Next Generation Internet: Architectures for Future Internet Evolution

RAJ JAIN

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

Qualcomm Distinguished Lecture at
International Conference on Computing, Networking, and Communications
Maui, Hawaii, January 30, 2012

These slides and audio/video recordings of this talk are available on-line at:

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

1. Why Next Gen?
2. Our research on next generation: open ADN
3. Future Internet Arch, Content Centric Networking
4. Network Virtualization (Industry efforts)
5. OpenFlow
6. Software Defined Networks

Future Internet Research (History)

- In 2005 US National Science Foundation started a large research and infrastructure program on next generation Internet
- Q: How would you design Internet today? Clean slate design.
  - "Future Internet Design" (FIND): Architecture research
  - "Global Environment for Networking Innovations" (GENI): Testbed
- European Union: 7th Framework program
  Japan, China, Korea, Australia, …20+ countries
- April 2010: Future Internet Architecture (FIA): 4 Extra-Large Projects ⇒ Future Internet Assembly (FIA) in Europe
- Network Virtualization, Software Defined Networking

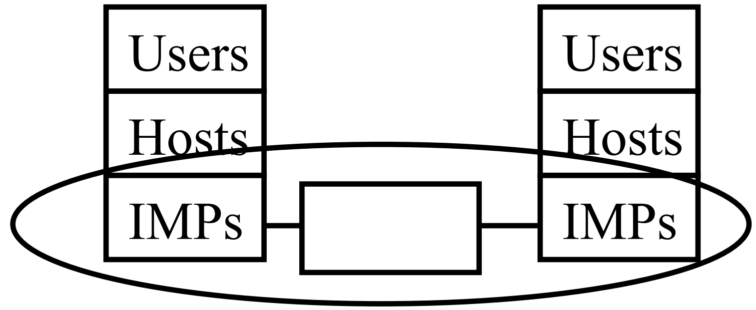
Ref: Jianli Pan, Subharthi Paul, and Raj Jain, "**A Survey of Research on Future Internet Architectures**," IEEE Comm. Magazine, Vol. 49, No. 7, July 2011, pp. 26-36, http://www1.cse.wustl.edu/~jain/papers/internet.htm
Future Internet http://en.wikipedia.org/wiki/Future_Internet

Why to worry about Future Internet?

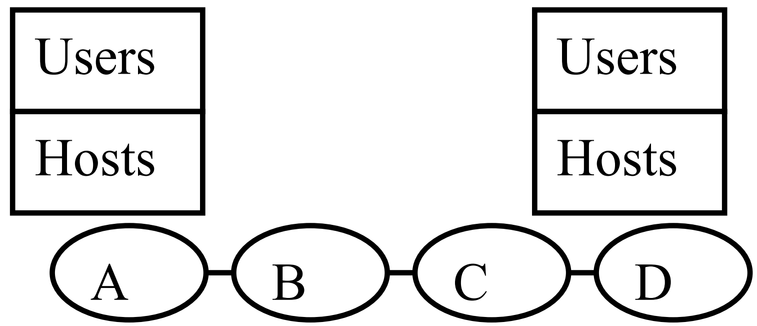
Billion dollar question!

Internet Generations

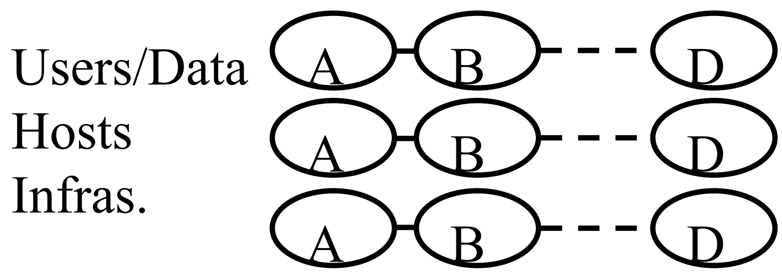
- **Internet 1.0** (1969 – 1989)
  - Single ownership ⇒ Trust
  - complete knowledge
  - Algorithmic optimality ⇒ RIP



- **Internet 2.0**(1989–2009) Commerce
  - Multiple ownership of infrastructure ⇒ Distrust, **Security**
  - No knowledge of internal topology and resources
  - ***Policy based*** routing ⇒ BGP



- **Internet 3.0**(2009–2029) Commerce
  - Users, Content, Host ownership
  - Requirements, Service Negotiation
  - Mobility of users and distributed data



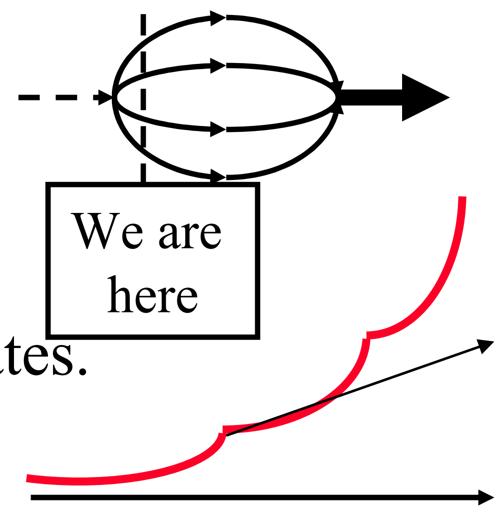
2012: Where are we now?

- At the knee of Mobile Internet age (paradigm shift)
  - Computing (IBM 360) ⇒ Mini-computing (PDP11) ⇒ Personal Computing (Desktop, PC+MAC) ⇒ Laptops ⇒ Netbooks ⇒ Smart Phones + Tablets
- Most valued companies in the stock market are generally those that lead the paradigm shift
  - Automotive (General Motors) ⇒ Electrical (GE, Edison Electric) ⇒ Networking (Cisco + 3Com in 80's) ⇒ Internet (Netscape + Yahoo in 90's) ⇒ Mobile Internet (Apple +MS+ Google, 2010's)
- Note: Apple ≠ PC (MAC) company (mobile device company)
  - Google ≠ search engine (mobile device company)
- Also Social Networking (Facebook), Internet Retail (Amazon)

5 Future Predictors

1. **Miniaturization**: Campus ⇒ Datacenter ⇒ Desktop ⇒ Laptop ⇒ Pocket ⇒ Multi-functional Pocket device
2. **Mobility**: Static ⇒ Mobile (1 km/hr) ⇒ Mobile (100 km/hr) ⇒ Mobile (600 km/hr)
3. **Distance**: PAN (5m) ⇒ LAN (500 m) ⇒ MAN (50 km) ⇒ WAN (500 km)
4. **Applications:** Defense ⇒ Industry ⇒ Personal
5. **Social Needs**: Energy, Environment, Health, Security

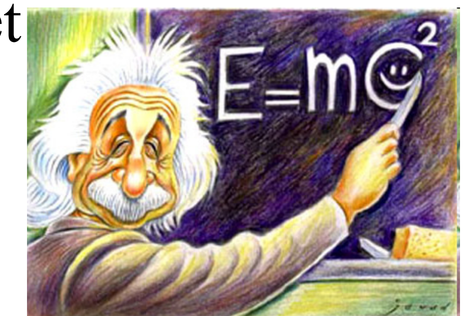
- Broadening and Aggregation: Research ⇒ Many Solutions ⇒ One Standard ⇒ General Public adoption, e.g., Ethernet
- Non-Linearity: Progress is not linear. It is exponential and bursty. Most predictions are linear ⇒ underestimates.



Trend: Moore's Law

- Computing Hardware is cheap
- Memory is plenty

⇒ Storage and computing (Intelligence) in the net



- Energy ⟷ Matter
- Space ⟷ Time
- Communication in Space ⟷ Communication in Time

Rural India

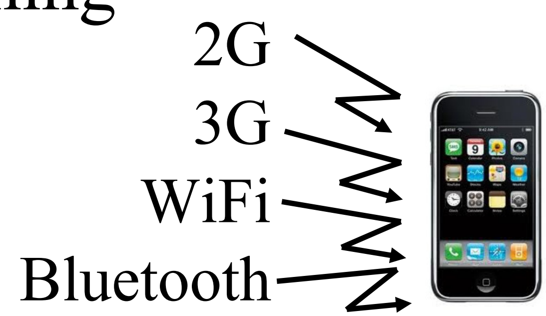
- Link
- Storage (USB, Caching,…)

Next Gen nets will use storage in networks, e.g., DTN, CCN

Trend: Multihoming + Mobility

- Centralized storage of info
- Anytime Anywhere computing
- Dynamically changing Locator
- User/Data/Host/Site/AS Multihoming
- User/Data/Host/Site Mobility

⇒ **ID/Locator Split**



Mobile Telephony already distinguishes ID vs. Locator
We need to bring this technology to IP.

