CompSci 514: Computer Networks
Lecture 15 Practical Datacenter Networks

Xiaowei Yang
Overview

• Wrap up DCTCP analysis

• Today
  – Google’s datacenter networks
    • Topology, routing, and management
  – Inside Facebook’s datacenter networks
    • Services and traffic patterns
The DCTCP Algorithm
Review: The TCP/ECN Control Loop

ECN = Explicit Congestion Notification

Sender 1

ECN Mark (1 bit)

Sender 2

Receiver
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.
Small Queues & TCP Throughput: The Buffer Sizing Story

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

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

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

➢ Adaptive window decreases:

\[
Cwnd \leftarrow (1 - \frac{\alpha}{2})Cwnd
\]

– Note: decrease factor between 1 and 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).

![Diagram showing window size changes over time with equations for window size: \( W^* + 1 \), \( W^* \), and \( (W^* + 1)(1 - \alpha/2) \).]
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).
Analysis

• \( Q(t) = NW(t) - C \times RTT \)

• The key observation is that with synchronized senders, the queue size exceeds the marking threshold \( K \) for exactly one RTT in each period of the saw-tooth, before the sources receive ECN marks and reduce their window sizes accordingly.

• \( S(W_1, W_2) = (W_2^2 - W_1^2)/2 \)

• Critical window size when ECN marking occurs: \( W^* = (C \times RTT + K)/N \)
• \( \alpha = S(W^*,W^* + 1)/S((W^* + 1)(1 - \alpha/2),W^* + 1) \)

• \( \alpha^2(1 - \alpha/4) = (2W^* + 1)/(W^* + 1)^2 \approx 2/W^* \)
  – Assuming \( W^* \gg 1 \)

• \( \alpha \approx \sqrt{2/W^*} \)

• Single flow oscillation
  – \( D = (W^* + 1) - (W^* + 1)(1 - \alpha/2) \)

\[
A = ND = N(W^* + 1)\alpha/2 \approx \frac{N}{2} \sqrt{2W^*} \\
= \frac{1}{2} \sqrt{2N(C \times RTT + K)}, \quad \text{(8)}
\]

\[
T_C = D = \frac{1}{2} \sqrt{2(C \times RTT + K)/N} \quad \text{(in RTTs).} \quad \text{(9)}
\]

Finally, using (3), we have:

\[
Q_{max} = N(W^* + 1) - C \times RTT = K + N. \quad \text{(10)}
\]
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).

\[ Q_{min} = Q_{max} - A \]
\[ = K + N - \frac{1}{2} \sqrt{2N(C \times RTT + K)}. \]  

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

Minimizing Qmin

85% Less Buffer than TCP
Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google’s Datacenter Network

Arjun Singh, Joon Ong, Amit Agarwal, Glen Anderson, Ashby Armistead, Roy Bannon, Seb Boving, Gaurav Desai, Bob Felderman, Paulie Germano, Anand Kanagala, Jeff Provost, Jason Simmons, Eiichi Tanda, Jim Wanderer, Urs Hölzle, Stephen Stuart, and Amin Vahdat
What’s this paper about

• Experience track

• How Google datacenter evolve over a decade
Key takeaways

- Customized switches built using merchant silicon
- Recursive Clos to scale to a large number of servers
- Centralized control/management
• Bandwidth demands in the datacenter are doubling every 12-15 months, even faster than the wide area Internet.
Top of Rack (ToR) switches serving 40 1G-connected servers were connected via 1G links to four 512 1G port Cluster Routers (CRs) connected with 10G sidelinks. 512*40 ~20K hosts
When a lot of traffic leaves a rack, congestion occurs.

