Scaling Switch-driven Flow Control with Aquarius

Wenxue Li, Chaoliang Zeng, Jinbin Hu , Kai Chen

iSingLab HKUST

Outline

- Introduction
- Background
- Motivation
- Aquarius Design
- Evaluation
- Summary

Outline

- **Introduction**
- **Background**
- **Motivation**
- **Aquarius Design**
- **Evaluation**
- **Summary**

Introduction

- **Key motivations:**
  - End-to-end congestion controls becomes increasingly challenging to maintain effective due to the inherent feedback delay.
  - Prior flow control (FC) mechanisms either lack fine-grained (i.e., per-flow granularity) control or require an impractical number of queues.
- **Solution:**

  Aquarius, a scalable solution that maintains fine-grained per-flow level control granularity with a practical number of queues.

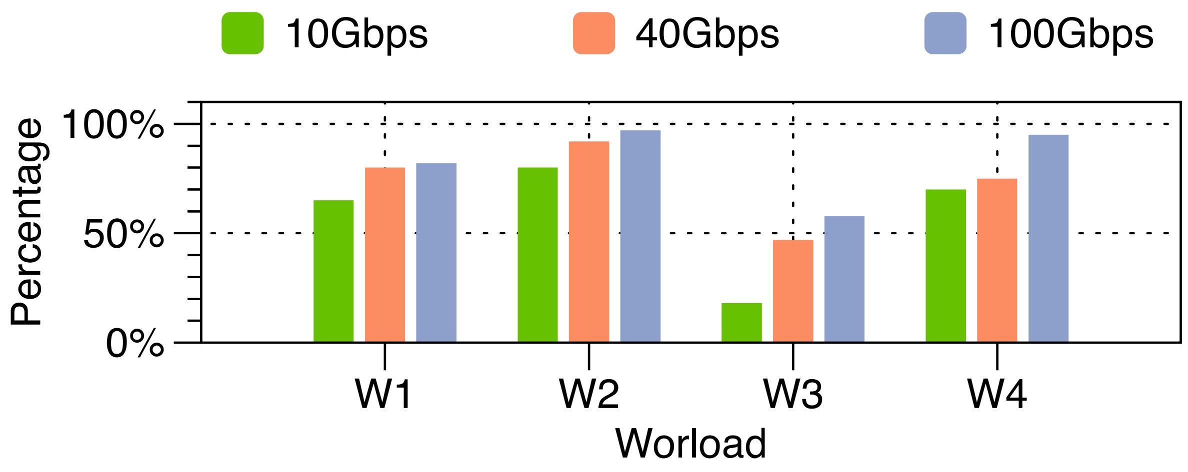
Outline

- Introduction
- **Background**
- Motivation
- Aquarius Design
- Evaluation
- Summary

Background

- **Rising link speeds result in increasingly bursty traffic**

  Representative production datacenter workloads:
  (W1) Web Server [2], (W2) Alibaba Storage [3],
  (W3) Web Search [1], (W4) Facebook Hadoop [2]



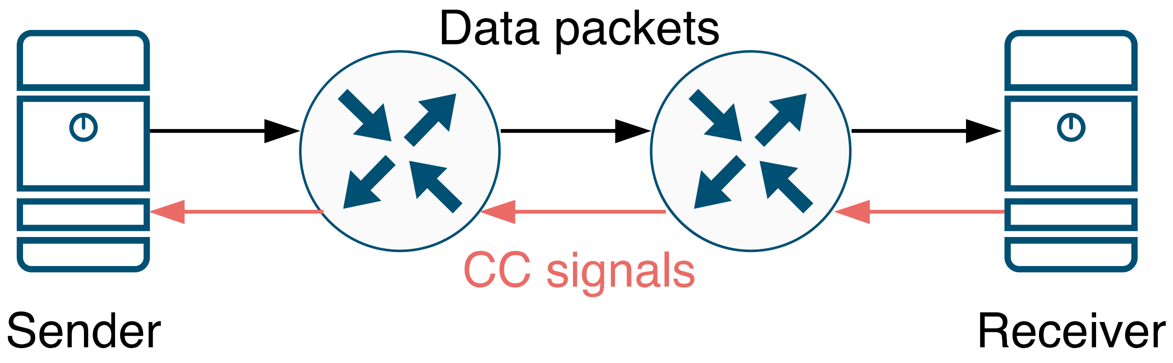
Percentage of flows that finish in a single RTT (12us)

