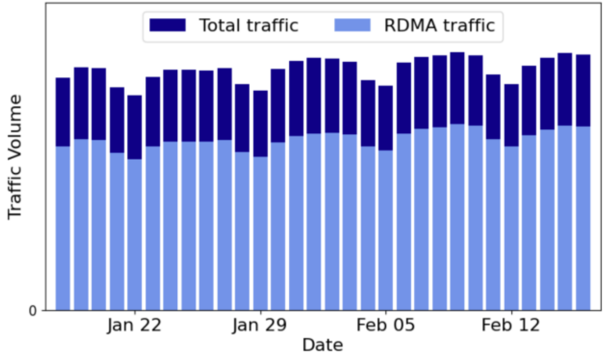
Revisiting Congestion Control for Lossless Etherent

Yiran Zhang, Qingkai Meng, Chaolei Hu, Fengyuan Ren

The 21th USENIX Symposium on Networked Systems Design and Implementation (NSDI '24)
April, 2024

Lossless Ethernet



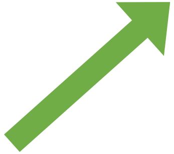
Wide adoption of RDMA

[1] Wei Bai, Shanim Sainul Abdeen, Ankit Agrawal et al. Empowering Azure Storage with RDMA. NSDI 2023

RDMA + PFC: lossless Ethernet

No packets dropping

Full potential of RDMA



But PFC comes with side effects!

Head of line (HoL) blocking,

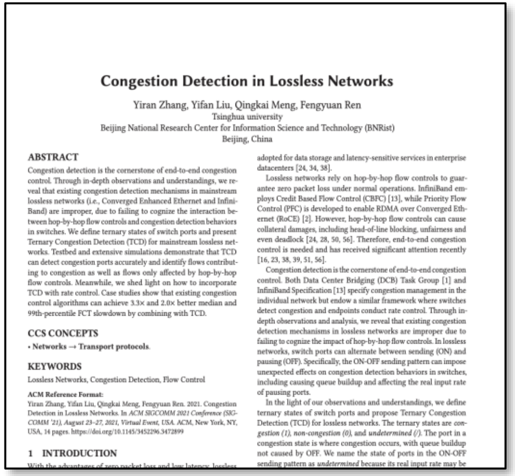
Deadlock, etc

Congestion control is a key enabler for lossless Ethernet at scale

Congestion Control in Lossless Ethernet

	Congestion Detection	Rate Control
DCQCN [SIGCOMM'15] TIMELY [SIGCOMM'15]	Following lossy networks (ECN, RTT)	Traditional heuristic rules
HPCC [SIGCOMM'19]	Advanced telemetry technique	Larger overhead

TCD: Ternary Signal for Lossless Networks



Congestion Detection in Lossless Networks

Yiran Zhang, Yifan Liu, Qingkai Meng, Fengyuan Ren
Tsinghua university
Beijing National Research Center for Information Science and Technology (BNRist)
Beijing, China

ABSTRACT

Congestion detection is the cornerstone of end-to-end congestion control. Through in-depth observations and understandings, we reveal that existing congestion detection mechanisms in mainstream lossless networks (i.e., Converged Enhanced Ethernet and InfiniBand) are improper, due to failing to cognize the interaction between hop-by-hop flow controls and congestion detection behaviors in switches. We define ternary states of switch ports and present Ternary Congestion Detection (TCD) for mainstream lossless networks. Testbed and extensive simulations demonstrate that TCD can detect congestion ports accurately and identify flows contributing to congestion as well as flows only affected by hop-by-hop flow controls. Meanwhile, we shed light on how to incorporate TCD with rate control. Case studies show that existing congestion control algorithms can achieve 3.3× and 2.0× better median and 99th-percentile FCT slowdown by combining with TCD.

CCS CONCEPTS

• **Networks → Transport protocols.**

KEYWORDS

Lossless Networks, Congestion Detection, Flow Control

ACM Reference Format:
Yiran Zhang, Yifan Liu, Qingkai Meng, Fengyuan Ren. 2021. Congestion Detection in Lossless Networks. In *ACM SIGCOMM 2021 Conference (SIGCOMM '21), August 23–27, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3452296.3472899

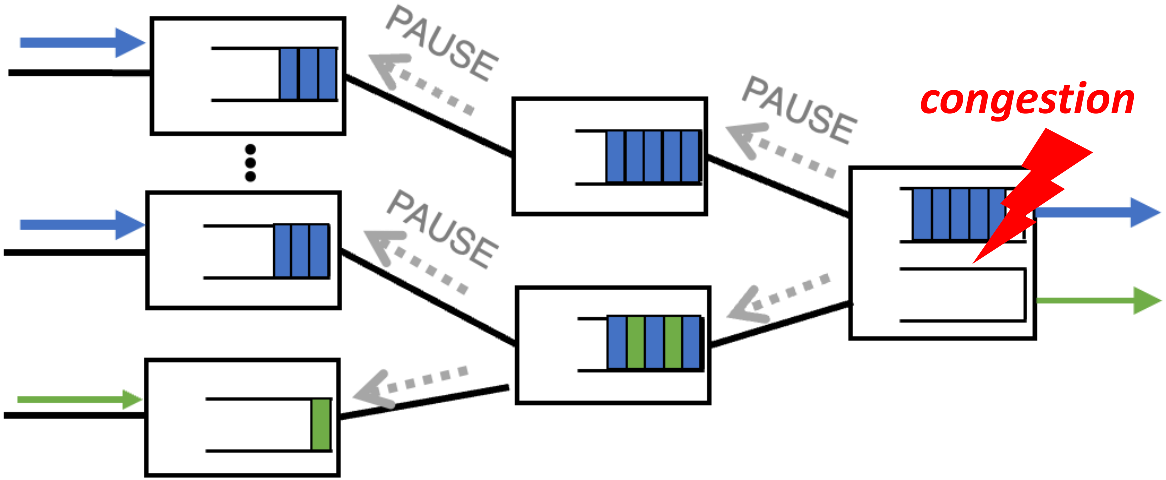
1 INTRODUCTION

adopted for data storage and latency-sensitive services in enterprise datacenters [24, 34, 38].