Figure 3: Mix of jobs in an example cluster with 12 blocks of servers (left). Fraction of traffic in each block destined for remote blocks (right).
Evolution over five generations

- **Firehose 1.0**
- **Watchtower**
- **Firehose 1.1**
- **Saturn**
- **Jupiter (1.3P)**

Bisection b/w (bps)

1000T
100T
10T
1T

(years)

- '04
- '05
- '06
- '07
- '08
- '09
- '10
- '11
- '12
- '13

(log scale)
• Use merchant silicon to build non-blocking/high port density switches
• Watchtower: 16*10G silicon

Figure 9: A 128x10G port Watchtower chassis (top left). The internal non-blocking topology over eight linecards (bottom left). Four chassis housed in two racks cabled with fiber (right).
Exercise

- 24*10G silicon
- 12-line cards
- 288 port non-blocking switch
Figure 12: Components of a Saturn fabric. A 24x10G Pluto ToR Switch and a 12-linecard 288x10G Saturn chassis (including logical topology) built from the same switch chip. Four Saturn chassis housed in two racks cabled with fiber (right).
• Dual redundant 10G links for fast failover
• Centauri as ToR
• Four Centauris made up a Middle Block (MB)
• Each ToR connects to eight MBs.
• Six Centauris in a spine plane block
Figure 14: Jupiter Middle blocks housed in racks.

- Four MBs per rack
- Two spine blocks per rack
Figure 10: Reducing deployment complexity by bundling cables. Stages 1, 2 and 3 in the fabric are labeled S1, S2 and S3, respectively.
Connecting to other networks

- Connect externally via border routers
- Massive external burst b/w
  - Enables (e.g.) cross-cluster MapReduce
- No need to keep old “cluster routers”
Summary

- Customized switches built using merchant silicon
- Recursive Clos to scale to a large number of servers
Inside the Social Network’s (Datacenter) Network

Arjun Roy, Hongyi Zeng†, Jasmeet Bagga†, George Porter, and Alex C. Snoeren
Motivation

• Measurement can help make design decisions
  – Traffic pattern determines the optimal network topology
  – Flow size distribution helps with traffic engineering
  – Packet size helps with SDN control
<table>
<thead>
<tr>
<th>Finding</th>
<th>Previously published data</th>
<th>Potential impacts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traffic is neither rack local nor all-to-all: low utilization (§4)</td>
<td>50–80% of traffic is rack local [12, 17]</td>
<td>Datacenter fabrics [4, 36, 21]</td>
</tr>
<tr>
<td>Demand is wide-spread, uniform, and stable, with rapidly changing, internally bursty heavy hitters (§5)</td>
<td>Demand is frequently concentrated and bursty [12, 13, 14]</td>
<td>Traffic engineering [5, 14, 25, 39]</td>
</tr>
<tr>
<td>Small packets (outside of Hadoop), continuous arrivals; many concurrent flows (§6)</td>
<td>Bimodal ACK/MTU packet size, on/off behavior [12]; &lt;5 concurrent large flows [8]</td>
<td>SDN controllers [1, 22, 28, 32, 34]; Circuit/hybrid switching [7, 20, 30, 39]</td>
</tr>
</tbody>
</table>

Table 1: Each of our major findings differs from previously published characterizations of datacenter traffic. Many systems incorporate one or more of the previously published features as design assumptions.
Service level architecture of FB

- Servers are organized into clusters
- Clusters may not fit into one rack
Measurement methodology

Figure 3: Fbflow architecture
Figure 4: Per-second traffic locality by system type over a two-minute span: Hadoop (top left), Web server (top right), cache follower (bottom left) and leader (bottom right) (Note the differing y axes)
Figure 6: Flow size distribution, broken down by location of destination

(a) Web servers
(b) Cache follower
(c) Hadoop

Figure 7: Flow duration distribution, broken down by location of destination

(a) Web servers
(b) Cache follower
(c) Hadoop
Figure 12: Packet size distribution
Summary

• Traffic is neither rack-local nor all-to-all; locality depends upon the service but is stable across time periods from seconds to days.

• Many flows are long-lived but not very heavy.

• Packets are small.
Today

• Wrap up DCTCP analysis

• Today
  – Google’s datacenter networks
    • Topology, routing, and management
  – Inside Facebook’s datacenter networks
    • Services and traffic patterns