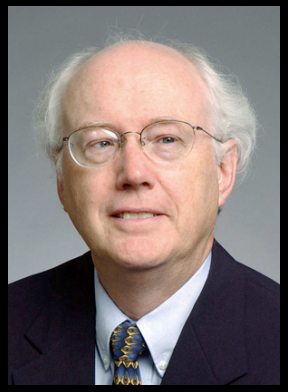
The evolving Internet industry

Disaggregation, SDN and NFV

Coming up later this week….

Wednesday: Jeff Mogul (Google) on
how Google used "SDN" in their data-center and WAN networks.

Friday: Omar Baldonado (Facebook) on Network Virtualization.



David D. Clark
Chief Protocol Architect, Internet
1981 - 1989

The Design Philosophy of the DARPA Internet Protocols

David D. Clark*
Massachusetts Institute of Technology
Laboratory for Computer Science
Cambridge, MA. 02139

(Originally published in Proc. SIGCOMM '88 Computer Communication Review Vol. 18, No. 4, August 1988, pp. 106–114)

Abstract

The Internet protocol suite, TCP/IP, was first proposed fifteen years ago. It was developed by the Defense Advanced Research Projects Agency (DARPA), and has been used widely in military and commercial systems. While there have been papers and specifications that describe how the protocols work, it is sometimes difficult to deduce from

architecture into the IP and TCP layers. This seems basic to the design, but was also not a part of the original proposal. These changes in the Internet design arose through the repeated pattern of implementation and testing that occurred before the standards were set.

The Internet architecture is still evol

Architectural Principles

1. Dumb and simple

↓

Streamlined, fast, low-cost, easy to maintain, infrequent upgrades.

2. Reliable in-order delivery can be built on top, at the end-points.

↓

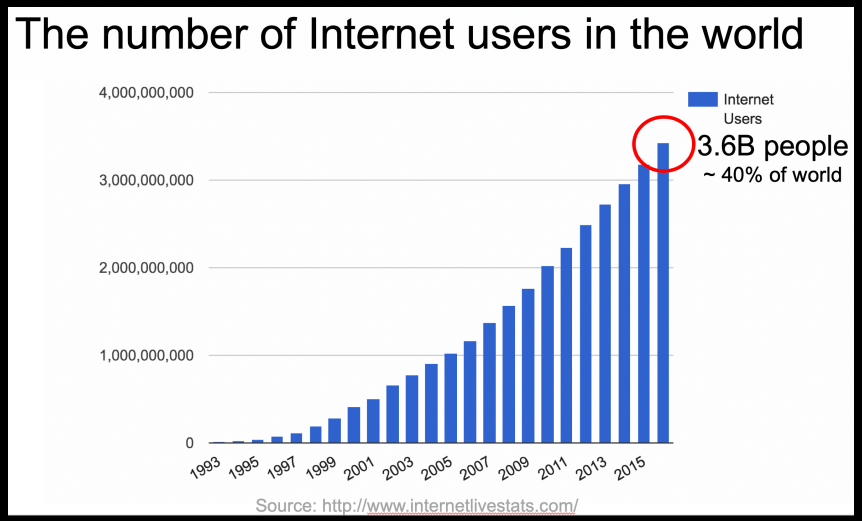
Intelligent end-points (computers).

Architectural Principles

3 Decentralized control

↓

Rapid organic growth



Tension

The Internet was carefully designed …

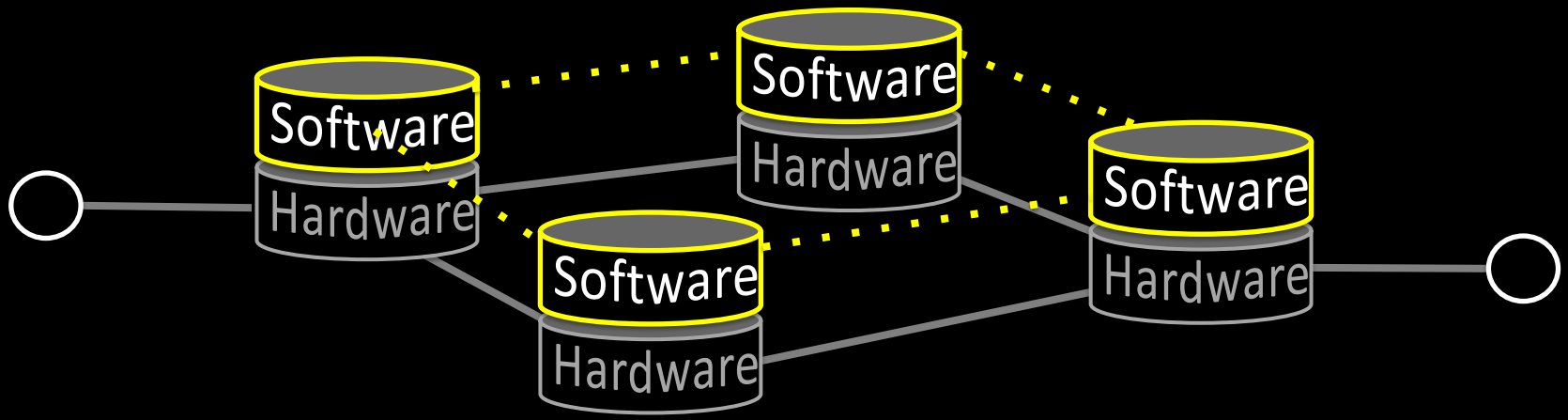
- To be simple and streamlined
- To have decentralized control: Lots of individually controlled pieces.

Which led to …

- Explosive organic growth of the Internet.
- A great business for companies selling routers.