Lossless networks rely on hop-by-hop flow controls to guarantee zero packet loss under normal operations. InfiniBand employs Credit Based Flow Control (CBFC) [13], while Priority Flow Control (PFC) is developed to enable RDMA over Converged Ethernet (RoCE) [2]. However, hop-by-hop flow controls can cause collateral damages, including head-of-line blocking, unfairness and even deadlock [24, 28, 50, 56]. Therefore, end-to-end congestion control is needed and has received significant attention recently [16, 23, 38, 39, 51, 56].

Congestion detection is the cornerstone of end-to-end congestion control. Both Data Center Bridging (DCB) Task Group [1] and InfiniBand Specification [13] specify congestion management in the individual network but endow a similar framework where switches detect congestion and endpoints conduct rate control. Through in-depth observations and analysis, we reveal that existing congestion detection mechanisms in lossless networks are improper due to failing to cognize the impact of hop-by-hop flow controls. In lossless networks, switch ports can alternate between sending (ON) and pausing (OFF). Specifically, the ON-OFF sending pattern can impose unexpected effects on congestion detection behaviors in switches, including causing queue buildup and affecting the real input rate of pausing ports.

In the light of our observations and understandings, we define ternary states of switch ports and propose Ternary Congestion Detection (TCD) for lossless networks. The ternary states are *congestion (1), non-congestion (0),* and *undetermined (/)*. The port in a congestion state is where congestion occurs, with queue buildup not caused by OFF. We name the state of ports in the ON-OFF sending pattern as *undetermined* because its real input rate may be



***Yiran Zhang**, Yifan Liu, Qingkai Meng, Fengyuan Ren.*
Congestion Detection in Lossless Networks.
SIGCOMM'21

- ✓ **Congested flows (CE)**
- ✓ **Victim/Undetermined flows (UE)**
- ✓ **Uncongested flows (NO)**

Congestion Control in Lossless Ethernet

	Congestion Detection	Rate Control
DCQCN[SIGCOMM'15] TIMELY[SIGCOMM'15]	Following lossy networks (ECN, RTT)	Traditional heuristic rules
HPCC [SIGCOMM'19]	Advanced telemetry technique	Larger overhead
TCD [SIGCOMM'21]	Precise ternary signal for lossless Ethernet	Traditional heuristic rules

Congestion Control in Lossless Ethernet

	Congestion Detection	Rate Control
DCQCN [SIGCOMM'15]		
TIMELY [SIGCOMM'15]		
HPCC [SIGCOMM'19]		
TCD [SIGCOMM'21]	Precise ternary signal for lossless Ethernet	Traditional heuristic rules

Sub-optimal performance

- ☹ **Suffering HoL blocking**
- ☹ **Hard to achieve low latency and high throughput, etc**

