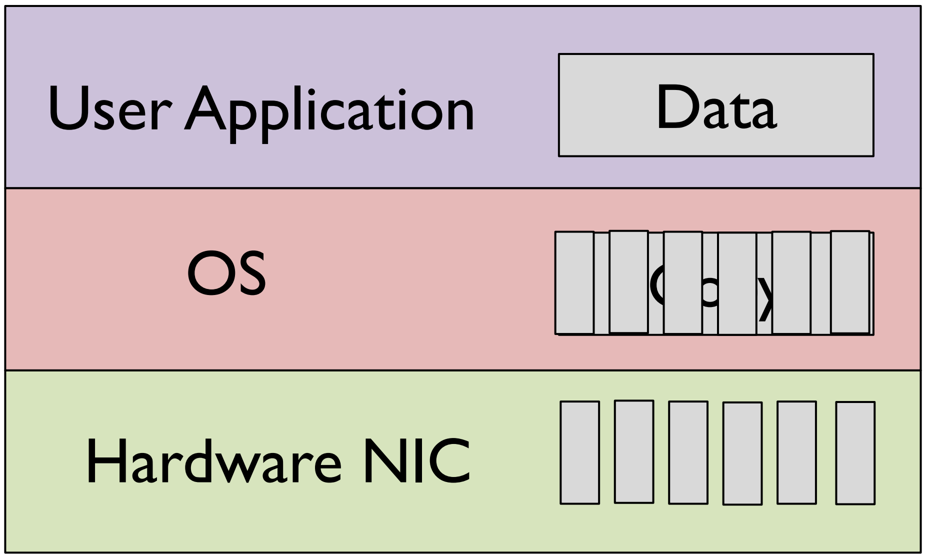
RDMA

ECE/CS598HPN

Radhika Mittal

Traditional Network Stack

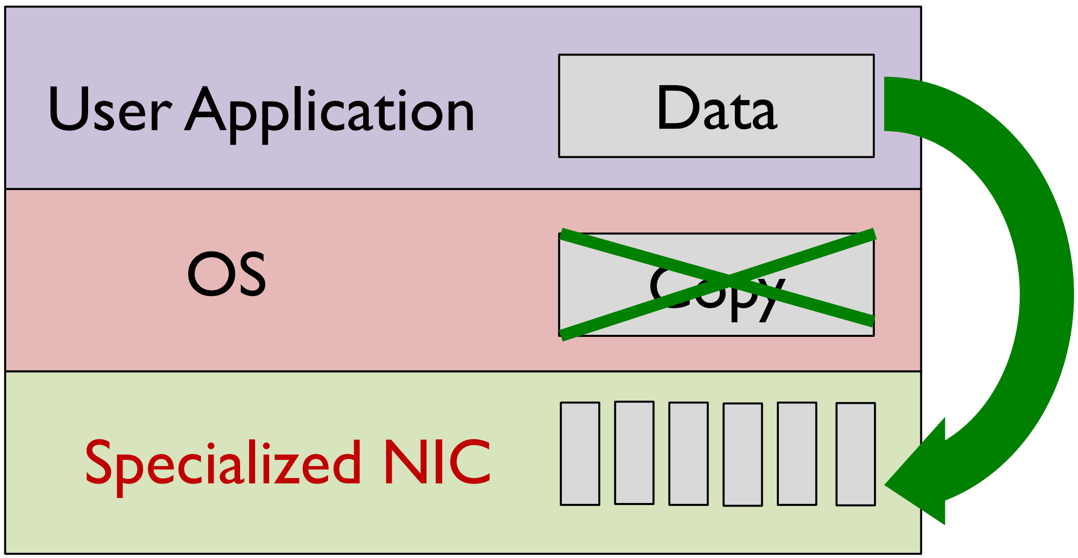


Packet processing in OS incurs high latency, cannot support high throughput, and leads to high CPU utilization.

