INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING
ICPP/2021/CHICAGO/USA
acm In-Cooperation
sighpc
AUGUST 9–12, 2021

Receiver-Driven Congestion Control for InfiniBand

Yiran Zhang, Kun Qian, Fengyuan Ren

Tsinghua University

InfiniBand Interconnect

- Lossless and low-latency communication
- Credit-based link-level flow control to guarrantee losslessness
- The side-effect: congestion spreading
  - Head-of-line blocking: waiting for credits
  - Increasing queueing delay and damaging the completion time of applications

IB Congestion Control in IB Specification

- Switch behavior
  - Congestion detection and identification: exceeding the selected threshold & there are available credits -> a root of congestion -> FECN notification

- Channel Adapter (CA) behavior
  - Rate regulation: On receiving CNP, regulates the sending rate with a fixed moving step according to a CCT table -> multiple round-trip times to converge

Traffic in HPC Systems

- Applications: Communication-intensive, I/O-intensive →

  short-lived MPI messsages long-lived I/O traffic

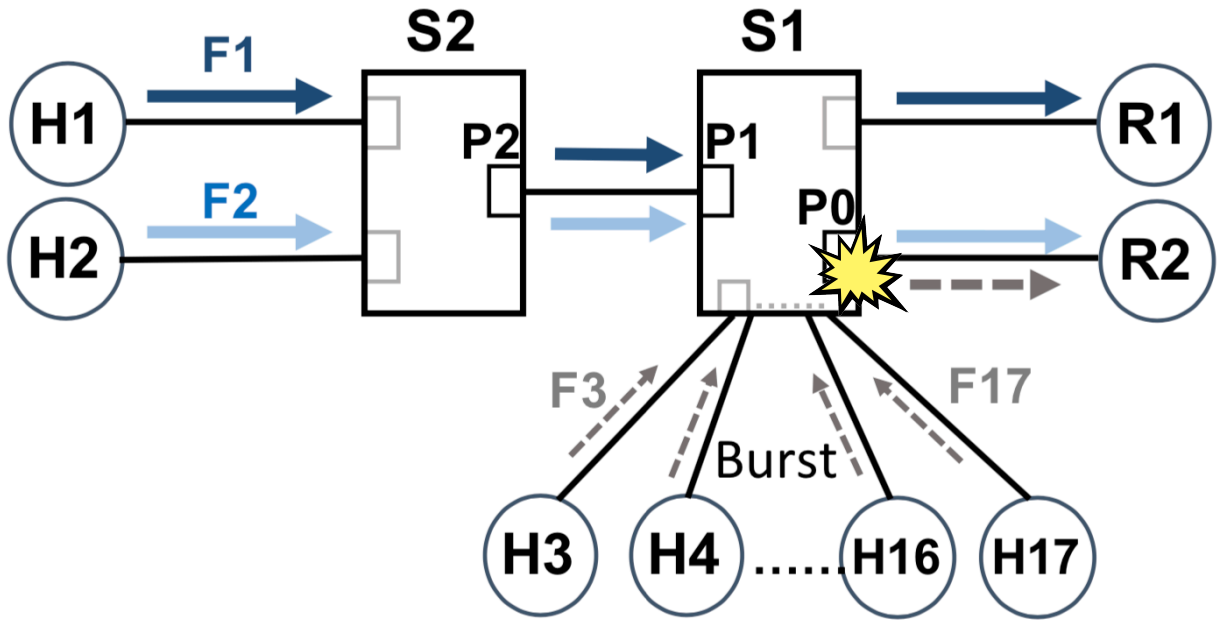
- Shared network infrastracture and file systems →

  increasingly diverse traffic patterns

Coexisting diverse traffic imposes great challenges to congestion control

Experimental observations

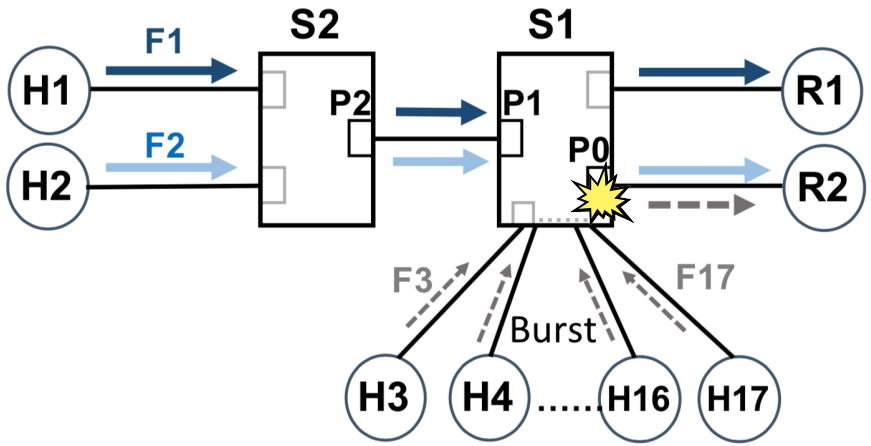
- Simulations
- Mixture of short messages (F3-F17) and long-lived messages (F1 F2)
- Congestion port: P0

