openSDN: A Dirty-Slate Network Architecture for Cloud-Based Services

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

NSF Science of Cloud Computing Workshop, March 17, 2011

Audio/Video Recordings of this talk are available at
http://www.cse.wustl.edu/~jain/talks/sdn_nsf.htm

Overview

1. Cloud-based Services and Internet
2. Ten Key Features for Cloud-based Services
3. Five Architecture Design Principles for Success
4. Five Key Components of Architecture
5. A brief overview of openSDN

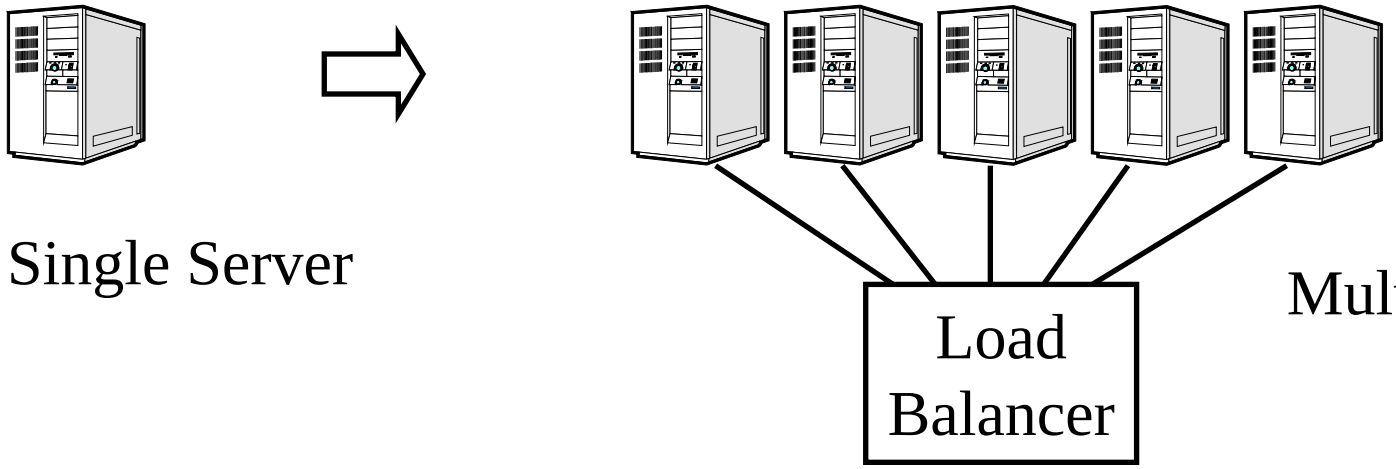
Trend: Profusion of Services

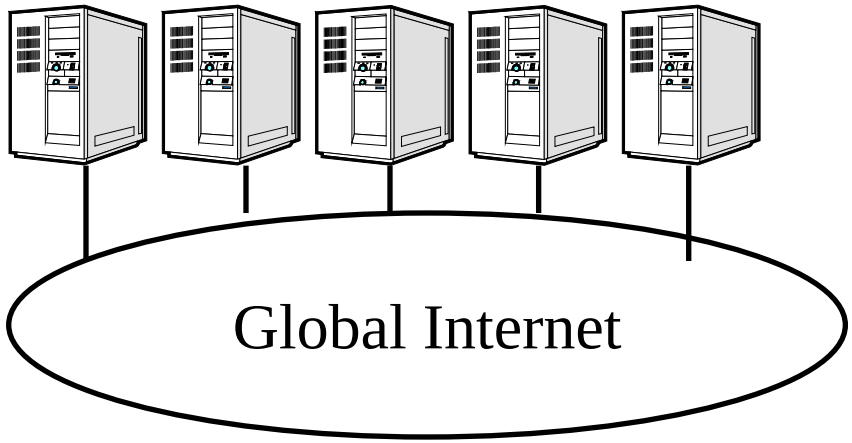
- Almost all top 50 Internet sites are services [Alexa]
- Smart Phones: iPhone, Android Apps
  ⇒ New globally distributed services, Games, …
  ⇒ More clouds, …

