Lecture 15: Datacenter TCP

CSE 222A: Computer Communication Networks
Arjun Roy

Thanks: Mohammad Alizadeh
Lecture 15 Overview

- Datacenter workload discussion
- DCTCP Overview
Roadmap

- What’s really going on in datacenters?
  - Interviews with developers and operators
  - Analysis of applications
  - Switches: shallow-buffered vs deep-buffered
  - Measurements

- A systematic study of transport in Microsoft’s DCs
  - Identify *impairments*
  - Identify *requirements*
Case Study: Microsoft Bing

- Measurements from 6000 server production cluster

- Instrumentation passively collects logs
  - Application-level
  - Socket-level
  - Selected packet-level

- More than 150TB of compressed data over a month
Partition/Aggregate Application Structure

- Time is money
  - Strict deadlines (SLAs)
- Missed deadline
  - Lower quality result

Picasso

"Everything you can imagine is real."
"Bad artists copy. Good artists steal."
"It is your work in life that is the ultimate seduction."
"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."

TLA

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

MLA

1. Deadline = 250ms
2. Deadline = 50ms
3. Deadline = 10ms

Worker Nodes
Partition/Aggregate

- The foundation for many large-scale web applications.
  - Web search, Social network composition, Ad selection, etc.

- Example: Facebook

Partition/Aggregate ~ Multiget
- Aggregators: Web Servers
- Workers: Memcached Servers
Workloads

- Partition/Aggregate (Query)
- Short messages [50KB-1MB] (Coordination, Control state)
- Large flows [1MB-50MB] (Data update)
Impairments

- Incast
- Queue Buildup
- Buffer Pressure
Incast

Worker 1

Worker 2

Worker 3

Worker 4

• Synchronized mice collide.
  ➢ Caused by Partition/Aggregate.

RTO_{min} = 300 ms

TCP timeout

CSE 222A – Lecture 15: Datacenter TCP
Incast in Bing

Requests are jittered over 10ms window.

Jittering switched off around 8:30 am.

Jittering trades off median against high percentiles.

99.9\textsuperscript{th} percentile is being tracked.

MLA Query Completion Time (ms)

Saturday, December 19, 2009
Queue Buildup

- Big flows buildup queues.
  - Increased latency for short flows.

- Measurements in Bing cluster
  - For 90% packets: RTT < 1ms
  - For 10% packets: 1ms < RTT < 15ms
1. High Burst Tolerance
   – Incast due to Partition/Aggregate is common.

2. Low Latency
   – Short flows, queries

3. High Throughput
   – Continuous data updates, large file transfers

The challenge is to achieve these three together.
Tension Between Requirements

High Throughput
High Burst Tolerance
Low Latency

Deep Buffers:
- Queuing Delays Increase Latency

Shallow Buffers:
- Bad for Bursts & Throughput

Objective:
Low Queue Occupancy & High Throughput

Reduced RTO_{min} (SIGCOMM ‘09)
- Doesn’t Help Latency

AQM – RED:
- Avg Queue Not Fast Enough for Incast
ECN Control Loop

Sender 1

ECN Mark (1 bit)

Sender 2

Receiver
Small Queues & TCP Throughput:

- Bandwidth-delay product rule of thumb:
  - A single flow needs $C \times RTT$ buffers for 100% Throughput.
Small Queues & TCP Throughput:

- Bandwidth-delay product rule of thumb:
  - A single flow needs $C \times RTT$ buffers for 100% Throughput.

- Appenzeller rule of thumb (SIGCOMM '04):
  - Large # of flows: $C \times RTT/\sqrt{N}$ is enough.
Small Queues & TCP Throughput:

- **Bandwidth-delay product rule of thumb:**
  - A single flow needs $C \times RTT$ buffers for **100% Throughput**.

- **Appenzeller rule of thumb (SIGCOMM ‘04):**
  - Large # of flows: $C \times RTT/\sqrt{N}$ is enough.

- Can’t rely on stat-mux benefit in the DC.
  - Measurements show **typically 1-2 big flows** at each server, at most 4.
Small Queues & TCP Throughput:

- Bandwidth-delay product rule of thumb:
  - A single flow needs $C \times RTT$ buffers for 100% Throughput.

- Appenzeller rule of thumb (SIGCOMM ‘04):
  - Large # of flows: $C \times RTT/\sqrt{N}$ is enough.