[1] M. Alizadeh, et al. "Data center tcp (dctcp)," in Proceedings of the ACM SIGCOMM 2010 Conference.
[2] Arjun Roy, et al. "Inside the social network's (datacenter) network", in Proceedings of the ACM SIGCOMM 2015 Conference.
[3] Yuliang Li, et al. "HPCC: High precision congestion control", in Proceedings of the ACM SIGCOMM 2019 Conference.

Background

- **End-to-end CC alone is insufficient for managing transient congestion**
  - End-to-end CCs rely on receiver-echoed signals to adjust sending rates.
  - sender requires at least one RTT to receive feedback and loses control of flows that can complete within the first RTT.

- **Per-hop flow control is necessary for handling transient congestion**



Outline

- Introduction
- Background
- **Motivation**
- Aquarius Design
- Evaluation
- Summary

Motivation

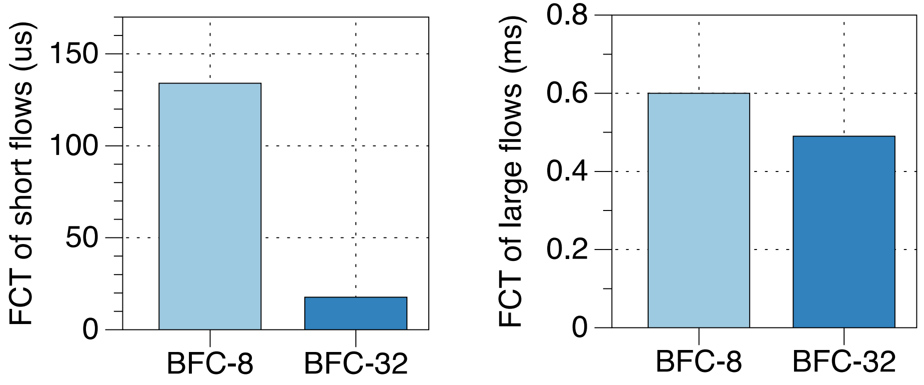
***Prior* flow control schemes are insufficient; they either lack fine-grained control or require an impractical number of queues.**

- **PFC is coarse-grained**
- **Ideal Flow Control is fine-grained but impractical**
  The ideal flow control allocates a dedicated queue to every flow, thus providing per-flow level control. However, the per-flow queue is impractical.
- **Scalability issues persist in BFC [NSDI'22]**
  BFC assigns a dedicated queue to each active flow if possible and enables multiple flows sharing a queue when there are no available queues.

BFC Scalability

- **BFC requires more physical queues than the common switch can accommodate**
  - BFC uses 32/128 queues per port.
  - (1) Majority of switches are usually equipped with 8 or fewer queues
  - (2) Physical queues are critical resources and are typically reserved for strong physical isolation between different tenants.

- **BFC experiences considerable performance degradation when queues are limited**