What Internet routers do

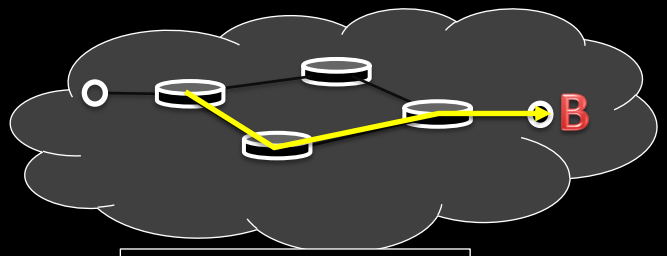
Large Public Internet Router


Software
Hardware
Software
Hardware
Software
Hardware
Software
Hardware

Edsger Dikjstra
1930-2002

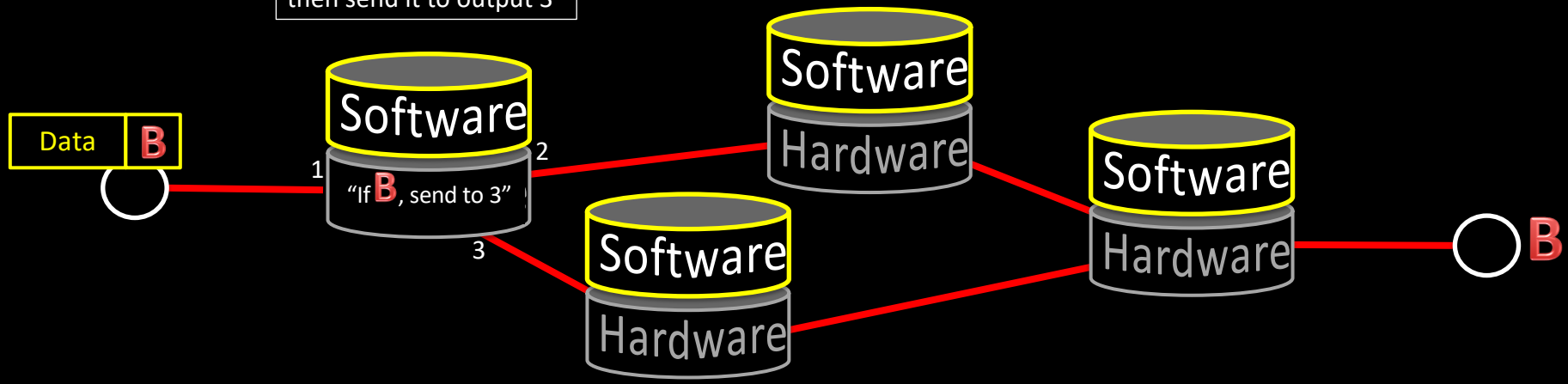

```
function Dijkstra(Graph, source):
for each vertex v in Graph:
       dist[v]  := infinity ;
       previous[v]  := undefined;
dist[source]  := 0 ;
Q := the set of all nodes in Graph ;
while Q is not empty:                  // The main loop
    u := vertex in Q with smallest distance in dist[] ;
    remove u from Q ;
    if dist[u] = infinity:
        break ;

    for each neighbor v of u:
        alt := dist[u] + dist_between(u, v) ;
        if alt < dist[v]:
            dist[v]  := alt ;
            previous[v]  := u ;
            decrease-key v in Q;
return dist[], previous[];
end function
```



1. Figure out which routers and links are present.
2. Run Dijkstra's algorithm to find shortest paths.

"If a packet is going to B, then send it to output 3"



95% → 1. Figure out which routers and links are present.

5% → 2. Run Dijkstra's algorithm to find shortest paths.

Network Working Group
Request for Comments: 2328
STD: 54
Obsoletes: 2178
Category: Standards Track