- Can’t rely on stat-mux benefit in the DC.
  - Measurements show typically 1-2 big flows at each server, at most 4.

Real Rule of Thumb:
Low Variance in Sending Rate $\rightarrow$ Small Buffers Suffice
Two Key Ideas

1. React in proportion to the **extent** of congestion, not its **presence**.
   ✓ Reduces **variance** in sending rates, lowering queuing requirements.

<table>
<thead>
<tr>
<th>ECN Marks</th>
<th>TCP</th>
<th>DCTCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0 1 1 1 1 0 1 1 1</td>
<td>Cut window by 50%</td>
<td>Cut window by 40%</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0 0 1</td>
<td>Cut window by 50%</td>
<td>Cut window by 5%</td>
</tr>
</tbody>
</table>

2. Mark based on **instantaneous** queue length.
   ✓ Fast feedback to better deal with bursts.
Data Center TCP Algorithm

Switch side:
- Mark packets when Queue Length > K.

Sender side:
- Maintain running average of fraction of packets marked ($\alpha$).

In each RTT:

$$F = \frac{\# \text{ of marked ACKs}}{\text{Total} \# \text{ of ACKs}}$$

$$\alpha \leftarrow (1 - g)\alpha + gF$$

- Adaptive window decreases:
  - Note: decrease factor between 1 and 2.

$$Cwnd \leftarrow (1 - \frac{\alpha}{2})Cwnd$$
DCTCP in Action

Setup: Win 7, Broadcom 1Gbps Switch
Scenario: 2 long-lived flows, K = 30KB
Why it Works

1. High Burst Tolerance
   - Large buffer headroom → bursts fit.
   - Aggressive marking → sources react before packets are dropped.

2. Low Latency
   - Small buffer occupancies → low queuing delay.

3. High Throughput
   - ECN averaging → smooth rate adjustments, low variance.
Analysis

- How low can DCTCP maintain queues without loss of throughput?
- How do we set the DCTCP parameters?

> Need to quantify queue size oscillations (Stability).

Window Size

\[ W^* + 1 \]

\[ W^* \]

\[ (W^* + 1)(1 - \alpha/2) \]

Time
Analysis

- How low can DCTCP maintain queues without loss of throughput?
- How do we set the DCTCP parameters?

➢ Need to quantify queue size oscillations (Stability).

Window Size

\[ W^* + 1 \]

\[ W^* \]

\[ (W^*+1)(1-\alpha/2) \]

Packets sent in this RTT are marked.

Time
Analysis

- How low can DCTCP maintain queues without loss of throughput?
- How do we set the DCTCP parameters?

> Need to quantify queue size oscillations (Stability).

\[ K > \frac{1}{7} C \times RTT \]

85% Less Buffer than TCP
Emulated Traffic Experiment

- Emulate traffic within 1 Rack of Bing cluster
  - 45 1G servers, 10G server for external traffic

- Generate query, and background traffic
  - Flow sizes and arrival times follow distributions seen in Bing

- Metric:
  - Flow completion time for queries and background flows.

\[ RTO_{\text{min}} = 10 \text{ ms} \text{ for both TCP & DCTCP.} \]
Results

- Low latency for short flows.
- High throughput for long flows.
- High burst tolerance for query flows.
10x Traffic Increase

Completion Time (ms)

- DCTCP/ShallowBuf
- TCP/ShallowBuf
- TCP-RED/ShallowBuf
- TCP/DeepBuf

Short messages

Query
Queue length vs. flow count

Cumulative Fraction

Queue Length (packets)

Figure 13: Queue length CDF (1Gbps)
Bandwidth vs. queue limits

Figure 14: Throughput (10Gbps)
Figure 16: Convergence test

(a) DCTCP

(b) TCP
Figure 18: DCTCP performs better than TCP and then converges at 35 senders (log scale on Y axis; 90% confidence intervals for the means are too small to be visible).
Facebook link utilization

(b) Link utilization, 10-minute average
Facebook buffer utilization

(a) Normalized buffer occupancy, 10-microsecond resolution
Facebook buffer utilization

(b) Link utilization, 10-minute average

(a) Normalized buffer occupancy, 10-microsecond resolution
For Next Class…

- Read and review MP-TCP paper

- Keep going on projects!
  - Checkpoint 2 only 1 week away