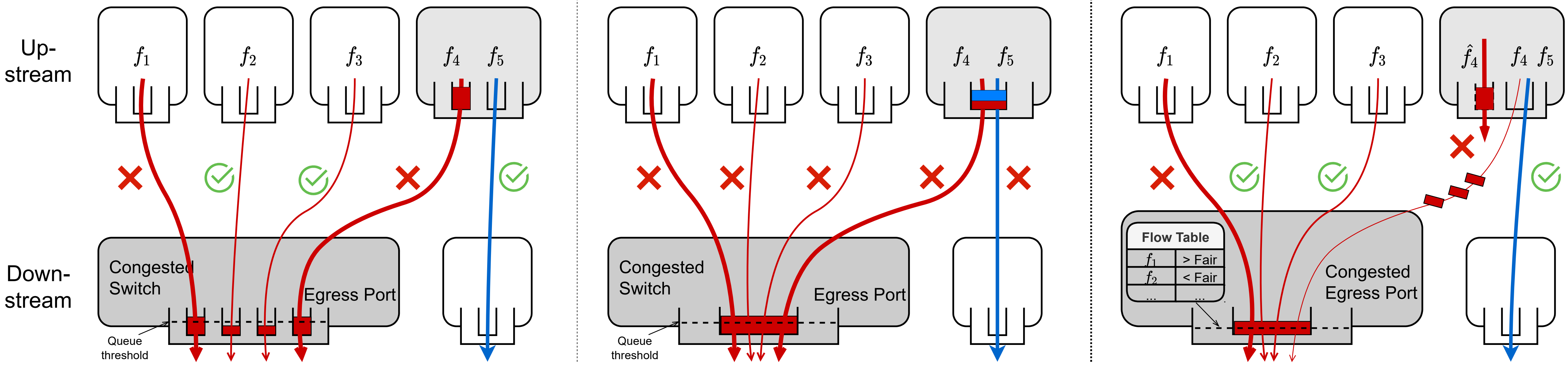
Average FCT of BFC under Web Server distribution.

Outline

- Introduction
- Background
- Motivation
- **Aquarius Design**
- Evaluation
- Summary