J. Moy
Ascend Communications, Inc.
April 1998

OSPF Version 2

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

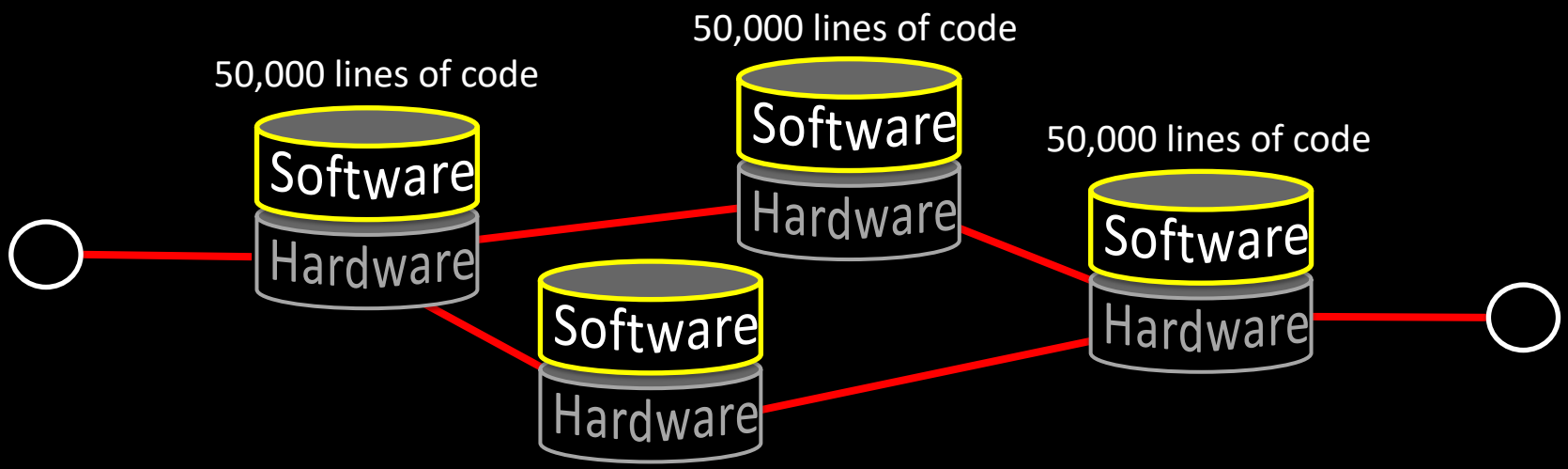
Copyright Notice

Abstract

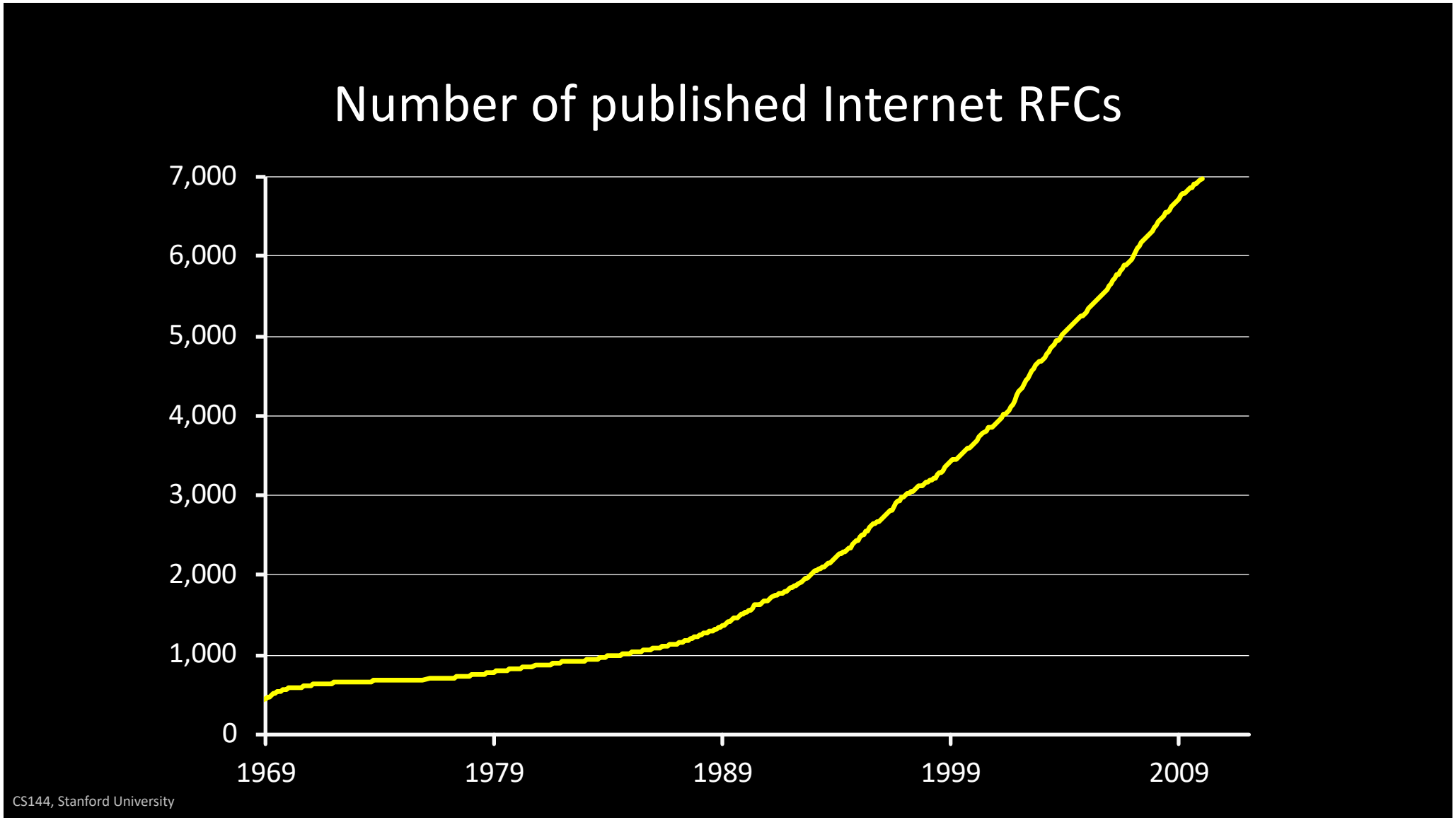
This memo documents version 2 of the OSPF protocol. OSPF is a link-state routing protocol. It is designed to be run internal to a single Autonomous System. Each OSPF router maintains an identical

95% → 1. Figure out which routers and links are present.

5% → 2. Run Dijkstra's algorithm to find shortest paths.



Building a reliable distributed system is really hard!


Number of published Internet RFCs
7,000
6,000
5,000
4,000
3,000
2,000
1,000
0
1969
1979
1989
1999
2009

By 2005

- Internet routers were complex: Over 100M lines of source code
- They were closed and proprietary, vertically integrated, expensive.
- Very hard to manage or control as a network. No clean APIs.
- ISPs felt under a “stranglehold” by the router manufacturers
- Internet research community was frustrated:
  - Internet was “ossified”
  - Easy to invent new ideas; but hard to test and deploy

Two parallel trends

1 Research community

- New programs: Clean Slate Program, NSF programs
- New research: Questioning closed ecosystem
- New switch API: OpenFlow
- New open source: Control planes, switches, emulators

2 Networking industry

Rise of data centers

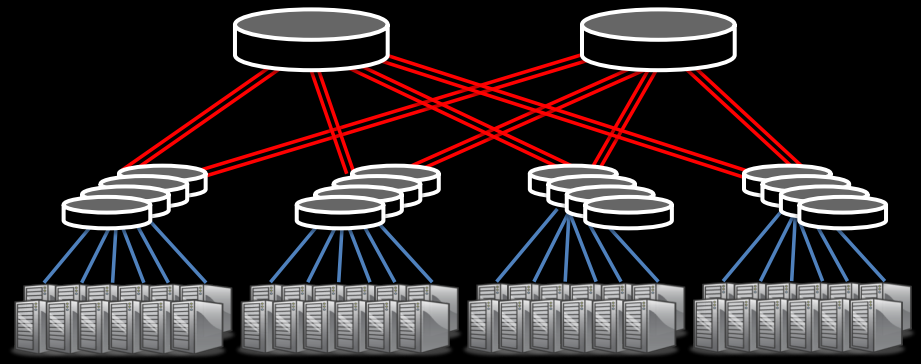
- Rise of Linux
- Availability of merchant switch chips
- Disaggregation of servers

