



# Advanced Networking and Cybersecurity

IBI Darmajaya

2022-2023



6.888

# Advanced Topics in Networking

Introduction to Data Center  
Networking

Mohammad Alizadeh

Spring 2016

✧ Includes material from lectures by Nick McKeown (Stanford), Jennifer Rexford (Princeton), and George Porter (UCSD)

# The Internet: An Exciting Time

## One of the most influential inventions

- A research experiment that escaped from the lab
- ... to be the global communications infrastructure

## Ever wider reach

- Today: 2 billion users, 15 billion devices
- Tomorrow: more users, content, sensors, “things”, 40 billion devices by 2020

## Constant innovation

- Web, P2P, video, online shopping, social networks, cloud, ...

# Transforming Everything

## The ways we do business

- E-commerce, advertising, cloud computing, ...

## The way we have relationships

- E-mail, IM, Facebook friends, virtual worlds

## The way we think about law and govern

- Interstate commerce, national boundaries?
- Censorship and wiretapping

## The way we fight

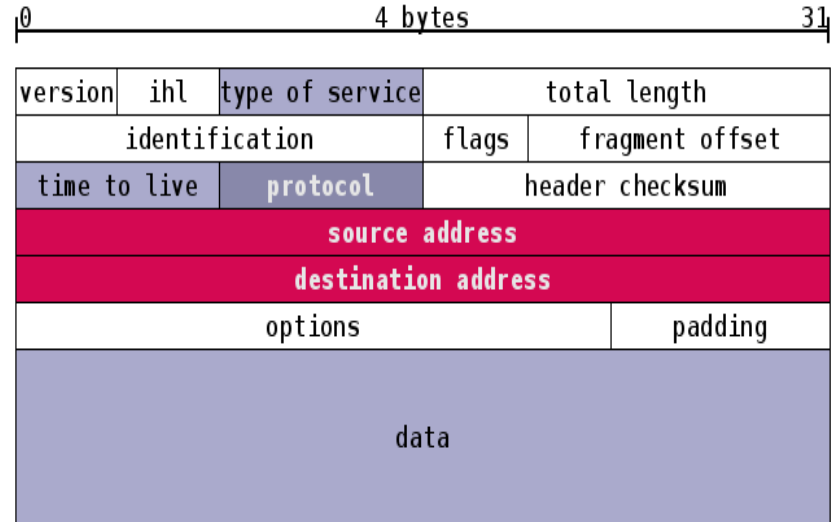
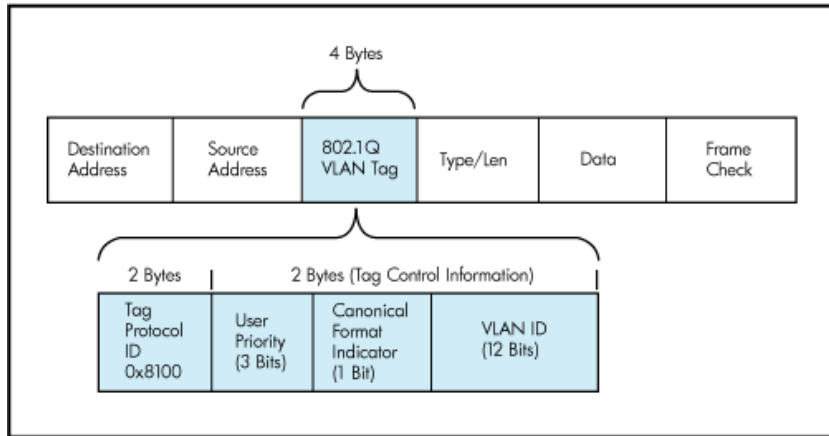
- Cyber-attacks, including nation-state attacks

But what *is* networking?

# A Plethora of Protocol Acronyms?

SNMP WAP SIP PPP IPX MAC  
LLDP FTP UDP ICMP IMAP IGMP HIP  
OSPF RTP BGP HTTP ARP ECN  
PIM RED BGP HTTP ARP ECN  
RIP IP MPLS TCP RTCP  
SMTP RTSP BFD CIDR  
NNTP SACK TLS NAT STUN  
DNS SACK SSH TLS NAT STUN  
POP VLAN LISP VTP TFTP DHCP LDP<sub>6</sub>

# A Heap of Header Formats?

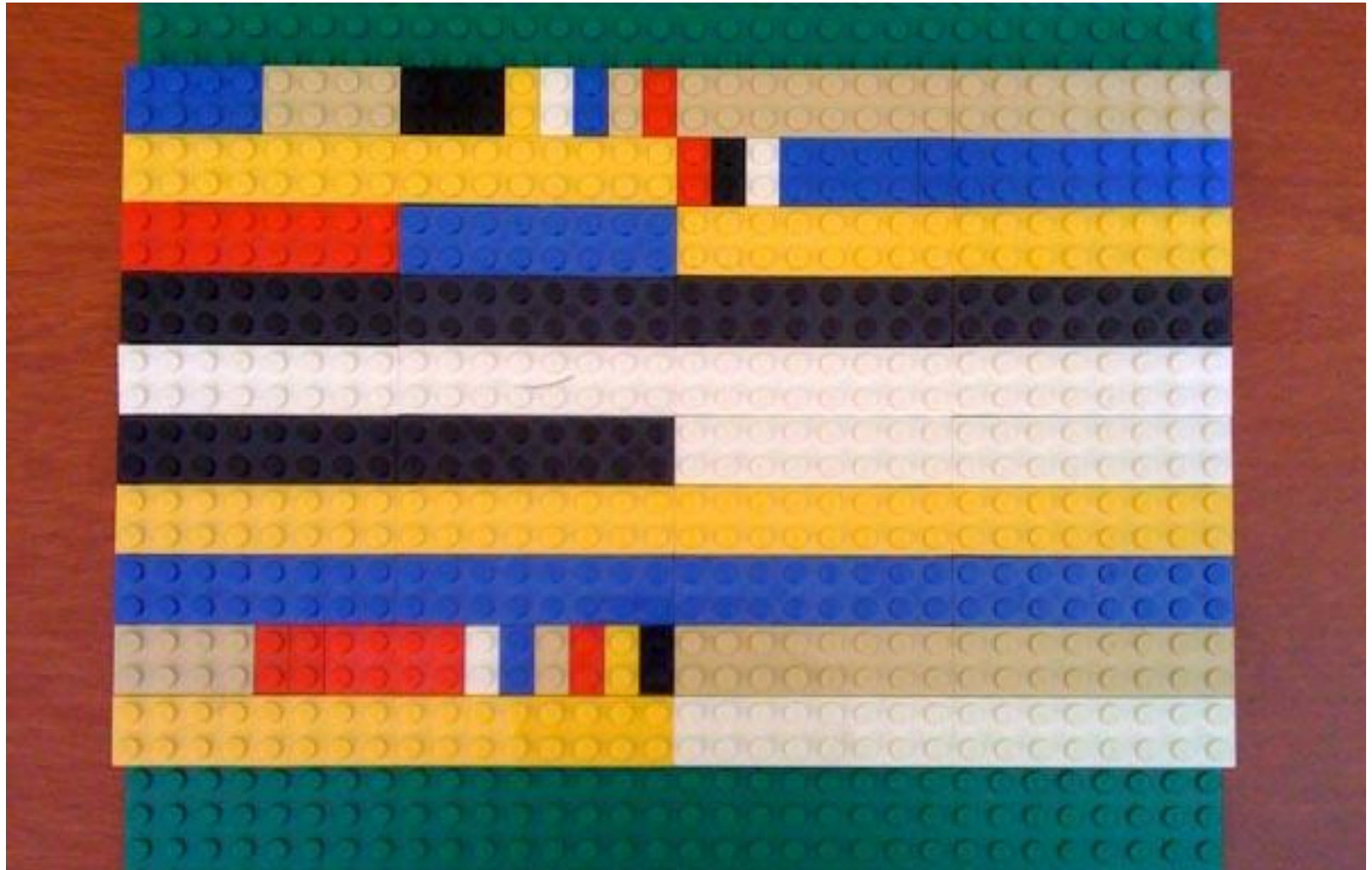


<b>Source Port</b>		<b>Destination Port</b>						
<b>Sequence Number</b>								
<b>Acknowledgment Number</b>								
<b>Data Offset</b>	<b>Reserved</b>	<b>URG</b>	<b>ACK</b>	<b>PSH</b>	<b>RST</b>	<b>SYN</b>	<b>FIN</b>	<b>Window</b>
<b>Checksum</b>				<b>Urgent Pointer</b>				
<b>Options</b>				<b>Padding</b>				