A Desirable Congestion Control for Lossless Ethernet

Congestion Detection + **Rate Control** = **Congestion Control**

Tailored for lossless Ethernet

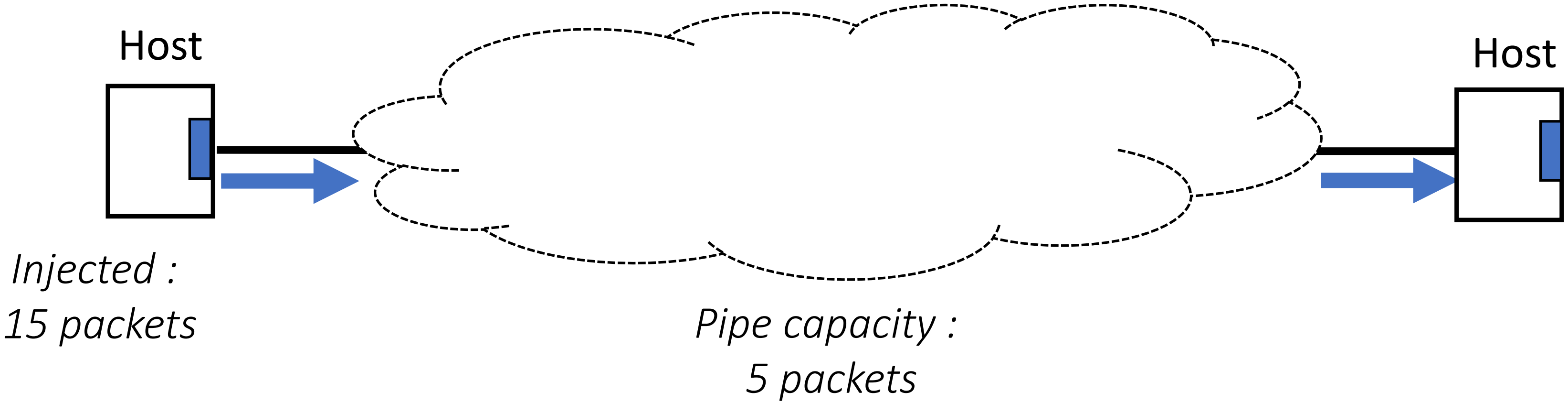
Tailored for lossless Ethernet

High-performance for lossless Ethernet

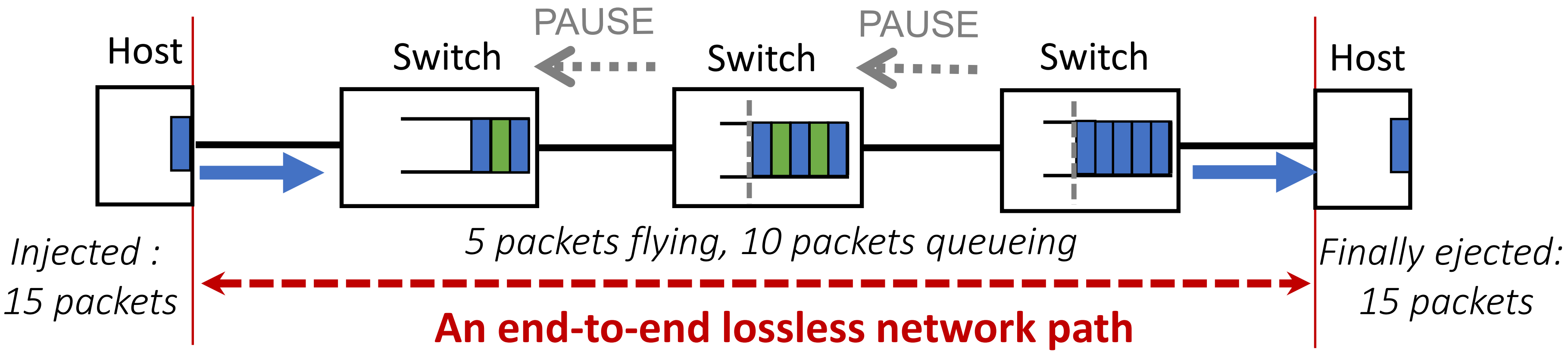
- ✓ Fast convergence to alleviate HoL blocking, deadlock etc.
- ✓ Low latency
- ✓ High throughput

Can we rethink congestion control for lossless Ethernet by taking full advantage of its intrinsic properties?

Revisiting the Impact of PFC



Revisiting the Impact of PFC



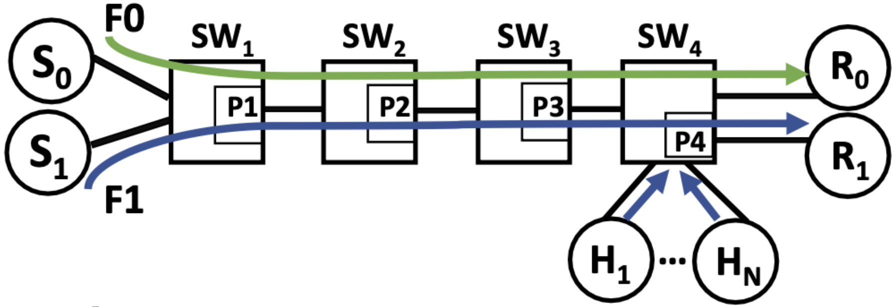
Packet Conservation Property

- Number of injected packets = number of ejected (acked) packets
- All injected packets are either flying or queueing

Packet Conservation Empowers ACK-Driven

Simulation:

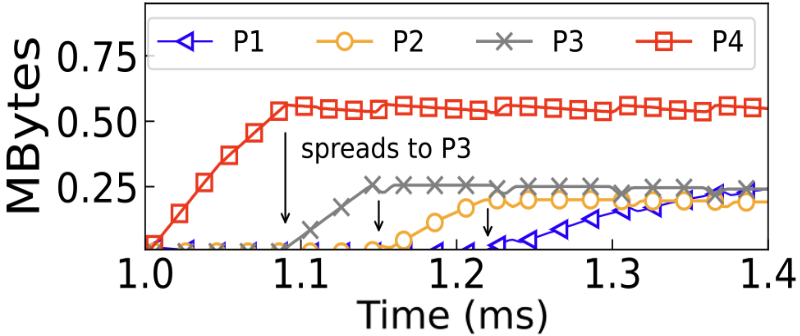
H_1-H_N: concurrent burst



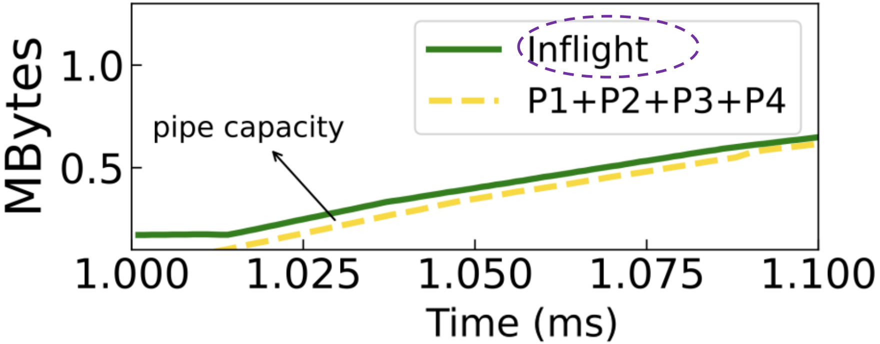
F0: victim flow

F1: congested flow

Queue size



Total queue size

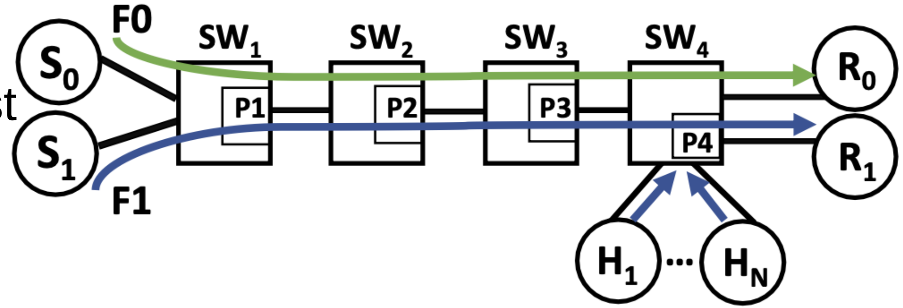


Excessive packets queueing in switches = inflight packets - network pipe capacity

Packet Conservation Empowers ACK-Driven

Simulation:

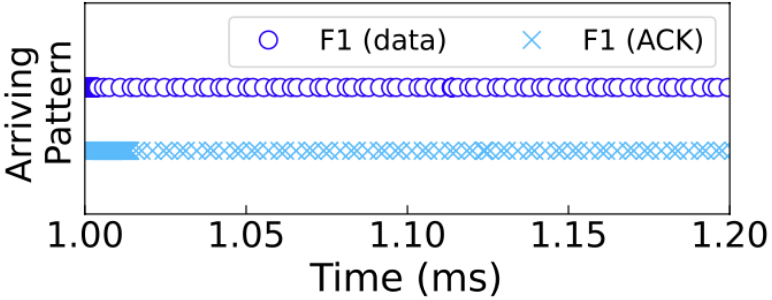
H_1-H_N: concurrent burst



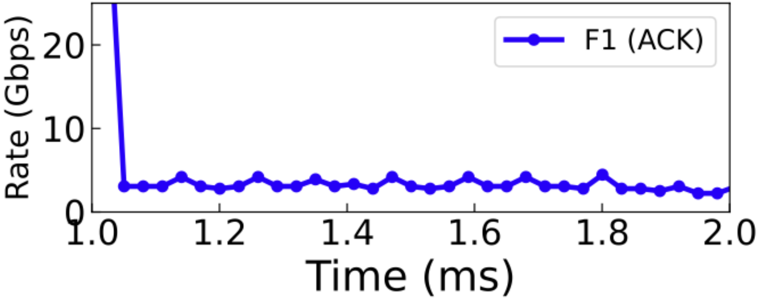
F0: victim flow

F1: congested flow

ACK arriving pattern



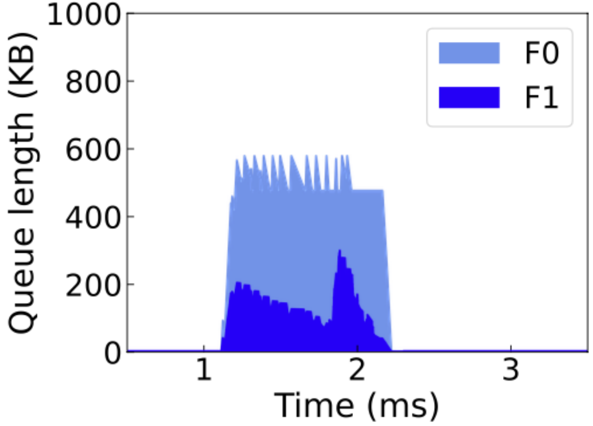
ACK arriving rate



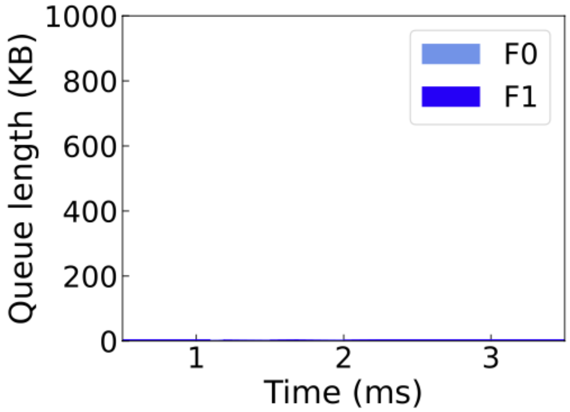
The ACK arrival rate can imply the available bandwidth

Handling HoL Blocking

Queue occupancy at a HoL blocked port.
F0: victim flow. F1: congested flow



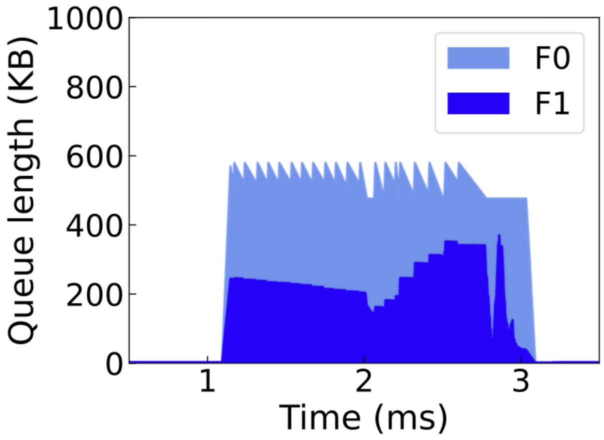
Stopping only congested flow F1
enough can eliminate HoL blocking



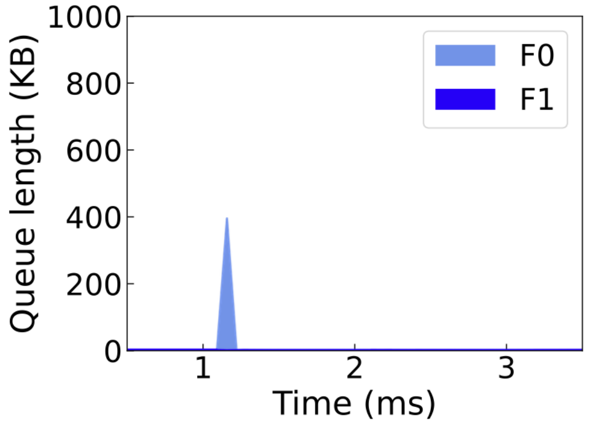
Stopping congested flows sufficiently long can empty buffers as soon as possible

Handling HoL Blocking

If HoL blocking is more severe:
F0: victim flow. F1: congested flow



Only stopping congested flow F1
can **not** eliminate HoL blocking:



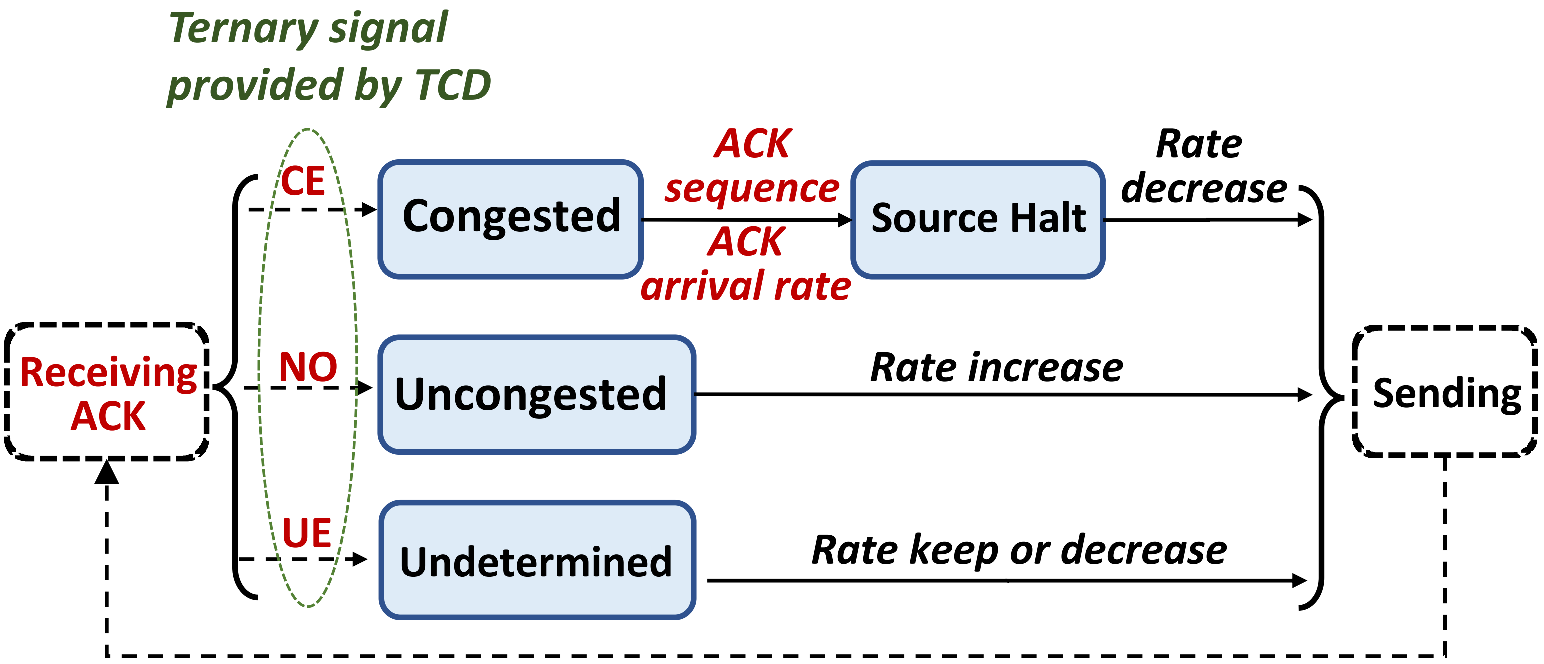
Whether should throttling victim flows depends on the extent of congestion

Summary of Principles

✓ **The ACK-driven paradigm should be renewed**:
Inferring the proper throttled rate and the precise number of excessive packets for congested flows

✓ **Handling HoL blocking needs individual rules**:
Stopping congested flows sufficiently long is the foremost means to suppressing HoL blocking
Victim flows should adapt to the severity of congestion

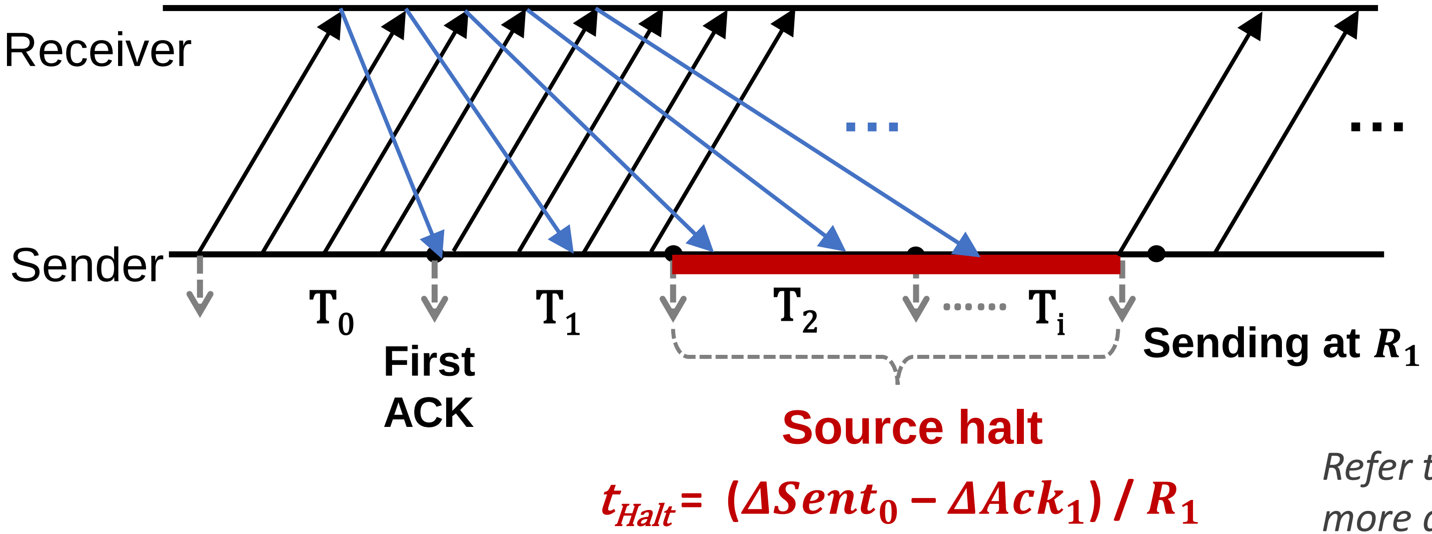
ACK-Driven Congestion Control (**ACC**)