Trend: Profusion of Services

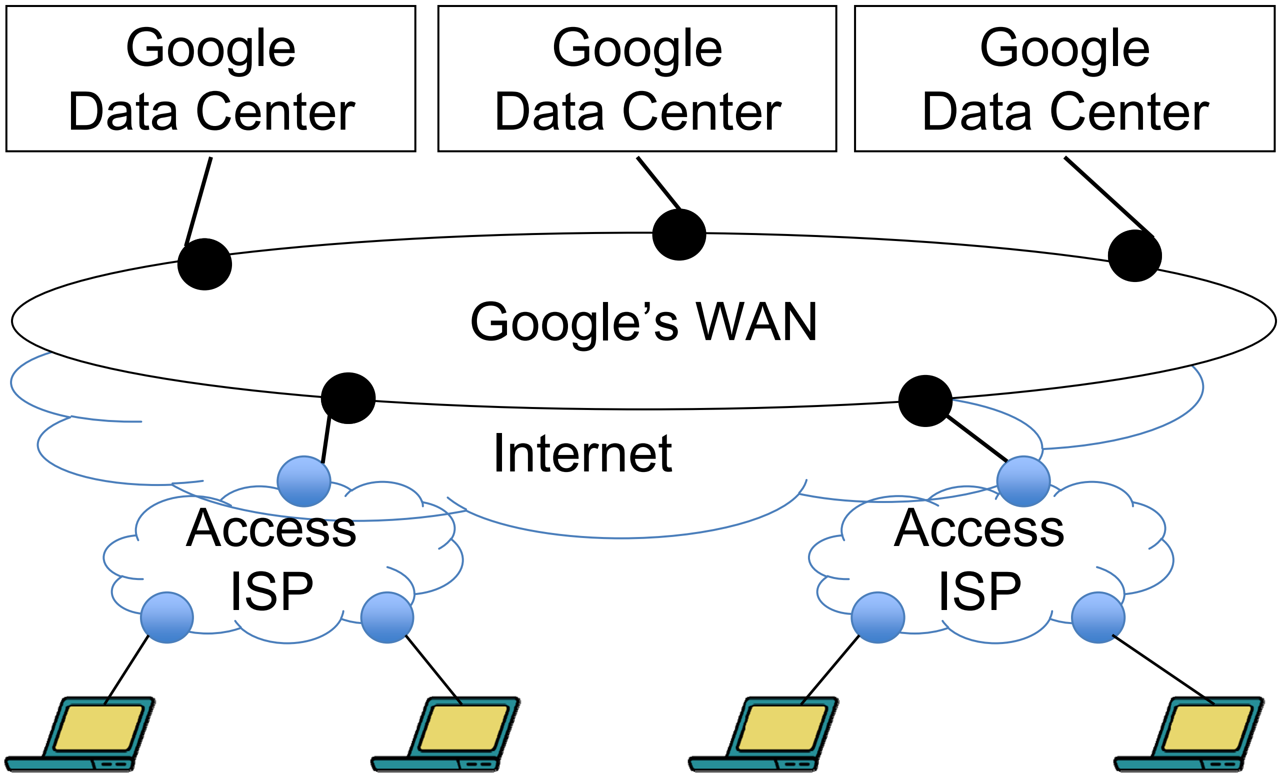
- Almost all top 50 Internet sites are services [Alexa]
- Smart Phones: iPhone, Android Apps
  ⇒ New globally distributed services, Games, …
  ⇒ More clouds, …

Networks need to support efficient service setup and delivery

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

Private Smart WANs

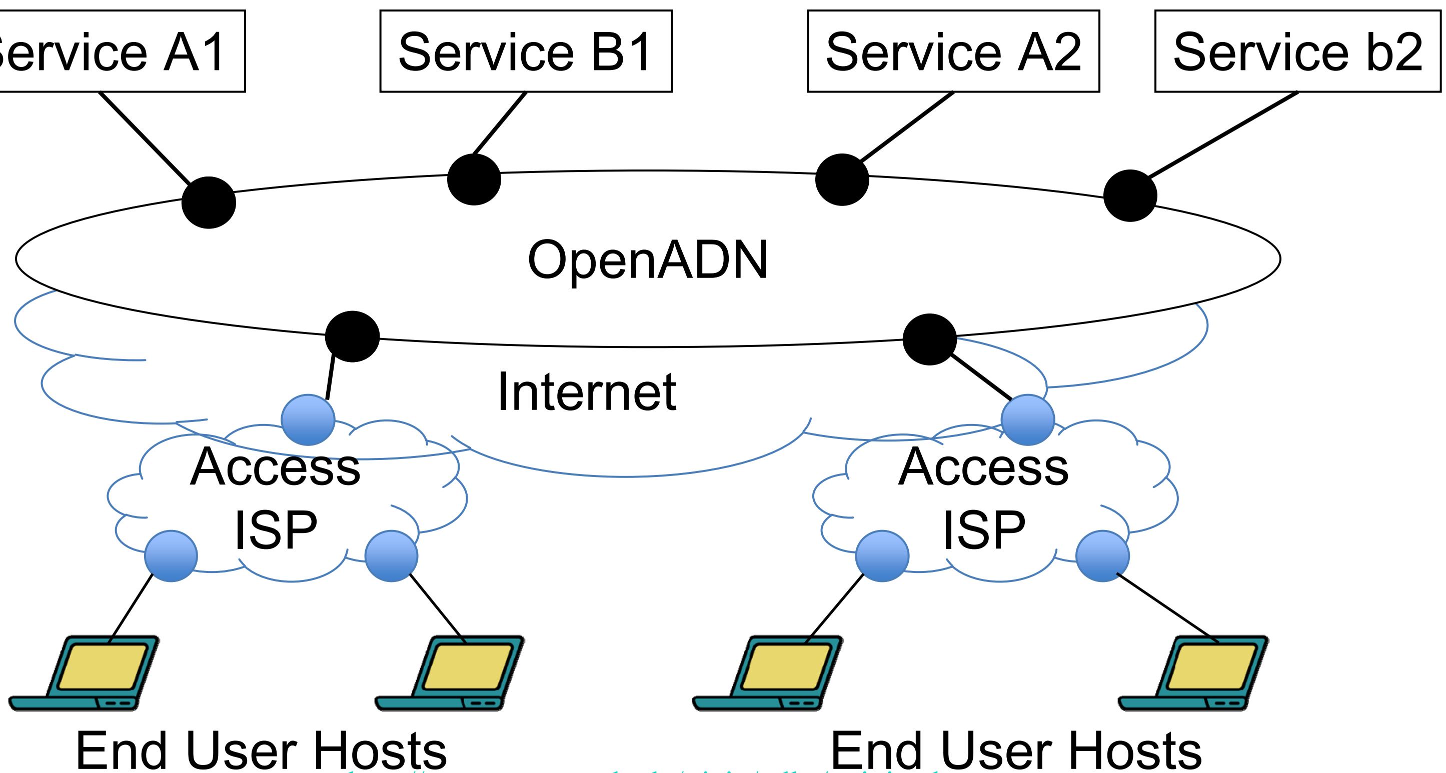
- Services totally avoid the Internet core ⇒ Many private WANs
- Google WAN, Akamai ⇒ Rules about how to connect users



Opportunity for ISPs to offer these types of WAN services

OpenADN

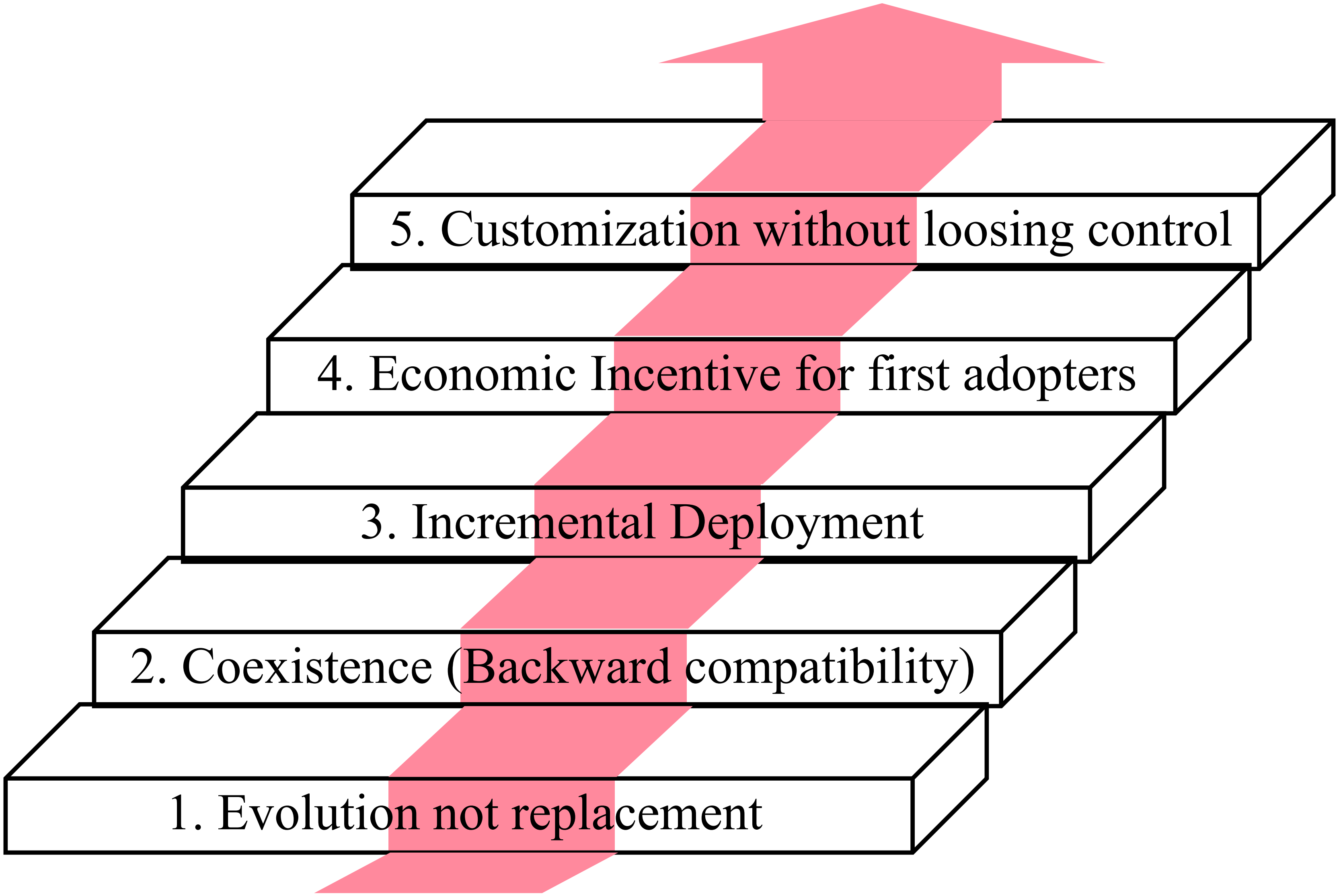
- High-Speed WAN for Application Service Delivery.
- Allows ASPs to quickly setup services



Ten Key Features that Services Need

1. **Replication**: Multiple datacenters appear as one
2. **Fault Tolerance**: Connect to B if A is down
3. **Load Balancing**: 50% to A, 50% to B
4. **Traffic Engineering**: 80% on Path A, 20% on Path B
5. **Flow based forwarding**: Movies, Storage Backup, …
   ATMoMPLS, TDMoMPLS, FRoMPLS, EoMPLS, …
   Packets in Access, Flows in Core
6. **Security**: Provenance, Authentication, Privacy, ...
7. **User Mobility**: Gaming/Video/… should not stop as the user moves
8. **Service composition**: Services using other services
9. **Customization**: Every service has different needs
10. **Dynamic Setup** ⇒ Networking as a Service

Five Arch Design Principles for Success



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.