Not acceptable in today's datacenters:

- few microseconds of latency
- tens to hundred Gbps bandwidth
- cpu = $$$

Remote Direct Memory Access

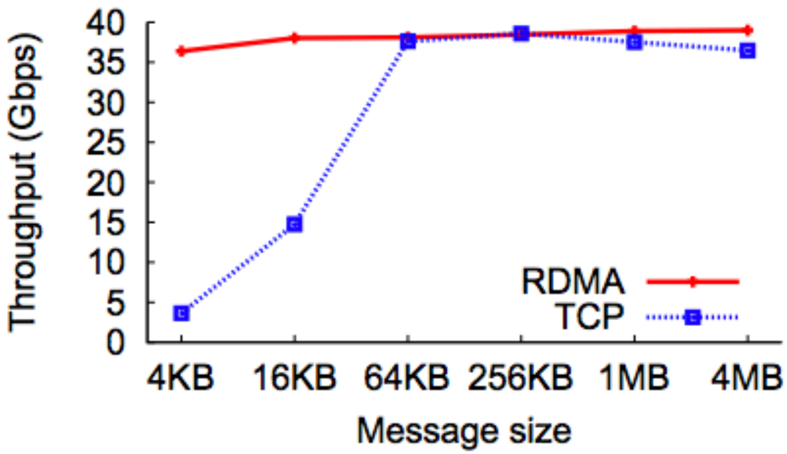


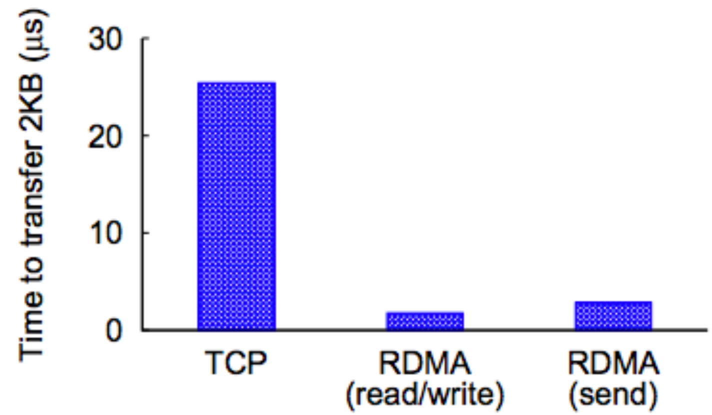
- Protocol offload
- Zero-copy
- OS by-pass

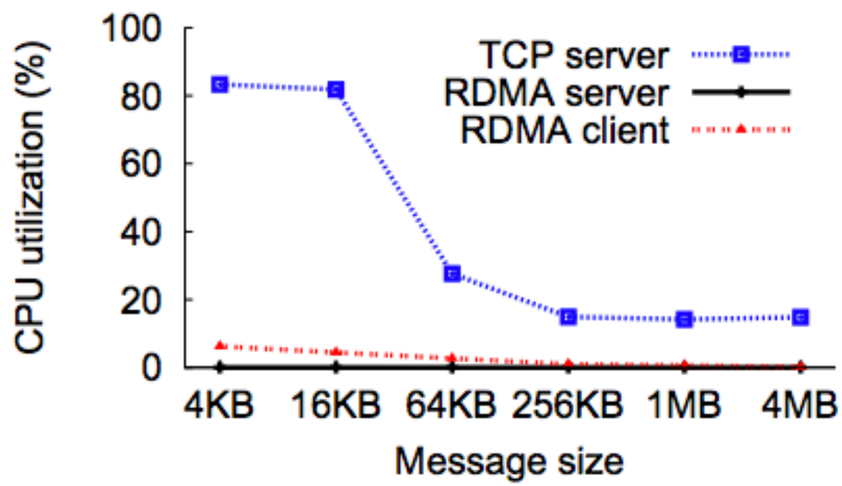
Traditionally used in Infiniband clusters for HPC.

Achieves low latency, high throughput and negligible CPU utilization.