## HTTP Response Header

Name	Value
HTTP Status Code: HTTP/1.1 200 OK	
Date:	Thu, 27 Mar 2008 13:37:17 GMT
Server:	Apache/2.0.55 (Ubuntu) PHP/5.1.2
Last-Modified:	Fri, 21 Mar 2008 13:57:30 GMT
Etag:	"358a4e4-56000-ddf5c680"
Accept-Ranges:	bytes
Content-Length:	352256
Connection:	close
Content-Type:	application/x-msdos-program

# TCP/IP Header Formats in Lego



# A Big Bunch of Boxes?

Router  
Label Switched Router  
Load balancer  
Switch  
Gateway  
Intrusion Detection System  
Bridge  
Route Reflector  
Deep Packet Inspection  
DHCP server  
Packet shaper  
NAT  
Firewall  
Hub  
Packet sniffer  
WAN accelerator  
DNS server  
Base station  
Proxy

# An Application Domain?

# A place to apply theory?

Algorithms and data structures

Control theory

Queuing theory

Optimization theory

Game theory and mechanism design

Formal methods

Cryptography

Programming languages

Graph theory

# A place to build systems?

Distributed systems

Operating systems

Computer architecture

Software engineering

...

# So, Why is Networking Cool?

## Relevant

- Can impact the real world
- Can measure/build things

## Interdisciplinary

- Well-motivated problems + rigorous solution techniques

## Widely-read papers

- Many of the most cited papers in CS are in networking
- Congestion control, distributed hash tables, resource reservation, self-similar traffic, multimedia protocols

# So, Why is Networking Cool?

## Young, relatively immature field

- Tremendous intellectual progress is still needed
- *You* can help decide what networking really is

## Defining the problem is a big part of the challenge

- Recognizing a need, formulating well-defined problem
- ... is at least as important as solving the problem.

## Lots of platforms for building your ideas

- Testbeds: Emulab, PlanetLab, Orbit, GENI
- Programmability: Click, Mininet, NetFPGA, Switch chips

# This course

... is about the latest in networking research

Main goal:

Prepare for high quality research in this field

# We'll focus mostly on...

Data Center Networking

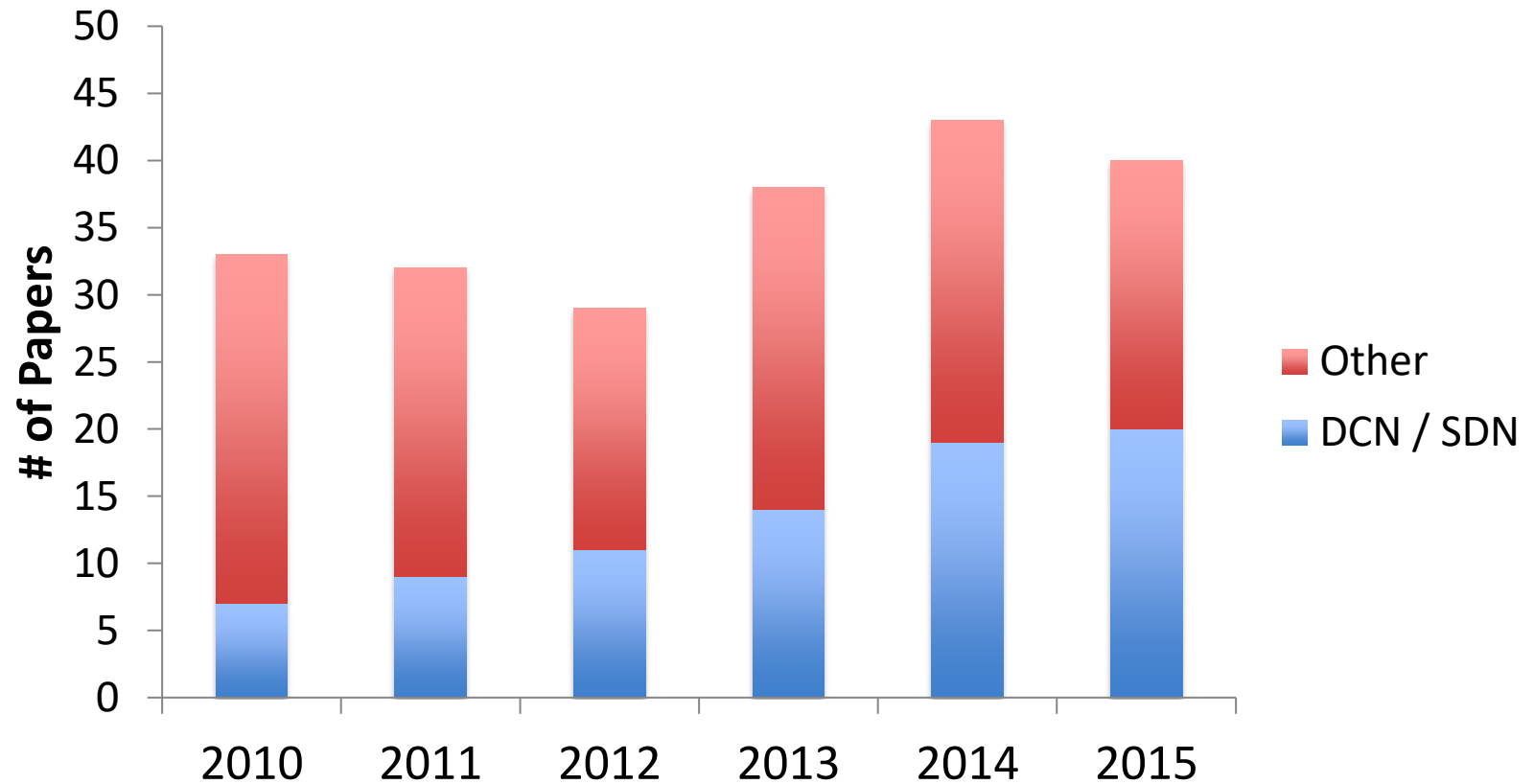
Software Defined Networking

Two “hot” areas of research

Significant interest in both academia & industry

Lots of opportunities for impact

# DCN/SDN Papers at SIGCOMM



# Readings & Presentations

Each group of two students will present *one* paper on Last Lecture (offline class)

- Read paper and relevant references
- 15 minute talk

# Data Center Networking

# What are Data Centers?

Large facilities with 10s of thousands of networked servers

- Compute, storage, and networking working in concert
- “Warehouse-Scale Computers”



