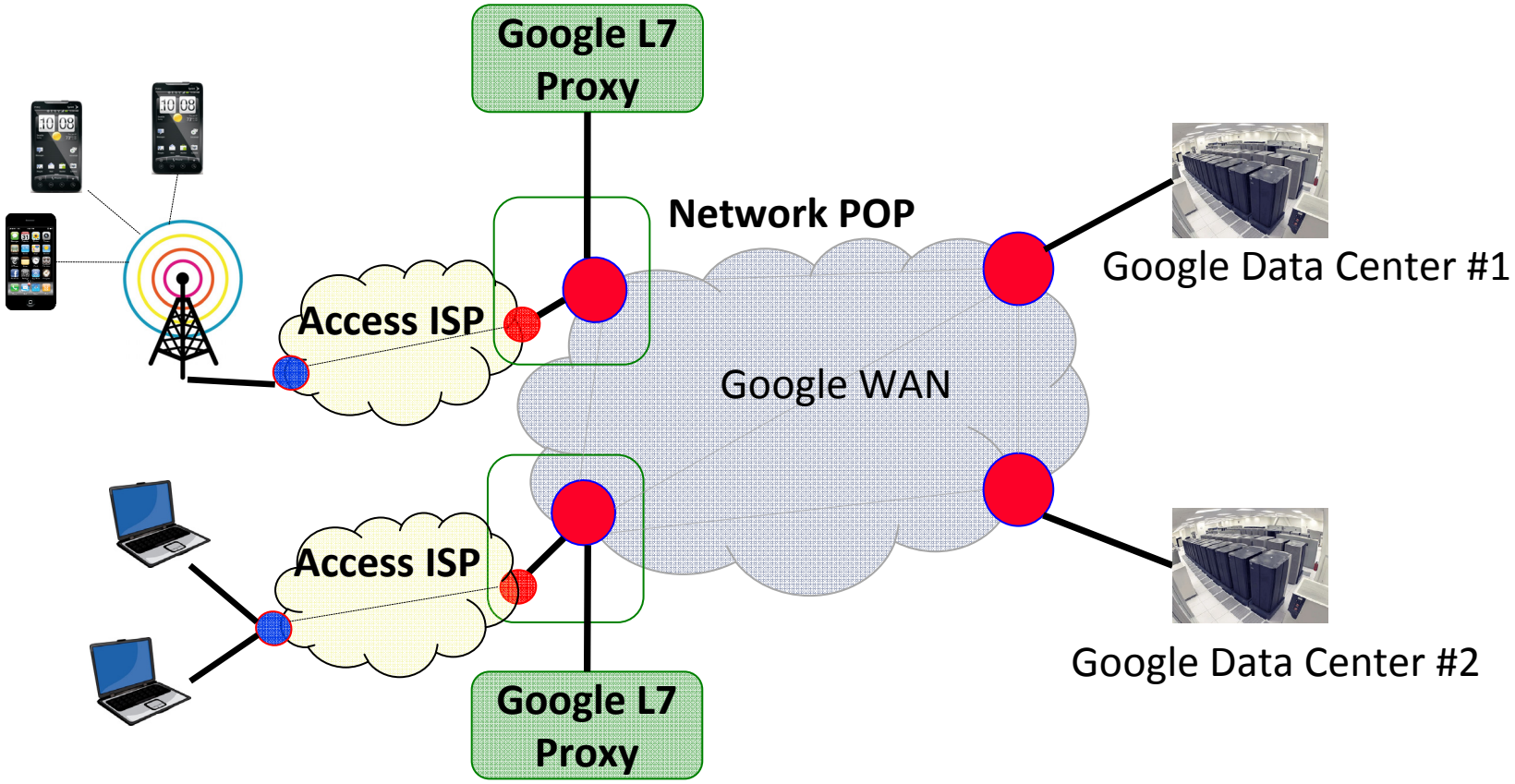
Ref: Top 500 sites on the web, http://www.alexa.com/topsites