Performance Benefits of RDMA







From *"Congestion Control for Large-Scale RDMA Deployments"*, Zhu et. al., SIGCOMM 2015

RDMA usecases in datacenters

- Distributed storage:
  - Distributed key-value stores
    - *Pilaf (ATC'13), FaRM (NSDI'14, SOSP'15), HERD (SIGCOMM'14), FASST(OSDI'16),…*
  - Distributed file systems
  - NVMe over Fabric

- Applications requiring low latency
  - Search queries, ML applications

- Other proposals
  - Resource disaggregation *(OSDI'16)*, Remote swapping *(NSDI'17)*, …

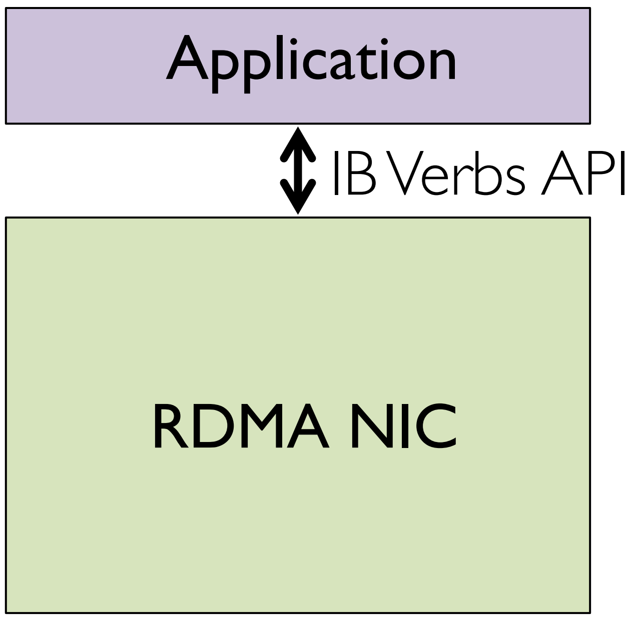
Focus of today's lecture

- Overview of RDMA
- RDMA deployment in today's datacenters

Focus of today's lecture

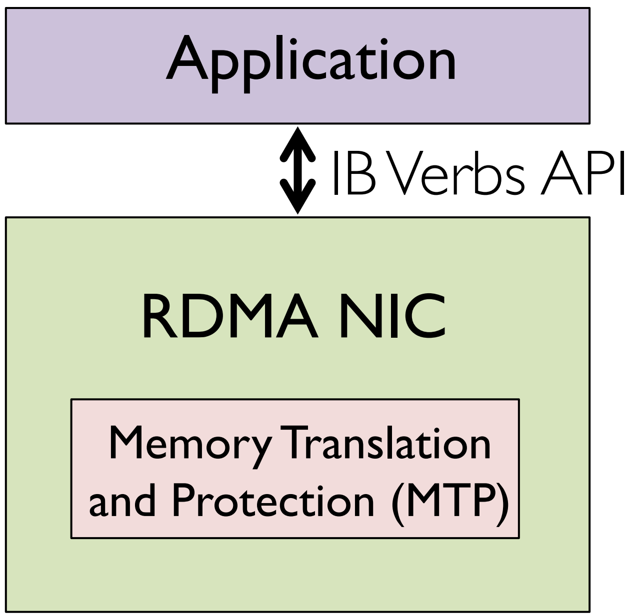
- Overview of RDMA
- RDMA deployment in today's datacenters