ACC State Machine

Halting and Throttling Congested Flows

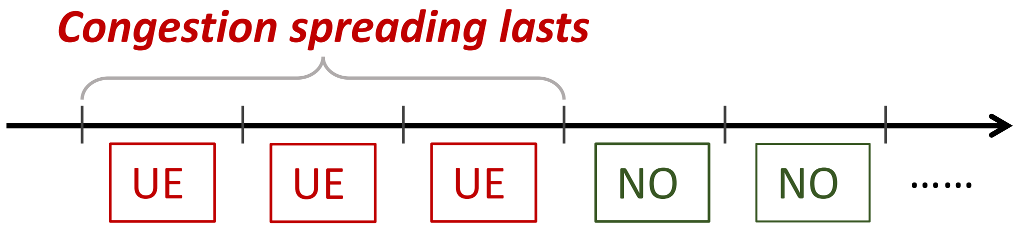
- ***First*** **halting to wait for the excessive packets to drain out**
- ***Then*** **matching the rate to the pipe capacity**



Refer to paper for more details

Victim Flows & Uncongested Flows

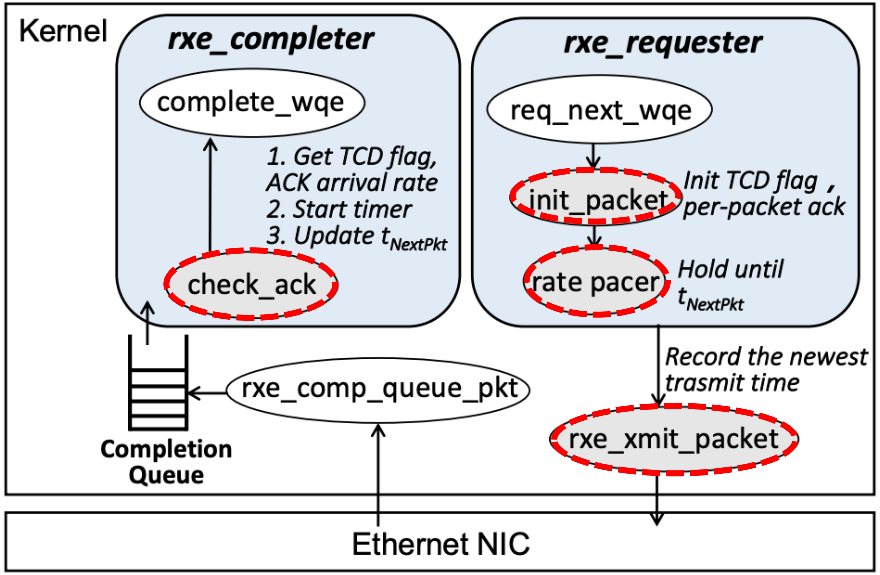
- **Victim flows: adapt to the severity of congestion spreading**
  Indicator: the number of consecutive periods with UE marks
  Exceeding a threshold P_{thresh}: decreasing the injection rate



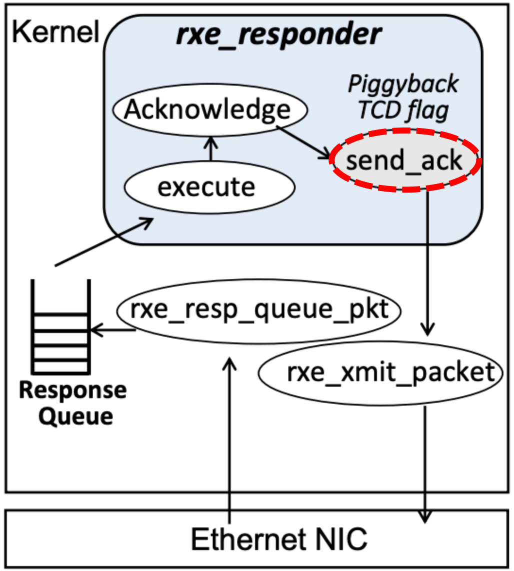
- **Uncongested flows:**
  increasing gradually at first and then aggressively

Implementation

SoftRoCE: software implementation of RDMA



ACC sender



ACC receiver

Implementation

Testbed

- 242 and 3 lines of code added in SoftRoCE sender and receiver
- 119 lines of code added to SoftRoCE common library
- 5 hosts with Intel 82599ES 10GbE NIC + 1 Tofino switch

Simulator

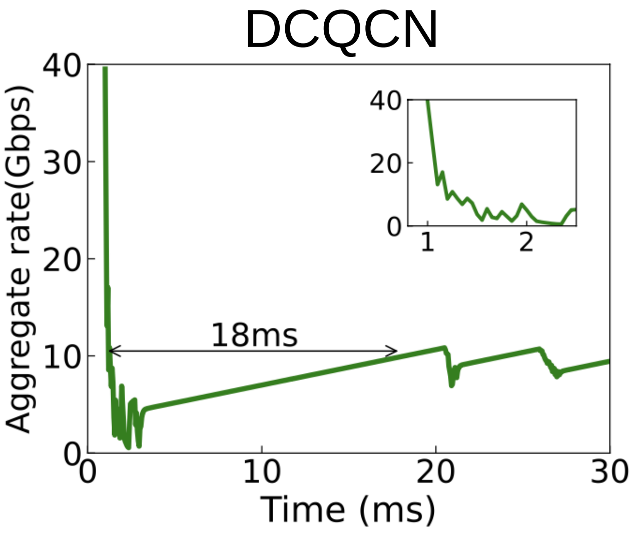
- Customized NS3 packet simulator
- Fat-tree network with 320 servers in 20 racks
- 100Gbps/400Gbps