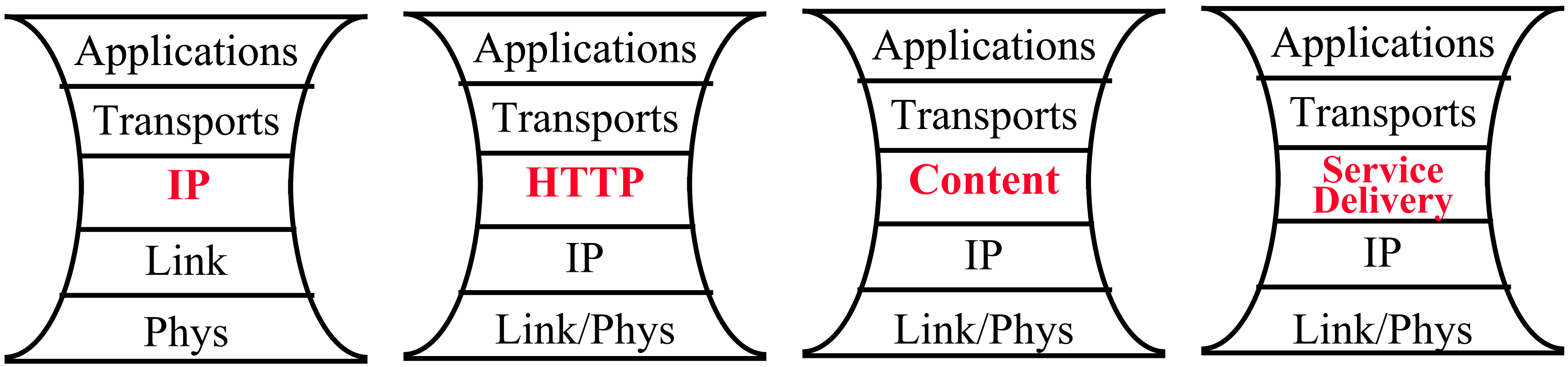
Key Features of openADN

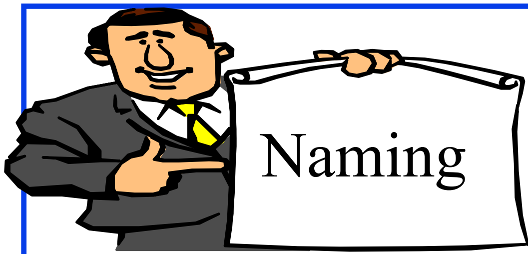
1. Edge devices only.
   Core network can be current TCP/IP based or future SDN based
2. Coexistence (Backward compatibility):
   Old on New. New on Old
3. Incremental Deployment
4. Economic Incentive for first adopters

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

The Narrow Waist

- Everything as a service over service delivery narrow waist
- IP, HTTP, Content, Service delivery, …





Naming Fundamentals

- Name, ID, and Locator
- ID/Locator Split
- Self-Certifying IDs

Names, IDs, Locators

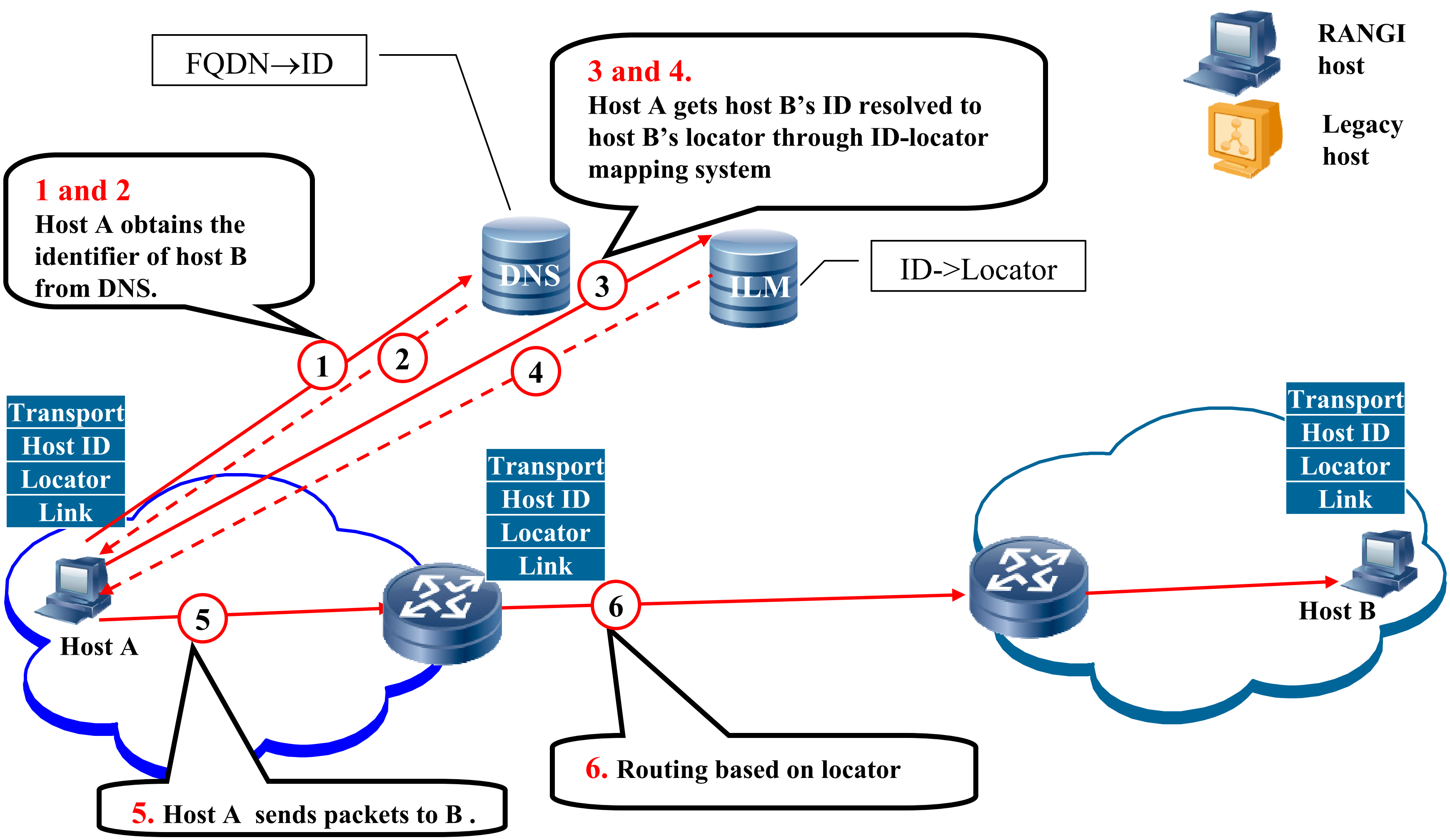
Name: John Smith

ID: 012-34-5678

Locator:
1234 Main Street
Big City, MO 12345
USA

- Locator changes as you move, ID and Names remain the same.
- **Examples**:
  - Names: Company names, DNS names (Microsoft.com)
  - IDs: Cell phone numbers, 800-numbers, Ethernet addresses, Skype ID, VOIP Phone number
  - Locators: Wired phone numbers, IP addresses

ID-Locator Split



ID-Locator Split (Cont)

- Allows hosts to move
- Allows entire organizations to move
  Allows organizations to change providers
- No need to use "Provider Independent (PI)" addresses
- Provider Aggregatable (PA) addresses are preferred since they result in shorter BGP tables
  ⇒ Scalable
- Several proposals for host-based ID-locator split:
  HIP, Shim6, I3, and HI3
- All hosts have ID and global locators
- Allow mobility, multihoming, renumbering

Secure IDs: HIP

- Host Identity Protocol
- 128-bit Host ID tag (HIT)
- TCP is bound to HIT. HIT is bound to IP address in the kernel
- Uses flat cryptographic based identifier
- Two Methods:
  - Locator registered using Update packets to DNS
    ⇒ Does not allow fast mobility
  - Use rendezvous servers
    ⇒ Does not adhere to organizational boundary
- Requires changes to end hosts

Ref: R. Moskowitz, P. Nikander and P. Jokela, ``Host Identity Protocol (HIP) Architecture," IETF RFC4423, May 2006.