# Types of Data Centers



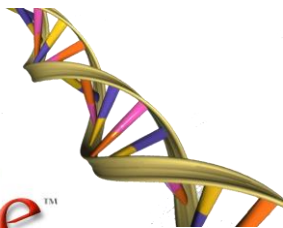
Specialized data centers built for one big app

- Social networking: Facebook
- Web Search: Google, Bing

bing™



NETFLIX



Google™

amazon®

“Cloud” data centers

- Amazon EC2, Windows Azure
- Google App Engine

# Cloud Computing

## On-demand

- Use resources when you need it; pay-as-you-go

## Elastic

- Scale up & down based on demand

## Multi-tenancy

- Multiple independent users share infrastructure
- Security and resource isolation
- SLAs on performance & reliability (sometimes)

## Dynamic Management

- Resiliency: isolate failure of servers and storage
- Workload movement: move work to other locations

# Data Centers with 100,000+ Servers



Microsoft



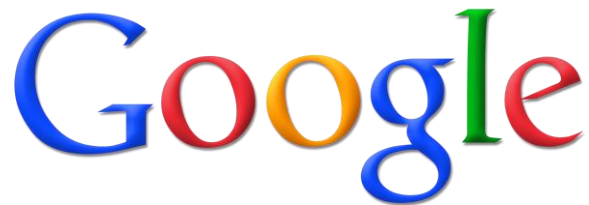
Google



Facebook<sup>23</sup>



# These things are really big

The Google logo, consisting of the word "Google" in its characteristic multi-colored font (blue, red, yellow, blue, green, red).The Facebook logo, featuring the word "facebook" in white lowercase letters on a dark blue rectangular background.The Amazon.com logo, with the text "amazon.com" in black lowercase letters and a curved orange arrow underneath the "a" and "z" characters.

10-100K servers

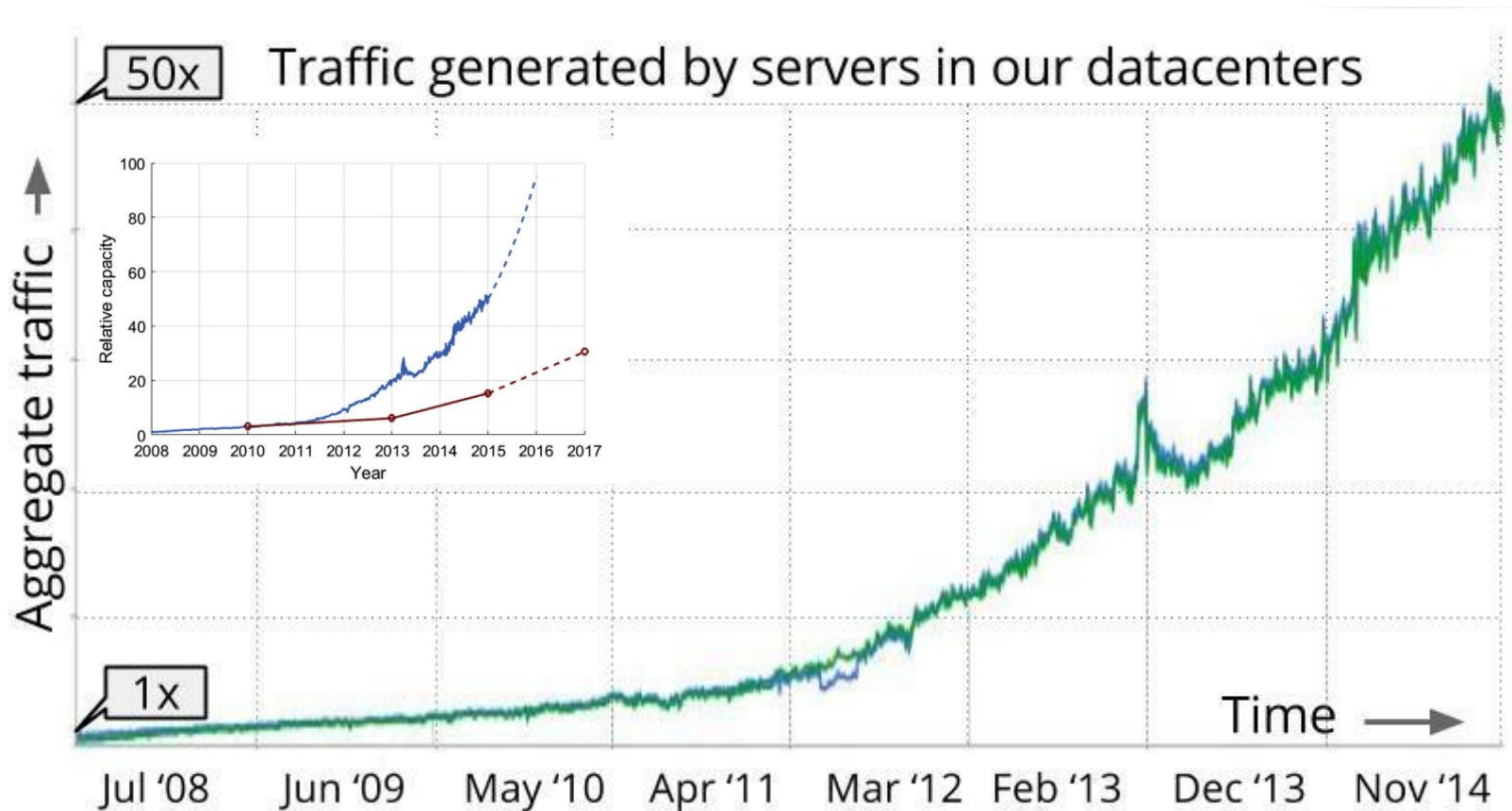
100s of Petabytes of storage

100s of Terabits/s of Bw  
(more than core of Internet)

10-100MW of power  
(1-2 % of global energy consumption)

