6.888
Advanced Topics in Networking

Lecture 1: Introduction

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?
A Heap of Header Formats?

HTTP Response Header

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP Status Code:</td>
<td>HTTP/1.1 200 OK</td>
</tr>
<tr>
<td>Date:</td>
<td>Thu, 27 Mar 2008 13:37:17 GMT</td>
</tr>
<tr>
<td>Server:</td>
<td>Apache/2.0.55 (Ubuntu) PHP/5.1.2</td>
</tr>
<tr>
<td>Last-Modified:</td>
<td>Fri, 21 Mar 2008 13:57:30 GMT</td>
</tr>
<tr>
<td>ETag:</td>
<td>&quot;359a4e4-56000-ddf5c680&quot;</td>
</tr>
<tr>
<td>Accept-Ranges:</td>
<td>bytes</td>
</tr>
<tr>
<td>Content-Length:</td>
<td>352256</td>
</tr>
<tr>
<td>Connection:</td>
<td>close</td>
</tr>
<tr>
<td>Content-Type:</td>
<td>application/x-msdos-program</td>
</tr>
</tbody>
</table>
TCP/IP Header Formats in Lego
A Big Bunch of Boxes?
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

We will read 1-2 papers per class
  – Every expected to read the papers in advance
  – Submit a short review of the required reading by midnight the night before class
  – Say something that is not in the paper

Submit reviews here:

Each student will also present one paper
  – Read paper and relevant references
  – 25 minute talk; instructions on website
Projects

Research project of your choice

Work alone or in groups of two

Project ideas

– Explore your own
– I will also suggest some ideas
– Can involve system implementation, algorithms, theory, ...
– Can be related to your research (come talk to me)
Project Timeline

Proposal (1 page) | March 1
Midterm Review | April 1
Final Presentation | May 10 (tentative)
Final Report (6-8 pages) | May 18 (tentative)
Grading

Class participation 15%

Paper Reviews 15%

Paper Presentation 20%

Project 50%
Other Logistics

Time/Location
  – Tue/Thr 2:30-4pm, room 36-112

Mohammad’s office hours
  – Tuesday 5-6pm at 32-G920

Course webpage

Piazza forum, sign up here:
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

“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

Google
Microsoft
Facebook
These things are **really** big

- 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

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

- Computers are useless. They can only give you answers.

Deadline = 250ms
Deadline = 50ms
Deadline = 10ms

1. Art is a lie...
2. The chief...
3. ...

1. Art is a lie...
2. The chief...
3. ...

“Everything you can imagine is real.”
“Bad artists copy. Good artists steal.”
“It is your work in life that is the ultimate education.”
“The chief enemy of creativity is good sense.”
“Inspiration does exist, but it must find you working.”
“I'd like to live as a poor man with lots of money.”
“Art is a lie that makes us realize the truth.”
“Computers are useless. They can only give you answers.”
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
Software Defined Networking
Software Defined Network (SDN)

Control Plane

Global Network Map

Control Program

Packet Forwarding

Control
Forwarding

Control
Forwarding

Control
Forwarding

Control
Forwarding
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)
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
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: 44
Obsoletes: 2178
Category: Standards Track

50,000 lines of code
J. Moy
Ascend Communications, Inc.
April 1998

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

Hundreds of protocols
6,500 RFCs

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

Billions of gates.
Power hungry and bloated.
Vertically integrated
Closed, proprietary
Slow innovation
Small industry

Horizontally integrated
Open interfaces
Rapid innovation
Huge industry
Vertically integrated
Closed, proprietary
Slow innovation

Horizontal
Open interfaces
Rapid innovation
A Major Trend in Networking

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
VL2: A Scalable and Flexible Data Center Network

Albert Greenberg
Srikant Kandula
David A. Maltz

James R. Hamilton
Changhoon Kim
Parveen Patel

Navendu Jain
Parantap Lahiri
Sudipta Sengupta

Microsoft Research

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) Vlant 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 in 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 measurements, analysis, and experiments. Our VL2 prototype sustains a rate that is 94% of the maximum possible.

Categories and Subject Descriptors: C.2.2 [Computer-Communication Network]: Network Architecture and Design

General Terms: Design, Performance, Reliability

Keywords: Data center network, commodization

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 1:20 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 their original VLAN while keeping the same IP, and makes it difficult for network administrators to monitor and manage network traffic.