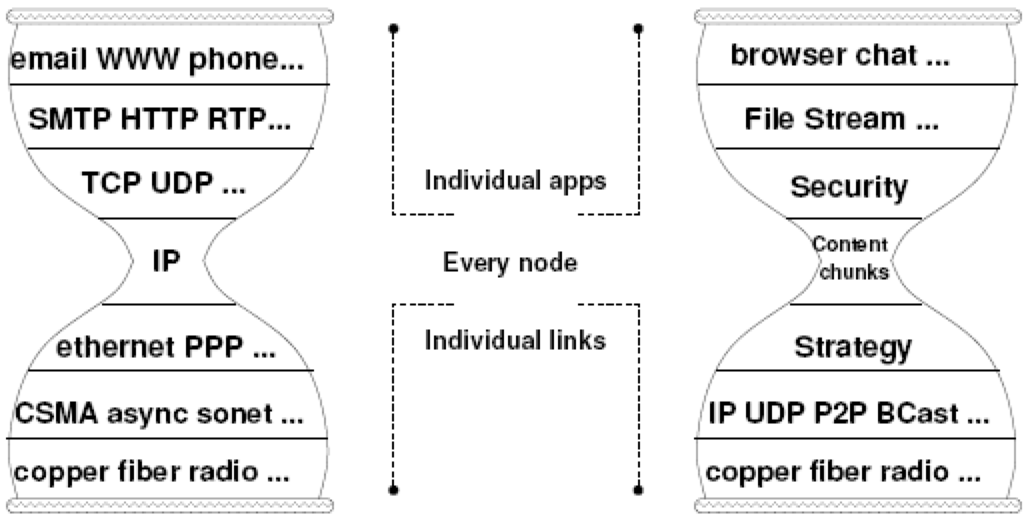
NSF FIA Winners

- **Named Data Networking**: CCN
  - Routing scalability, Fast forwarding, Trust models, Network security, Content protection and privacy
- **Mobility First**: Generalized Delay Tolerant Networking with self-certifying public key addresses
- **Nebula** (Latin for Cloud): Trustworthy data, control and core networking for cloud computing
- **eXpressive Internet Architecture (XIA):** Application programming interface (API) for communication, flexible context-dependent mechanisms for establishing trust

Ref: NSF Announces Future Internet Architecture Awards, August 27, 2010, http://www.nsf.gov/news/news_summ.jsp?cntn_id=117611

Named Data Networking aka CCN

- IP cares about "**Where**": forward packets from A to B
- Users care about "**What**": Movie X
- Replace "packets" with "Data Objects" or "Interests" (requests)
- Replace "Addresses" with "Names of Objects"

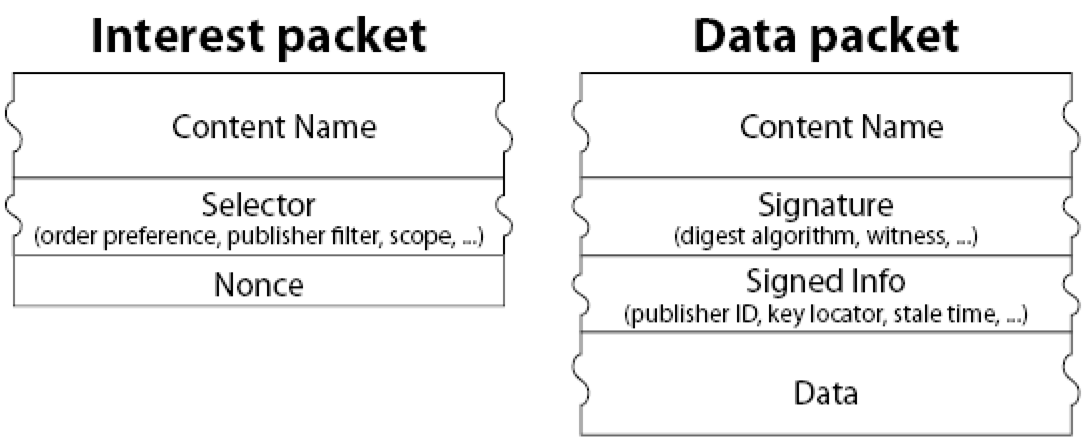


[JAC09]

Ref: [JAC09] V. Jacobson, et al, "Networking Named Content," CoNEXT 2009, December 2009

CCN Packets

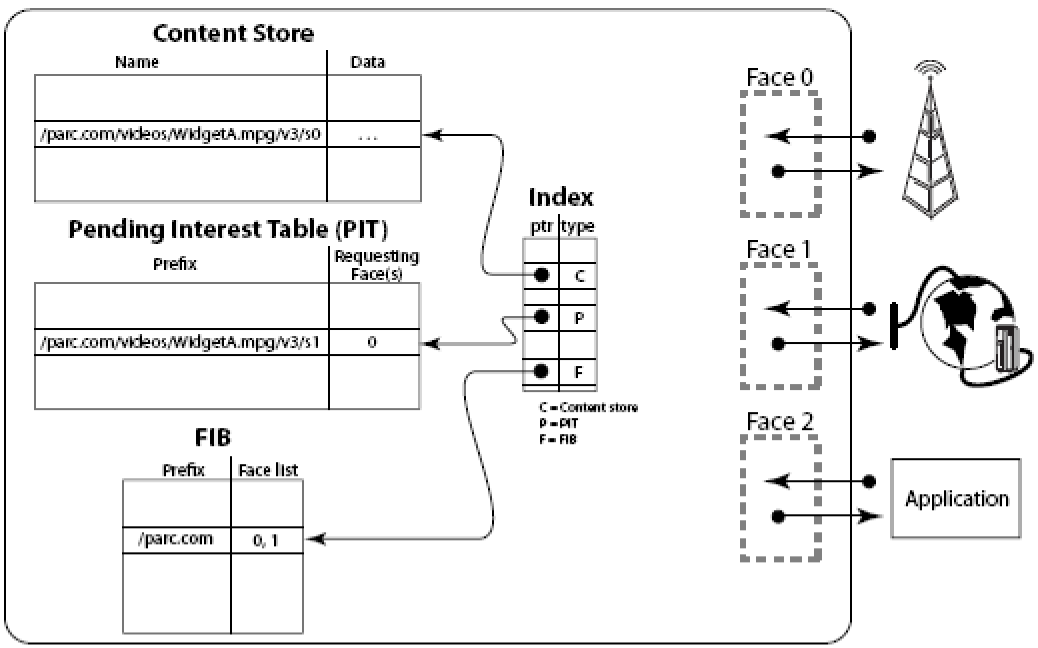
- Interest Packets: Request for Data
- Data Packets: Signed Data
- Longest prefix match is used as in IP addresses
  http://www.cse.wustl.edu/~jain/talks/ftp/in3_video matches
  http://www.cse.wustl.edu/~jain/talks/ftp/in3_video/V00/S00



[JAC09]

CCN Capable Routers Operation

- **Content Store**: Local cache of data
- **Pending Interest Table (PIT):** Recent requests forwarded
- **Forwarding Information Base (FIB):** Known data locations
- **Faces**: Requesting processes and hardware interfaces



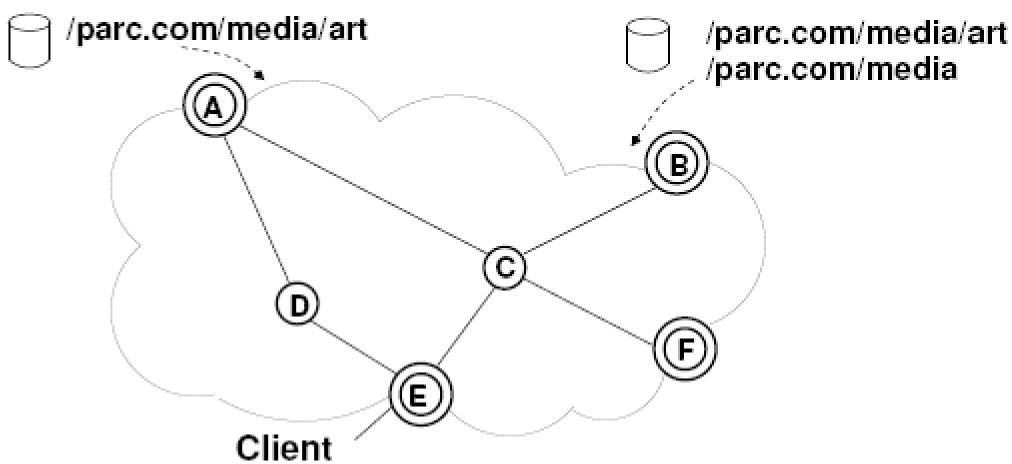
[JAC09]

Routers Operation (Cont)

- Applications send "Interest" in data X
- Router looks up in local store and sends if found
- Router looks up in PIT, if entry already exists (someone requested it recently) , adds the interest, face to the same entry
- Router looks up in FIB, if entry exists (data location is known), a PIT entry is made and the interest is multicasted to all faces in the FIB entry
- If there is no FIB entry, interest is discarded (router does not know how to get the data)
- When data arrives, Content Store match $\Rightarrow$ duplicate, discard
  PIT match $\Rightarrow$ Forward to all faces
  FIB match $\Rightarrow$ No PIT $\Rightarrow$ Unsolicited $\Rightarrow$ Discard
- Data providers register their data $\Rightarrow$ Creates FIB entries

CCN Security

- Data-Centric Security ⇒ Protections travel with the data
- All data is signed and versioned.
- IP and CCN routers can coexist. Public domain code available.
- Current Issues: Origin/requester privacy, Economic benefits to owners and network, Scalability



/parc.com/media	B
/parc.com/media/art	A,B

Trend: Network Virtualization

- Virtual Memory ⇒ L1, L2, L3, ... ⇒ Recursive
- Virtual Desktop ⇒ Virtual Server ⇒ Virtual Datacenter
  Thin Client ⇒ VMs ⇒ Cloud
- 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 virtualized
- In this part, we limit it to L2 Network (LAN)

Why Virtualize?