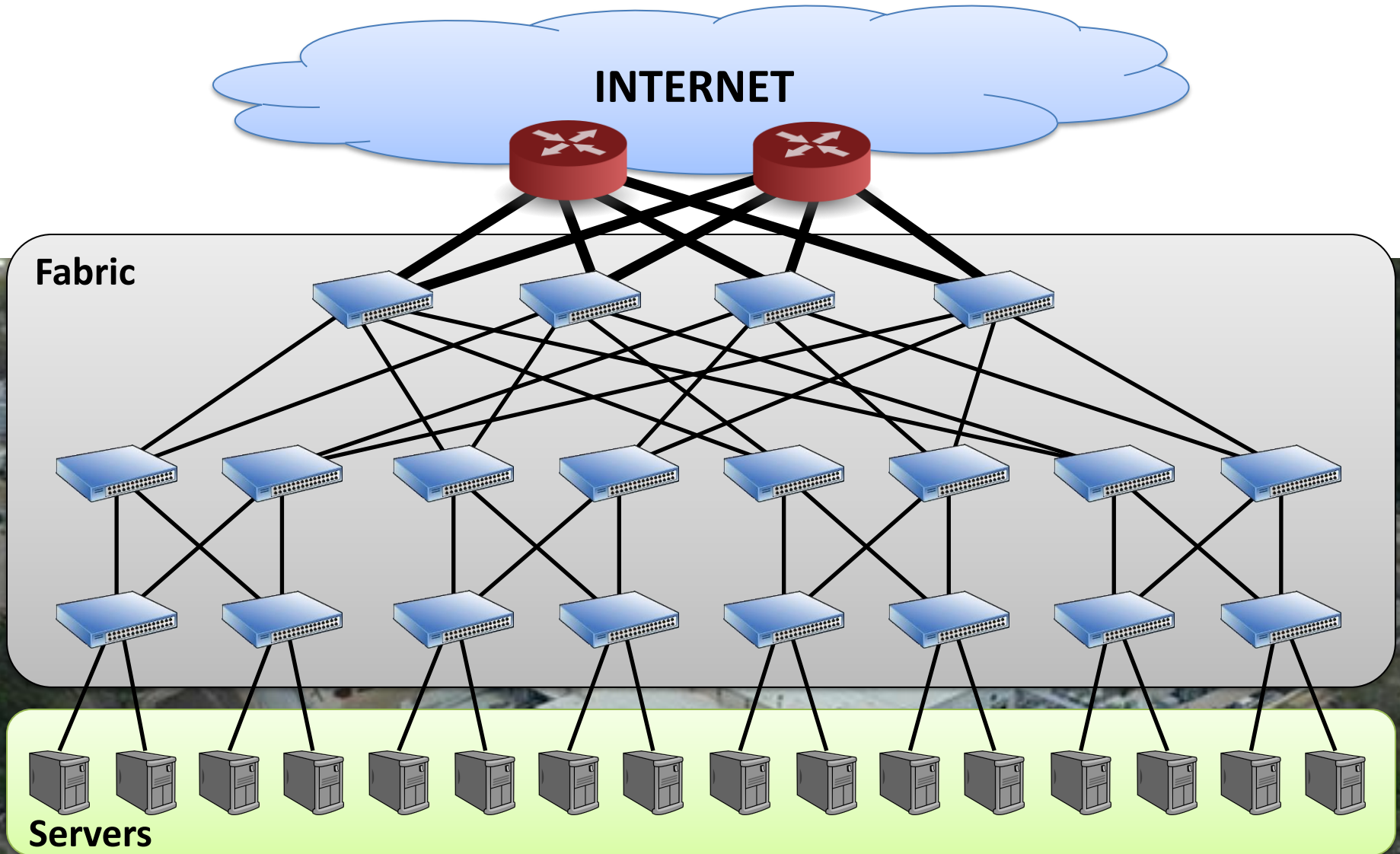
100s of millions of dollars

# Datacenter Traffic Growth



❖ Source: “Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google’s Datacenter Network”, SIGCOMM 2015.

# What's Different about DCNs?



# What's Different about DCNs?

Single administrative domain

No need to be compatible with outside world

Tiny round trip times (microseconds)

Latency/tail latency critical

Massive multipath topologies

Shallow buffers

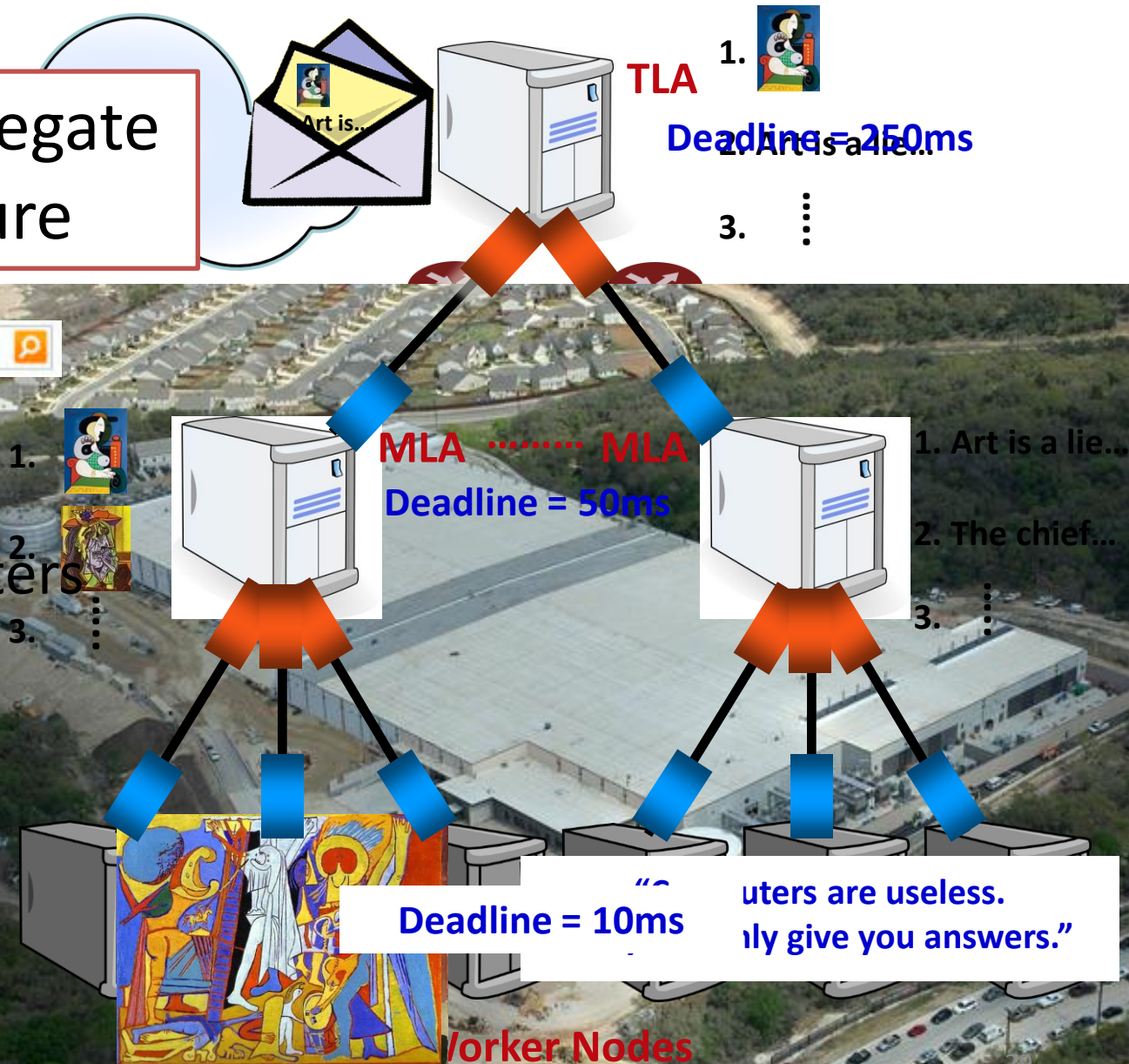
Backplane for large-scale parallel computation

# Example: Web Search

## Partition/Aggregate App Structure

- Strict deadlines
- Tail Latency Matters

Picasso



# Data Center Challenges

## Massive bisection bandwidth

- Topologies
- Load balancing
- Optics

## Ultra-Low latency (<10 microseconds)

- Rate-control or packet scheduling?
- Centralized or distributed?