ISPs under stress

- Falling consumer prices
- Need to differentiate

Somehow, control needed to move from the equipment vendors to those who own and operate networks

Example: Data Center Owner



Cost

500,000 servers

25,000 switches

$10k per switch = $250M

$2k commodity switch = $50M

Savings in 5 data centers = $1B

Control

More flexible control of traffic

Tailor network as needed

Quickly improve and innovate

e.g. Google B4 network from 50% to 95%

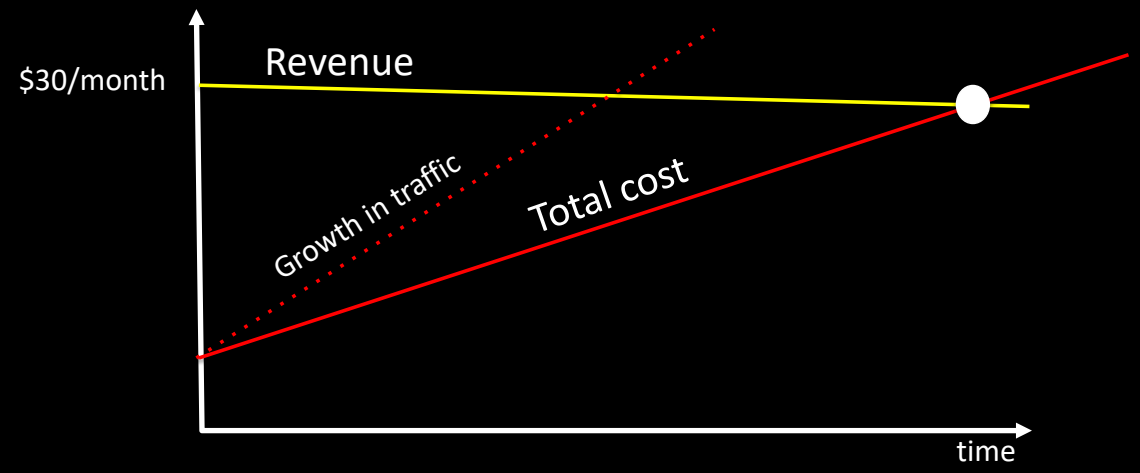
Example: Internet Service Providers

Global IP traffic growing 50% per year

End-customer monthly bills stay the same

Therefore, cost of ownership needs to reduce 50% per Gb/s per year

But in practice, reduces by <20% per year



Research community in 2006

1. Stanford Clean Slate Program
   "What would we do if we started (the Internet) again?"

2. Martin Casado and the Ethane Project

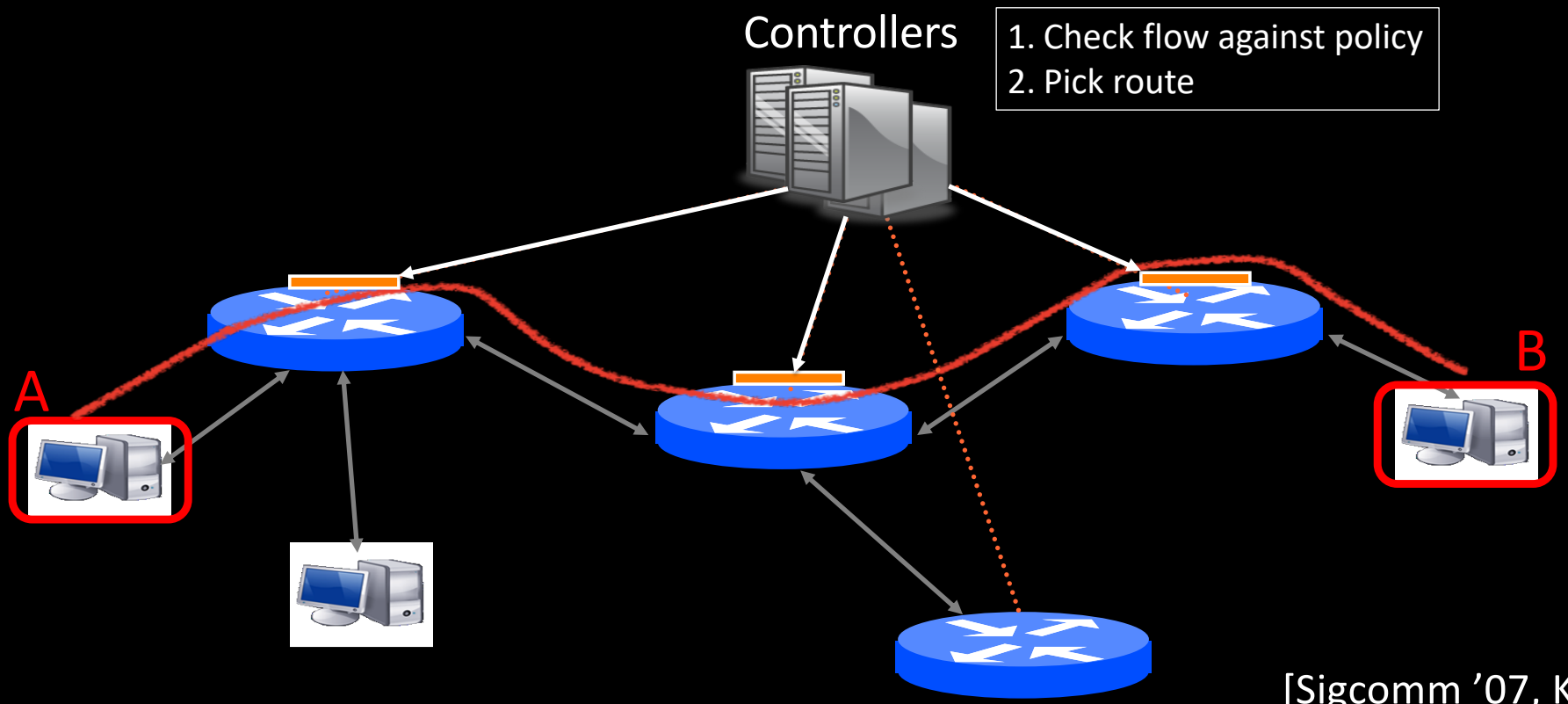
How difficult is it to define all network operations in software, outside the datapath?