Networks need to support efficient service setup and delivery

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

Service Center Evolution



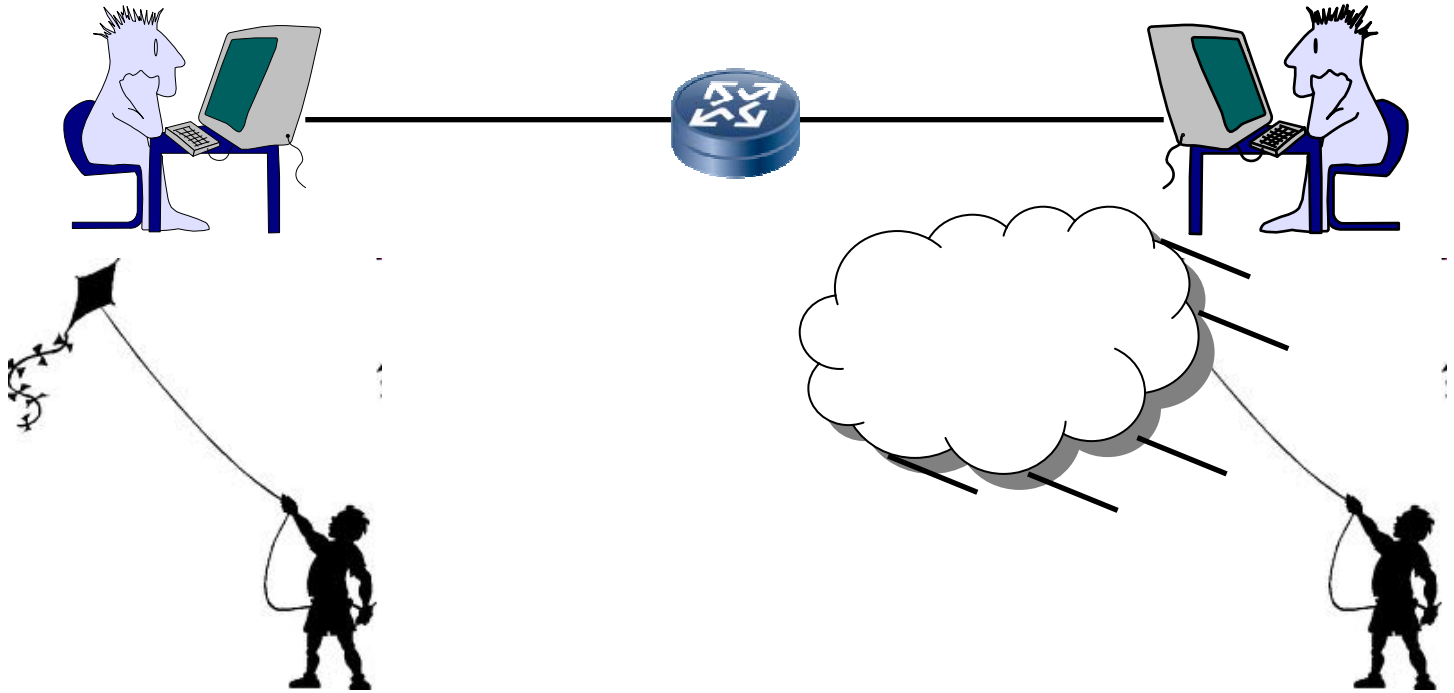


Distributed
Services

Need a distributed load balancer

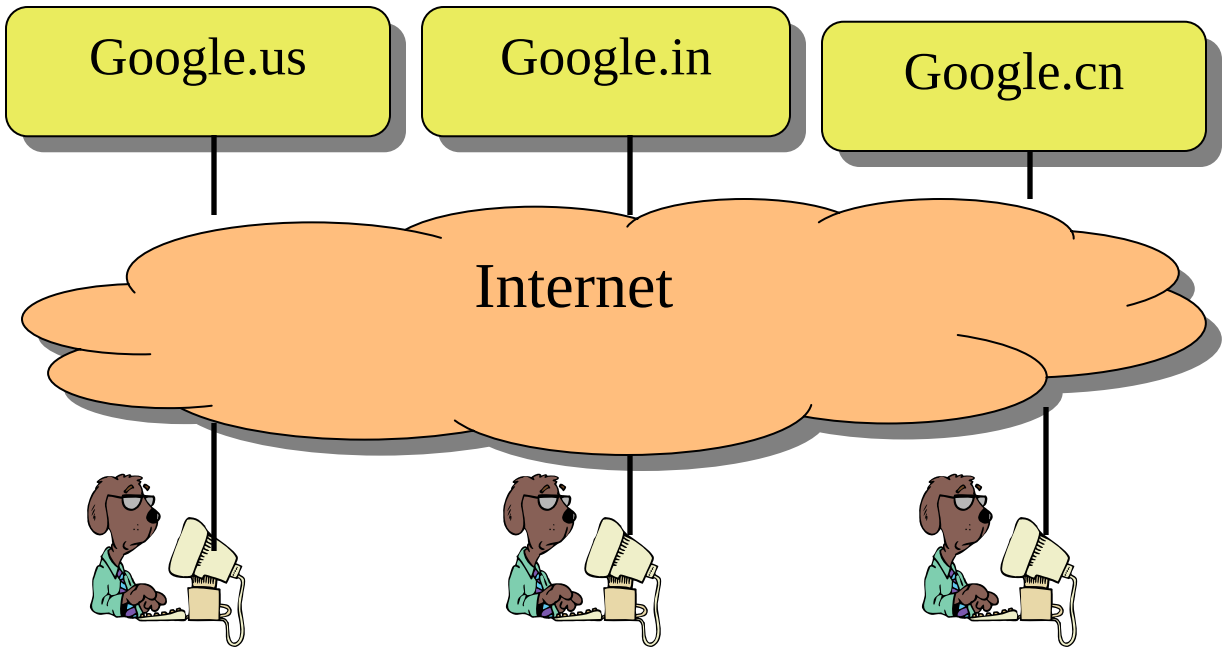
Globally Distributed Cloud Based-Services