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



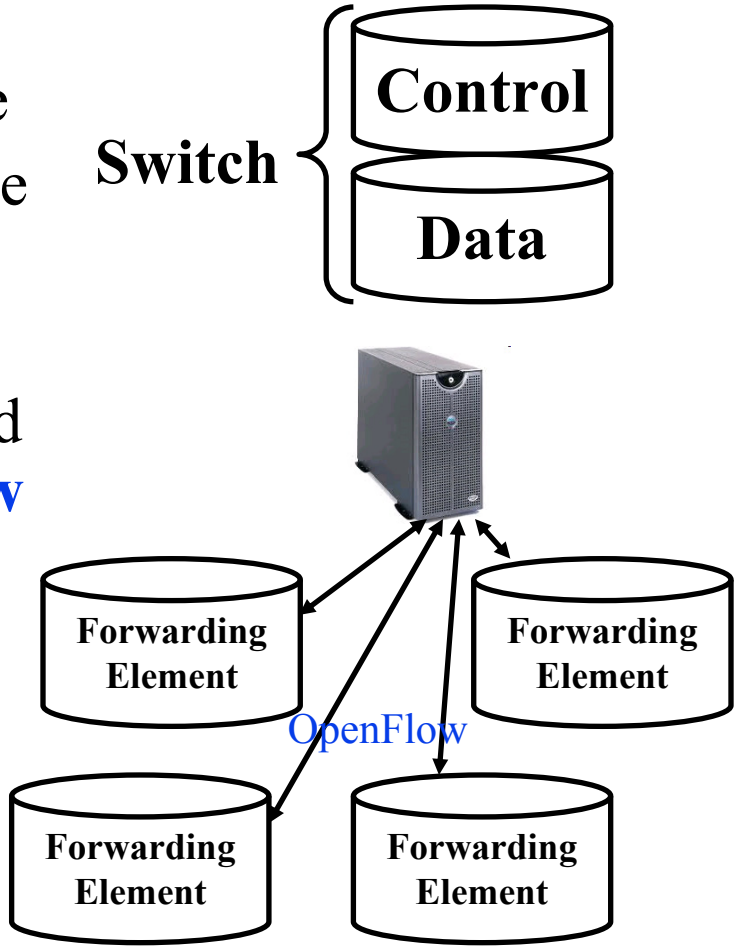
Google WAN



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

Trend: Separation of Control and Data Planes

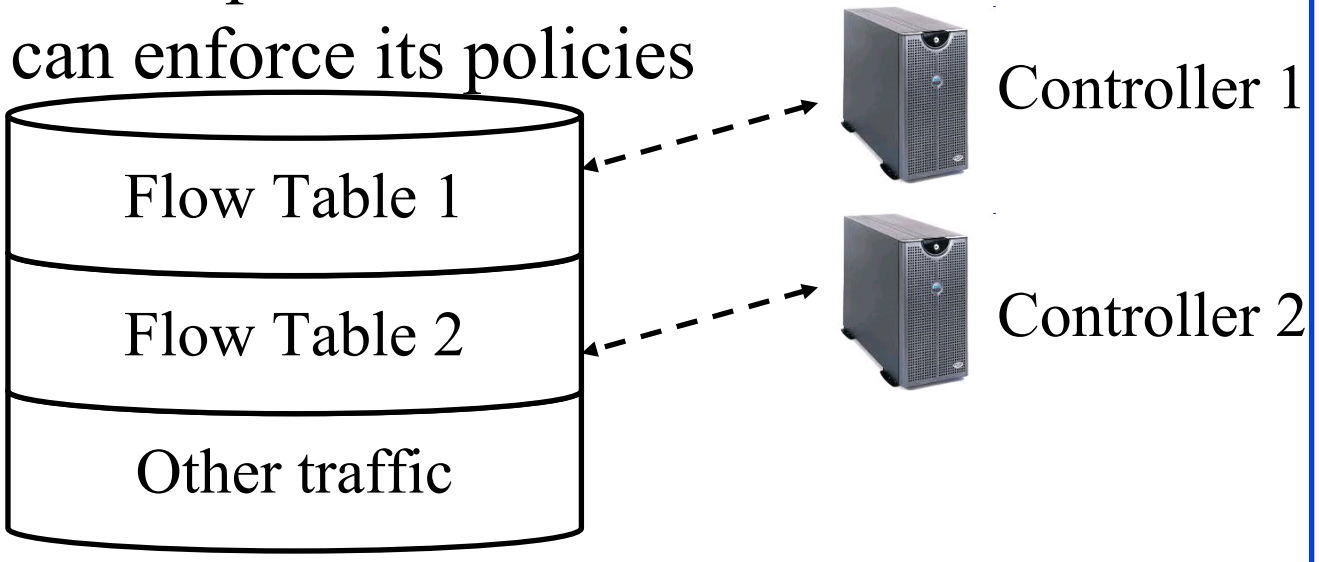
- Control = Prepare forwarding table
- Data Plane: Forward using the table
- Forwarding table is prepared by a central controller
- Protocol between the controller and the forwarding element: **OpenFlow**
- Centralized control of policies
- Switches are simple.
  Controller can be complex
  Can use powerful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

Trend: Multi-Tenants Clouds

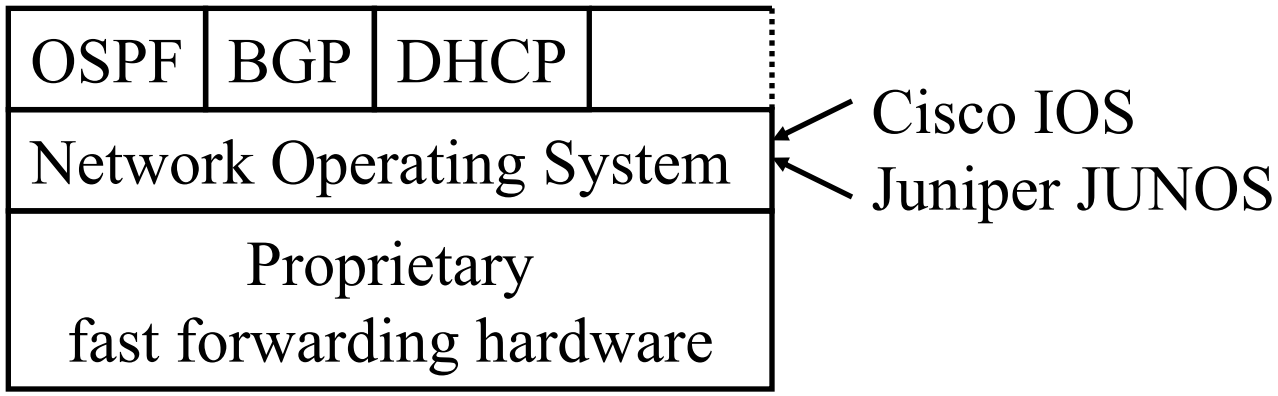
- Problem: Multiple tenants in the datacenter
- Solution: Use multiple controllers.
  Each tenant can enforce its policies



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

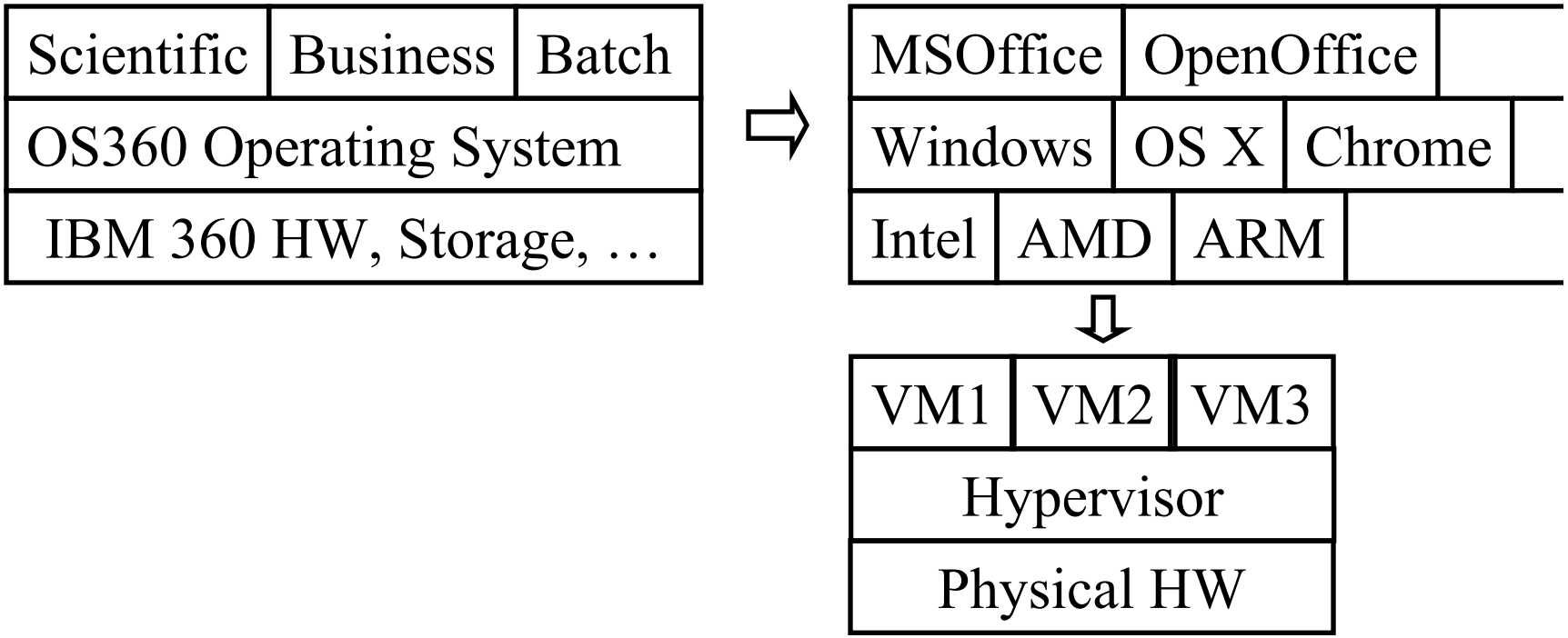
Problem: Complex Routers

- 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).