2006

35,000 users
10,000 new flows/sec
137 network policies

2,000 switches
2,000 switch CPUs

Crazy question: What if software decides whether to accept each flow, and how to route it?



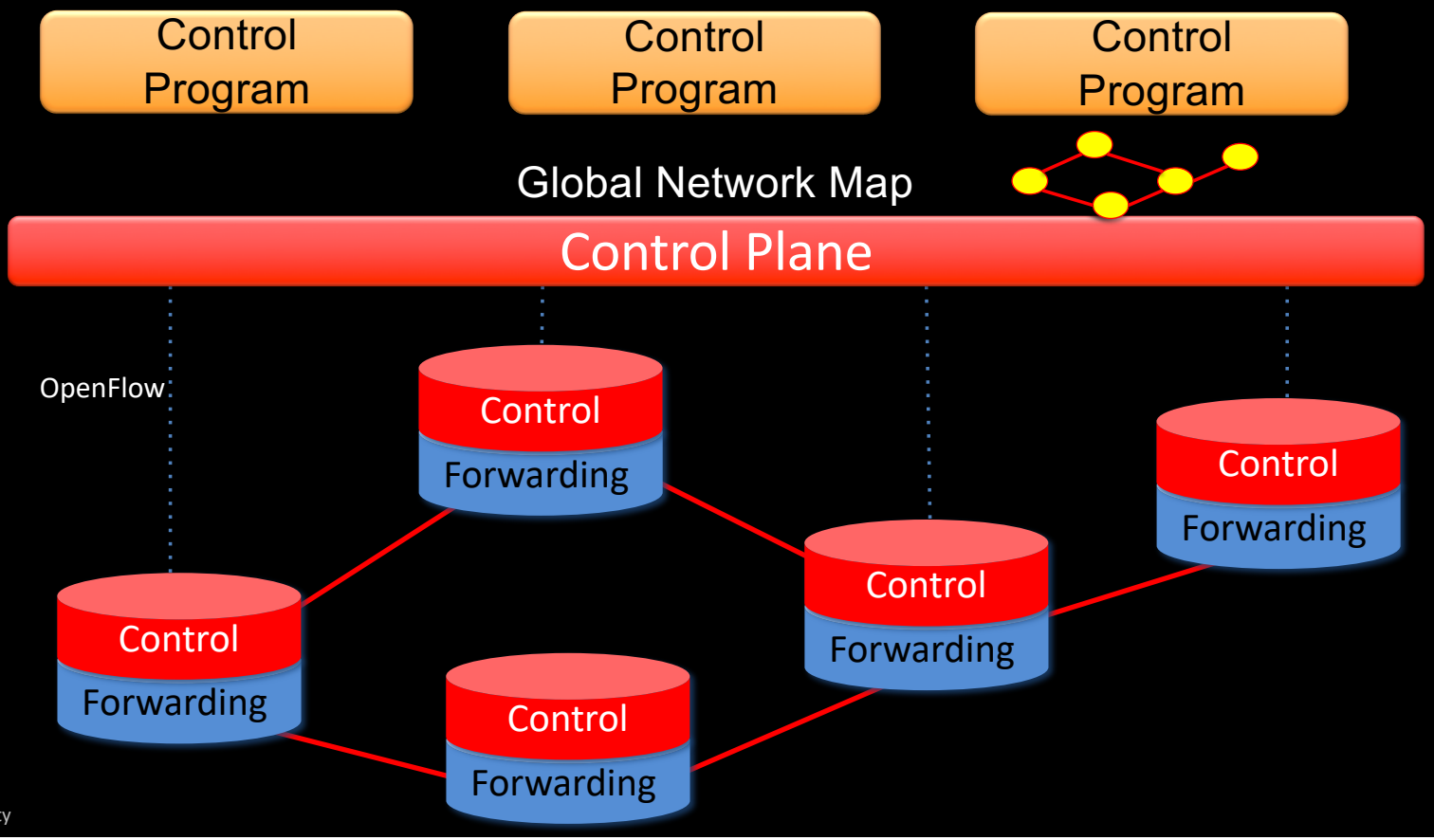
[Sigcomm '07, Kyoto]

