# RoCC: Robust Congestion Control for RDMA

Parvin Taheri et. Al.
Cisco Systems and Purdue University

ACM CoNEXT 2020

# Outline

- Introduction
- Design
- Evaluation
- Conclusion

# Introduction

- Remote Direct Memory Access (**RDMA**) provides ultra-low latency (~1μs) and high throughput (40/100Gbps) with little CPU overhead

- Recently, RDMA has been deployed in datacenters at scale with RDMA over Converged Ethernet (**RoCE**) **v2**

# Congestion control's goal

- Congestion control has clear goal
  : **Reducing Flow Completion Time** (FCT)
    - Low latency for small flows (mice)
    - High throughput for large flows (elephants)

- Datacenter congestion control has more goal
  : **Minimizing Priority Flow Control** (PFC) activation
    - PFC increases FCT
    - PFC causes routing deadlocks

# Requirements of RoCC

- Fairness
  - Flows on a congested link must **equally share** the link bandwidth or **max-min fairness**

- Rapid convergence
  - **React quickly** to increasing and decreasing congestion levels

# Requirements of RoCC

- Stability
  - Congestion control has to be **stable** regardless of the number of flows creating congestion

- Efficiency
  - Congestion control should not be performed at the expense of link **under-utilization** resulting in low throughput

# Design

# Design consideration

- Categorization of congestion controls (based on entity)

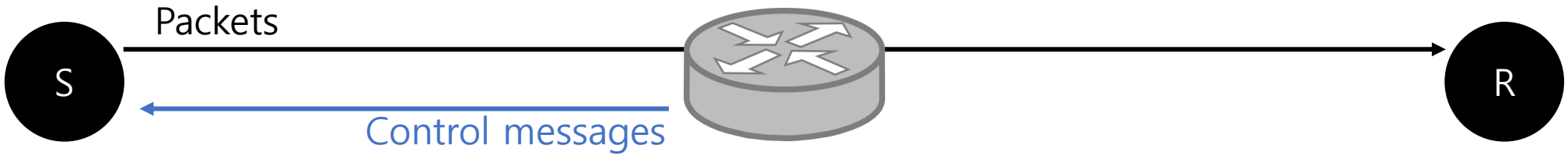
: Source-driven or switch-driven

- Source-driven
  - Source paces packets based on congestion signals

- **Switch-driven**
  - Switch computes pacing information and sends it to the source

# RoCC overview

- RoCC is **switch-driven** congestion control



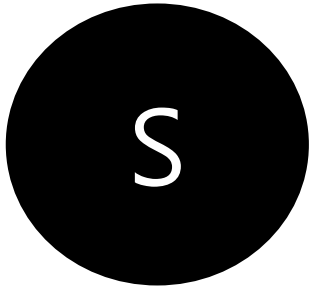

Receive signals from switch
Pace rate of packets

Compute rate for flows
Send signals to sources

Packet sink

# Key components

- 1. Congestion Point, **CP**
  - **Rate calculator**
  - Feedback message generator
  - Flow table




- 2. Reaction Point, **RP**
  - **Rate controller**

# Components on CP

- **Rate calculator**
  - Periodically reads current **queue size**
  - Calculates fair rate (FAIR)
- Feedback message generator
  - Creates the control message
  - Sends it to sources
- Flow table
  - Keeps track of flows

# Rate calculation

- Multiplicative decrease
  - Reduce rate **exponentially** based on **queue growth** threshold
  - Reduce rate to **minimum** based on **queue size** threshold

- Proportional integral
  - Rate control with queue size (Q) as input
  - Stable queue → arrival rate = drain rate → fair rate

$$F \leftarrow F - \alpha \times (Q_{\text{cur}} - Q_{\text{ref}}) - \beta \times (Q_{\text{cur}} - Q_{\text{old}})$$