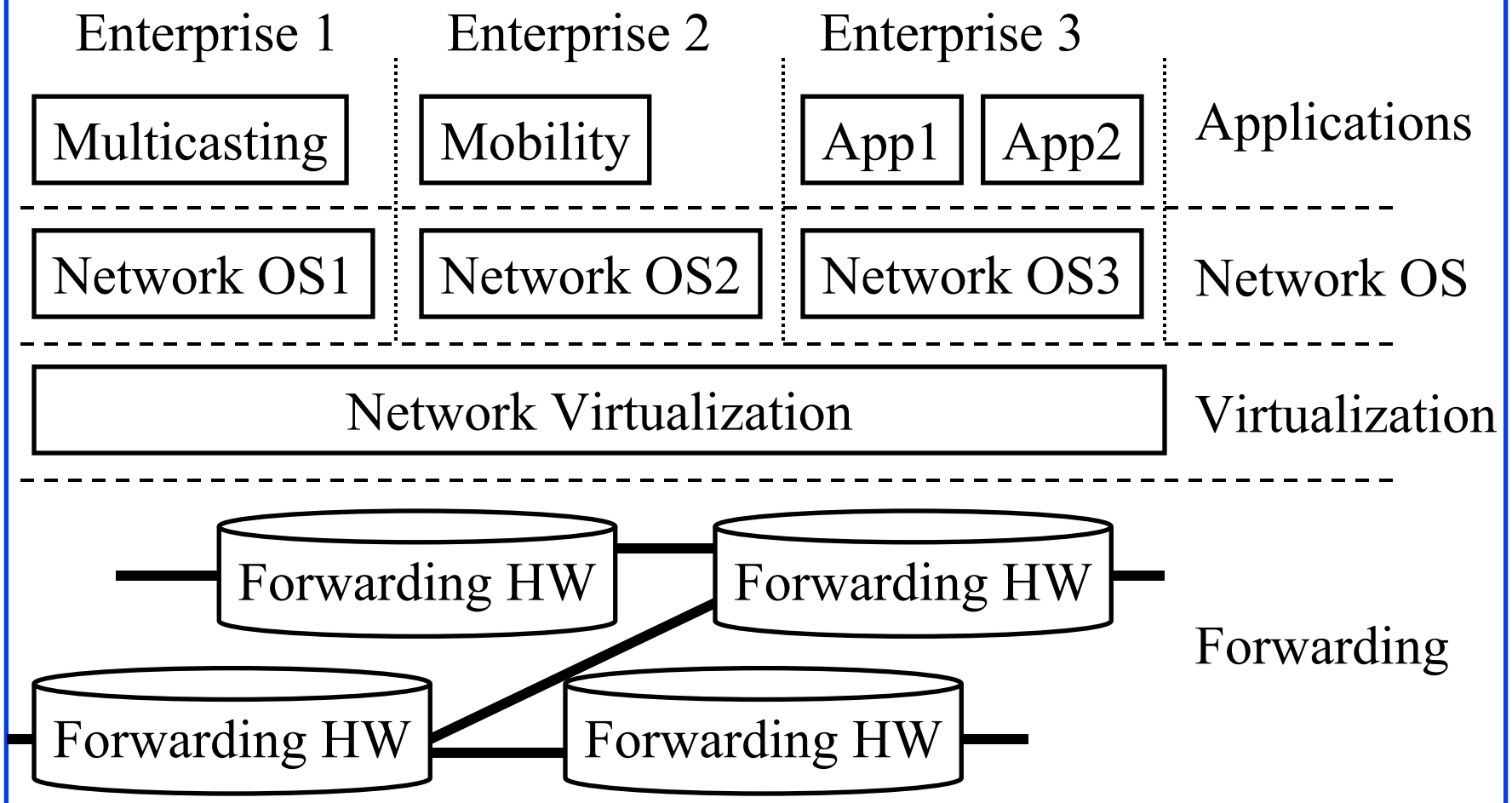
Solution: Divide, Simplify and Standardize