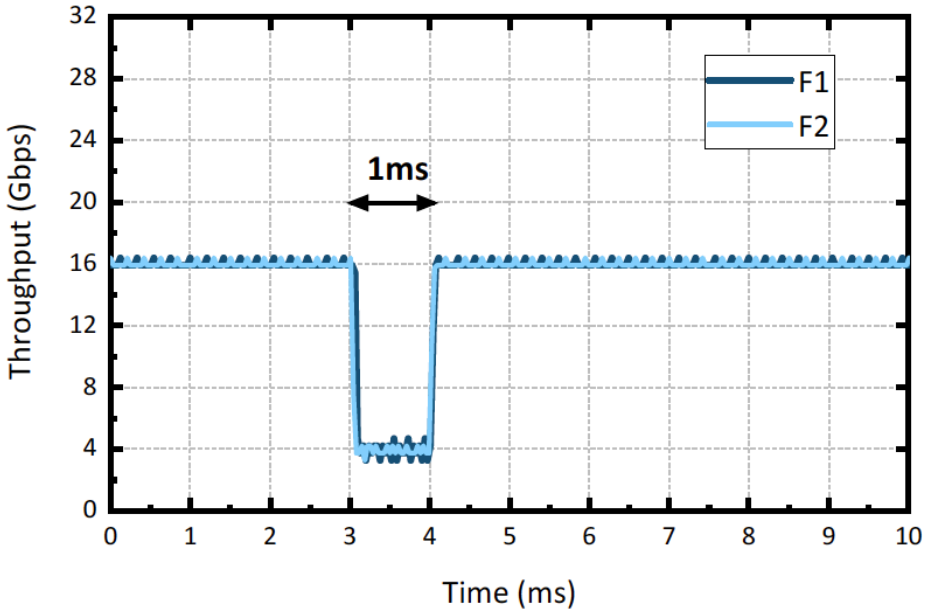


Burst last for about 1ms

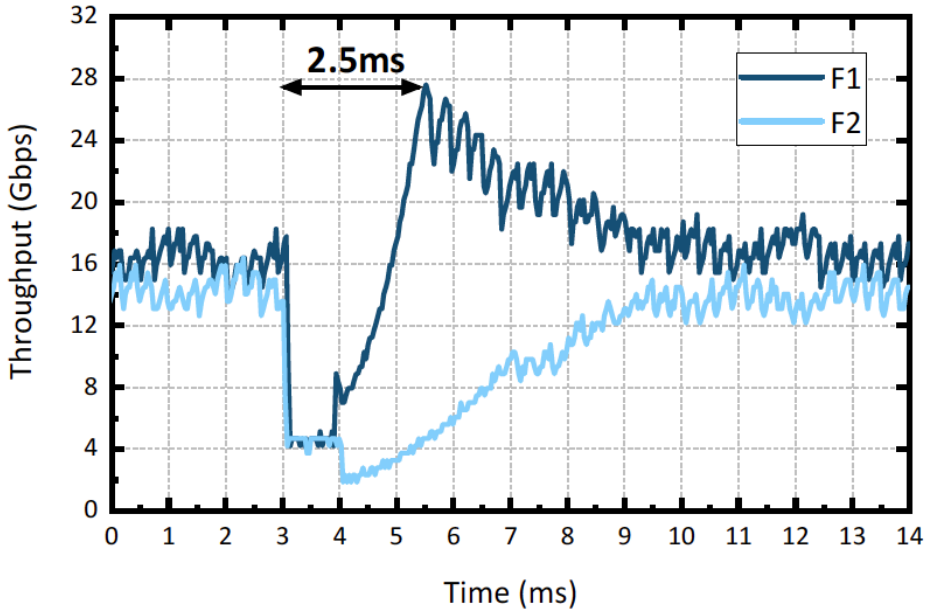
Experimental observations

- Overall throughput performance





No CC

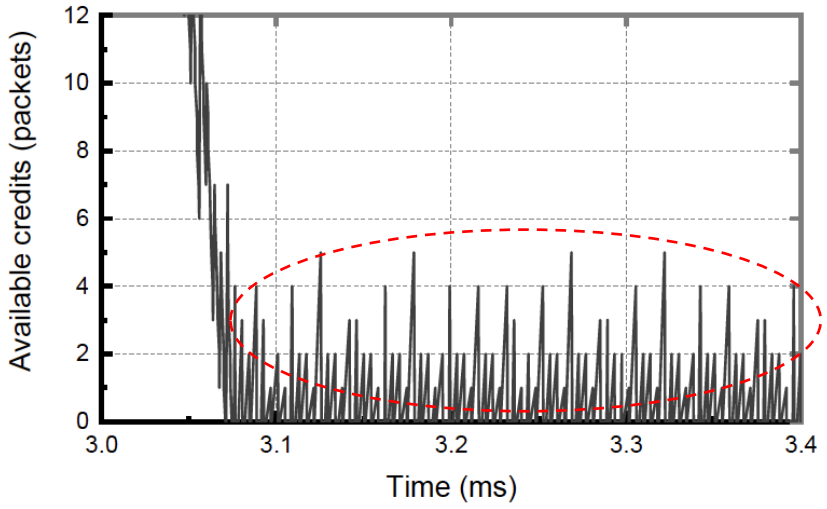


IB CC 2.5ms under-utilized

F1 should not be throttled
F2 should be throttled.