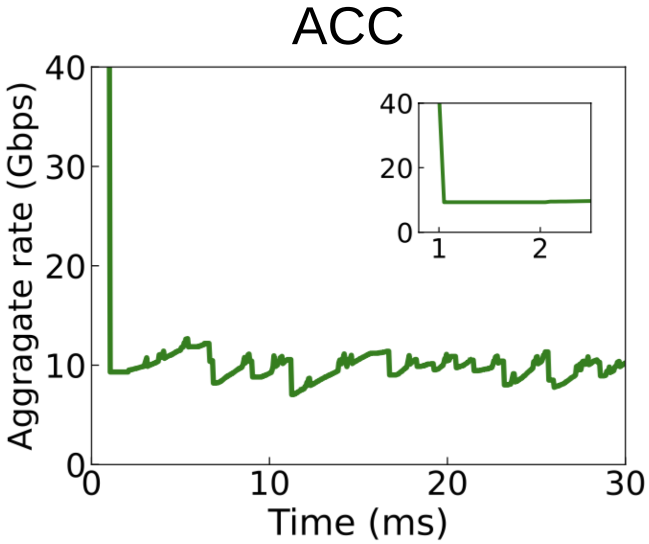
Evaluation Summary

- **Basic Properties**
  - ✓ **25X faster** Convergence
  - ✓ **Full** link utilization
  - ✓ **1.35X faster** emptying the queues and **suppressing HoL blocking**
  - ✓ Effectively **prevent deadlocks**
  - ✓ Good fairness
  - ✓ Proper parameters (UE periods threshold for victim flows, etc)
- **FCT Performance**
  - ✓ **1.3~3.3X** and **1.4~11.5X** better FCT (avg and P99) of small flows
  - ✓ Not sacrificing throughput of large flows
  - ✓ Source halt greatly benefits low latency and reduces PFC PAUSEs