- 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

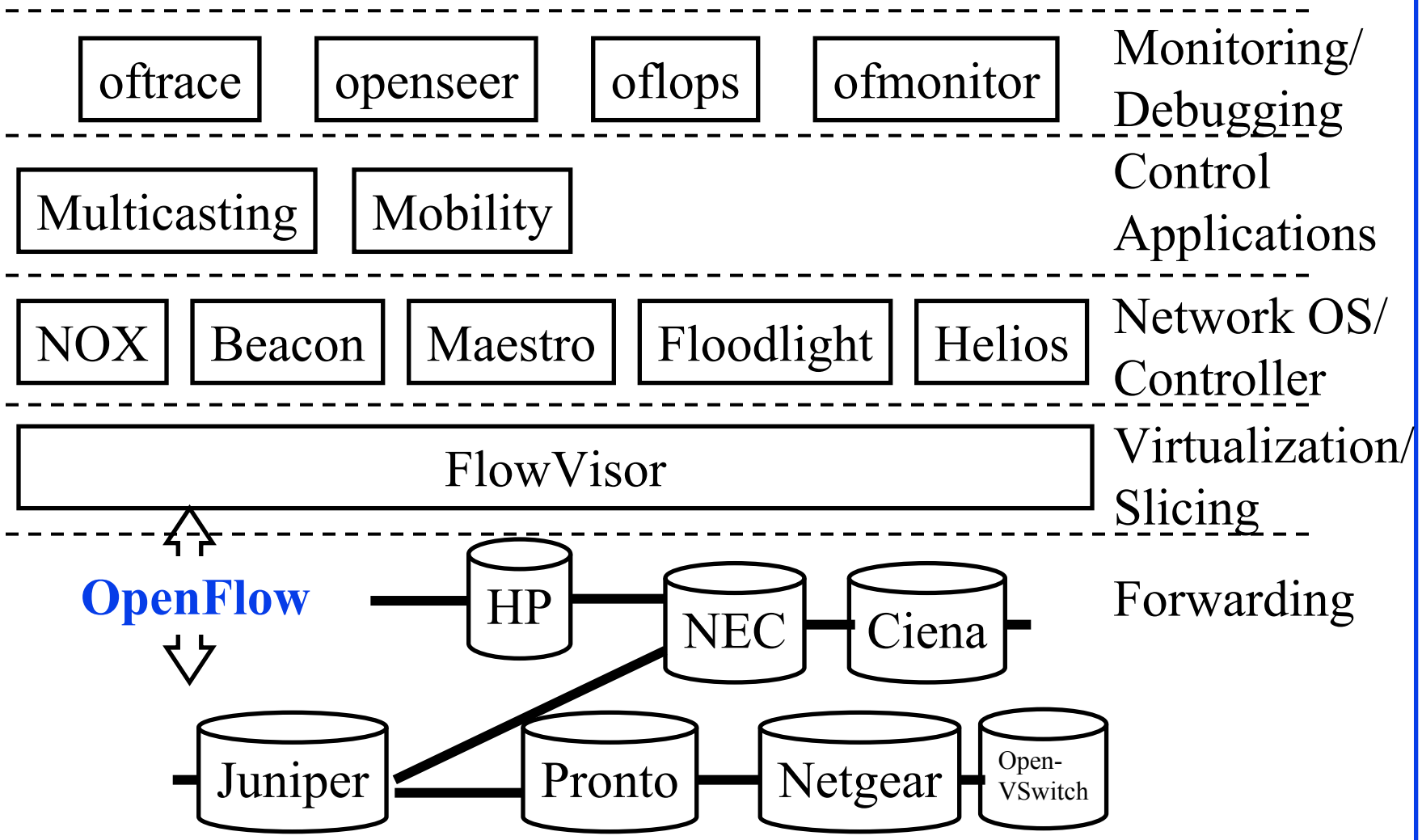


Trend: Software Defined Networking

- Layered abstractions with standardized APIs



SDN Architecture Component Examples

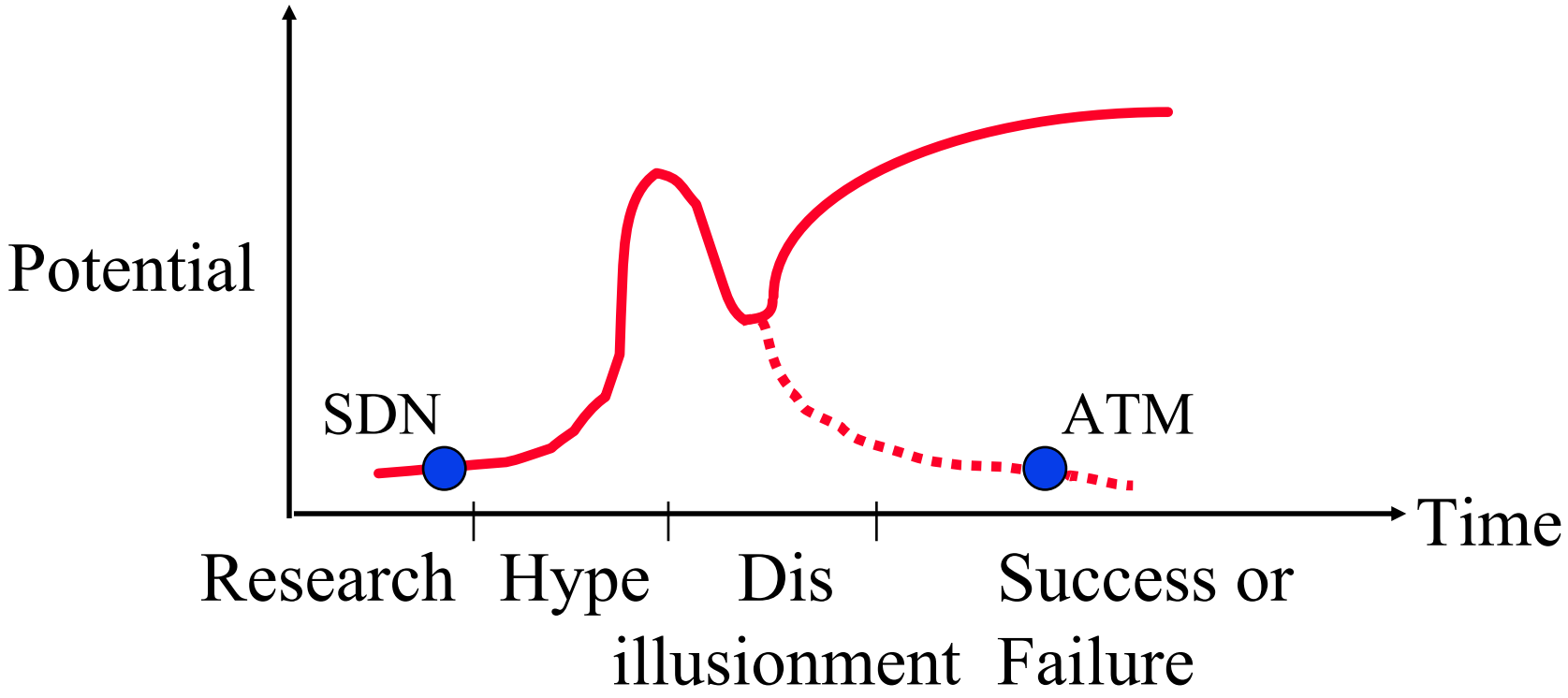


Ref: https://courses.soe.ucsc.edu/courses/cmpe259/Fall11/01/pages/lectures/srini-sdn.pdf

SDN Impact

- Why so much industry interest?
  - Commodity hardware
    ⇒ Lots of cheap forwarding engines ⇒ Low cost
  - Programmability ⇒ Customization
  - Sharing with Isolation ⇒ Networking utility
  - Those who buy routers, e.g., Google, Amazon, Docomo, DT will benefit significantly
- Opens up ways for new innovations
  - Dynamic topology control: Turn switches on/off depending upon the load and traffic locality
    ⇒"Energy proportional networking"