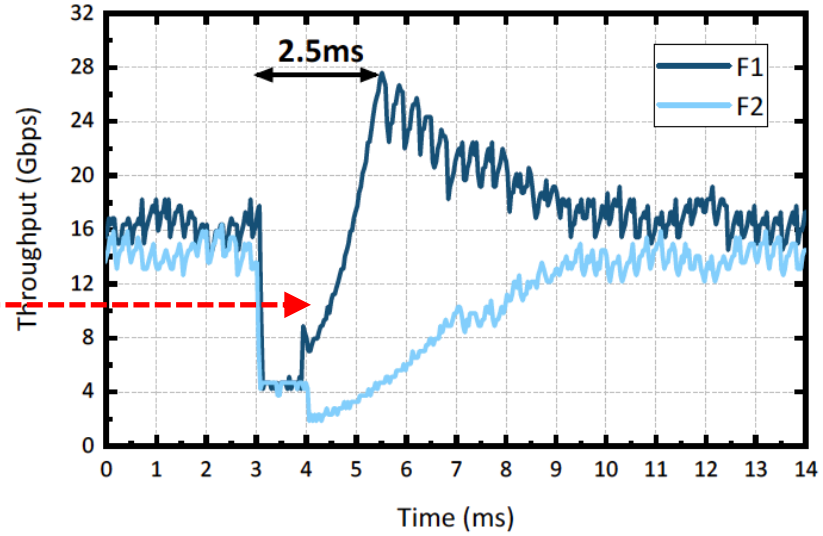
Insights (1)

- Periodical updated credits may confuse congestion detection.



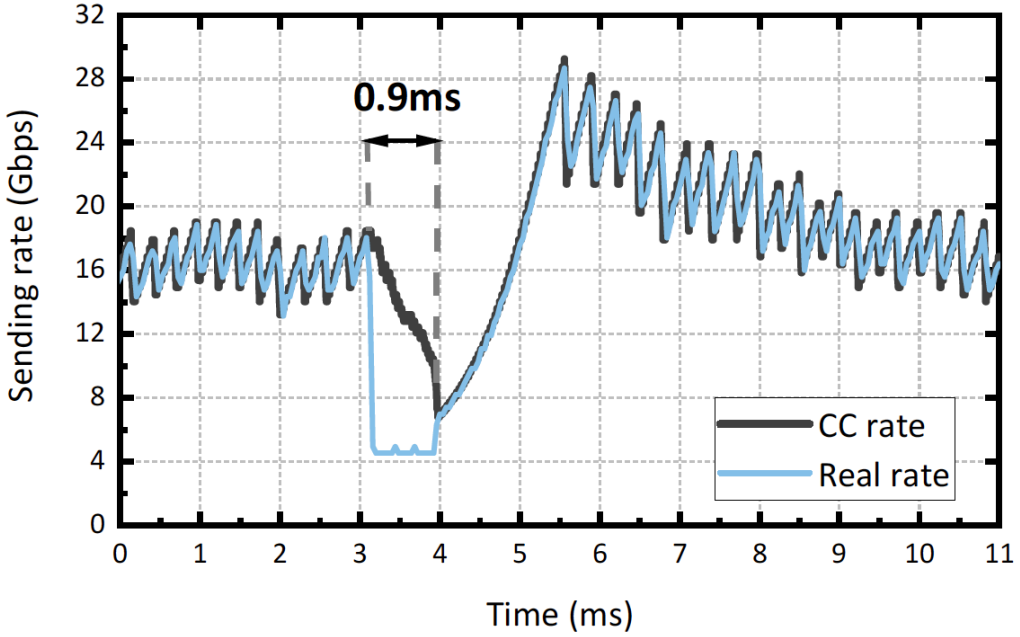
(a) Available credits at port P2

Partially packets of F1 is mistakenly marked

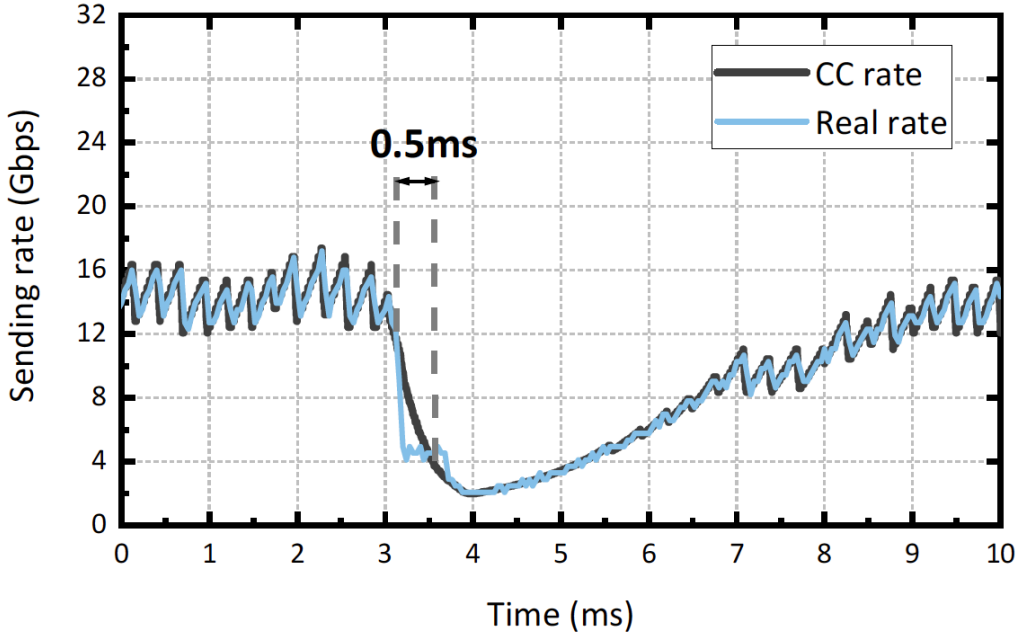


Insights (2)

- Periodical updated credits may confuse congestion detection.
- The sluggish rate adjustment of end-to-end IB CC mismatches the fast hop-by-hop credit-based flow control.



