# Recitation 11: DCTCP

MIT - 6.033

Spring 2021

Henry Corrigan-Gibbs

# Plan

* Feedback summary
* Recitation Qs
* Background on DCs
* Queue game
* DCTCP

## Logistics

* DP Prep report assignment at due TODAY 5pm!
* Participation check-in #2 out Saturday
  ↳ Explain how works
* Midterm on Tuesday, April 6
  ↳ example exams posted

# Feedback Summary

* Most people like the pace
* Request for course notes — now online
* Midterm
  ↳ See past midterms on course website
* Go over pre-rectation Qs in class
  ↳ Volunteers for next time? q3
* Breakout rooms
  ↳ pros & cons
* Awkward pauses waiting for questions
* Kahoot

# Recitation Qs

1. What is the goal of DCTCP?
   - ↳ Everyone type in chat, then send at the same time. (Idea from feedback form.)
   - * Better net utilization in DC

2. How does DCTCP differ from TCP?
   - * Management of buffers

3. Why does DCTCP differ from TCP?
   - * Different setting
     - ↳ Control endpoints
     - ↳ Very low RTT

# Why is this a great paper?

- Simple, important (?) problem
- Elegant solution
  - ↳ No new hardware
  - ↳ Simple changes to endpoints
- Works in practice (e.g Cisco)

# Types of Flows in Data Center

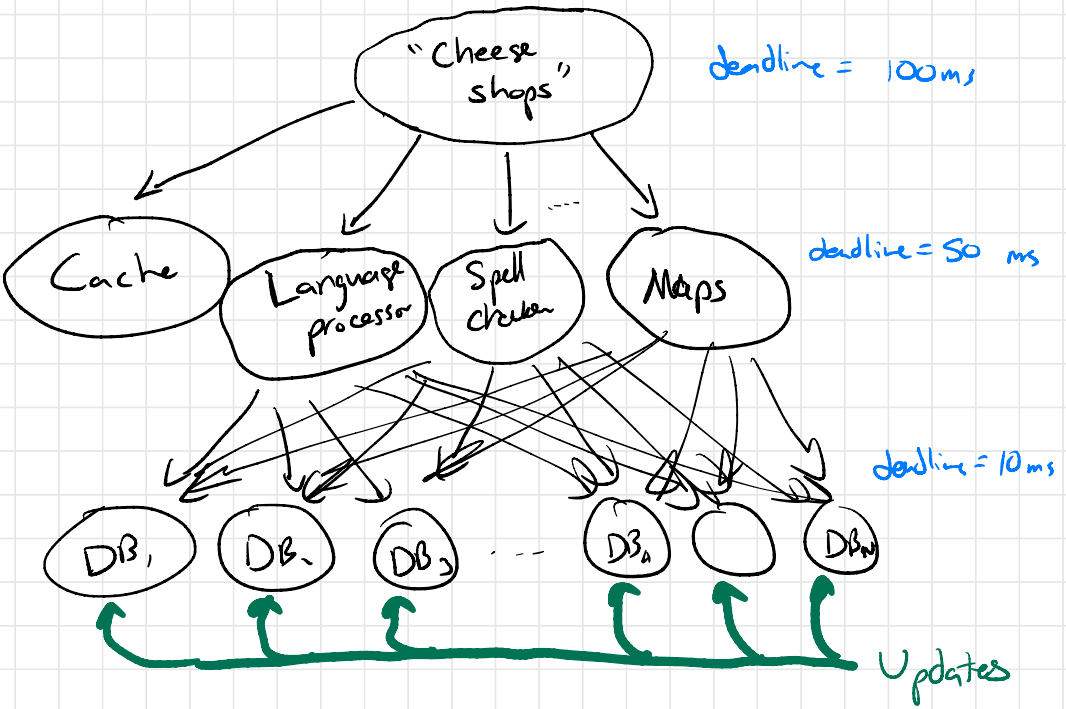
- Query traffic ~2KB
- Short message traffic ~1 MB