RDMA Overview and Components



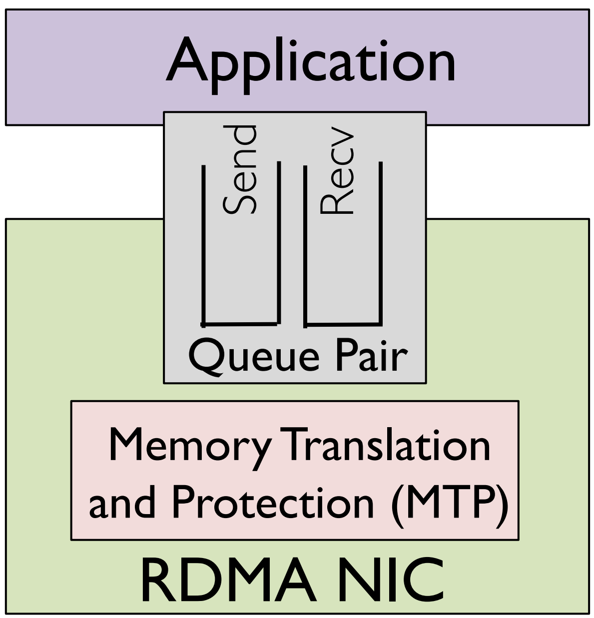
Applications bypass the kernel and interact directly with the RDMA NIC using the **IB verbs** API provided by the NIC driver.

Memory Translation and Protection



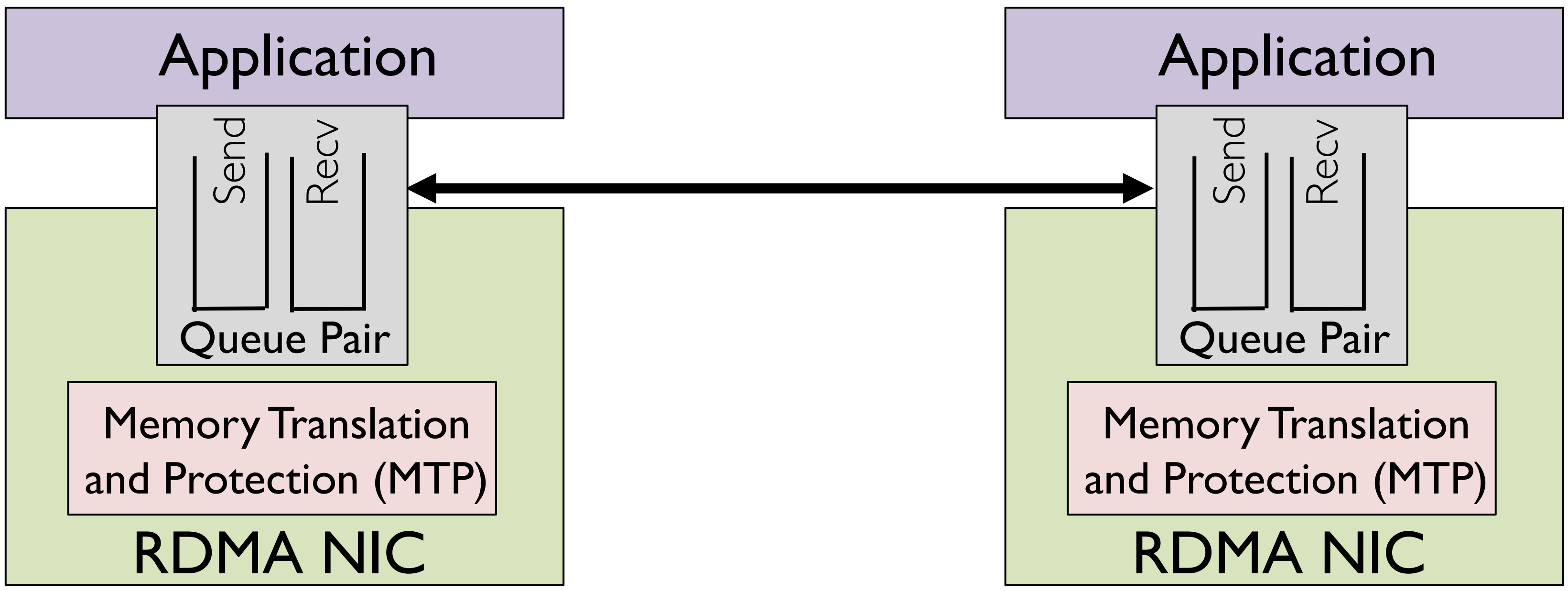
- Applications register *memory regions* with the NIC.
- **Translation:** MTP maintains *virtual address* to *physical address* mapping.
- **Protection:** MTP assigns local and remote access keys to memory region.