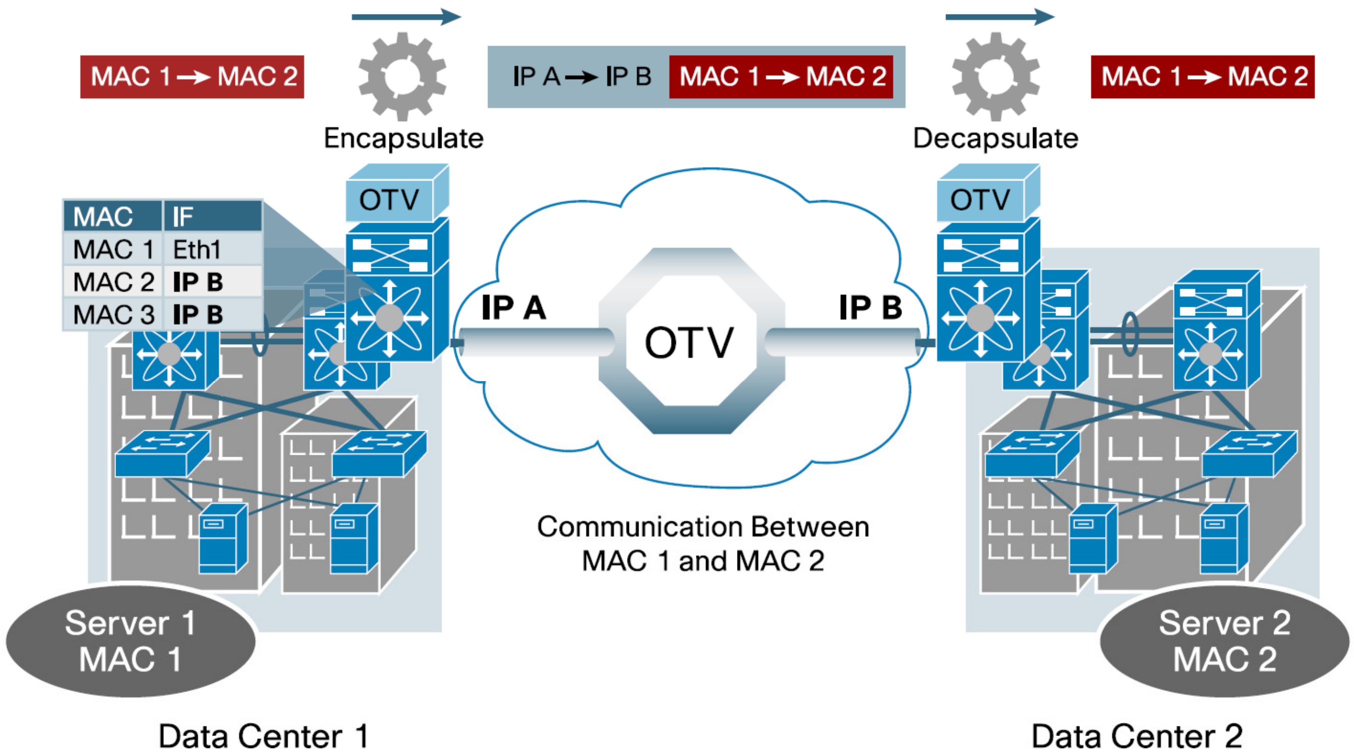
- Ease of Management ⇒ Centralization
- Sharing ⇒ Carrier Hotels = Sharing buildings
- Cost Savings
- Isolation ⇒ Protection
- Dynamics: Replication, load balancing
- Mobility for fault tolerance

LAN Virtualization Technologies

- Problem: LANs were **not** designed for:
  1. Long distances
  2. Dynamic on-demand connectivity
  3. Very large number of nodes
  4. Multiple tenants in a **cloud**
- Solutions:
  1. Overlay Transport Virtualization
  2. VXLAN
  3. Software defined networks

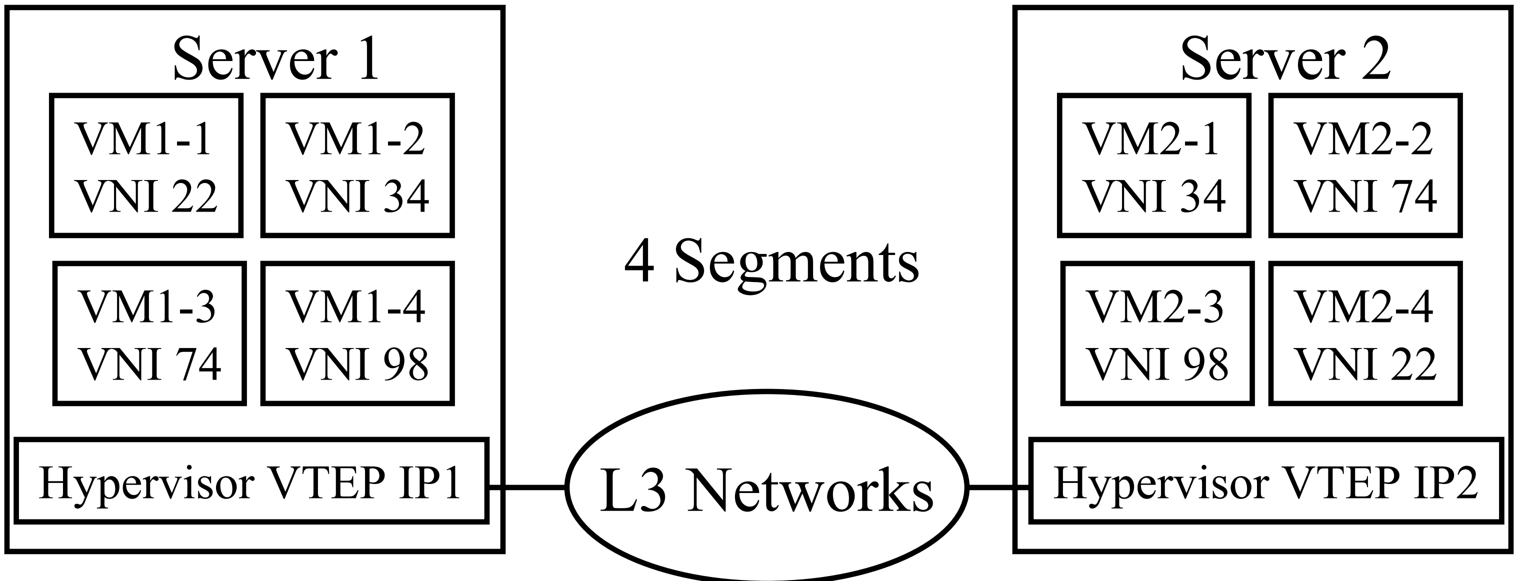
Overlay Transport Virtualization (OTV)

- Cisco technology to allow a single LAN to span multiple datacenters located far apart
- Encapsulates L2 frames and sends using L3



Ref: [Cisco-OTV] Cisco, "Enhance Business Continuance with Application Mobility Across Data Centers," http://www.cisco.com/en/US/prod/collateral/switches/ps9441/ps9402/white_paper_c11-591960.pdf

VXLAN

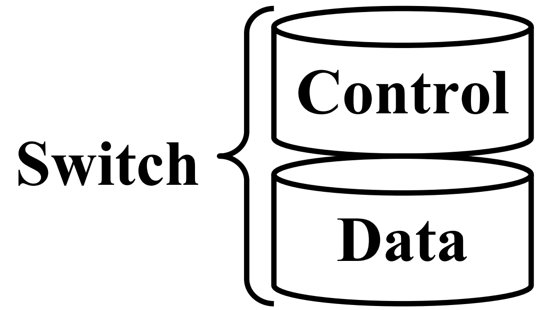


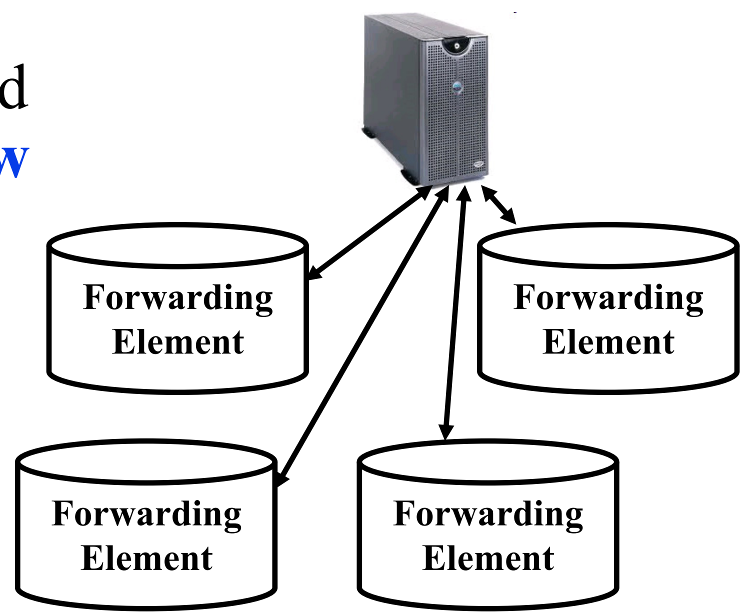
- Virtual Extensible Local Area Networks. By VMware.
- Solves the problem of multiple tenants in a cloud environment.
- A server may have VMs belonging to different tenants
- Each tenant has its own VLANs that connect its VMs
- Uses tunneling to overlay Segments over L3
  Tunnels end points (VTEP) in hypervisors

