Lecture 15: Datacenter TCP

CSE 222A: Computer Communication Networks
Alex C. Snoeren

Thanks: Mohammad Alizadeh
Lecture 15 Overview

- Datacenter workload discussion
- DC-TCP Overview
Datacenter Review

- Large purpose-built DCs
  - Huge investment: R&D, business

- Transport inside the DC
  - TCP rules (99.9% of traffic)

- How’s TCP doing?
TCP in the Data Center

- TCP is challenged to meet demands of apps.
  - Suffers from bursty packet drops, Incast [SIGCOMM ‘09], ...
  - Builds up large queues:
    - Adds significant latency.
    - Wastes precious buffers, esp. bad with shallow-buffered switches.

- Operators work around TCP problems.
  - Ad-hoc, inefficient, often expensive solutions
  - No solid understanding of consequences, tradeoffs
Roadmap

- What’s really going on?
  - 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

Art is a lie...
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)

- Delay-sensitive
- Delay-sensitive
- Throughput-sensitive
Impairments

- Incast
- Queue Buildup
- Buffer Pressure
Incast

Worker 1

Worker 2

Worker 3

Worker 4

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

Aggregator

RTO_{\text{min}} = 300 \text{ 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.

Jitter 99.9\textsuperscript{th} percentile is being tracked.

MLA Query Completion Time (ms)

CSE 222A – Lecture 15: Datacenter TCP
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 $RTO_{\text{min}}$ (SIGCOMM ‘09)
- Doesn’t Help Latency

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

CSE 222A – Lecture 15: Datacenter TCP
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 (SIGCOM ‘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 (α).

In each RTT:

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

\[
\alpha \leftarrow (1 - g)\alpha + gF
\]

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

\[
\text{Cwnd} \leftarrow (1 - \frac{\alpha}{2})\text{Cwnd}
\]
DCTCP in Action

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

Queue Length (Kbytes)

Time (seconds)

DCTCP, K=20, 2 flows
TCP, 2 flows

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).

![Diagram showing window size and time with equations: Window Size = W\*+1, W\*, (W\*+1)(1-\alpha/2).]
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

Packets sent in this RTT are marked.
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.

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

Background Flows

- Low latency for short flows.
- High throughput for long flows.
- High burst tolerance for query flows.

Query Flows

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

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

Completion Time (ms)

Short messages vs. Query

CSE 222A – Lecture 15: Datacenter TCP
For Next Class…

- Read and review MP-TCP paper

- Keep going on projects!
  - Checkpoint 2 only 1.5 weeks away

- See at talk by the author of today’s paper:
  - Wednesday 3/19 at 11am EBU3b 1201