## Managing resources across network & servers

- Multi-tenant performance isolation
- App-aware network scheduling (e.g. for big data)

## Next-generation hardware

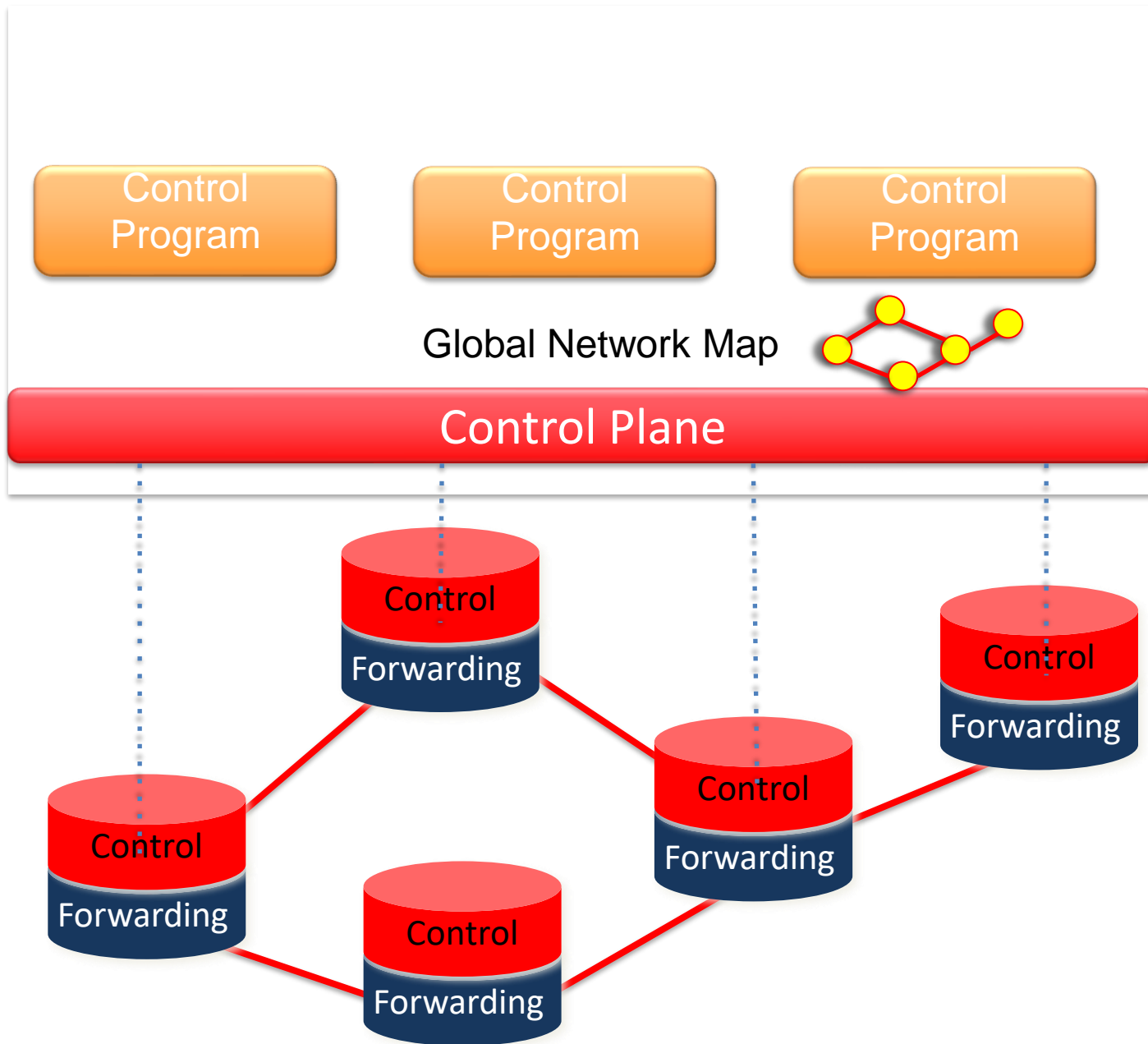
- RDMA, Rack-Scale Computing



**SDN**

# Software Defined Networking

# Software Defined Network (SDN)



# Software Defined Network

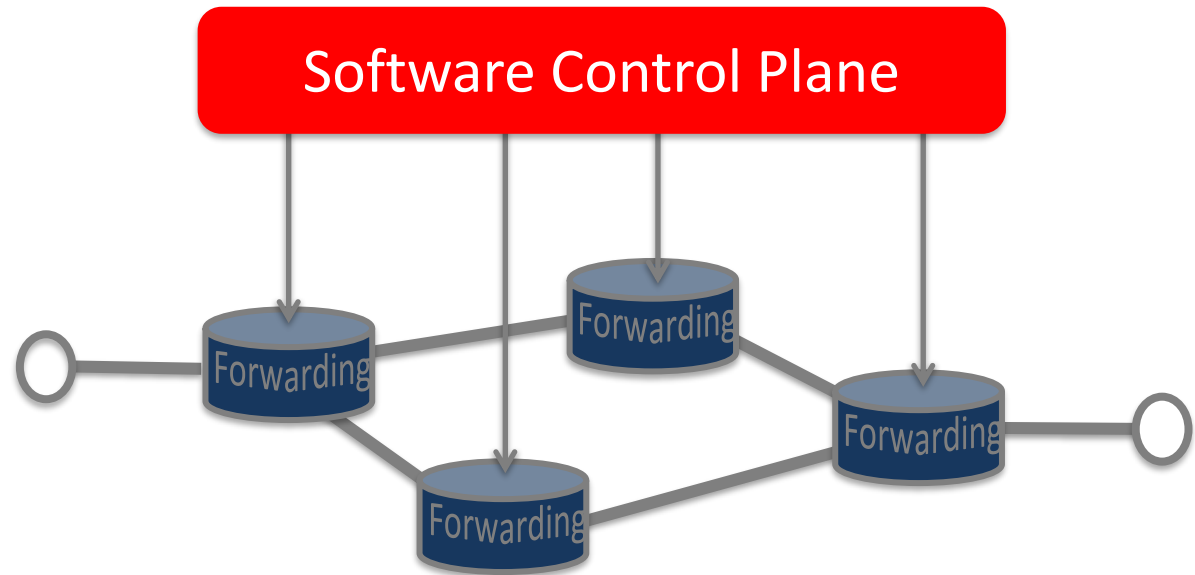
A network in which the control plane is physically separate from the forwarding plane.

*and*

A single control plane controls several forwarding devices.

(That's it)

# SDN



Intended consequences...

1. Put network owners and operators in control.
2. Networks that are more reliable and more secure.
3. Networks that cost less: simpler, streamlined hardware.
4. Networks that cost less to operate (fewer features).