- Scale ⇒ Global ⇒ Distributed ⇒ Multihomed
- Cloud computing makes it easy to set up computing part of the global services.
- Internet 1.0 is designed for point-to-point communication
- Significant opportunities for improvement for global services



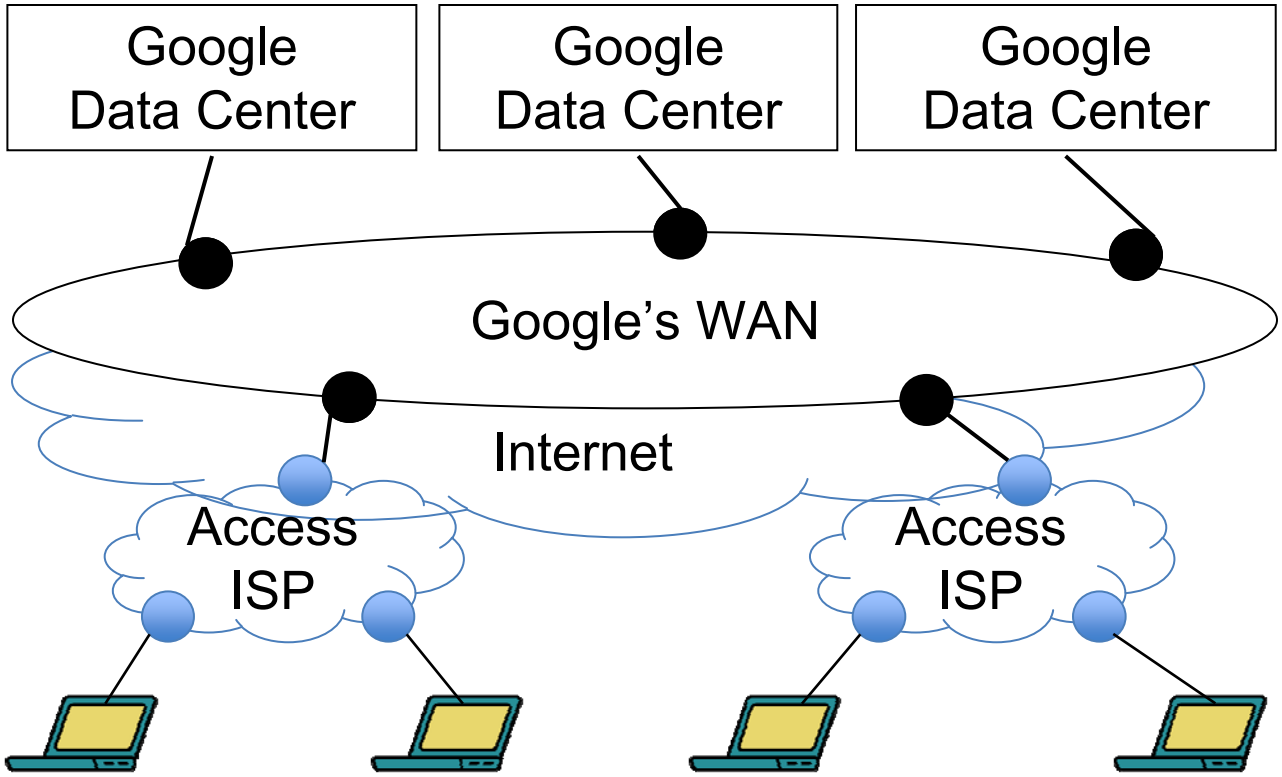
Globally Distributed Services (Cont)