Queue Pairs (QP)



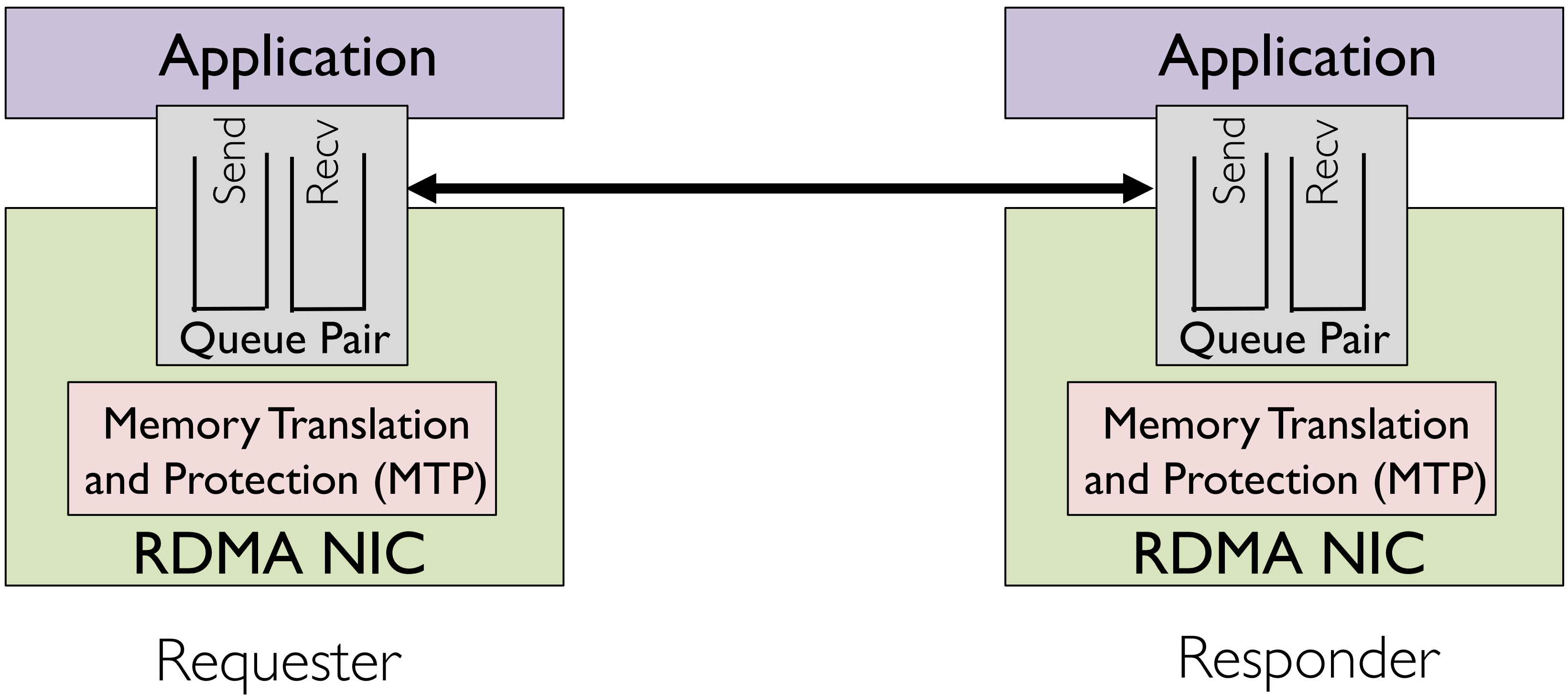
- QPs are interfaces between the application and the NIC.
- Different types:
  - Connection-oriented vs Datagram
  - Reliable vs unreliable.
- Reliable Connected (RC) QPs
  - Analogous to a TCP connection.
  - Support all types of operations.