IB CC: F1



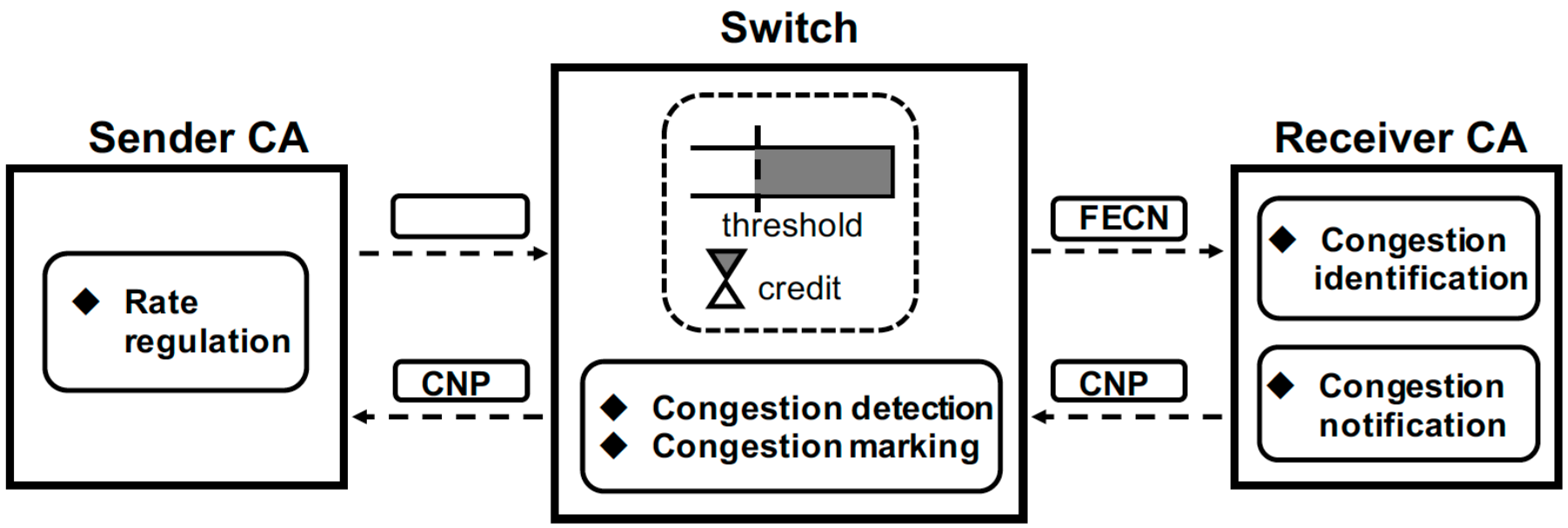
IB CC: F2

Hop-by-hop Flow control reacts faster

Insights Summary

- Periodical updated credits may confuse congestion detection.
- The sluggish rate adjustment of end-to-end IB CC mismatches the fast hop-by-hop credit-based flow control.
- Congestion-unaware rate increase contradicts with rate decrease: parameter tuning is tough

Design: Overview



RR CC

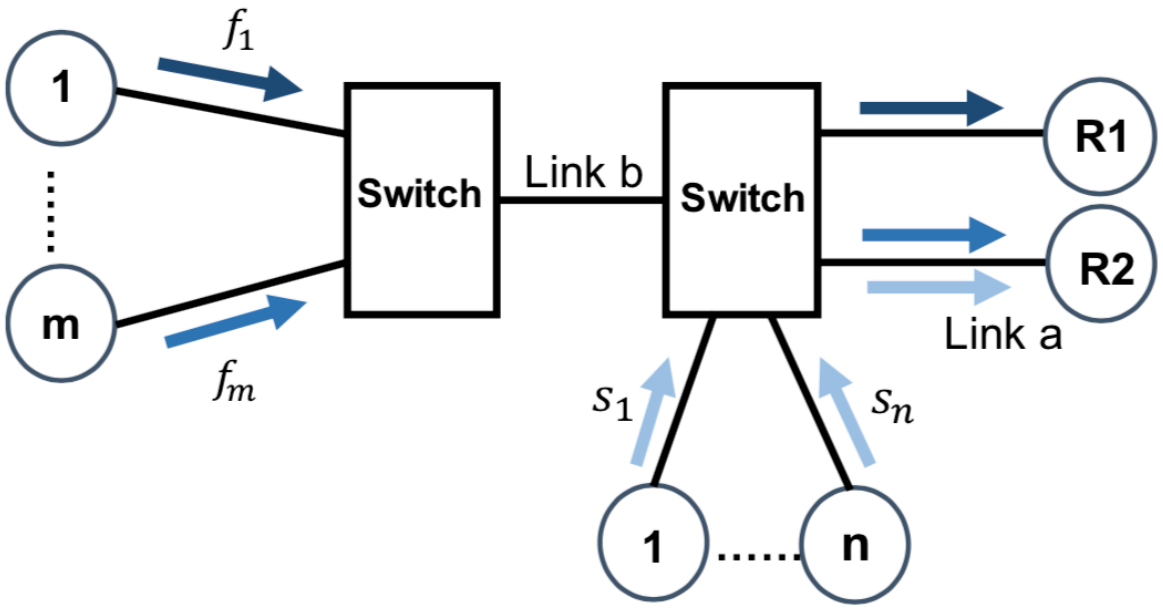
Key idea (1): Receiver-Driven Congestion Identification

- **The receiver-side has the packets marking pattern to guide congestion identification:**
  - Congested flows: *all* packets passing through the congestion root and are marked with FECN
  - Victim flows: only *fractional* packets have available credits to send and may be marked with FECN.