Life Cycles of Technologies



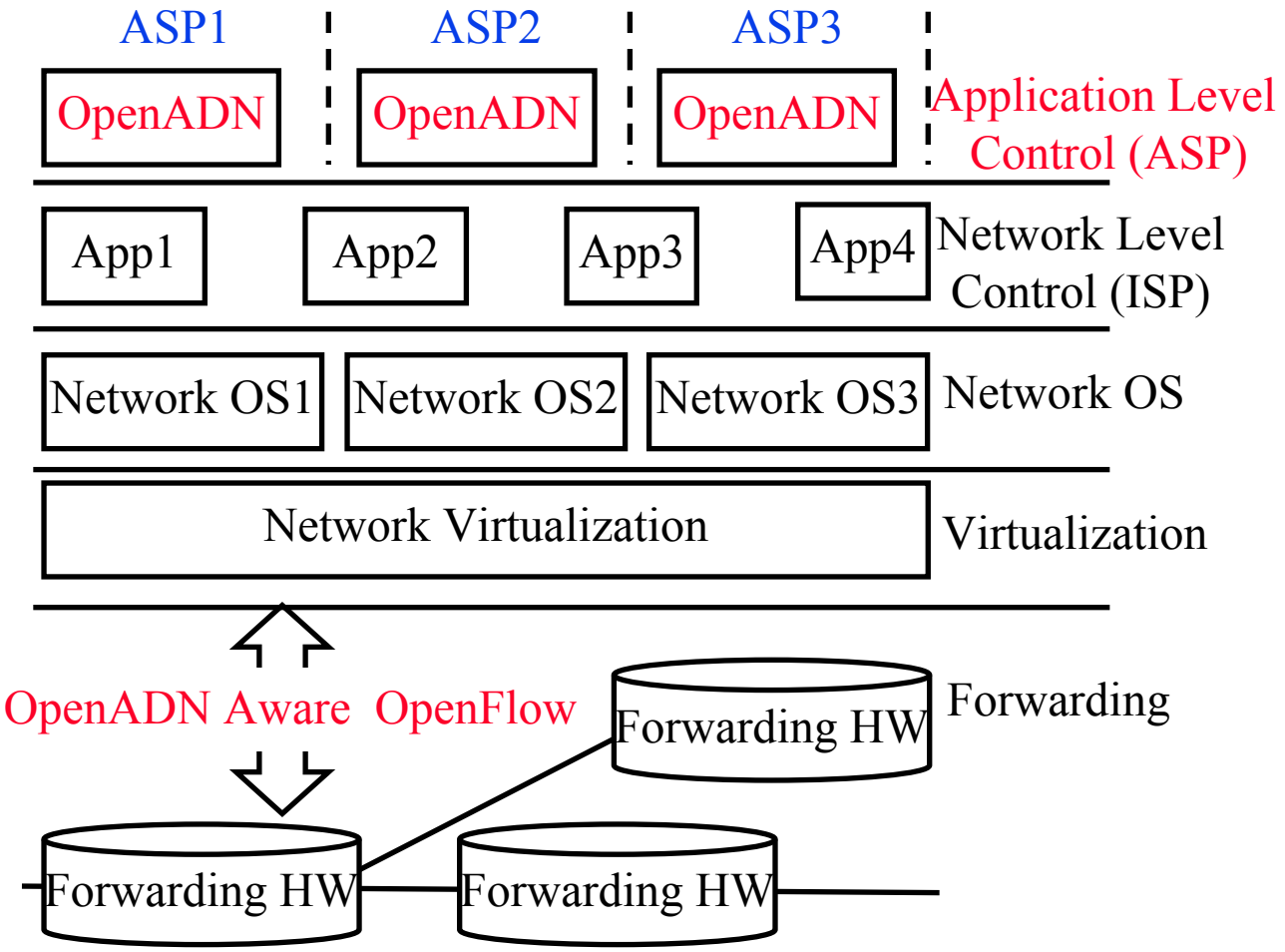
Industry Growth: Formula for Success

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

OpenFlow and SDN: Key Features

- OpenFlow:
  - Classify packets into flows by header fields
  - Apply policies to flows
  - Policies are defined in the central controller
  - Open Flow protocol allows communicating policies from control plane to data plane
  - Multi-tenant: Multiple controllers
- SDN:
  - Standardized abstraction layers and APIs
  - Standardized view of the distributed network
  - Hardware independent networking applications

OpenADN in SDN's Layered Abstraction



- SDN provides standardized mechanisms for distribution of control information

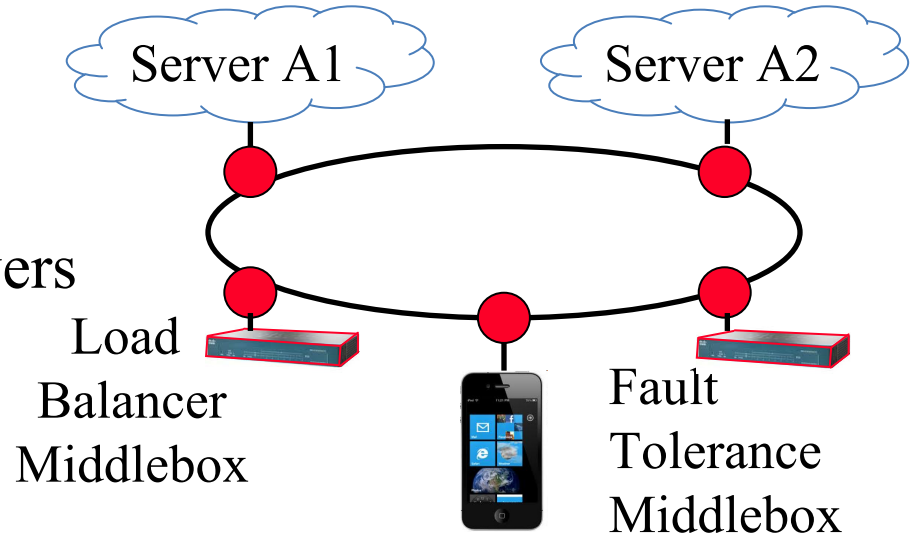
OpenADN Innovations

1. Extended OpenFlow flow-based handling, centralized policy control
2. Software Defined Networking: Standardized abstractions, Multi-Tenants, Control Plane programming for data plane
3. ID/Locator Split
4. Layer 7 Proxies (Similar to Google's proxies in Access PoPs)
5. Cross-Layer Communication
6. MPLS like Labels

Extension 1: Application Level Policies

ASPs want:

- Server selection
- Load balancing between servers
- Fault tolerance
- Server mobility
- User Mobility
- Secure L5-L7 headers and data (rat hole)
- Middlebox services: Intrusion detection, Content based routers, application firewalls, …



Extension 2: Application vs. Network Flow Classes

- Network-level Policies:
  - Specified by ISPs, e.g., routing, traffic engineering, congestion control, …
  - Applied to all packets that belong to a network flow Class, e.g., by source-destination addresses and MPLS tags
  - Easily enforced by routers.
    Packet header is sufficient to enforce network-level policies
- Application-level Policies:
  - Specified by ASPs, e.g., Send all voice + video messages to Server group 1, accounting messages to Server 2
  - Enforced by middle boxes
  - Usual way: Look at application messages, requires message reassembly and decryption in middle boxes.
    Requires terminating TCP.
  - Our solution: Use cross-layer communication

Extension 3: ID/Locator Split

- All servers are addressed by 32-bit ID.
  OpenADN appliances translate 32-bit ID to 32-bit locators
- A group of servers have a group ID
- Group ID allows OpenADN to select a particular server

Extension 4: Sender and Receiver Policies

- No distinction between users and servers. Both can have control over application traffic destined to them
  - Senders specify sender policies (enforced in sender domain)
  - Receivers specify receiver policies (enforced in the receiver domain)

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"

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

Past Failures

- 1986: MAP/TOP (vs Ethernet)
- 1988: OSI (vs TCP/IP)
- **1990: Active Networks**
- 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, …