Ref: VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks, draft-mahalingam-dutt-dcops-vxlan-00, 2011-08-27

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

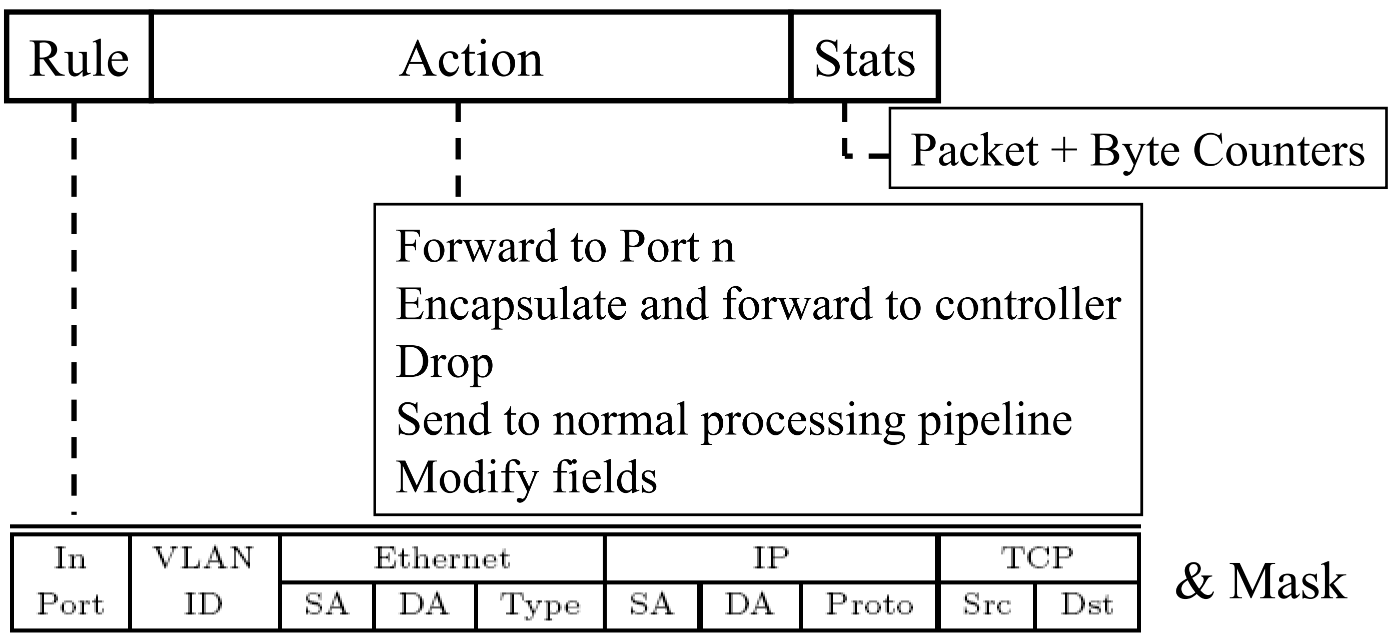




Ref: [MCK08] ``OpenFlow: Enabling Innovation in Campus Networks," OpenFlow Whitepaper, March 2008
http://www.openflow.org/documents/openflow-wp-latest.pdf

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



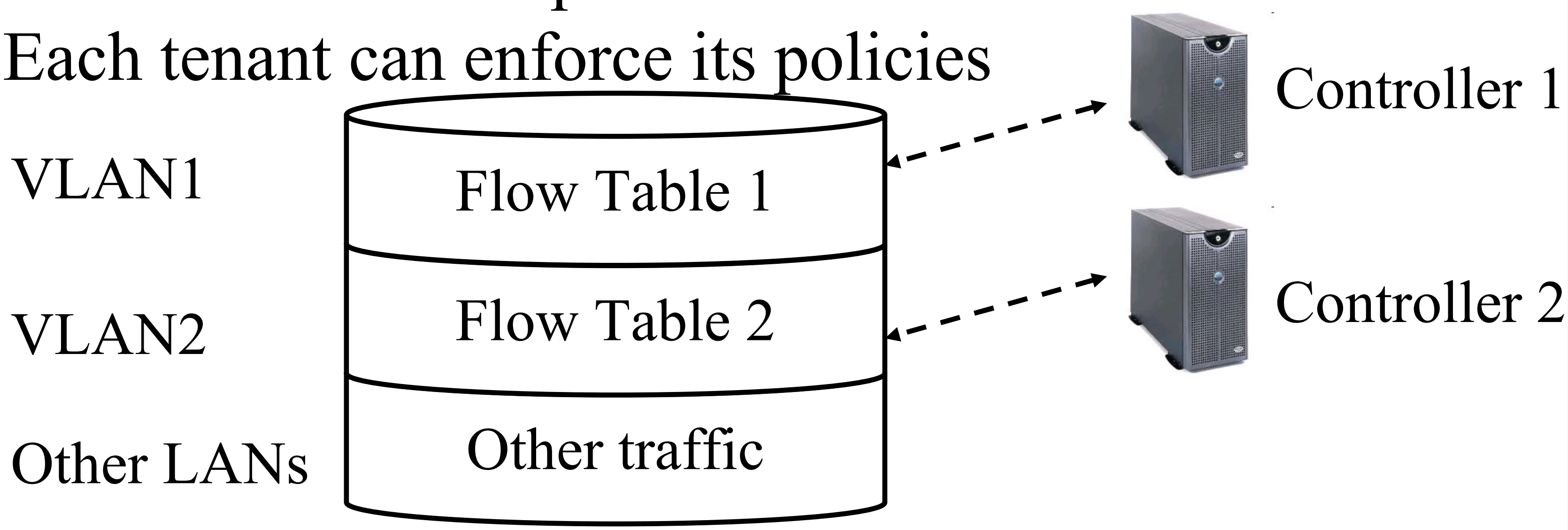
OpenFlow (Cont)

- 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

Trend: Software Defined Networks

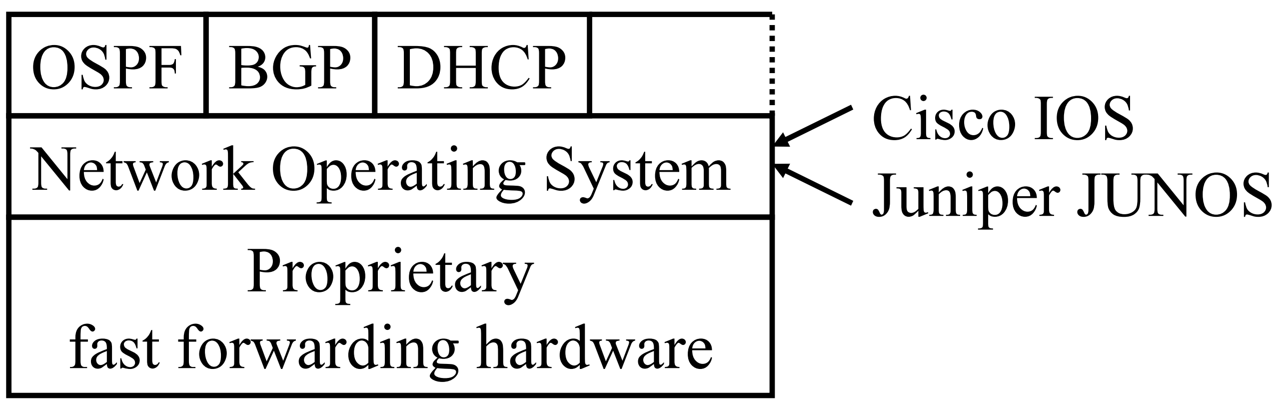
- Problem: Multiple tenants in the datacenter
- Solution: Use multiple controllers.
  Each tenant can enforce its policies



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

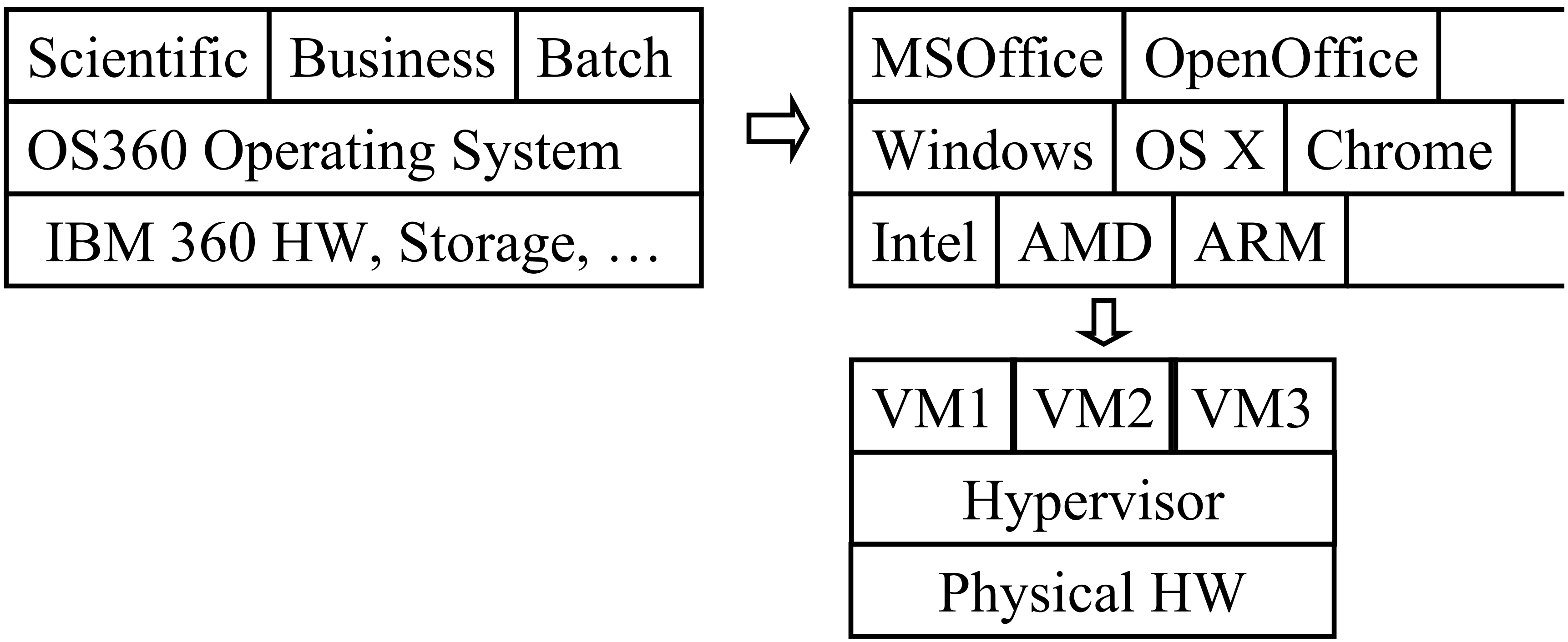
Problem: Complex Routers

- 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).