# Feedback and flow table

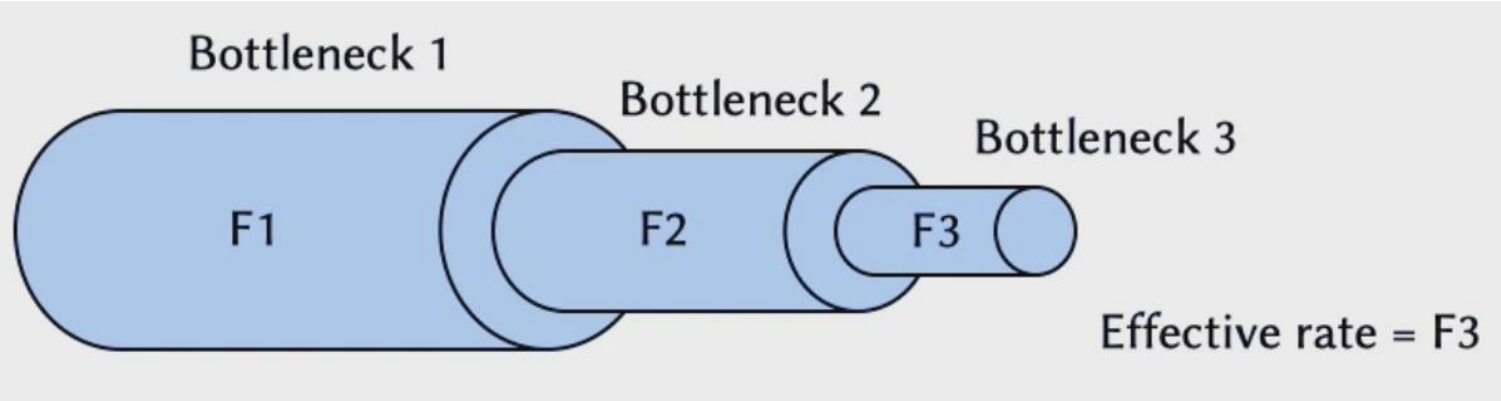
- Feedback message includes
  - Fair rate value
  - Flow information (identifier)

- Flow table keeps track of the recipients of the feedback messages
  - Maintaining a table of the flows currently in the queue

# Components on RP

- Fast recovery
  - Exponential rate self-rise in the absence of rate messages

- Multi-feedback handler
  - Use minimum available rate along path of flow



# Implementation

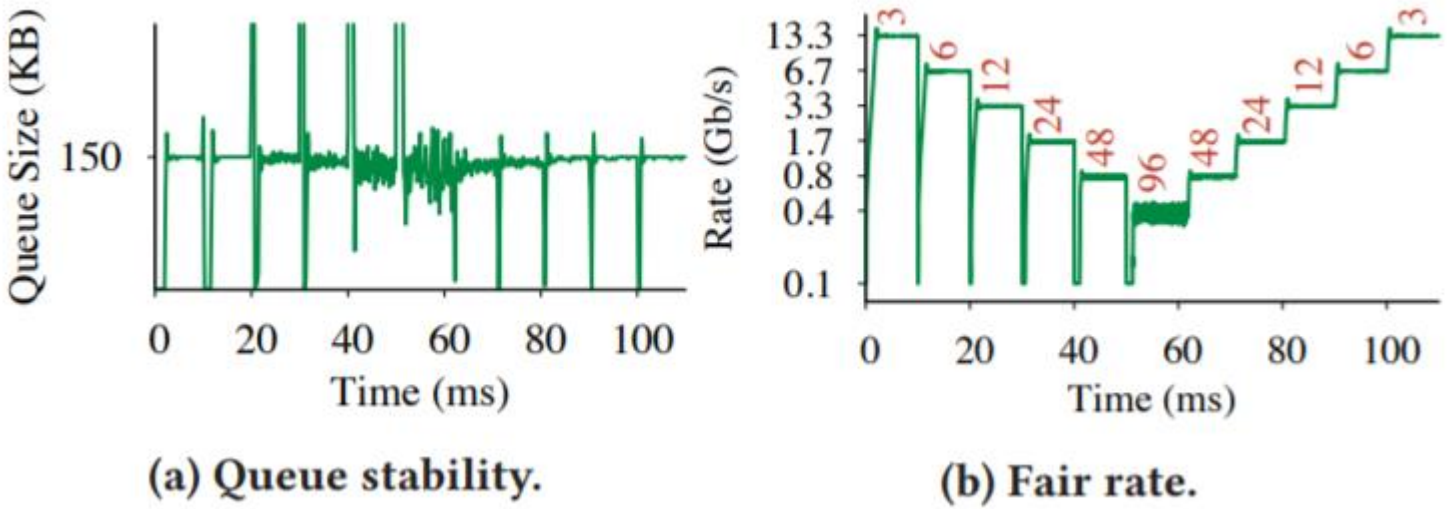
- RoCC can be implemented by P4
  - Programmable switch
- They use “v1model” for simulation
  - P4 simulation model
- It seems that RoCC is hard to be deployed without P4

# Evaluation

# Experiment environments

- 1. Small scale **simulations**
- 2. Evaluation with **DPDK** to confirm the properties of RoCC on a real network and validate our simulations
- 3. **Larger scale** evaluation using a **simulation**
  - Setup resembling a real datacenter network in terms of topology, number of nodes, and traffic patterns