- It's the service responsibility to find the right server for the client



Trend: Private Smart WANs

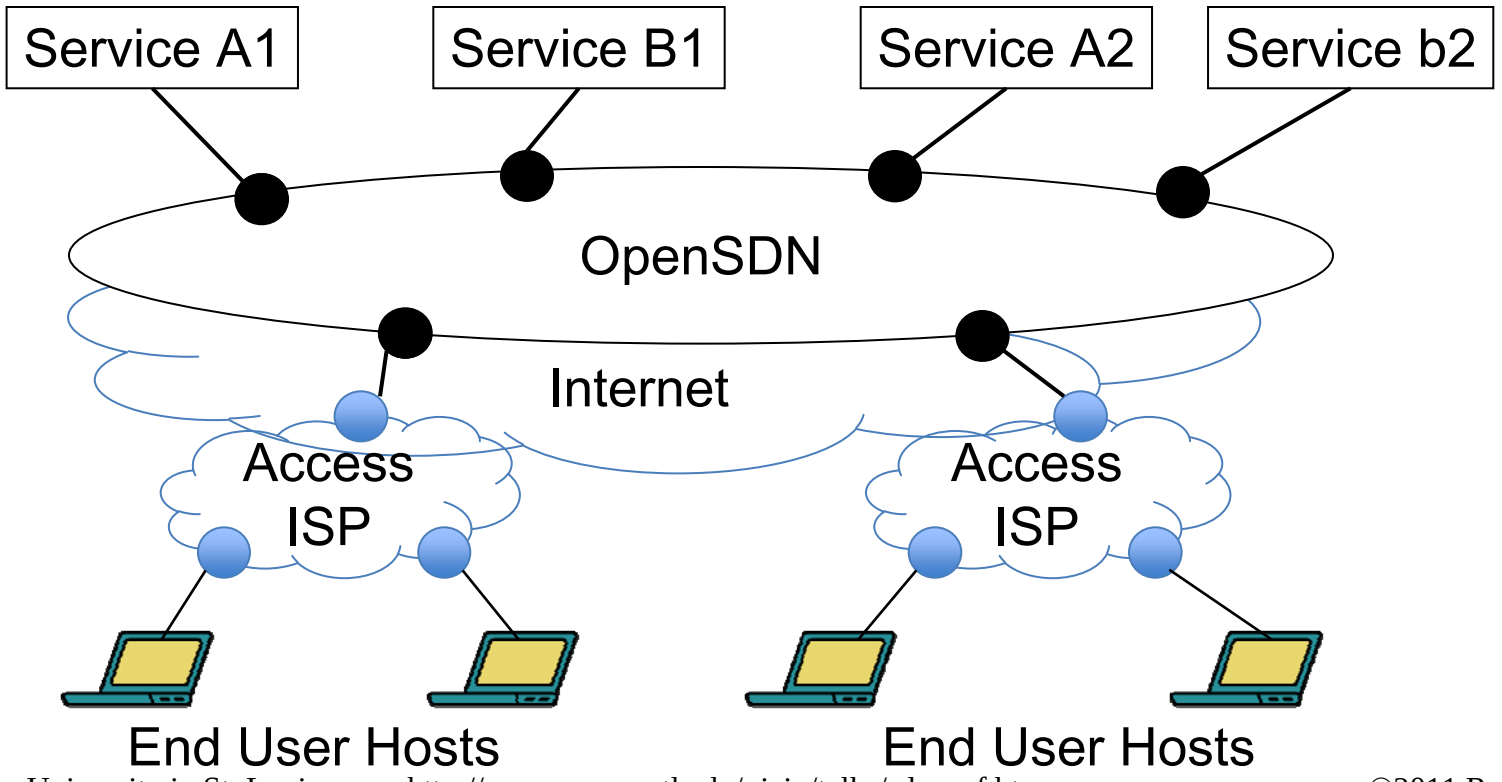
- Services totally avoid the Internet core ⇒ Many private WANs
- Google WAN, Akamai ⇒ Rules about how to connect users