Key idea (2): Receiver-Driven Rate Regulation

- **The receiver-side has the *receiving rate* information to guide rate regulation**:

  The *receiving rate* is the maximum rate that the congested flow can pass through the bottleneck link.



- Endpoint congestion
- In-network congestion
- Coexisting multiple congestion points

Design: Receiver – Congestion Identification and Notification

Algorithm 1 Receive side CA behavior

```
1:  // for each Queue Pair or Service Level
2:  while during each period T do
3:     if received packet with FECN bit set then
4:        fecn_count ← fecn_count + 1
5:        recv_count ← recv_count + 1
6:     else
7:        recv_count ← recv_count + 1
8:     end if
9:  end while
10: if period T expires then
11:    P_FECN ← fecn_count/recv_count
12:    receive_rate ← recv_count ∗ MTU/T
13:    recv_count ← 0
14:    fecn_count ← 0
15:    if P_FECN ≥ 0.95 then
16:       Send CNP with BECN bit set and receive_rate
17:    else
18:       if receive_rate < line_rate then
19:          Send CNP without BECN bit set
20:       end if
21:    end if
22: end if
```

Receiving rate calculation

Generating Congestion Notification Packets to guide rate increase/decrease explicitly (BECN bit)

Design: Sender – Rate Regulation

Algorithm 2 Send side CA behavior

```
1:  // for each Queue Pair or Service Level
2:  if received CNP with BECN bit set then
3:      target_rate ← current_rate
4:      current_rate ← receive_rate ∗ d
5:  else
6:      if consecutive 3 CNPs without BECN bit set then
7:          target_rate ← target_rate + R_AI
8:          current_rate ← (current_rate + target_rate)/2
9:      else
10:         current_rate ← (current_rate + target_rate)/2
11:     end if
12: end if
```

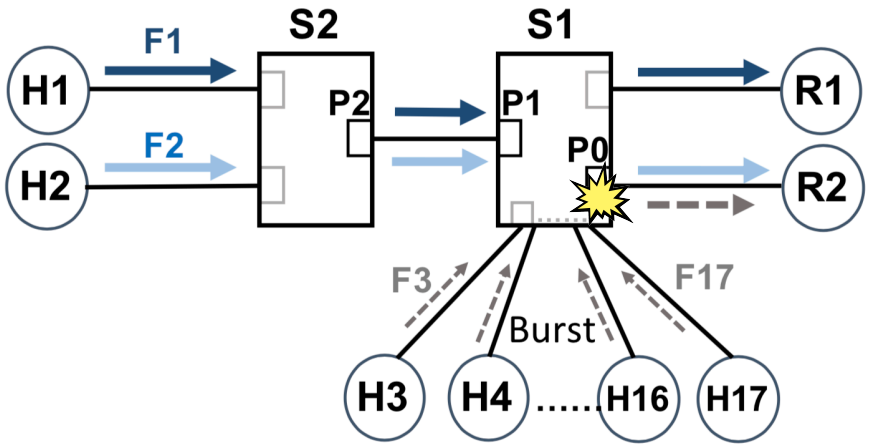
- Simple yet effective rate adjustment rules for congested flows and uncongested/victim flows:

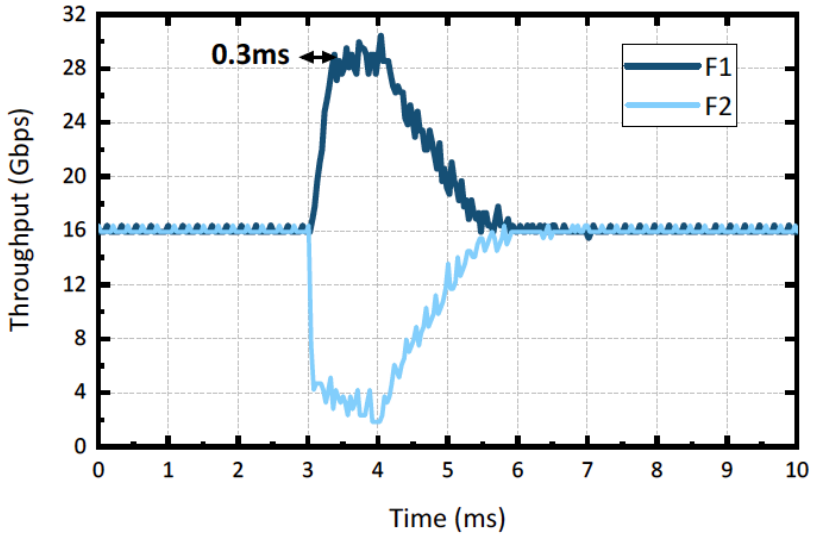
→ Rate decrease according to piggybacked receiving rate