# An example Routing



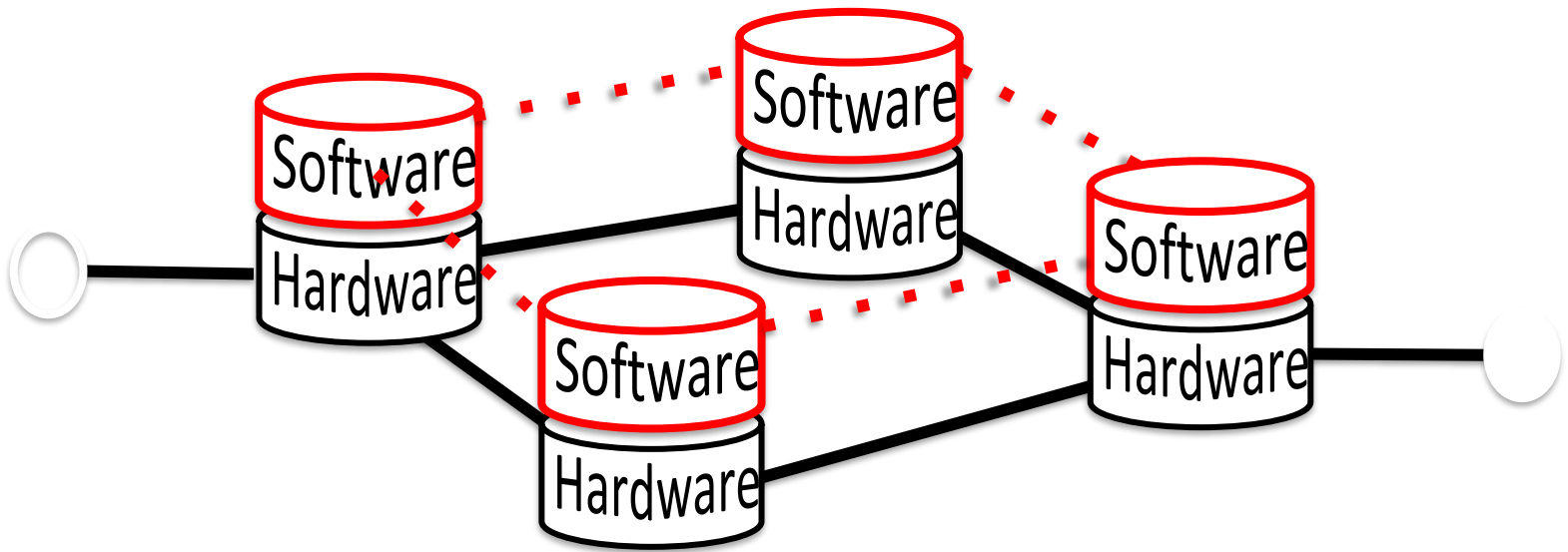
Edsger Dijkstra  
1930-2002

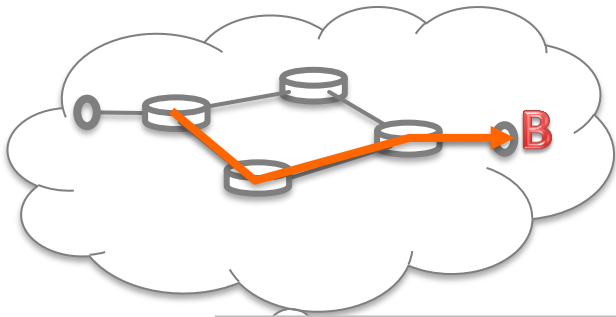
## function Dijkstra(Graph, source):

```
    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
```

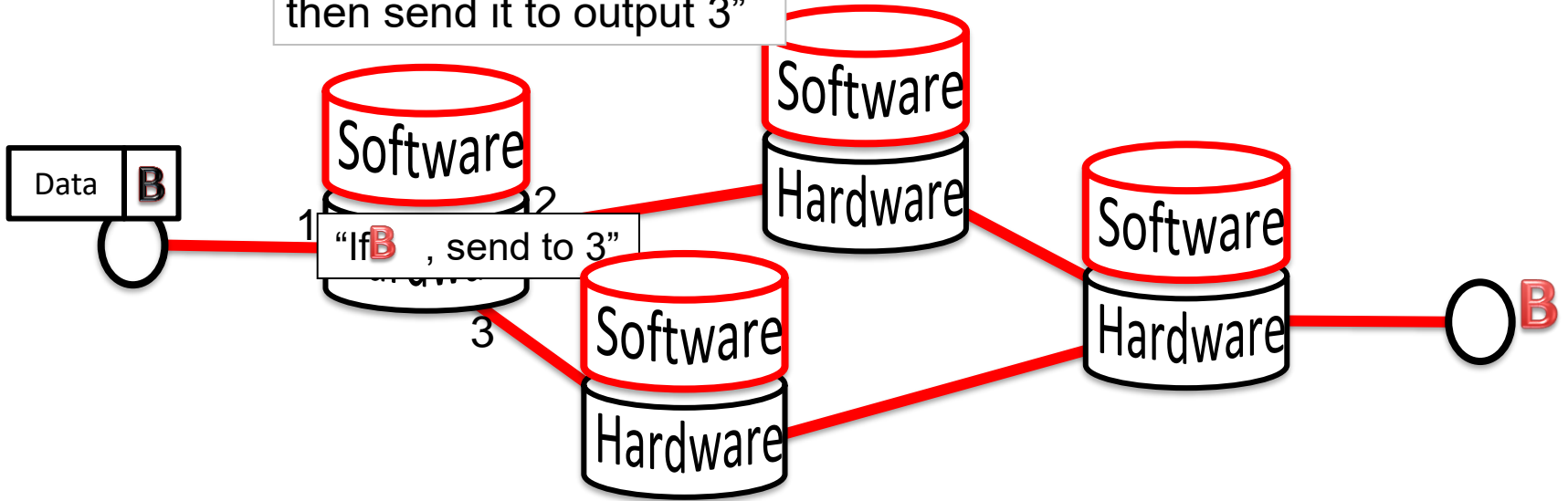






1. Figure out which routers and links are present.
2. Run Dijkstra's algorithm to find shortest paths.

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



95%

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

5%

Network Working Group  
Request for Comments: 2328  
STD. 14  
Obsoletes: [2178](#)  
Category: Standards Track

50,000 lines of code

J. Moy  
Ascend Communications, Inc.  
April 1998

50,000 lines of code

50,000 lines of code

**OSPF Version 2**

Status of this Memo

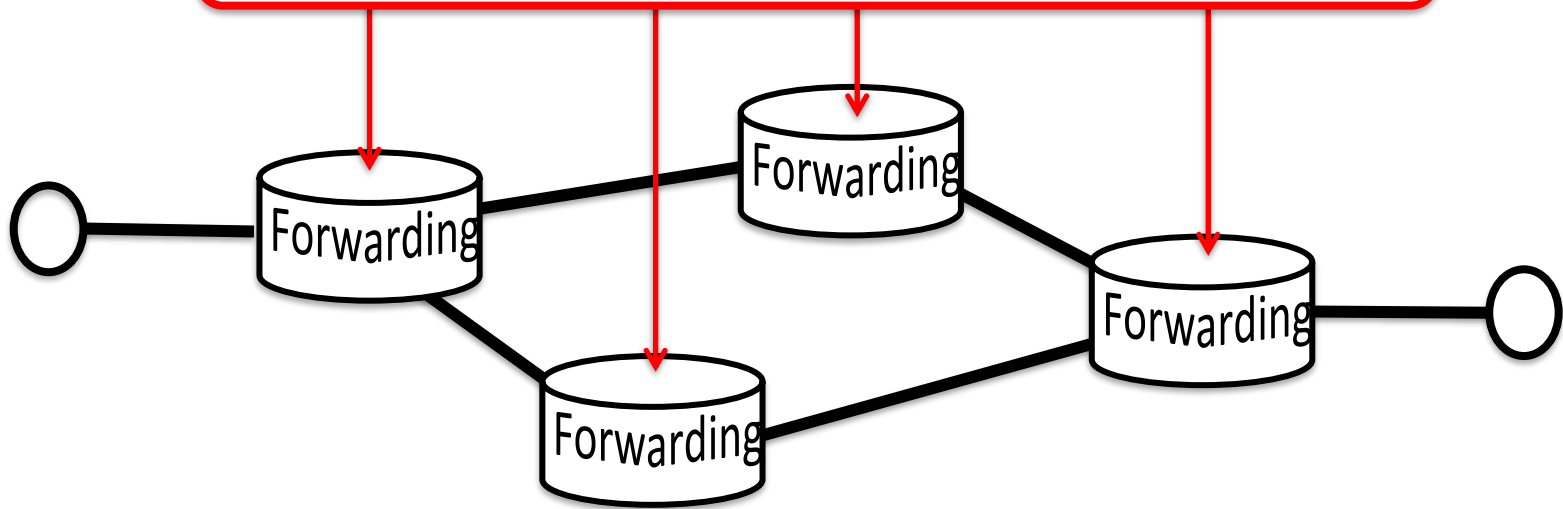
This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution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

