Credit-Scheduled Delay-Bounded Congestion Control for Datacenters

Inho Cho, Keon Jang, Dongsu Han
Presented by: Leonid Bloch

Software Defined Networking Seminar, May 18, 2018
Scope

- Introducing ExpressPass
  - Motivation
  - Overview
- Design
  - Design Principles
  - Optimizations
- Limitations
- Related Work
- Personal conclusions
- Paper remarks
- Discussion
Motivation

- The datacenter networks become faster
  - The paper mentions 10/40 Gbps with 100 Gbps on the horizon, but currently the horizon is beyond 400 Gbps.
- Faster networks, many connections → risk of congestion.
  - ...And the switches have limited buffer size!
- With some applications data loss is not acceptable (e.g. RDMA), with others – causes significant slowdown.
- Current congestion control methods cause latency and massive buffer use at switches.
Overview

Introducing ExpressPass: optimized credit-based congestion control.

- Prevents congestion *before* it occurs
- Avoiding data loss
- Can run on commodity switches
- Features (some) optimizations for real-life scenarios
A *seemingly* very simple concept:

- Receivers send credit packets
- The network limits the credit packets’ traffic as needed
- Senders send a data packet per each credit packet
Design

Basic rationale:
- One does not need to be “gentle” with credit packets
- “Natural” adaptation to the network conditions

Real life problems with the naïve design:
- Waste of bandwidth
- Does not guarantee fairness
- Credit and data packets may travel on different paths

These are solved by:
- Limiting the sending of credits as much as possible
  - “As much as possible” is not actually that much.
- Symmetric routing via hashing
- **Regularized feedback control** (senders report back!)
Limitations

- Does not account “other” traffic (e.g. ARP packets, link layer control), may interfere with it
- Does not allow intentional traffic prioritization (currently)
- A lot of redundant credit packets are sent (bandwidth waste)
Limitations

- Does not account “other” traffic (e.g. ARP packets, link layer control), may interfere with it
- Does not allow intentional traffic prioritization (currently)
- A lot of redundant credit packets are sent (bandwidth waste)
Similar approach was implemented previously in ultra-high-speed specialized networks before.
- On-chip networks
- Infiniband
- PCIe
- AMD Hypertransport, Intel QuickPath

Many different congestion control methods are employed currently in datacenters (based on different principles, including traffic prediction)

Hardware assisted congestion control (would be nice to see some dedicated hardware here as well)
Personal conclusions

- Simple and interesting idea
- **Implementable on commodity HW**
- It would be interesting to see some auxiliary network employed to address some challenges.
- Not addressing some real-life problems (prev. slides)
  - Suggests some solutions, but feels unfinished
- The testing description is unclear and quite artificial
Paper remarks

- Well presented
- Mentioning the shortcomings openly
- Written in a “pro jargon”
Let’s talk?