Connection Establishment

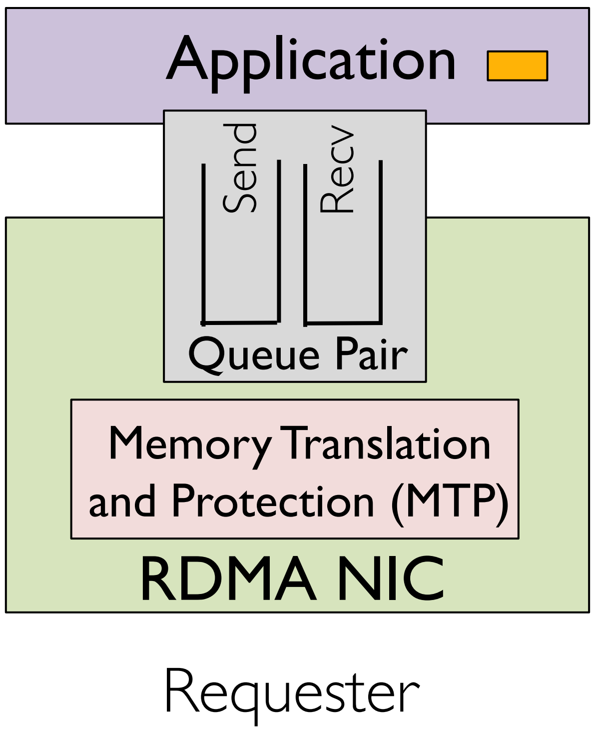


Connection establishment requires out-of-band exchange of node identifiers, QP id, and remote keys.

Work Requests

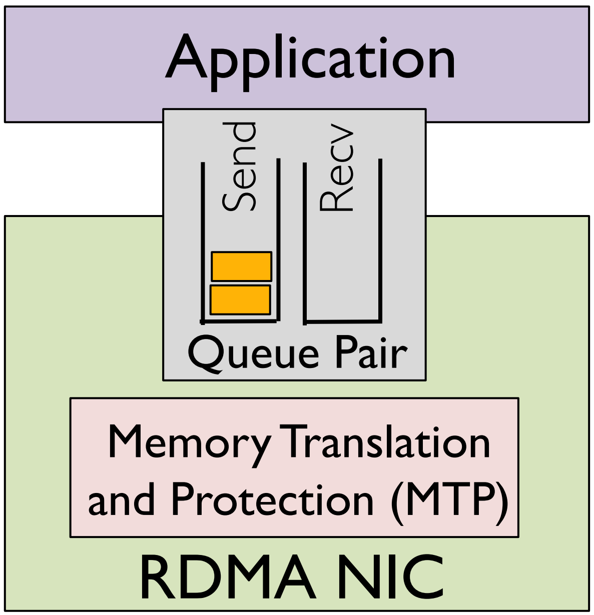


Work Requests



- Application issues a *work request* (WR) for a QP.
- WR contains all the metadata associated with a message transfer.

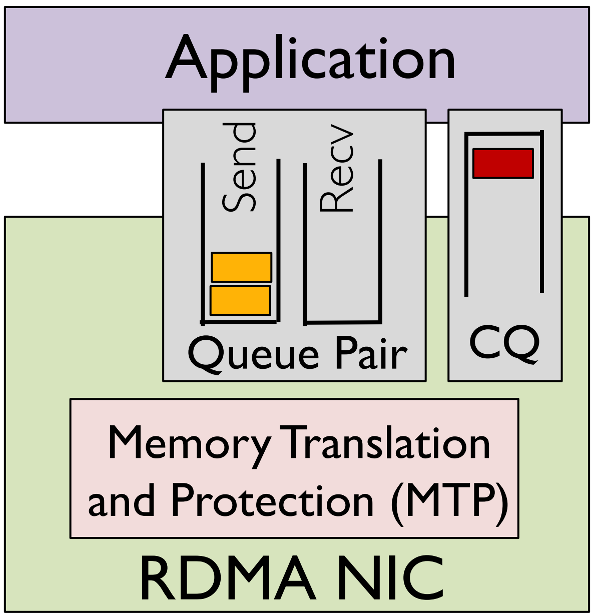
Work Queue Element (WQE)



Requester

- This WR gets stored as a *Work Queue Element (WQE)* at the QP's send queue.
- Multiple WQEs can get queued up in the send queue.
- RDMA NIC processes these WQEs one after another.