Opportunity for ISPs to offer these types of WAN services

OpenSDN

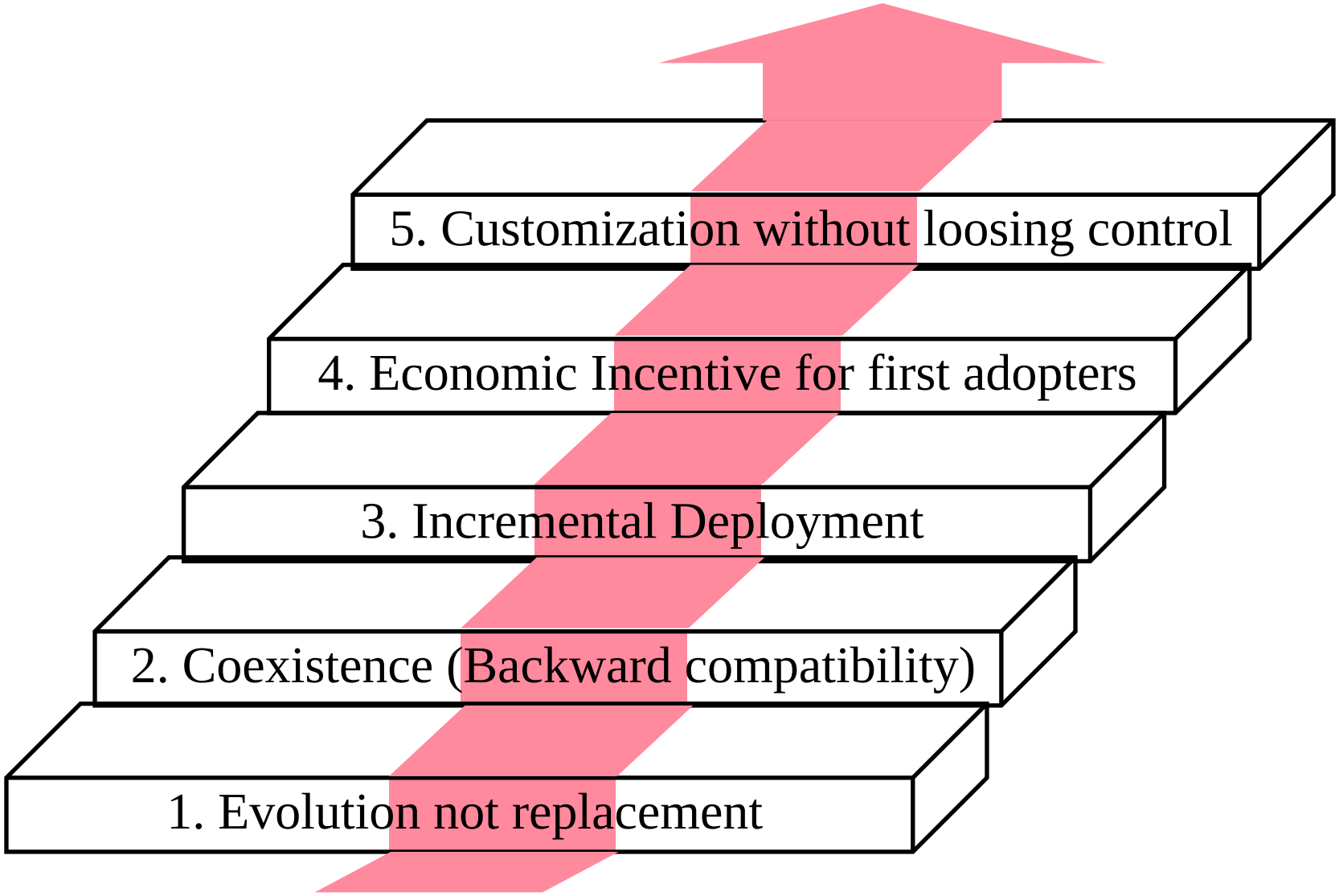
- High-Speed WAN architected for Service Delivery.
- Allows ASPs to quickly setup services