→ Rate increase following Averaging Principle

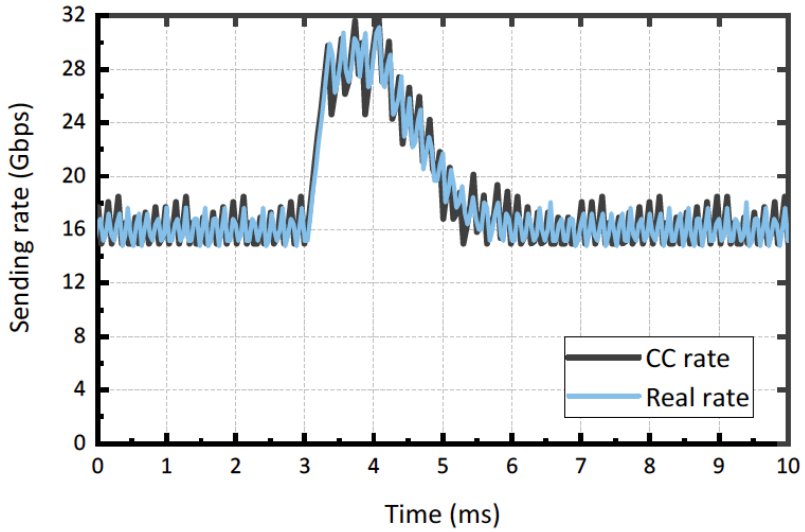
Evaluation

- Validation in the typical scenario (1)