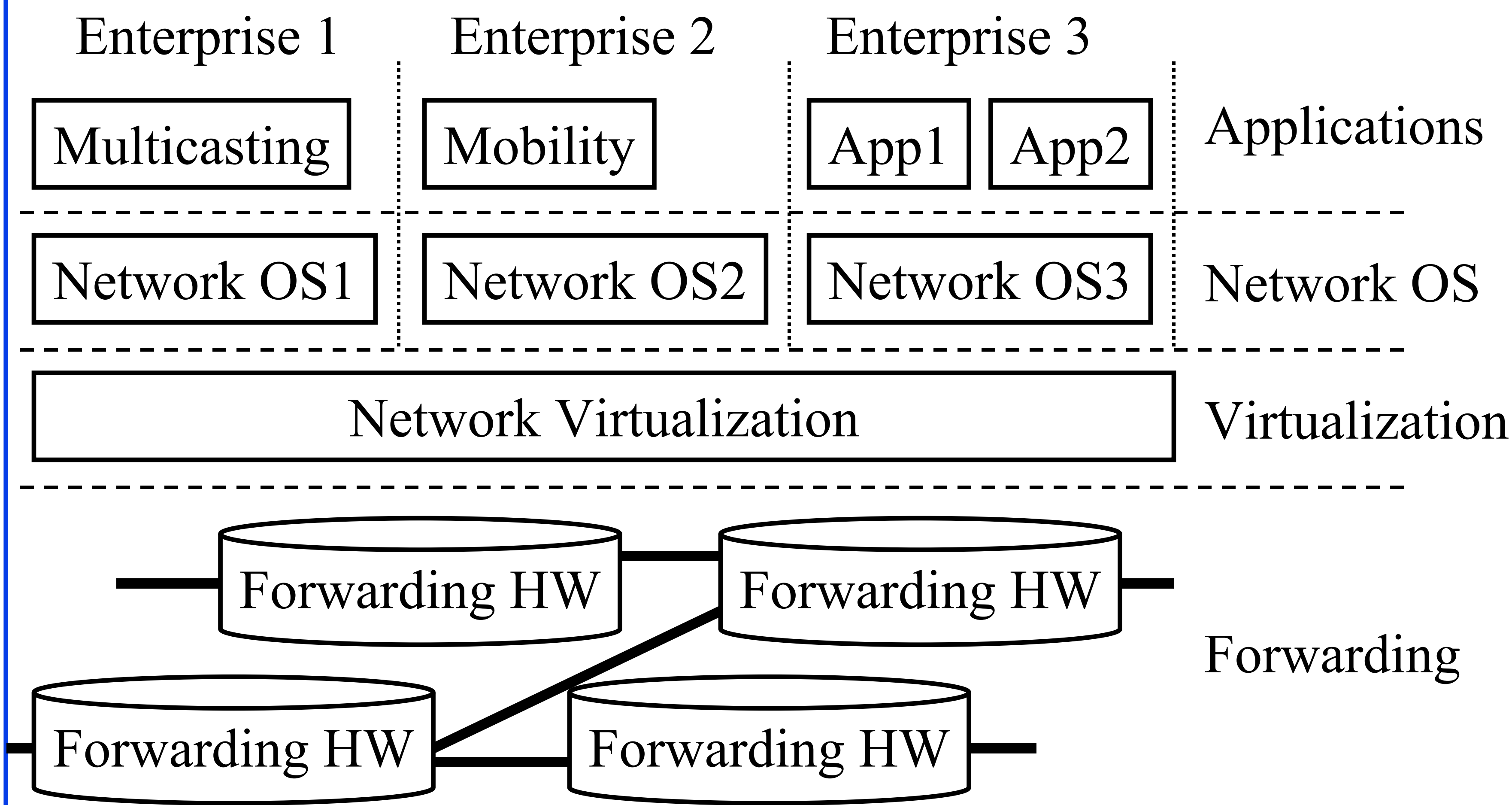


Solution: Divide, Simplify and Standardize

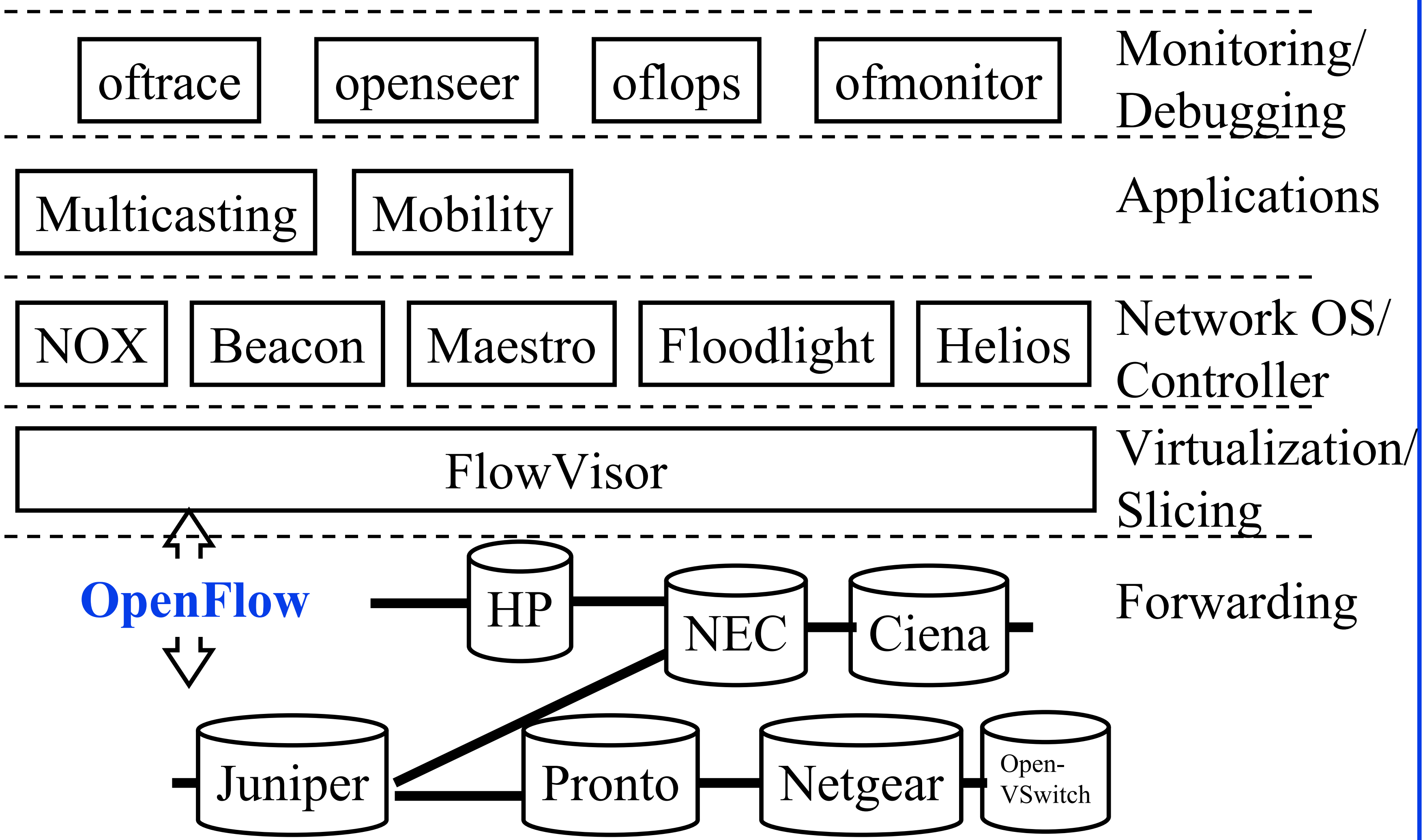
- Computing became cheaper because of clear division of hardware, operating system, and application boundaries with well defined APIs between them
- Virtualization ⇒ simple management + multi-tenant isolation



Multi-Tenant SDN Architecture



SDN Architecture Component Examples

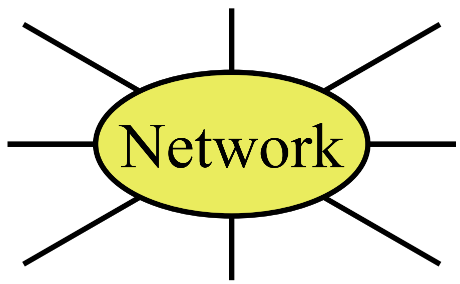


Ref: https://courses.soe.ucsc.edu/courses/cmpe259/Fall11/01/pages/lectures/srini-sdn.pdf

