CS434/534: Topics in Network Systems

Inter-datacenter Transport Scheduling: Software-Defined WAN (SDWAN)

Yang (Richard) Yang
Computer Science Department
Yale University
208A Watson
Email: yry@cs.yale.edu

http://zoo.cs.yale.edu/classes/cs434/
Road Map

- **Cloud data center (CDC) applications/services**
  - Fine-grained dataflow programming (e.g., Web apps)
  - Coarse-grained dataflow (e.g., data analytics)
  - Distributed machine learning using parameter server

- **DC scheduling**
  - DC cluster resource scheduling
  - DC transport scheduling
    - Overview
    - DCTCP
    - Fastpass
    - WAN scheduling
    - CoFlow scheduling
Instructor office hours

- Tuesday: 1:30-2:30
- Thursday: 3:00-4:00
- Fridays: 1:30-2:30 pm
Projects

- Milestones (exactly 4 weeks left)
  - 4/17 (T+1 week): Finished reading major related work; a google doc listing related papers (at least 4 papers)
  - 4/24 (T+2 weeks): Finished architecture design (slides/write up of architecture, including all key components)
  - 5/1 (T+3 weeks): Initial, preliminary evaluations (slides/write up, about experiment/analysis setup)
  - 5/8 (T+4 weeks) 5:30 pm, final report due
Current on-going student projects

1. Automatic, High-level SDN Compiler using Flow Algebra
2. SDN Update Algebra with Resource, Ordering Constraints
3. Multi Control Plane Switching using Distributed Verification in MANET using OLSv2
4. Toward High-Level Programming for 5G using Integrated SDN/NFV
5. Design and Implementation of Distributed, Consistent, Fast Data Store
6. Distributed Robots as Mobile Datacenters for Shared Learning
7. Scaling Automatic, Consistent Web App Programming using Noria
8. Distributed, Federated Learning through Autoencoder
9. Scaling Distributed Machine Learning for Geo-Distributed Data Analytics
10. Scaling Fastpass to PB using Partition and Aggregation
11. Distributed, Encrypted Data Store for Global, Community Data
Recap: DC Transport Challenges

- **Challenging setting:**
  - Mixed elephants and mice traffic with potential high burst traffic

- **Challenging requirement:**
  - Achieving high throughput for elephants and low latency for mice, despite potential high burst traffic
Recap: DC TCP

- Problem of TCP: large, full buffer to absorb drastic rate cut to achieve high tput, but result in long latency and bad tolerance of burst traffic

- Basic ideas
  - Switch using ECN: Mark packets when Queue Length > K (how big should K be?).
  - Host: Maintain running average of fraction of packets marked (α) and adjust using α

  \[
  F = \frac{\text{# of marked ACKs}}{\text{Total # of ACKs}} \quad \Rightarrow \quad \alpha \leftarrow (1-g)\alpha + gF
  \]

  \[
  W \leftarrow (1 - \frac{\alpha}{2})W
  \]
Recap: Fastpass

- **Goals:** low latency, high tput, multiple objectives using **centralized** scheduling

- **Key technical subproblems to solve**
  - rate allocation (how?)
    - scaling rate allocation (multi-core, pipelining)
  - path selection (how?)
    - scaling path selection (efficient matching alg using Euler paths)
Recap: Solution Space Covered So Far

Flow: Transfer of data from a source to a destination

Individual, independent flows do not reflect communication behaviors common in data-parallel applications