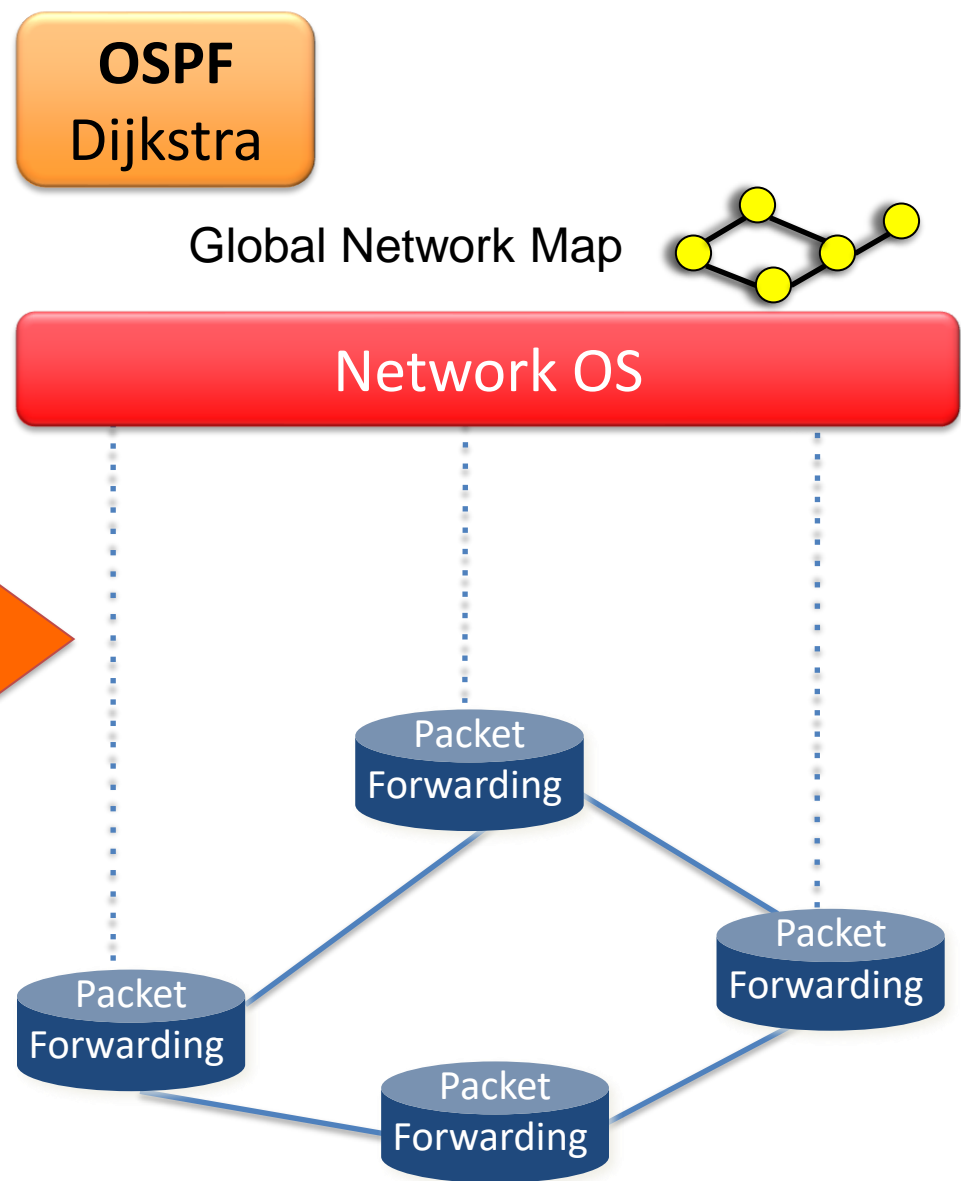
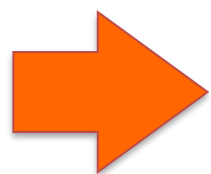
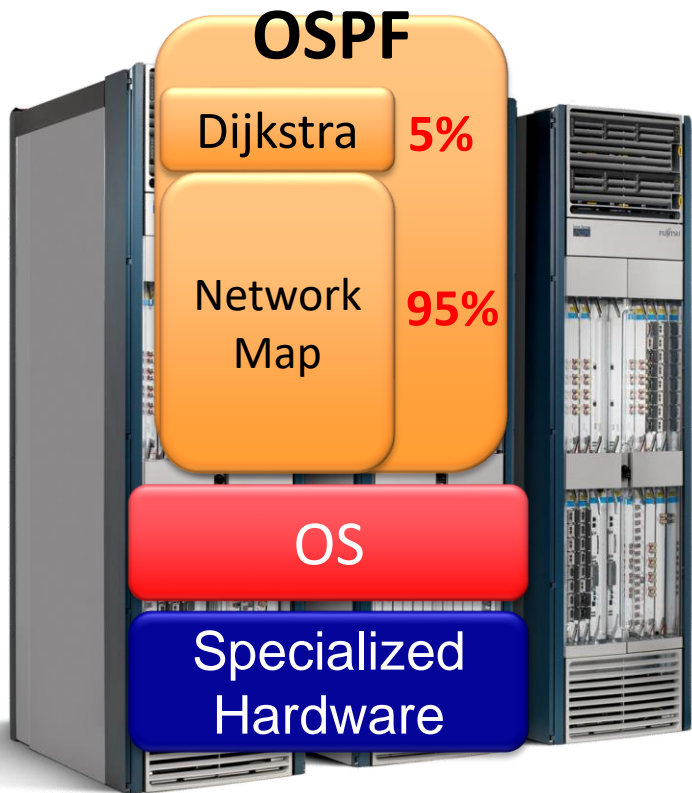
Copyright Notice

Copyright (C) The Internet Society (1998). All Rights Reserved.

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 database describing the Autonomous System's topology. From this







Specialized  
Features

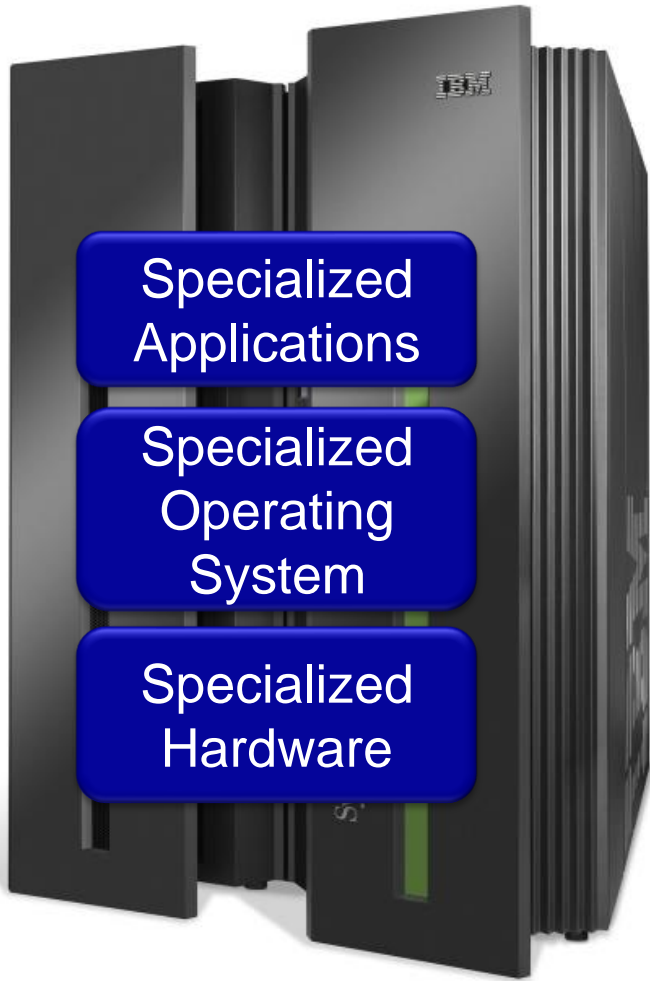
Hundreds of protocols  
6,500 RFCs

Specialized  
Control  
Plane

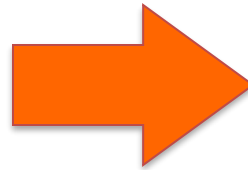
Tens of millions of lines of code.  
Closed, proprietary, outdated.