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



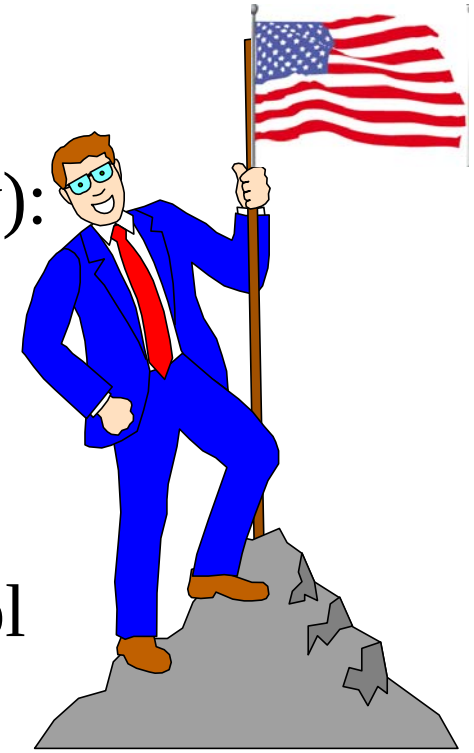
Networking: Failures vs Successes

- 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.

Five Architecture Design Principles

1. Evolution not replacement.
2. Coexistence (Backward compatibility): Old on New. New on Old
3. Incremental Deployment
4. Economic Incentive for first adopters
5. Customization without loosing control (No active networks)



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

Five Key Components of Architecture

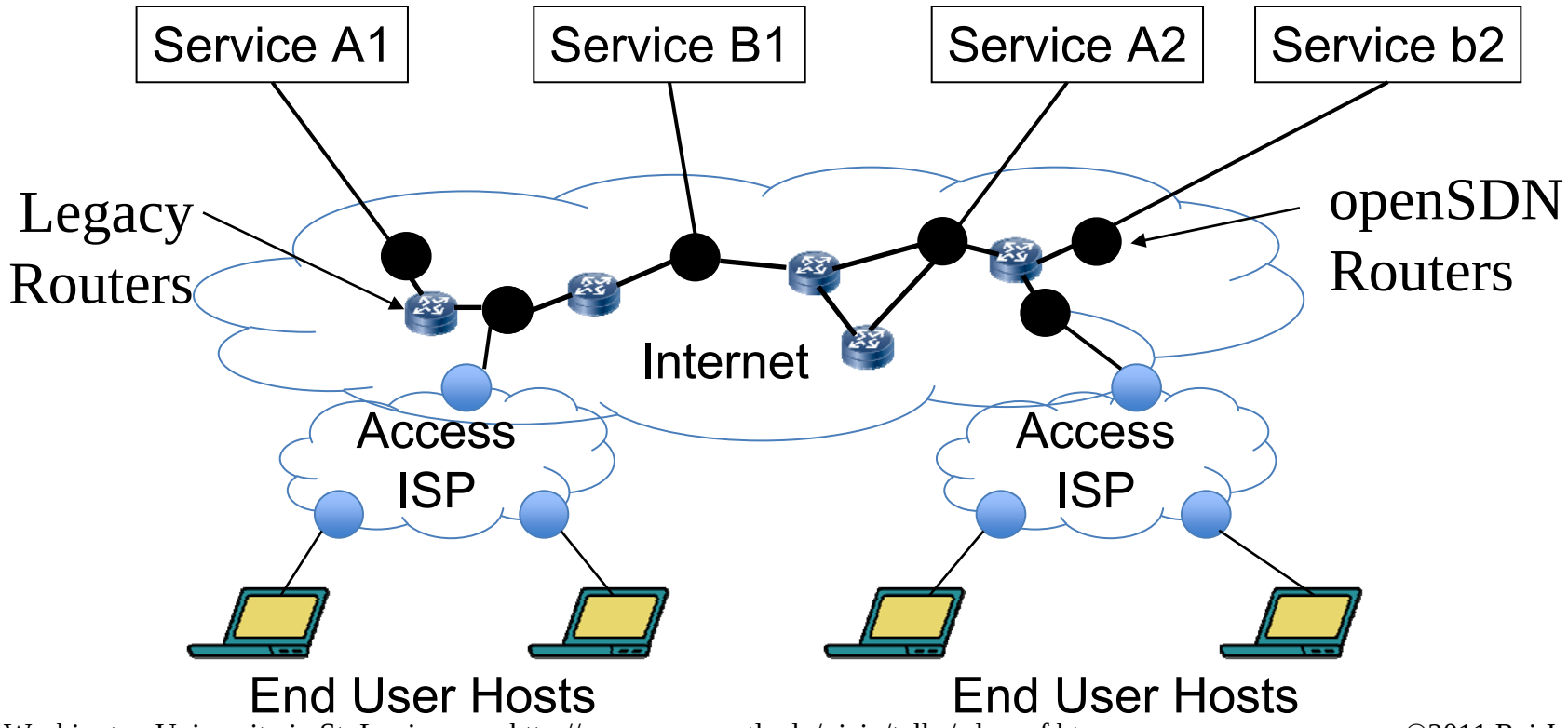
1. Naming
2. Data Plane (Forwarding)
3. Control Plane (Routing)
4. Management Plane (Monitoring, Fault tolerance, …)
5. Security