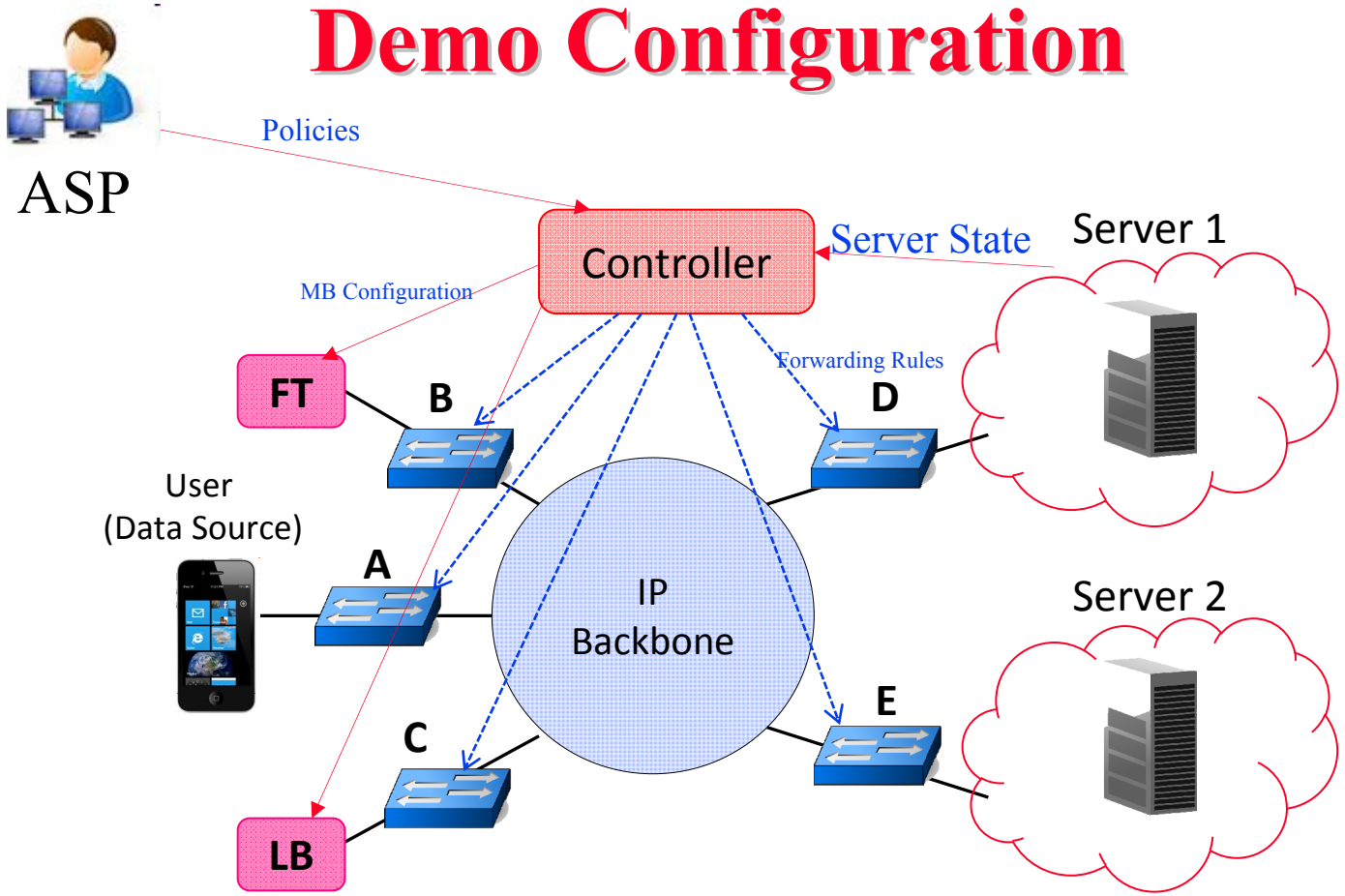
Technology alone does not mean success.

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

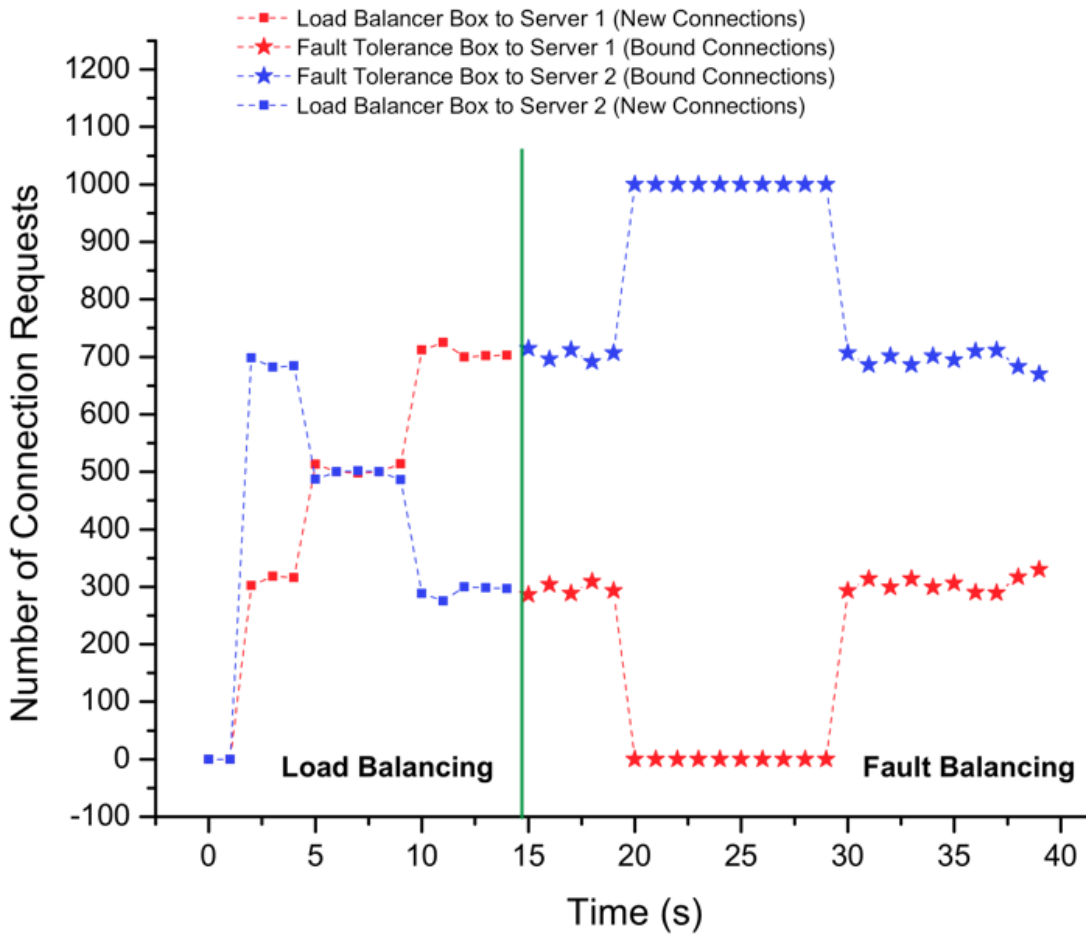
Most versions of Ethernet followed these principles.
Many versions of IP did not.