Key Idea

Approximate the ideal flow control behavior without requiring per-flow queues



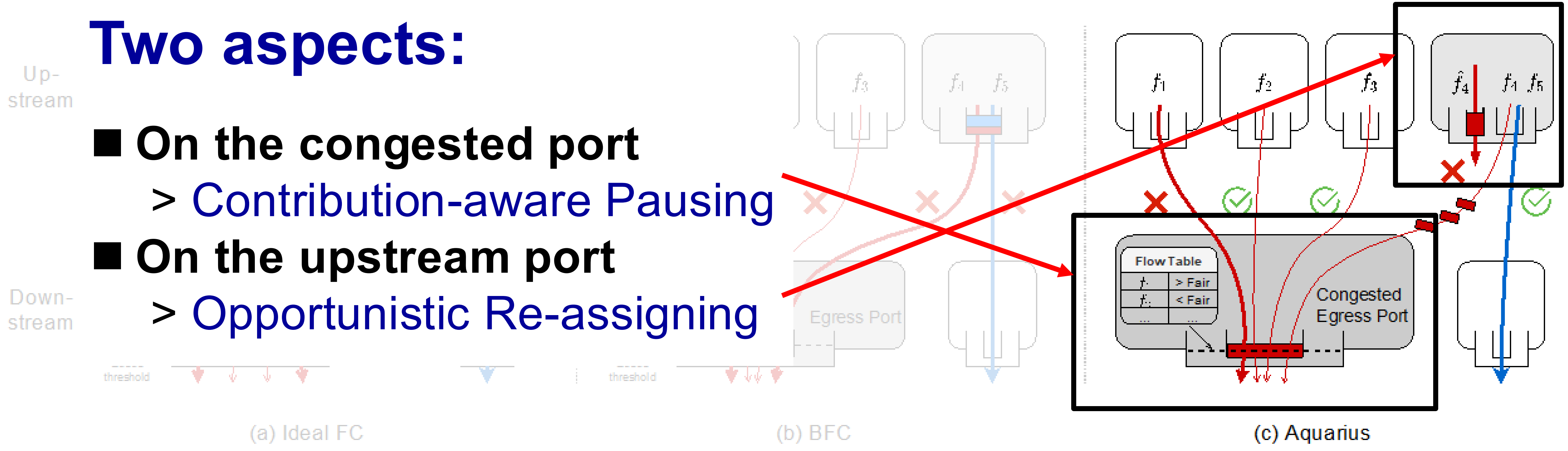
(a) Ideal FC (b) BFC (c) Aquarius

Key Idea

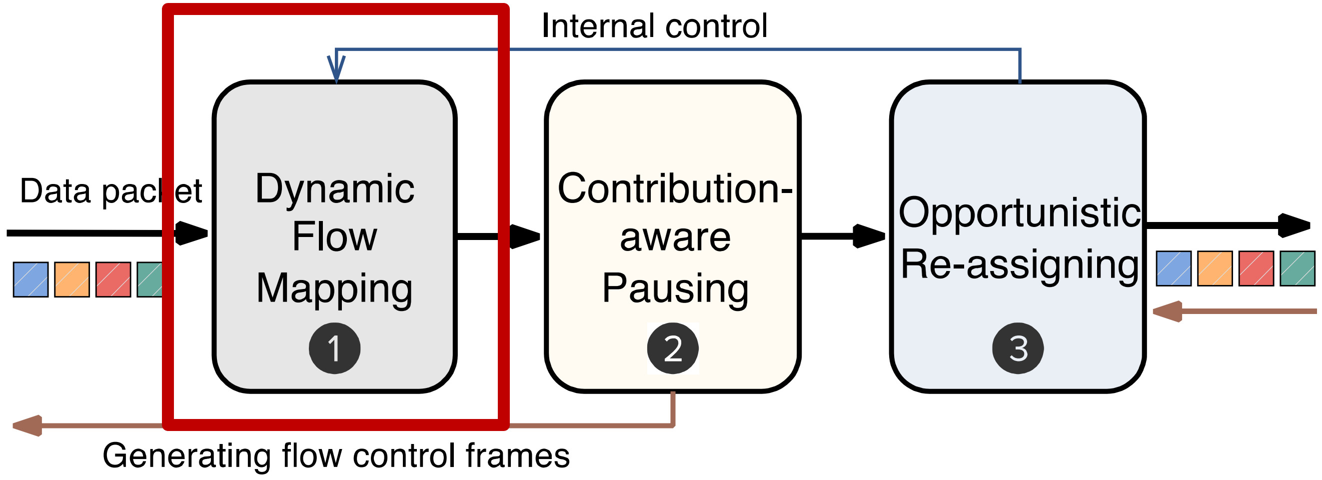
Approximate the ideal flow control behavior without requiring per-flow queues

Two aspects:

- **On the congested port**
  - > Contribution-aware Pausing
- **On the upstream port**
  - > Opportunistic Re-assigning