Naming

- Globally unique name with attributes
  $\Rightarrow$ Attribute based naming
- Attributes: Location, Type
- IDs: Service ID, Host ID, Data ID, User ID, Infrastructure Point-of-Attachment ID (= Locator)
- Applications are bound to IDs
- All IDs are 128-bit
  $\Rightarrow$ No changes to current applications

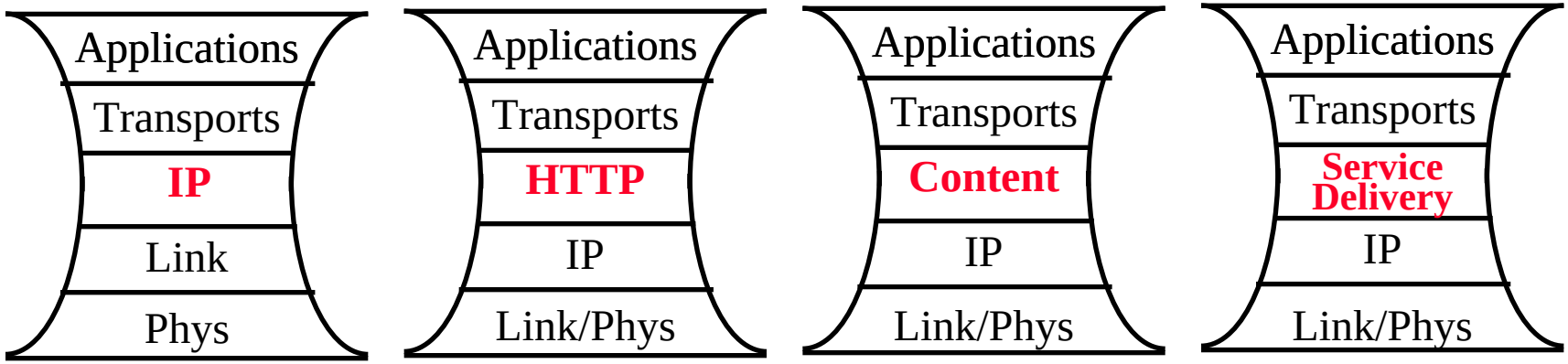
Embedded openSDN

- Co-exist openSDN routers and legacy routers

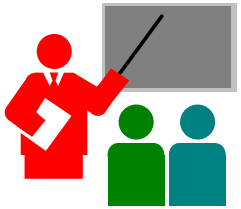


The Narrow Waist

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



Summary



1. Profusion of services on the Internet
2. It is easy to find computing resources for global services but appropriate networking architecture need to be designed.
3. Services need replication, fault tolerance, traffic engineering, security, …
4. New architectures need evolution, backward compatibility, incremental deployment, economic incentives, customization without loosing control for success. Clean slate will not work.
5. openSDN provides these features with naming for services, hosts, users, and data objects. Supports legacy nodes.

Need new evolutionary architecture for cloud-based service delivery