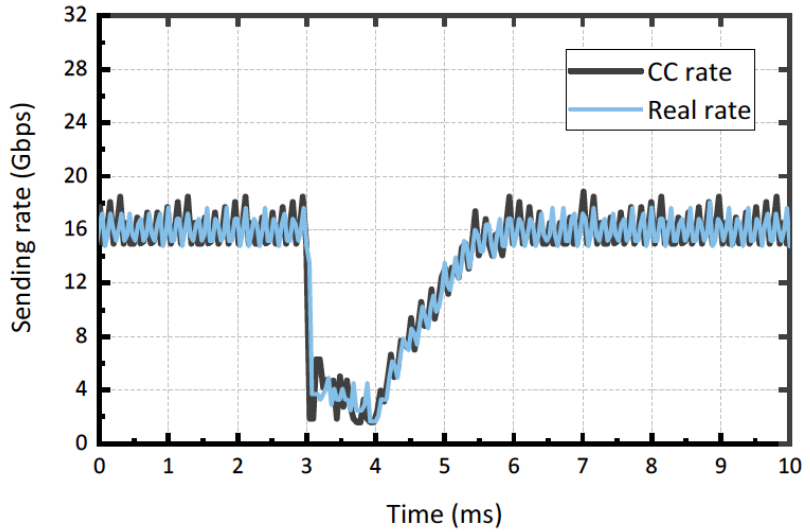




RR CC: overall throughput



RR CC: F1 rate

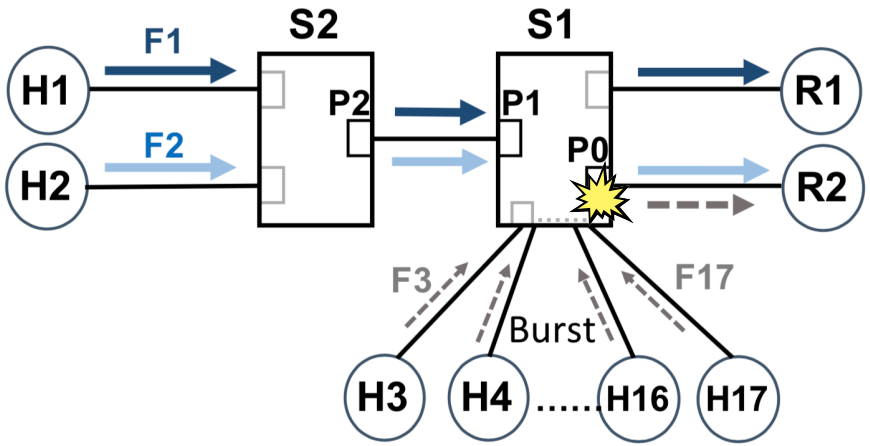


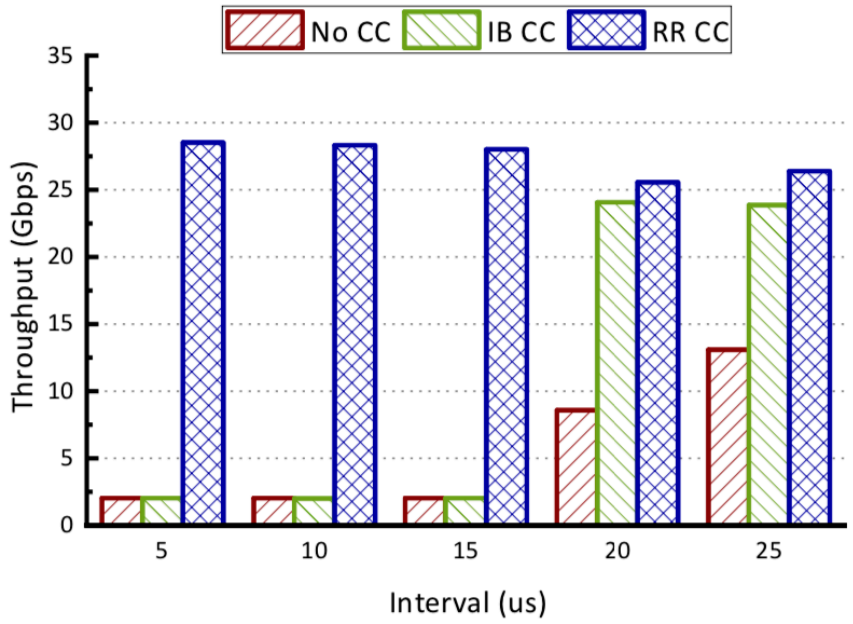
RR CC: F2 rate

Fast congestion elimination & short convergence time

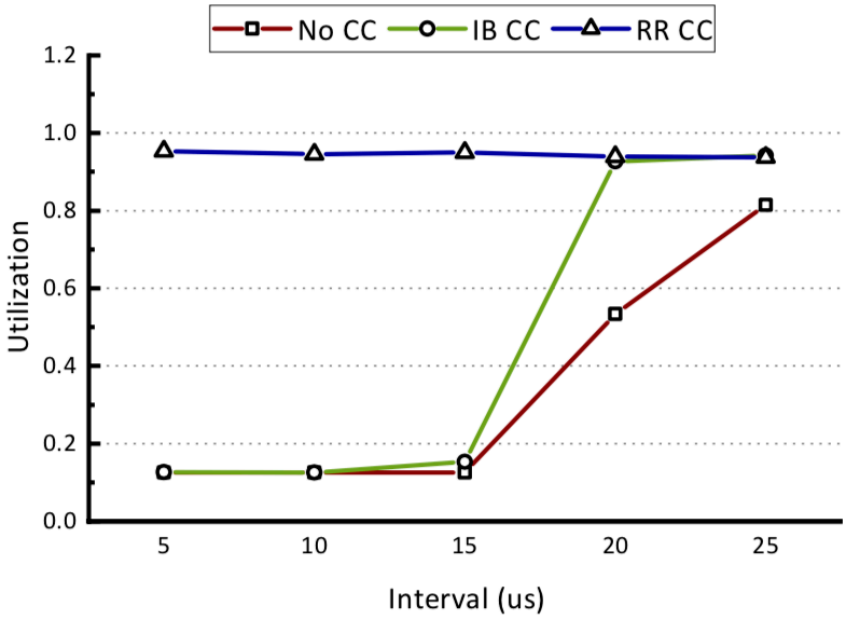
Evaluation

- Validation in the typical scenario (2)