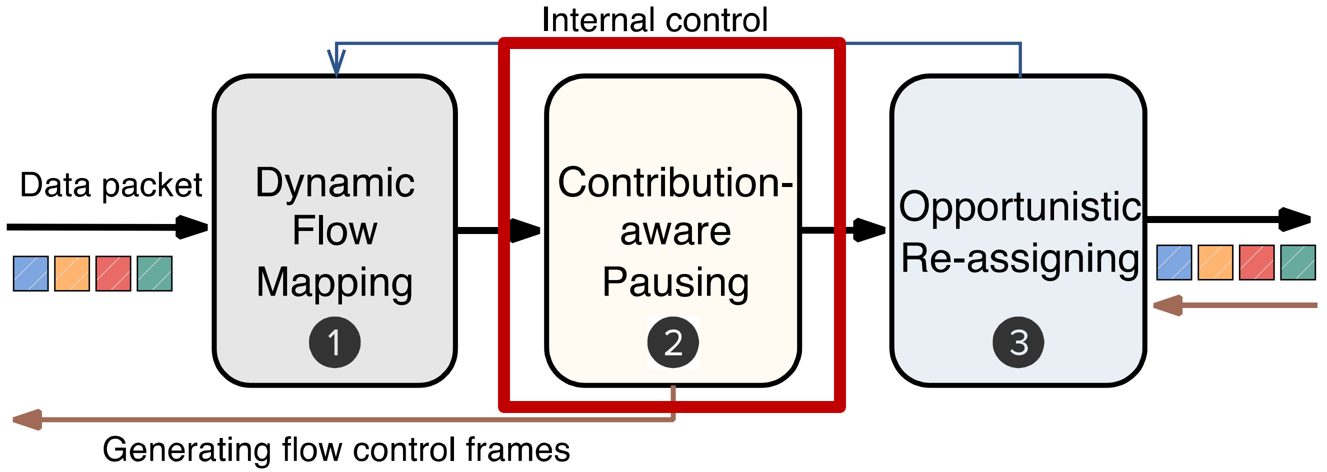
Aquarius



❶ ***Dynamic flow mapping* at every passed switch port**

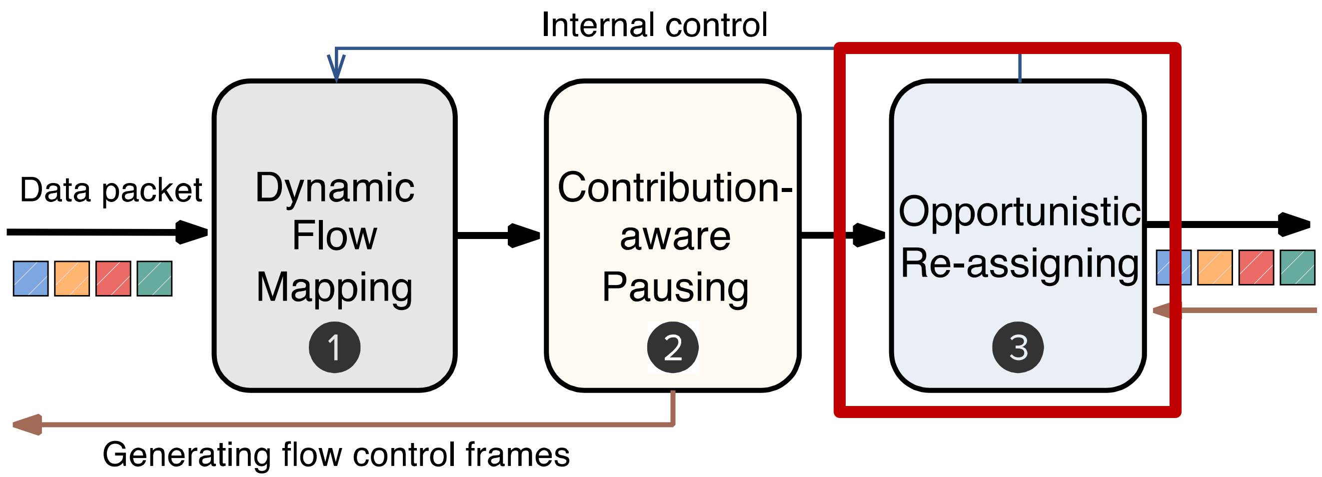
uniformly distribute all flows to available queues.

❷ Contribution-aware Pausing



- Records the size of each flow in Flow Table, indexed by *hash(FID)*
- Pausing Decision:
  - If Q_h > Q > Q_l: flow with size > *fair size* should be paused.
  - If Q > Q_h, all passed flow should be paused.
- *Fair size*: $[L >> [log_2 N]]$.

❸ Opportunistic Re-assigning



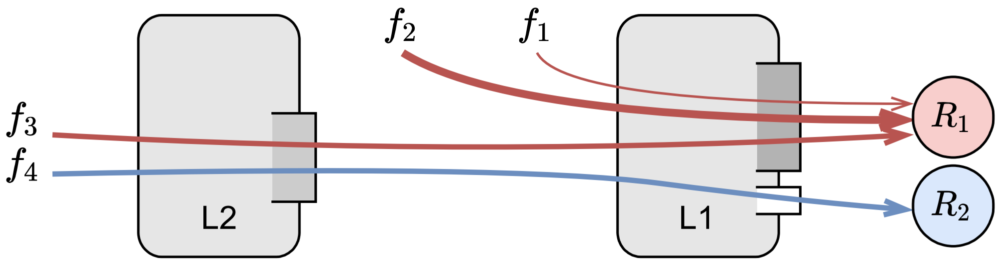
- ❑ PAUSE carries FID
- ❑ Re-direct all congested flows to a reserved isolation queue ($rsvQ$) by controlling the flow-to-queue mapping in ❶.
- ❑ Resuming condition of $rsvQ$:
  - ❑ 1) all isolated flows have received RESUME;
  - ❑ 2) previous buffered packets have been drained off