Specialized  
Hardware

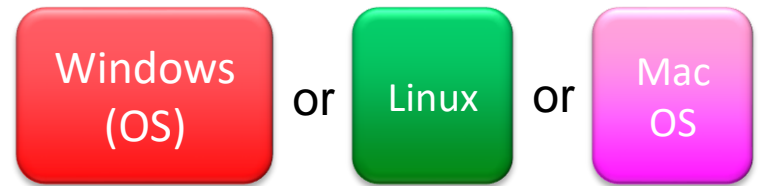
Billions of gates.  
Power hungry and bloated.



Vertically integrated  
Closed, proprietary  
Slow innovation  
Small industry



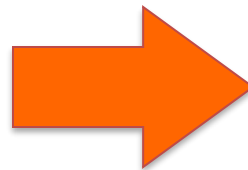
— Open Interface —

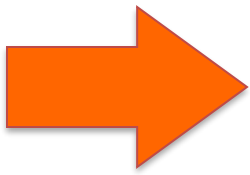


— Open Interface —



Horizontal  
Open interfaces  
Rapid innovation  
Huge industry





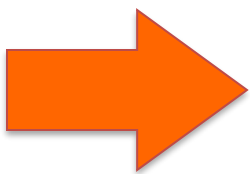
— Open Interface —



— Open Interface —

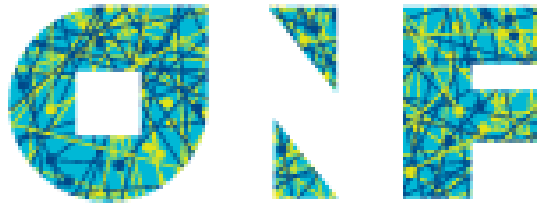


Vertically integrated  
Closed, proprietary  
Slow innovation

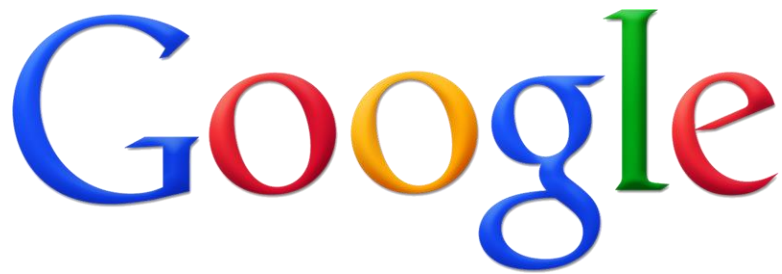


Horizontal  
Open interfaces  
Rapid innovation

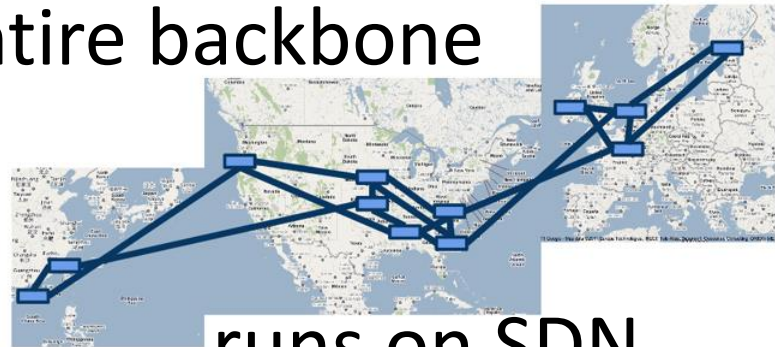
# A Major Trend in Networking



OPEN NETWORKING  
FOUNDATION



Entire backbone



runs on SDN

Bought for **\$1.2 billion**  
(mostly cash)



# An Opportunity to Rethink

How should future networks be

- Designed
- Managed
- Programmed

What are the right abstractions

- Simple
- Powerful
- Reusable

# For next time...

## VL2: A Scalable and Flexible Data Center Network

Albert Greenberg  
Srikanth Kandula  
David A. Maltz

James R. Hamilton  
Changhoon Kim  
Parveen Patel  
Microsoft Research

Navendu Jain  
Parantap Lahiri  
Sudipta Sengupta

### Abstract

To be agile and cost effective, data centers should allow dynamic resource allocation across large server pools. In particular, the data center network should enable any server to be assigned to any service. To meet these goals, we present VL2, a practical network architecture that scales to support huge data centers with uniform high capacity between servers, performance isolation between services, and Ethernet layer-2 semantics. VL2 uses (1) flat addressing to allow service instances to be placed anywhere in the network, (2) Valiant Load Balancing to spread traffic uniformly across network paths, and (3) end-system based address resolution to scale to large server pools, without introducing complexity to the network control plane. VL2's design is driven by detailed measurements of traffic and fault data from a large operational cloud service provider. VL2's implementation leverages proven network technologies, already available at low cost in high-speed hardware implementations, to build a scalable and reliable network architecture. As a result, VL2 networks can be deployed today, and we have built a working prototype. We evaluate the merits of the VL2 design using measurement, analysis, and experiments. Our VL2 prototype shuffles 2.7 TB of data among 75 servers in 395 seconds – sustaining a rate that is 94% of the maximum possible.

**Categories and Subject Descriptors:** C.2.1 [Computer-Communication Network]: Network Architecture and Design

**General Terms:** Design, Performance, Reliability

**Keywords:** Data center network, commoditization

Agility promises improved risk management and cost savings. Without agility, each service must pre-allocate enough servers to meet difficult to predict demand spikes, or risk failure at the brink of success. With agility, the data center operator can meet the fluctuating demands of individual services from a large shared server pool, resulting in higher server utilization and lower costs.

Unfortunately, the designs for today's data center network prevent agility in several ways. First, existing architectures do not provide enough capacity between the servers they interconnect. Conventional architectures rely on tree-like network configurations built from high-cost hardware. Due to the cost of the equipment, the capacity between different branches of the tree is typically oversubscribed by factors of 1:5 or more, with paths through the highest levels of the tree oversubscribed by factors of 1:80 to 1:240. This limits communication between servers to the point that it fragments the server pool — congestion and computation hot-spots are prevalent even when spare capacity is available elsewhere. Second, while data centers host multiple services, the network does little to prevent a traffic flood in one service from affecting the other services around it — when one service experiences a traffic flood, it is common for all those sharing the same network sub-tree to suffer collateral damage. Third, the routing design in conventional networks achieves scale by assigning servers topologically significant IP addresses and dividing servers among VLANs. Such fragmentation of the address space limits the utility of virtual machines, which cannot migrate out of their original VLAN while keeping their original IP addresses. The fragmentation of the address space also prevents the migration of services between server pools.