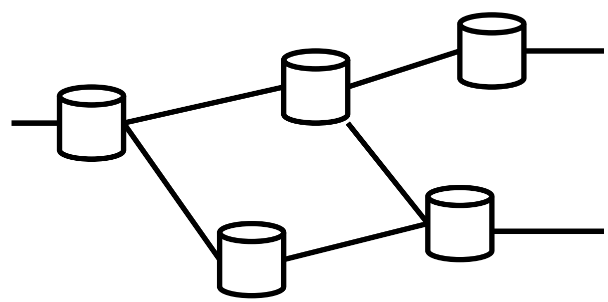
SDN Abstractions

- **Distribution State Abstraction**: No longer design a distributed control protocol. Design only centralized control.
- **Specification Abstraction**: Control program should specify "What" and not "how" ⇒ Virtualization

What



How



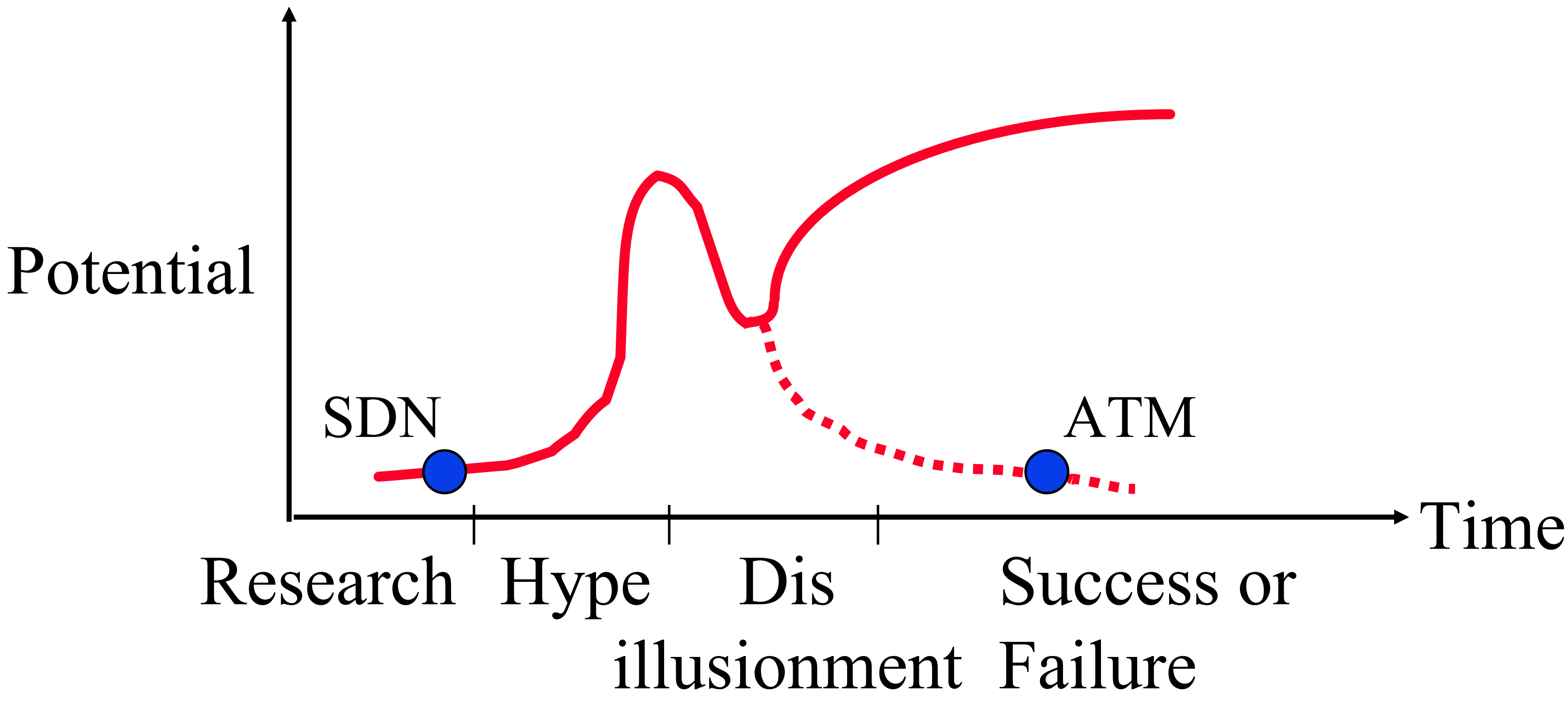
- **Forwarding Abstraction**: Map global view to physical forwarding elements ⇒ OpenFlow

Ref: Scott Shenker, http://inst.eecs.berkeley.edu/~ee122/fa11/notes/18-SDN122-lecture.pdf

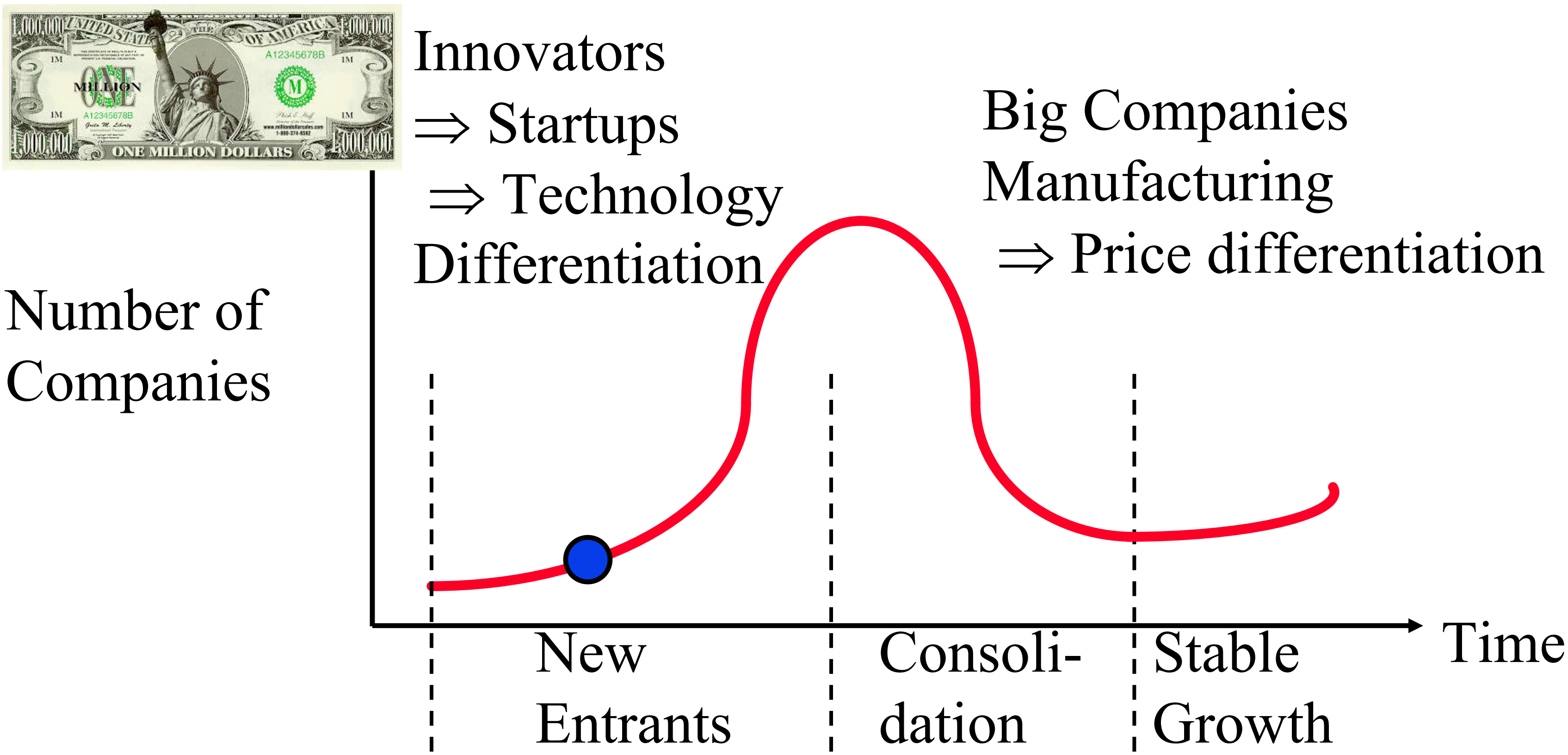
SDN Impact

- 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"

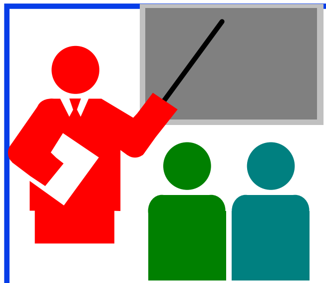
Life Cycles of Technologies



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



Summary

1. Profusion of **multi cloud-based applications** on the Internet. Application services need replication, fault tolerance, traffic engineering, security ⇒ **OpenADN**
2. Mobility and Security ⇒ ID/Locator Split, Self-Certifying IDs ⇒ FIA projects XIA, MobilityFirst, Nebula
3. Named Data Networking/Content Centric Networking allows routers to cache data ⇒ Need to resolve privacy/ownership, scalability, complexity, economic issues
4. L2 Virtualization is required to be able to move VMs from one cloud to another.
5. Centralization of Control ⇒ OpenFlow
6. **Simplification** and Commoditization of Networking Equipment ⇒ **Software Defined Networking** ⇒ Paradigm Shift ⇒ Winners/Loosers to be seen