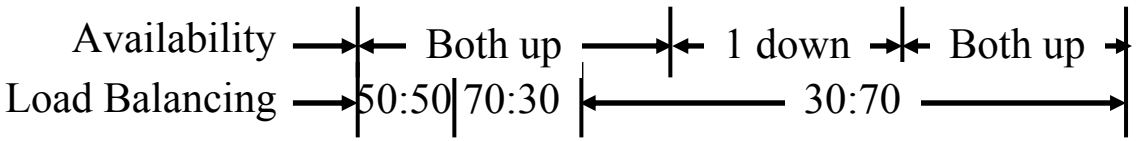
Demo Configuration



- Single user and single ASP with 2 servers
- OpenADN Appliances: A, B, C, D, E
- ISP offers ADN services: Fault tolerance and Load Balancing

Validation of Functionality



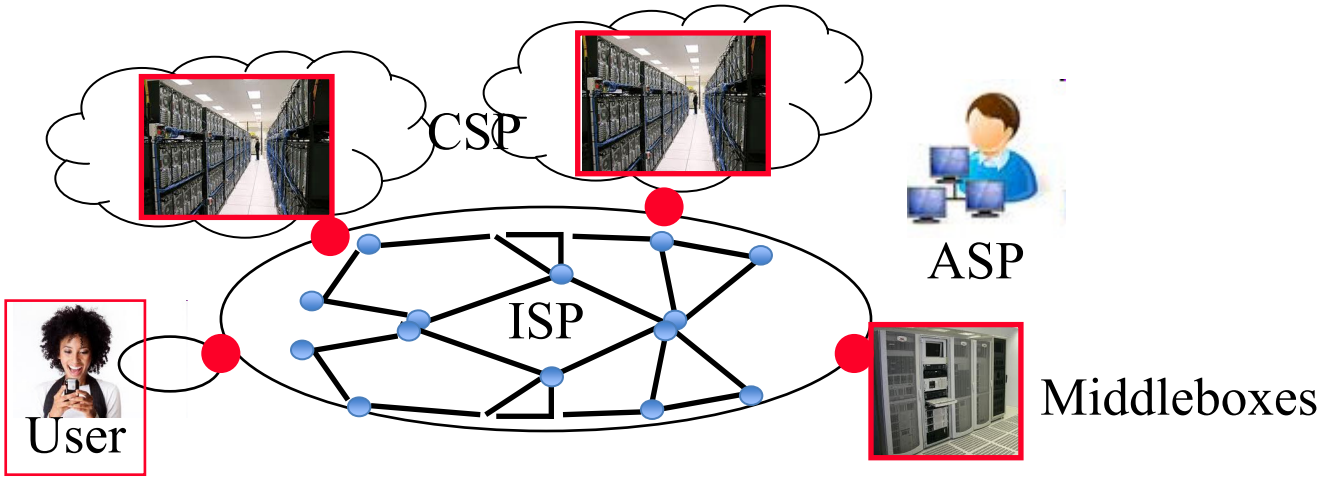


Resource Control

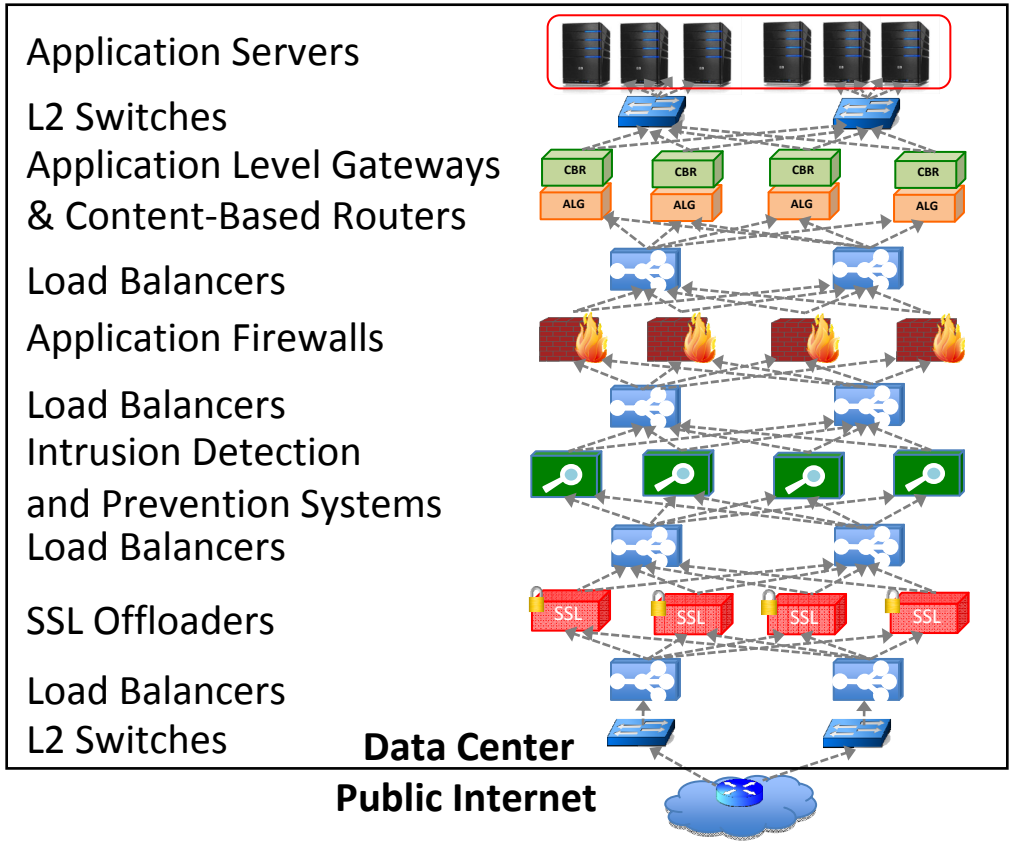
- ASPs keep complete control of their data.
  ISP does not have to look at the application headers or data to enforce application level policies
- ISPs keep complete control of their equipment.
  ASPs communicate their policies to ISP's control plane
- Middle boxes can be located anywhere on the global 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: Sell openADN appliances, openADN-aware applications
- ASPs: Deploy servers anywhere and move them anytime
- ISPs: Offer new services
- Cloud Service Providers (CSPs): Freedom to move VMs, Less impact of downtime