Completion Queue Element (CQE)



Requester

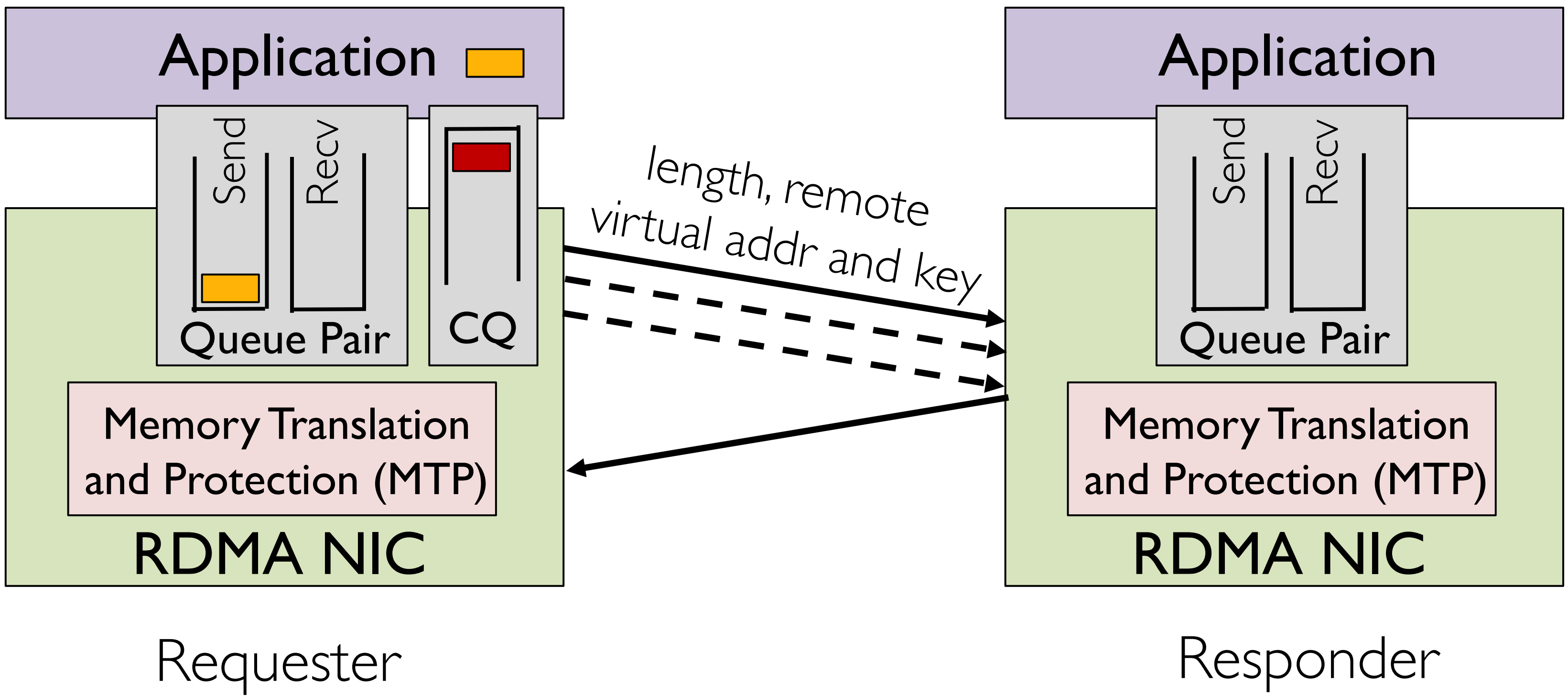
- Each QP is associated with a completion queue (CQ).
- Upon request completion,
  - The WQE expires.
  - CQE is created.
- CQE notifies request completion to application.