# Simulation results

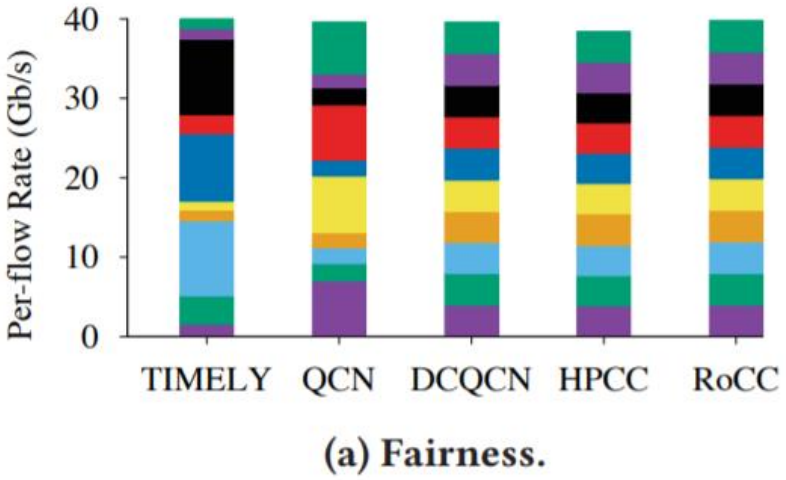



(a) Queue stability.

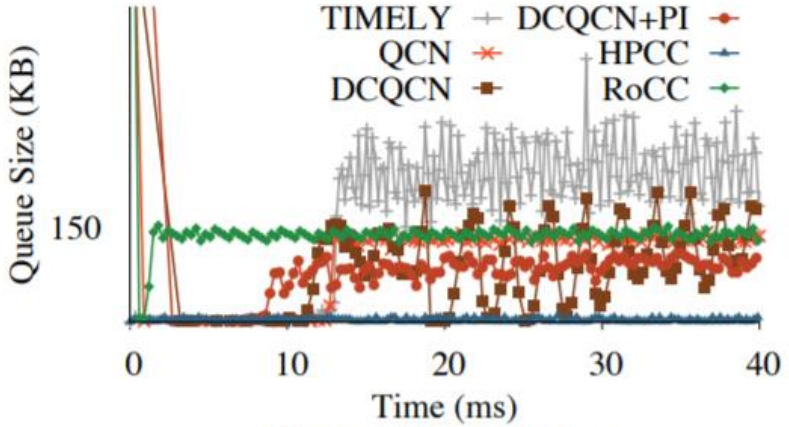
(b) Fair rate.

- Simulation results show fairness, stability, and convergence
  - n flows share bandwidth of congested link (fair)
  - Queue size and rate are not fluctuated (stable)
  - Queue size and rate quickly enter to stable state (conv)

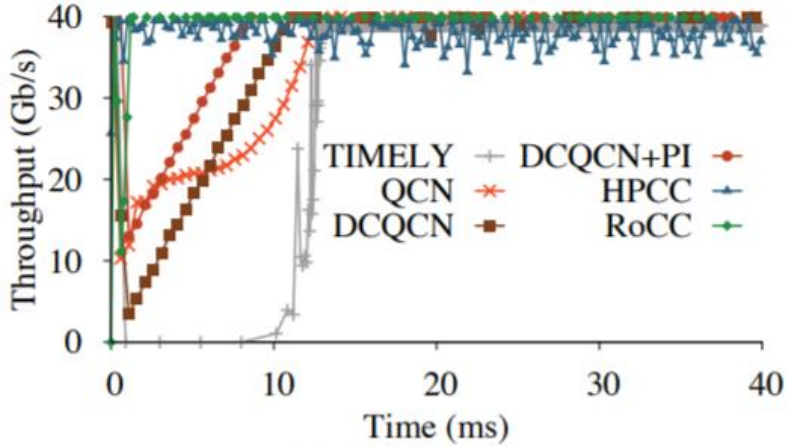
# Comparison results



(a) Fairness.



(b) Queue stability.



(c) Convergence.

- (a) shows RoCC has the best fairness performance
  - Due to fair rate computation on switch
- (b) and (c) show RoCC is more stable and converges quickly

# Conclusion

- Authors have proposed RoCC, a new switch-driven congestion control solution for RDMA
- RoCC employs rate control system that uses the egress queue size as input
- The authors show that RoCC is fair and efficient through evaluation