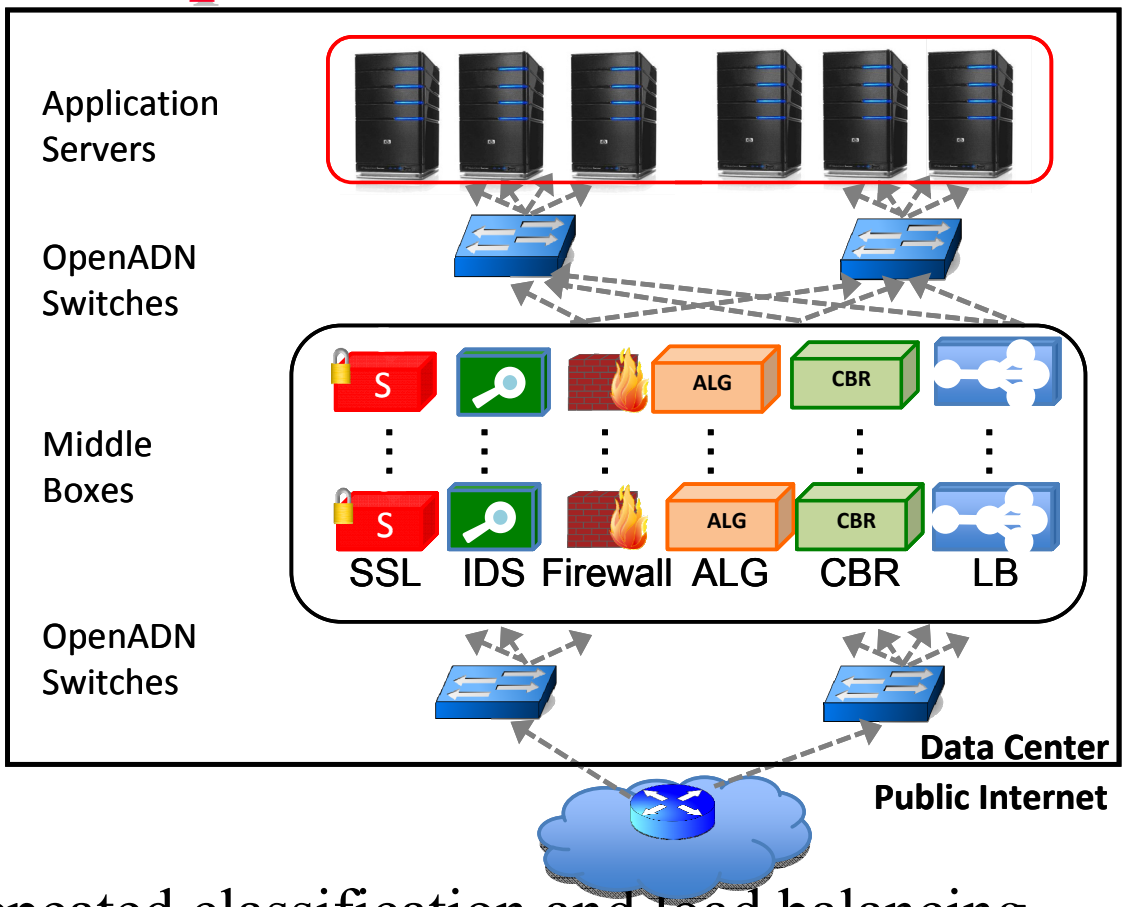


Data Center Applications



- Repeated classification and load balancing
- No application level control over MBs traversed
- Unnecessary traversals and reduced performance

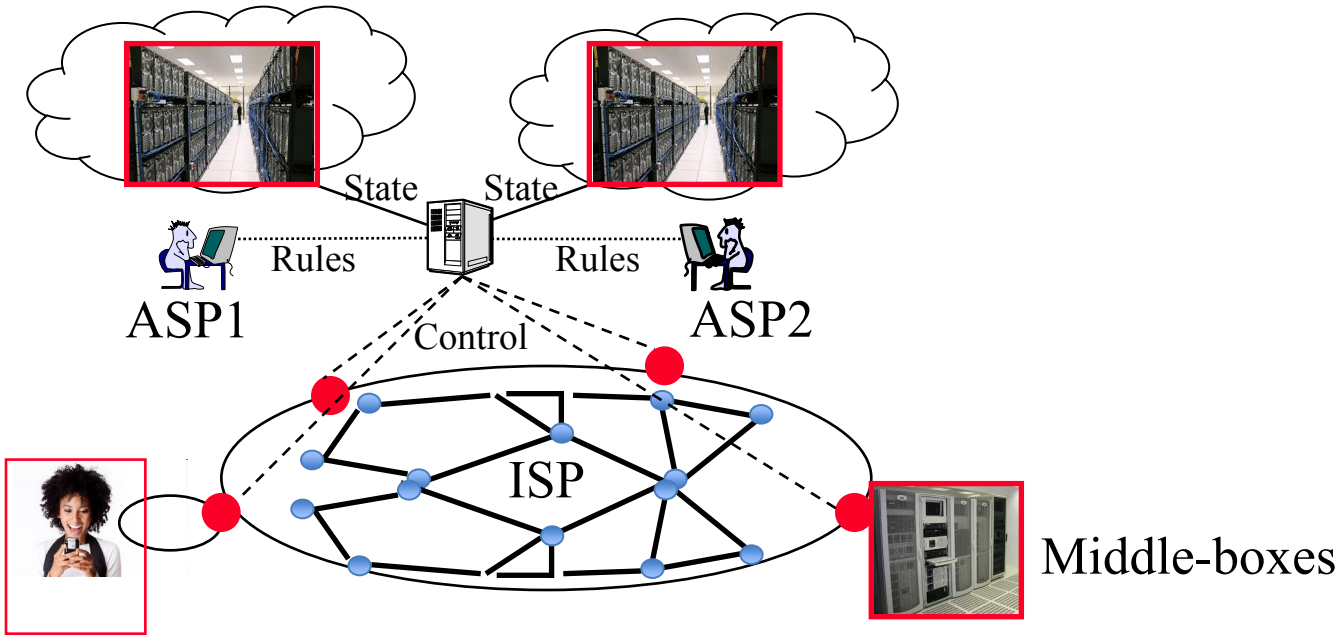
OpenADN in Data Center



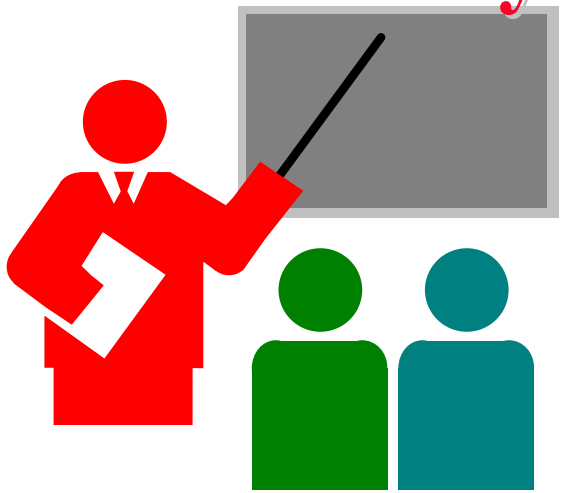
- No repeated classification and load balancing
- Application flow specific traversal through MBs
- Reduced number of appliances and increased performance

OpenADN Without OpenFlow/SDN

- OpenADN clients, servers, middle-boxes use only APLS labels.
- OpenADN aware devices need an API to communicate with controllers
- API can be vendor specific



Summary



1. Explosion of Apps using cloud services
2. OpenADN appliances can provide ASPs networking services they need
3. OpenADN extends using best of OpenFlow, SDN, MPLS, ID/Locator Split, Cross-layer communications, middle box appliances
4. Keeps resource control under resource owners
5. Can be implemented incrementally now