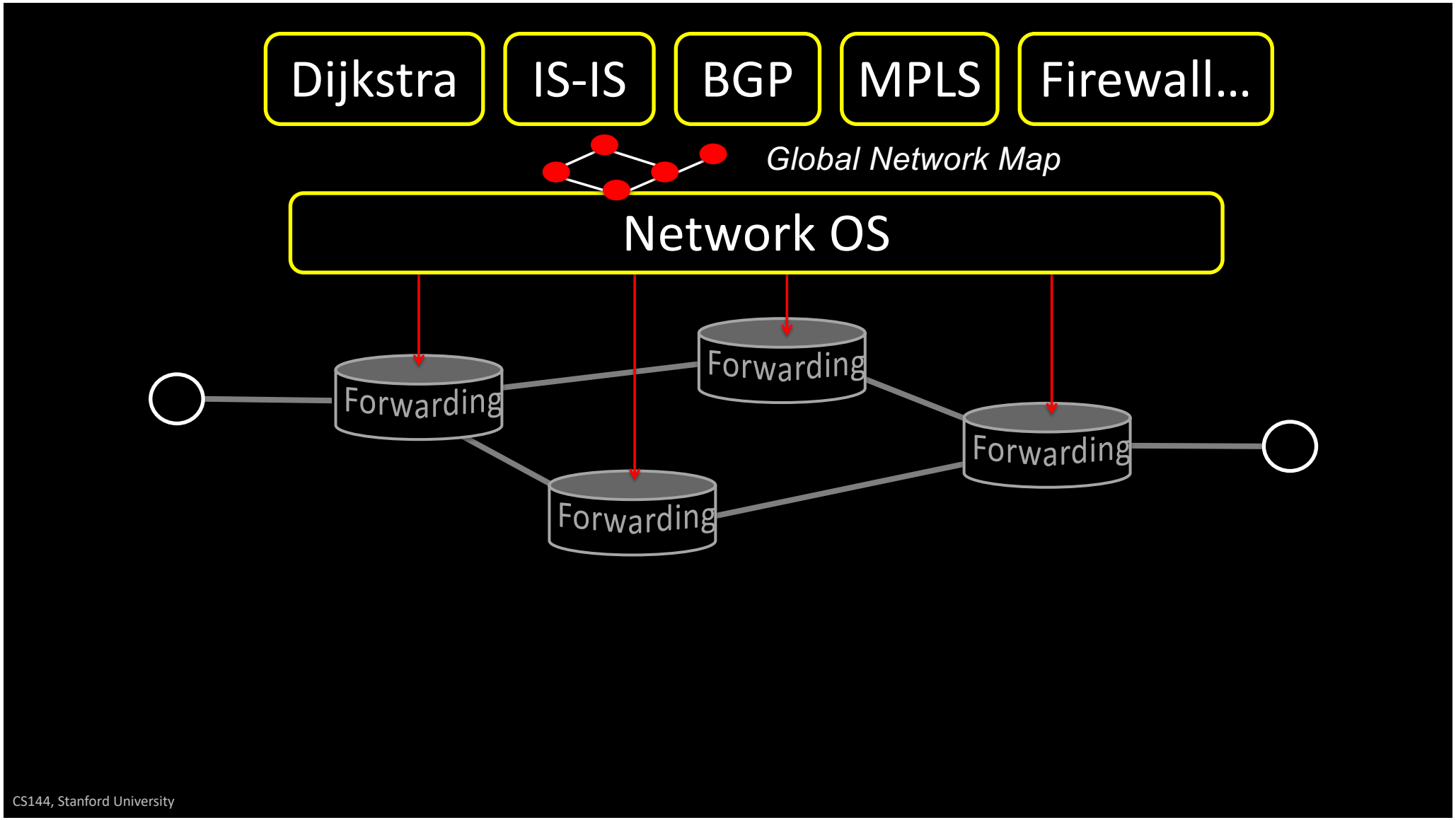
How many $400 servers do we need for 35,000 Stanford users?

Answer: Less than one!

If we can define network behavior outside the datapath, then eventually we will.

Software Defined Network (SDN)




Dijkstra
IS-IS
BGP
MPLS
Firewall…
Global Network Map
Network OS
Forwarding
Forwarding
Forwarding
Forwarding
Forwarding

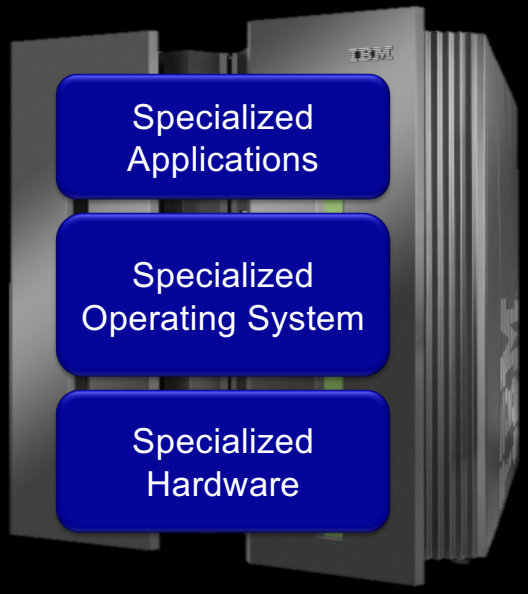
Quickly led to…

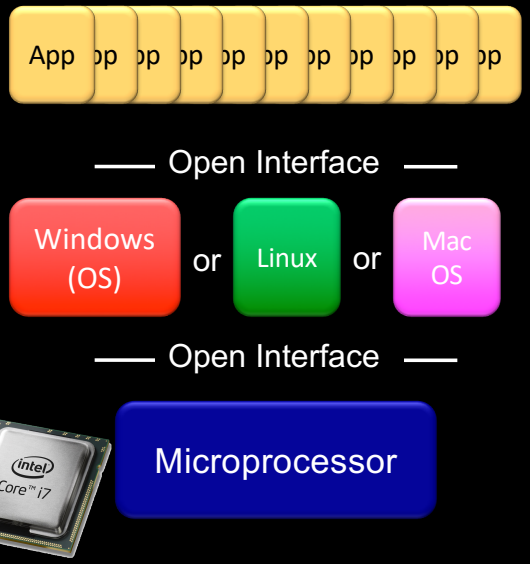
OpenFlow, SDN, Open vSwitch, network virtualization, …

In 2005, every data center and ISP was built using closed, proprietary routers.

