Plan

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

Logistics

* DP prelim report due TODAY 5pm.
* Participation check-in #2 out this weekend.
* Midterm on Tuesday, April 6
  example exams online
1. What is the goal of DCTCP?
   - Increase perf for DCs (latency, ...)
   - No new hardware

2. How does DCTCP differ from TCP?
   - ECN — early congestion feedback

3. Why does DCTCP differ from TCP?
   - Characteristics of DC traffic differ from non-traffic
What makes a great research result? (Spielman)

1. Beautiful theory.
2. Works in practice.
3. Solves a problem that people care about.
Types of Flows in DC

1. Query flows ~2kB
2. Bg. traffic ~50 MB
3. Short message ~1 MB

- throughput
- latency sensitive
e.g. Search

F.G.

"Cheese shops"

Cache

Language processor

Spell check

Maps

Deadline = 100 ms

50 ms

10 ms

Update
Two worries

1. Dropped packets
2.
1. One short flow

2. Many short flows same time “Incast”

3. Queue Delay
What does DCTCP do to fix?

**Switch**

- Explicit congestion notification.
  - Set bit on Packet
  - Difference w/ prior act:
    - Switch sets bit early

**Sender**

- Start getting ECN → slow down gradually
  - Normal TCP → really slows down → sharply
Why doesn't this work on Internet?

- Need to modify both ends and switch?

- Convergence time depends on RTT

- Feedback is too slow
Plan

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

Logistics

* DP prelim report due TODAY 5pm.
* Participation check-in #2 out this weekend.
* Midterm on Tuesday, April 6
  Exam exam online
DCTCP

1. What is the goal of DCTCP?
   - Improve on TCP in DCs: low latency, high throughput

2. How does DCTCP differ from TCP?
   - More clever use of explicit congestion notification (ECN)

3. Why does DCTCP differ from TCP?
   - Take advantage of unique properties of DC
     - 0.1 ms RTT
     - 500 ms RTT
What makes a great research result? (Spielman)

1. Beautiful theory
2. Works in practice
3. Solves a problem that people care about.
Types of Flows

1. Short/low-latency "Query" - 2 KB
2. Long background flows - (~100 MB)
3. Short msg traffic - (~1 MB)
E.g. Search

Front end

"Cheese Shops"

Cache

Language processing

Spell checker

Maps

DB_1

DB_2

Updates

\[ \text{deadline} = 100\,\text{ms} \quad \text{"Incast"} \]

\[ = 50\,\text{ms} \]

\[ \text{deadline} = 10\,\text{ns} \]
Queues

Front-end servers

DB

Servers
Queue Game

1. Drop packet
2. Queuing delay

Switch (Amir)

DB

us

=
DCTCP

**Switch**
- Early notification of congestion.
  - Normal: Wait until buffer full

**Sender**
- Keeps estimate of queue length
  - Adjust tx rate accordingly

How is this different from normal TCP?
- TCP: Cut window size in half
- Quantitative approach
Why doesn't this work on Internet?

- Scale: Convergence time = time before every endpoint tx at "right" rate
  - Depends on RTT
    - DC 0.1 ms
    - Net 50 ms

- Deployment

- Less structure