} Small, latency sensitive

- Background traffic ~50 MB

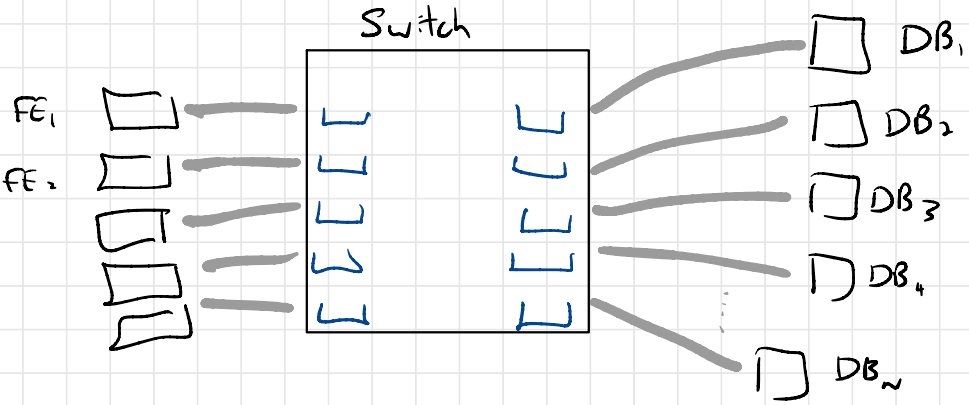
} big throughput is important

E.g. Search



# Queues

## Why do they exist?



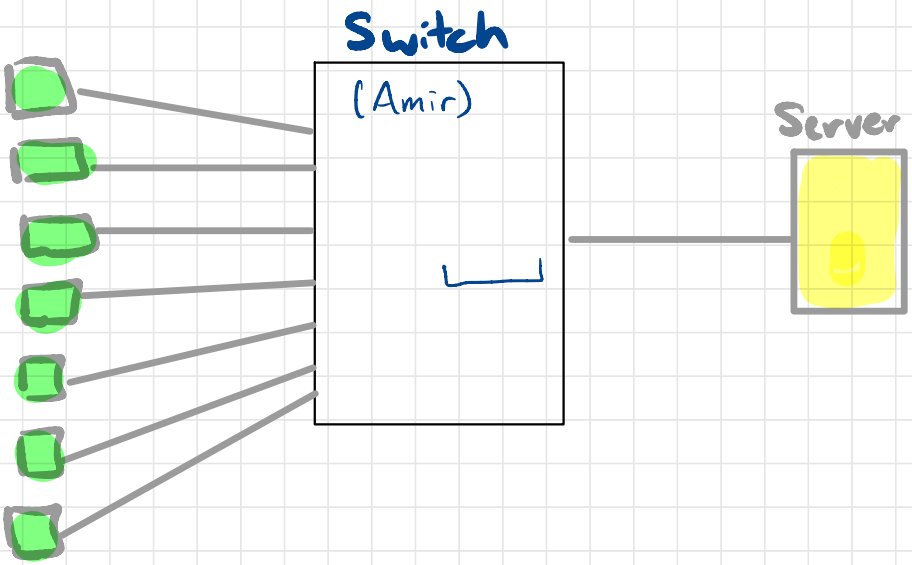

→ What happens when queue fills up?

↳ Dropped packets = worse search results (connection times out)

→ Even before queue fills up, can be problematic

↳ Latency grows!

# Queue Game




- Students: Type your initial in chat to simulate sending a packet
- Amir: Type $␣ to indicate processing and forwarding packet.

Trial run: One sender, every 3-5 seconds

↳ Show what this looks like in queue

# For real now...

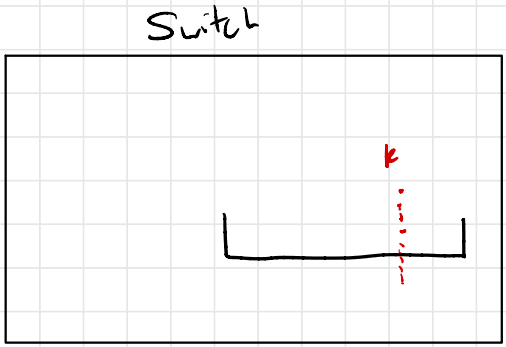
- Two flows, both send every 1-2 seconds

- Everyone sends at same time
  - ↳ lots of packets drop ("incast")
- One long flow, sending as quickly as possible
  - ↳ Some packets drop but throughput is good

- One long flow, one short flow
  - ↳ Queue length grows ⇒ latency!
    - "bufferbloat"

→ What does DCTCP do to fix this problem?

# DCTCP




## Switch

As soon as queue grows beyond threshold, K (small!)
switch sets flag on each packet
↳ ECN

Normal TCP: Wait until queue is full

## Sender

Uses ECN flags to estimate whether queue size is > K

↳ If so, gradually back off (adjust window size)

↳ If not, continue as normal

Normal TCP: Cut window in half

# Why would this not work on Internet?

- Practical: Need to modify both ends of the connection.
- Convergence time depends on RTT
  - ↳ Small in DC (0.1 ms)
  - ↳ Big in Internet (50 ms)

Time for new flow to get its fair share of bandwidth.

3-4× slower than TCP

- Less clarity about types of flows (?)
- Feedback is too slow... by the time sender gets CN, queue may be empty.