Outline

- Introduction
- Background
- Motivation
- Aquarius Design
- **Evaluation**
- Summary

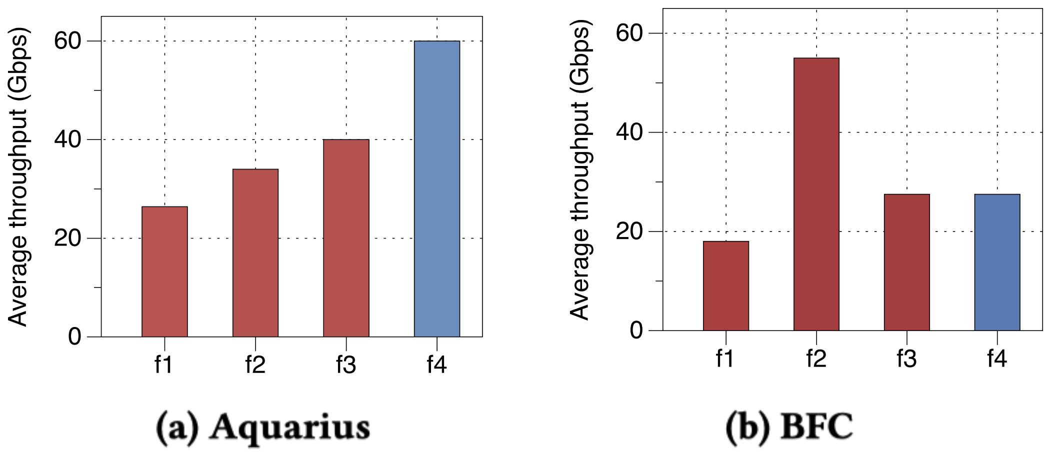
Micro-benchmark

- **Setting**
  - NS-3 simulator
  - f1: 33Gbps; f2~ f4: 100Gbps.
  - R1 becomes the bottleneck.



Micro-benchmark setting.

- **Aquarius achieves per-flow control granularity.**
  - Fair partition of bottleneck link capacity between f1 to f3.
  - The victim flow, f4, is not affected.



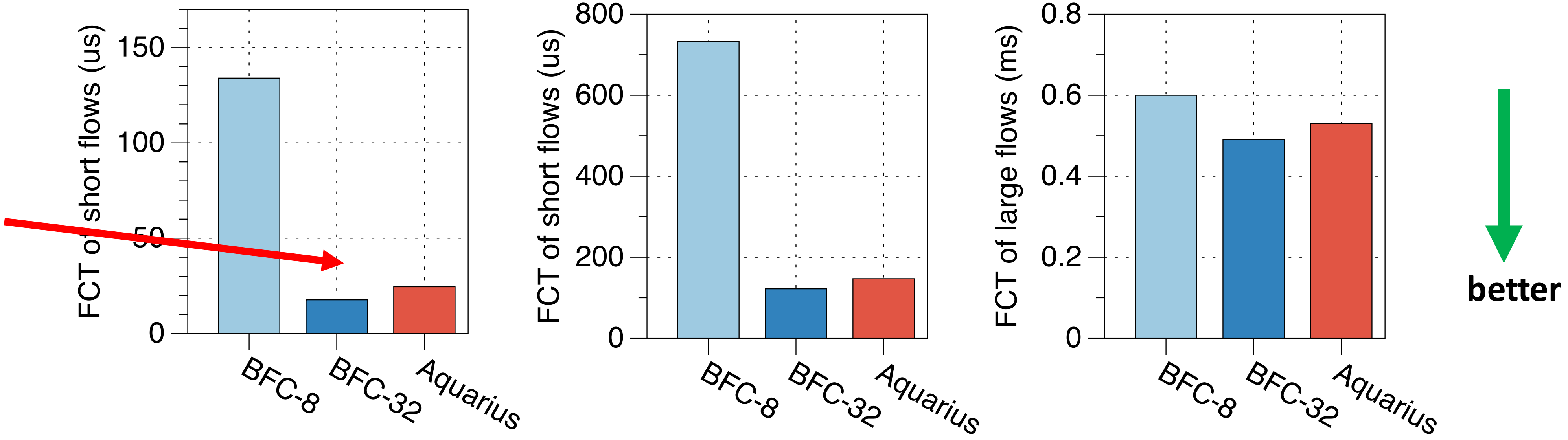
Average throughput for flows $f1 \sim f4$.