Today, the top 10 data center owners all build their own, and write their own software (or use open source)

Computer Industry





Networking Industry

App App App App App App App App App App App

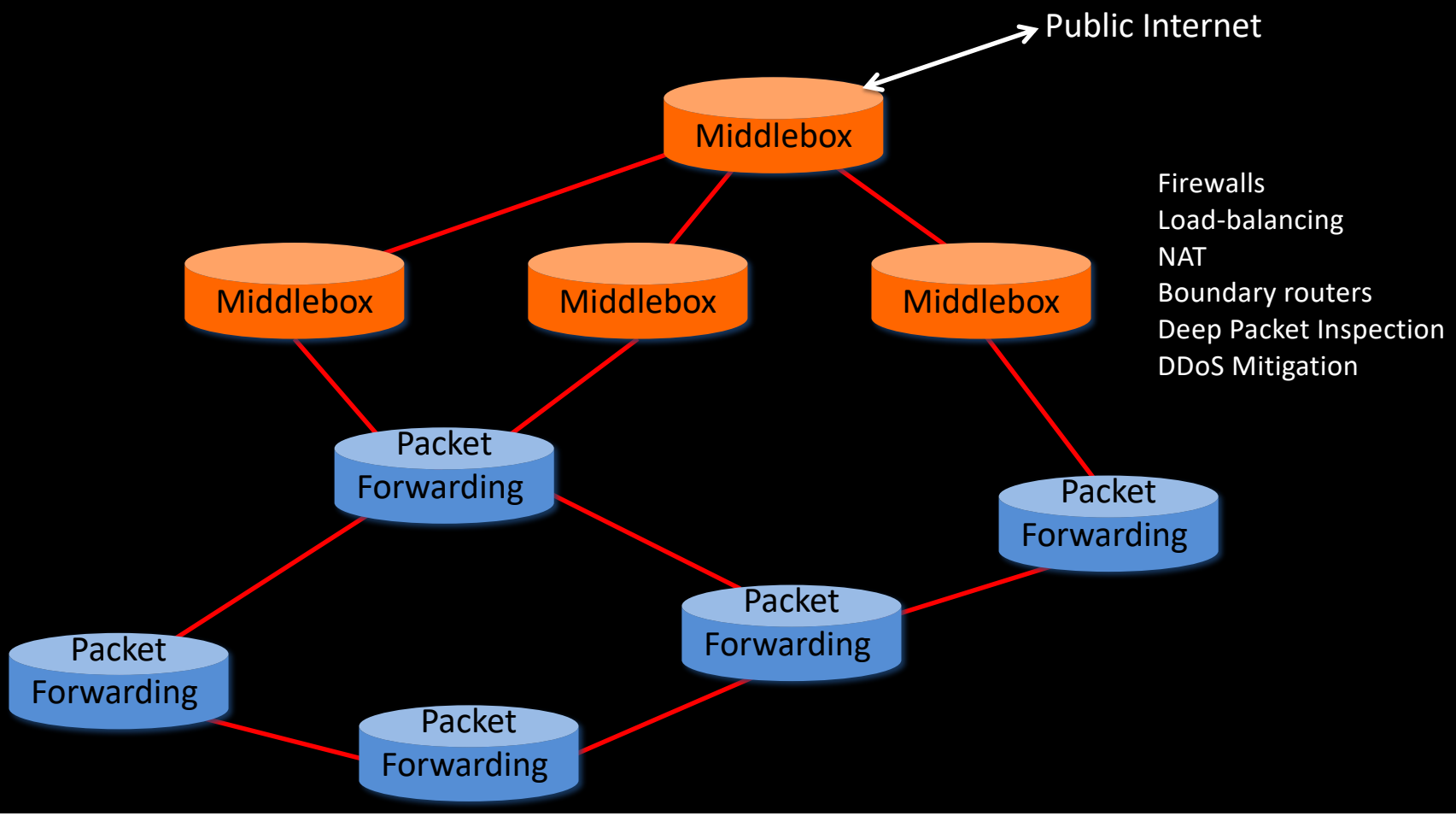
Open Interface

NOX Beacon ONIX POX ONOS Flood light Trema ODL Ryu

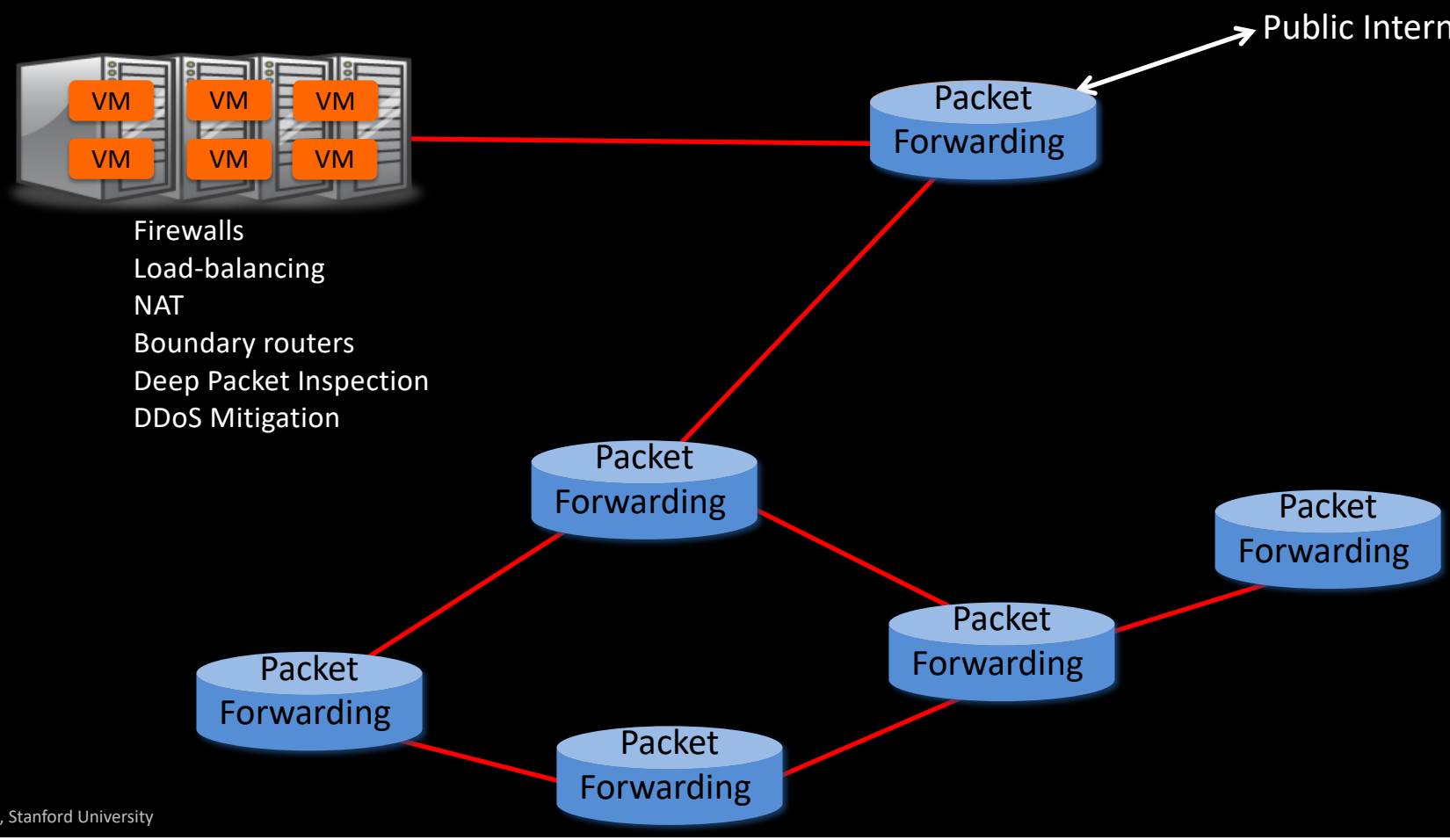
Open Interface

Switch Chips

Network Function Virtualization (NFV)



Network Function Virtualization (NFV)



With hindsight, Disaggregation, SDN and NFV were inevitable

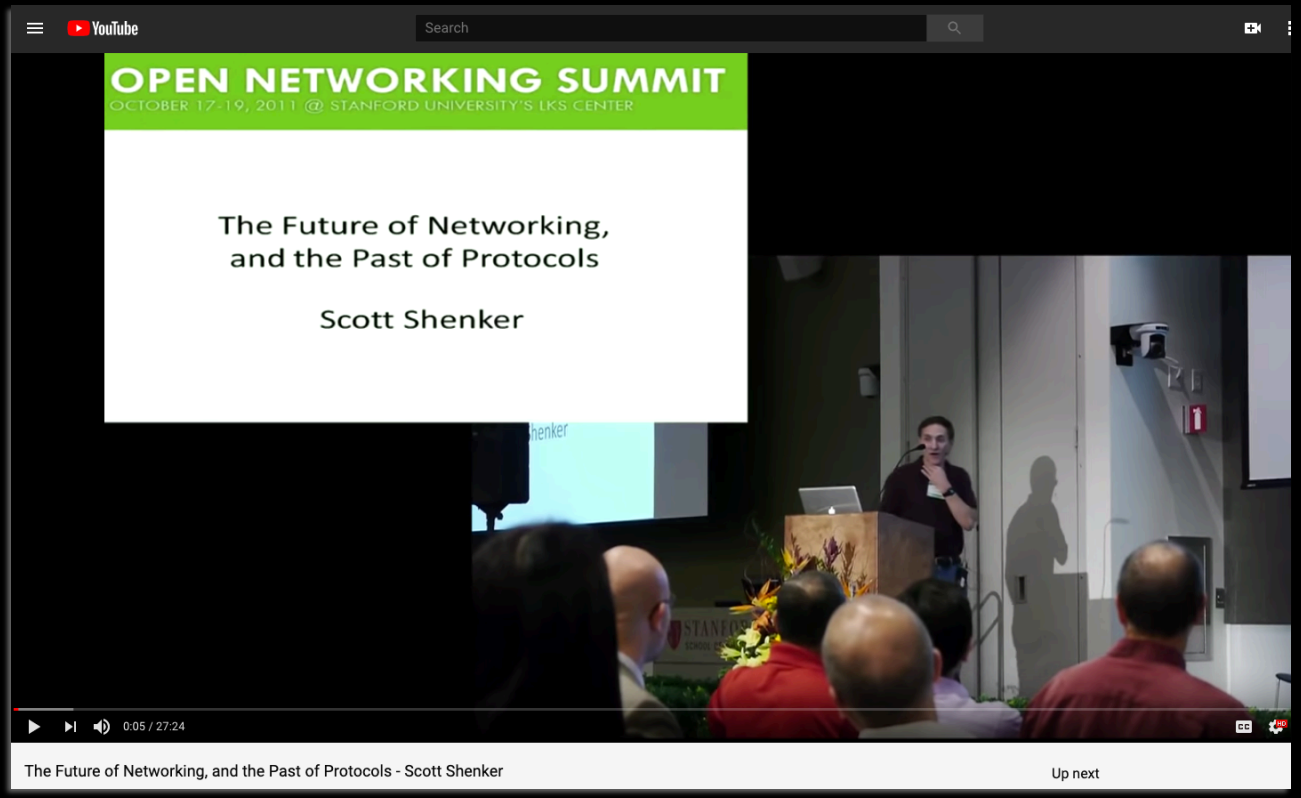
Part of a bigger trend towards the owners and operators of networks taking control of how they work

Inevitable because…

1. Rise of Linux.
2. Rise of baremetal servers and data centers.
3. SDN: Rise of merchant switching silicon.
4. NFV: Rise of computer virtualization.

“The Future of Networking and the Past of Protocols”

Scott Shenker 2011



link

Coming up later this week….

Wednesday: Jeff Mogul (Google) on
how Google used "SDN" in their data-center and WAN networks.

Friday: Omar Baldonado (Facebook) on Network Virtualization.