(a) F1 average throughput

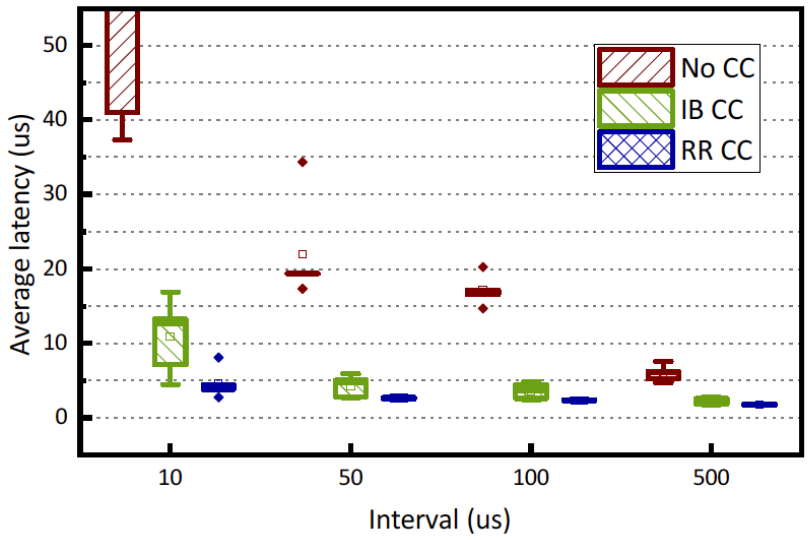


(b) Link P2-P1 utilization

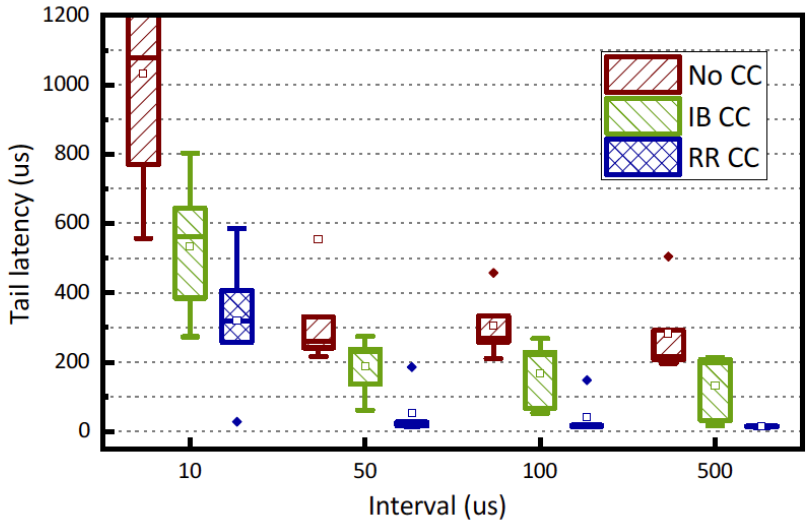
Evaluation

- Large-scale simluations

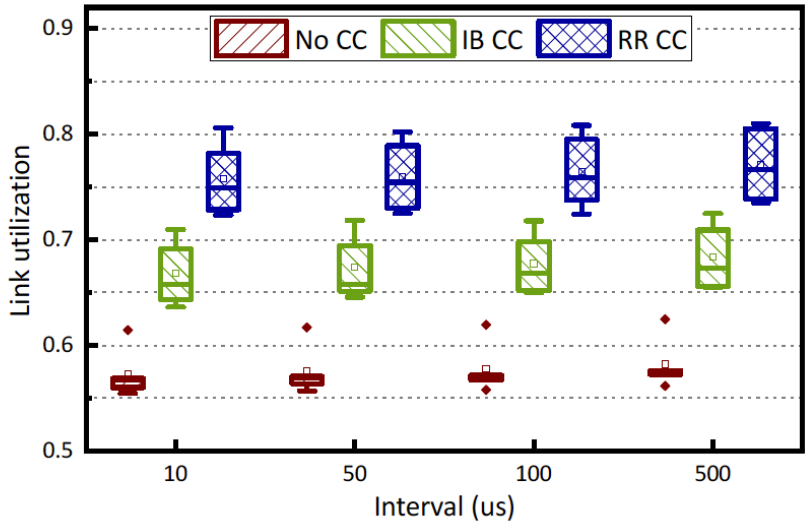
  1024-node Fat-tree (QDR rate), mixed MPI and I/O messages generated by MPI servers and I/O servers.



(a) Average latency of MPI messages



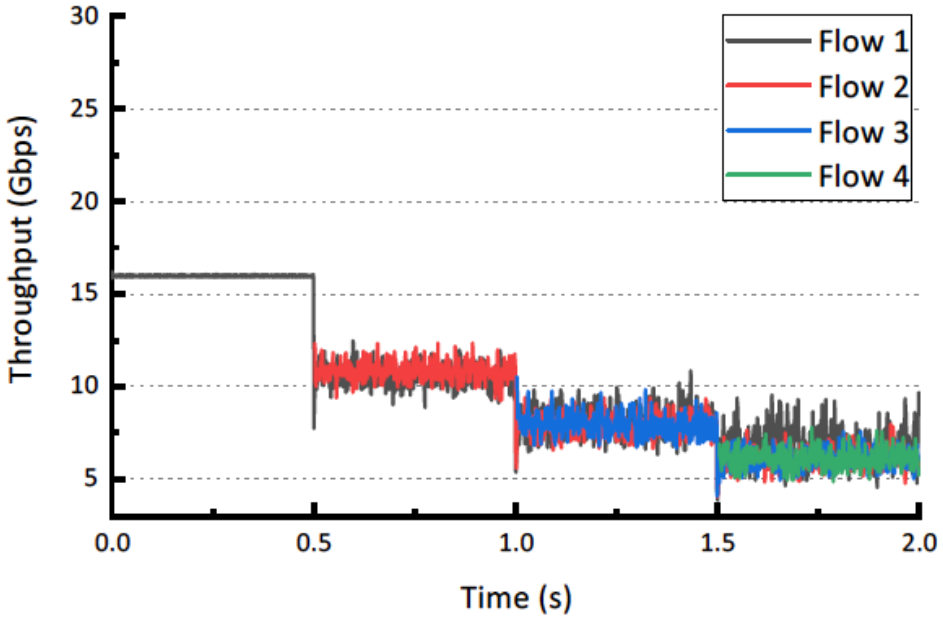
(b) Tail latency of MPI messages



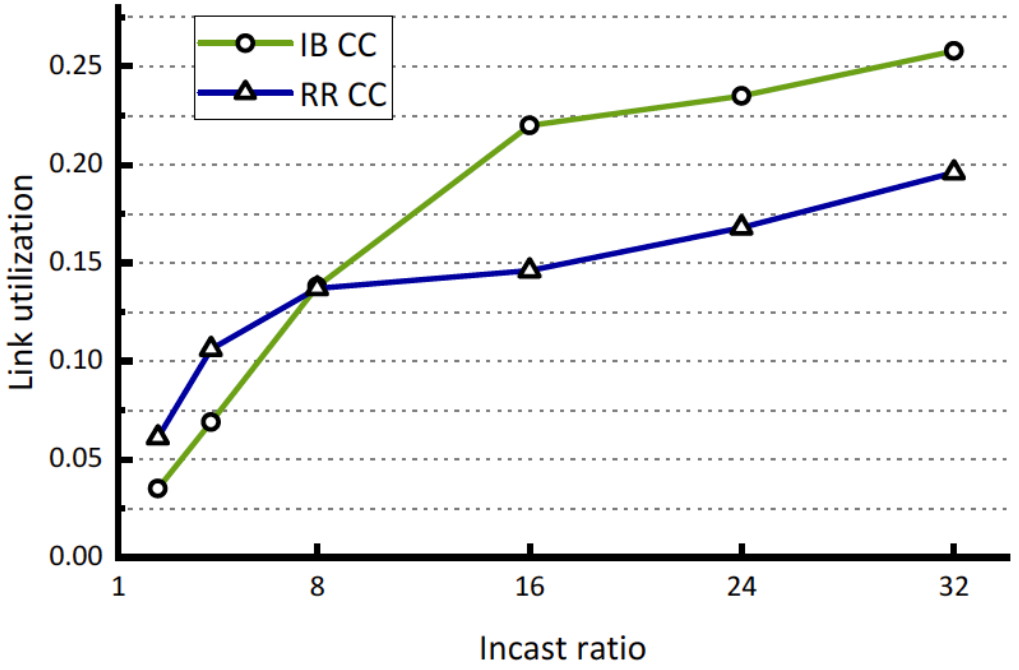
(c) Link utilization

Evaluation

- Other properties



Fairness



Overhead of CNP

Conclusion

- Investigating IB CC performance under typical mixed traffic scenario.
- Proposing RR CC for InfiniBand network, with much better average messages latency, fewer parameter tuning and compatible with commodity InfiniBand switches.

Thank you!

zyr17@mails.tsinghua.edu.cn