Fast Convergence

Testbed results

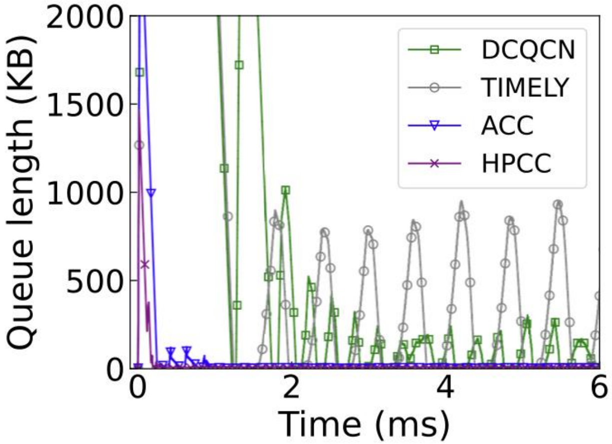




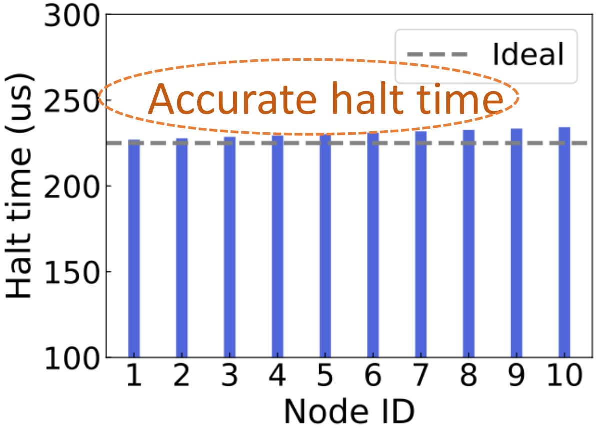
25X faster convergence than DCQCN

High Link Utilization and Low Queues

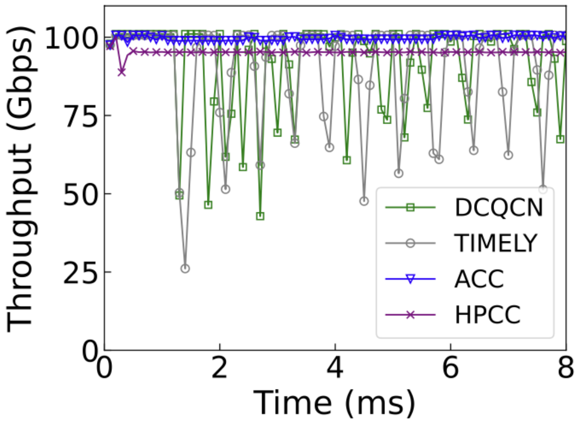
Simulation results



(a) Queue length



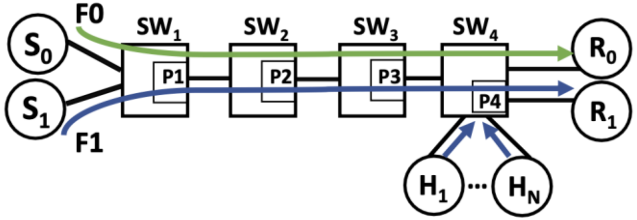
(b) Halt time in ACC



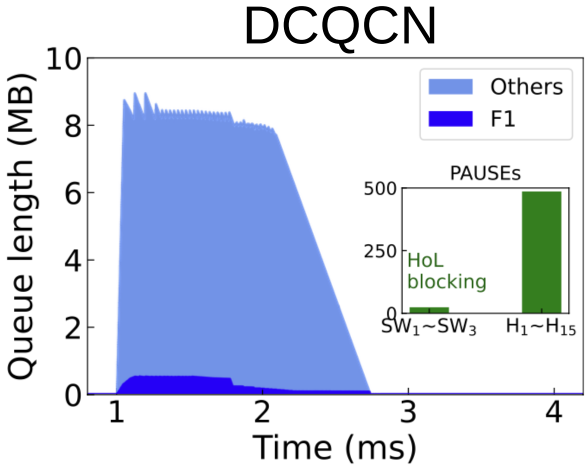
(c) Bottleneck link utilization

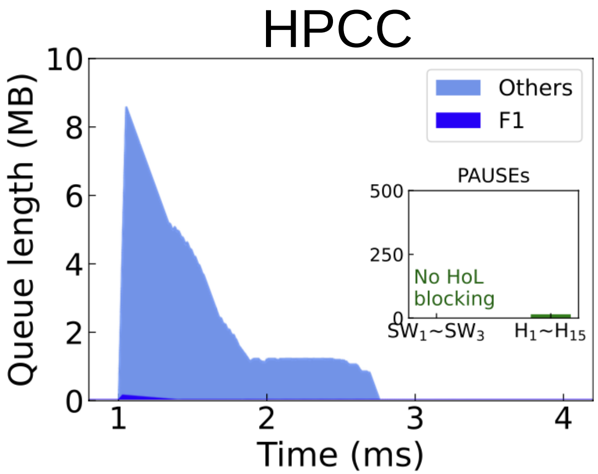
ACC can quickly eliminate congestion and maintain near full link utilization

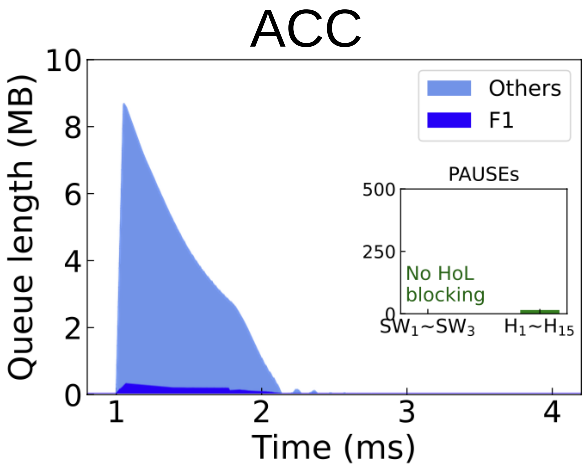
Suppressing HoL Blocking



Simulation results





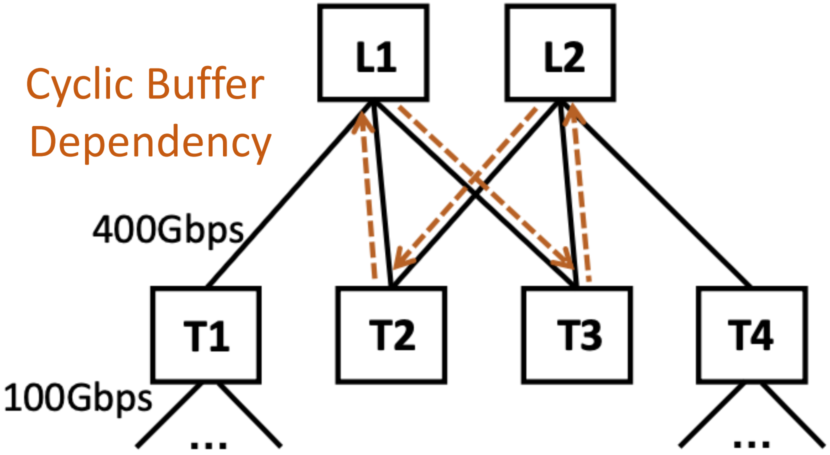


1.35X faster than HPCC

ACC can effectively alleviate HoL blocking and congestion spreading under bursty traffic

Resiliense to Deadlocks

Simulation results



Deadlock-prone topology

Fraction of deadlock runs

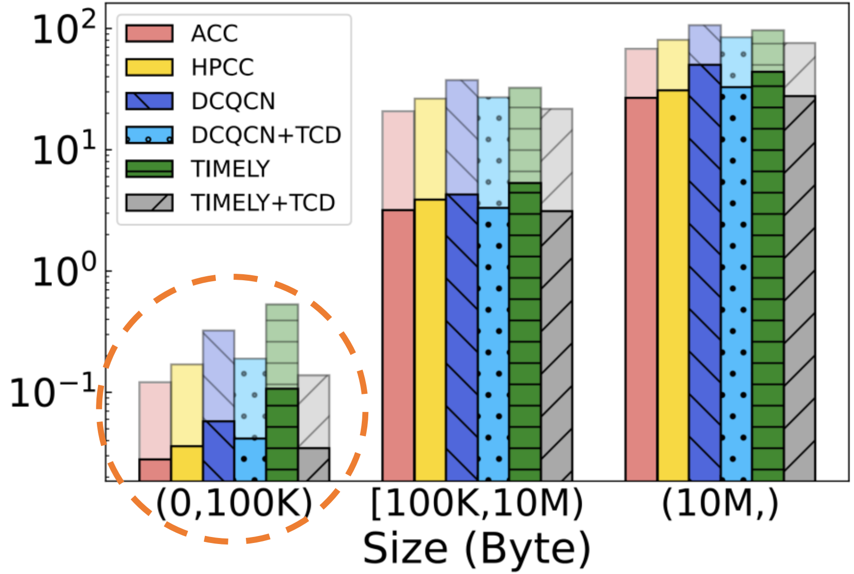
Scheme	Fraction
DCQCN	6%
TIMELY	74%
HPCC	0%
ACC	0%

No deadlocks in 50 runs

Fast convergence of CC does help prevent deadlocks

FCT Performance Gains

Avg/P99 FCT(ms)



Web Search 80% load

ACC vs. HPCC: reduces avg FCT by **29%,** P99 FCT by **40%**

ACC vs. DCQCN+TCD, TIMELY+TCD : **3.9X** and **5.1X** better P99 FCT

Conclusions

- **ACC pushes precise congestion control in lossless Ethernet by unlocking its intrinsic *packet conservation property***
  - ✓ *Only utilizing ACKs to infer the throttled rate, excessive packets and conduct accurate source halting*
- ACC well alleviates thorny issues (HoL blocking , congestion spreading and deadlock) and achieves lower FCT
- ACC can inspire congestion control or traffic management in other lossless interconnects