Realistic Traffic

■ Setting

- NS-3 simulator; 3-layer fat-tree topology; 48 switches; 128 servers
- 100Gbps link; 1us propagation delay; 12MB switch buffer
- Web Server with a 70% average load and 5% 100-to-1 incast traffic

Reducing FCT by up to 5X



FCT under Web Server distribution with 70% load and 5% 100-1 incast.

Outline

- Introduction
- Background
- Motivation
- Aquarius Design
- Evaluation
- **Summary**

Summary

- **Per-hop Flow Control is Necessary but Prior Scheme is Insufficient**
  - End-to-end CC alone is insufficient for managing transient congestion.
  - Prior flow control schemes either lack fine-grained control or require an impractical number of queues.
  - BFC experiences considerable performance degradation when queues are limited.
- **Key Idea of Aquarius**
  - To approximate the ideal flow control behavior without requiring per-flow queues.
- **Key points of Aquarius**
  - Contribution-aware pausing, that accurately identifies the set of congested flow, mimics the behavior of ideal FC at the congested port.
  - Opportunistic Re-assigning, that isolates congested flows from normal queues, mimics the behavior of ideal FC at the upstream port.

Thank you !

Contact email: wlicv@connect.ust.hk

Order Mark

-

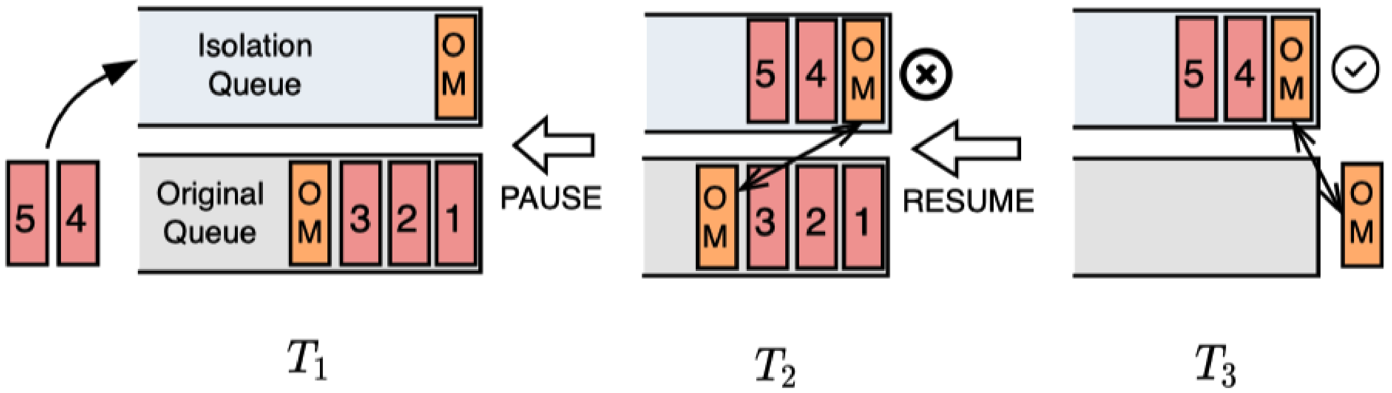


Fig. 7: A high-level view of how Order Mark (OM) packets support in-order delivery.