Four Types of RDMA Operations

- **RDMA Write:** Write data from local node to specified address at remote node.
- **RDMA Read:** Read data from specified address at remote node to local node.
- **RDMA Atomic:** Atomic fetch-add and compare-swap operations at specified location at remote node.
- **Send/Receive:** Send data to a remote node.

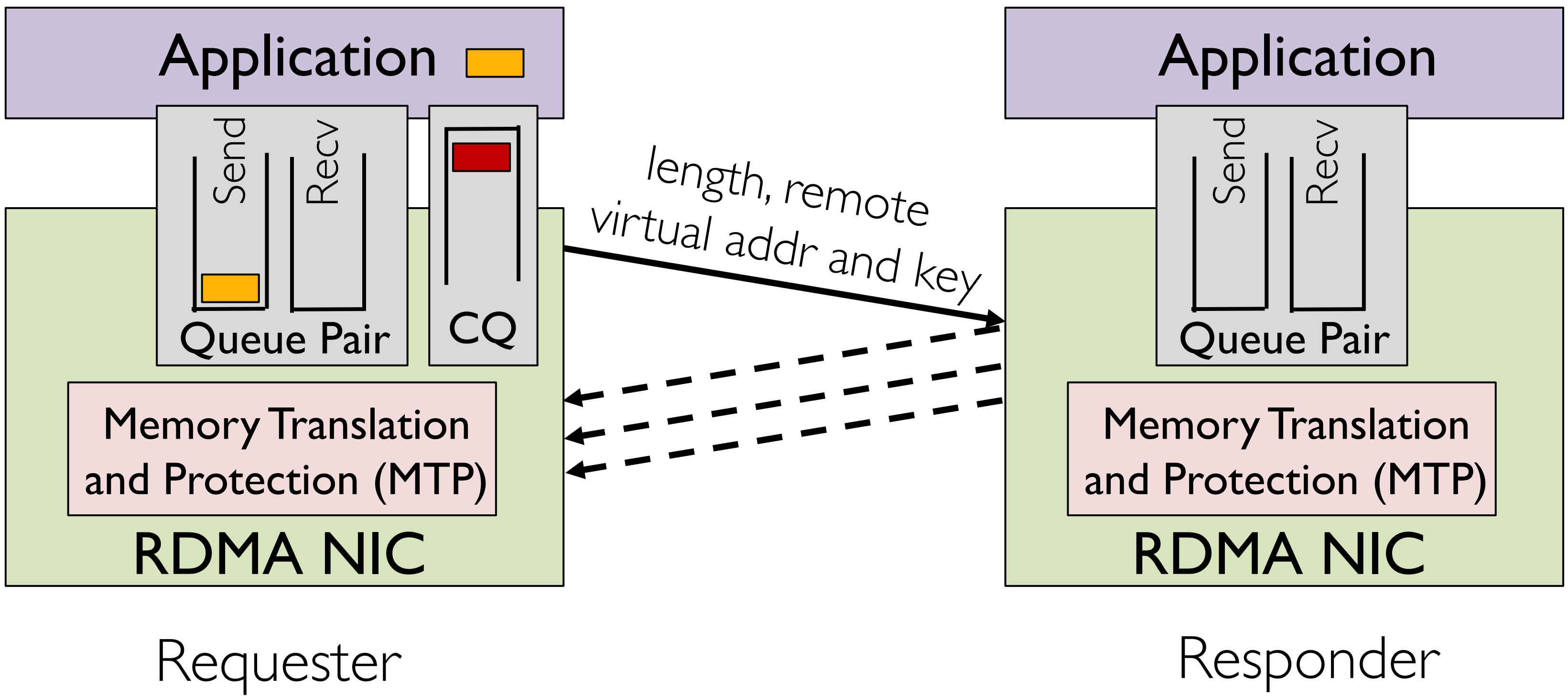
RDMA Write

WR/WQE metadata: local source virtual addr, local key, data length, remote sink virtual addr, remote key



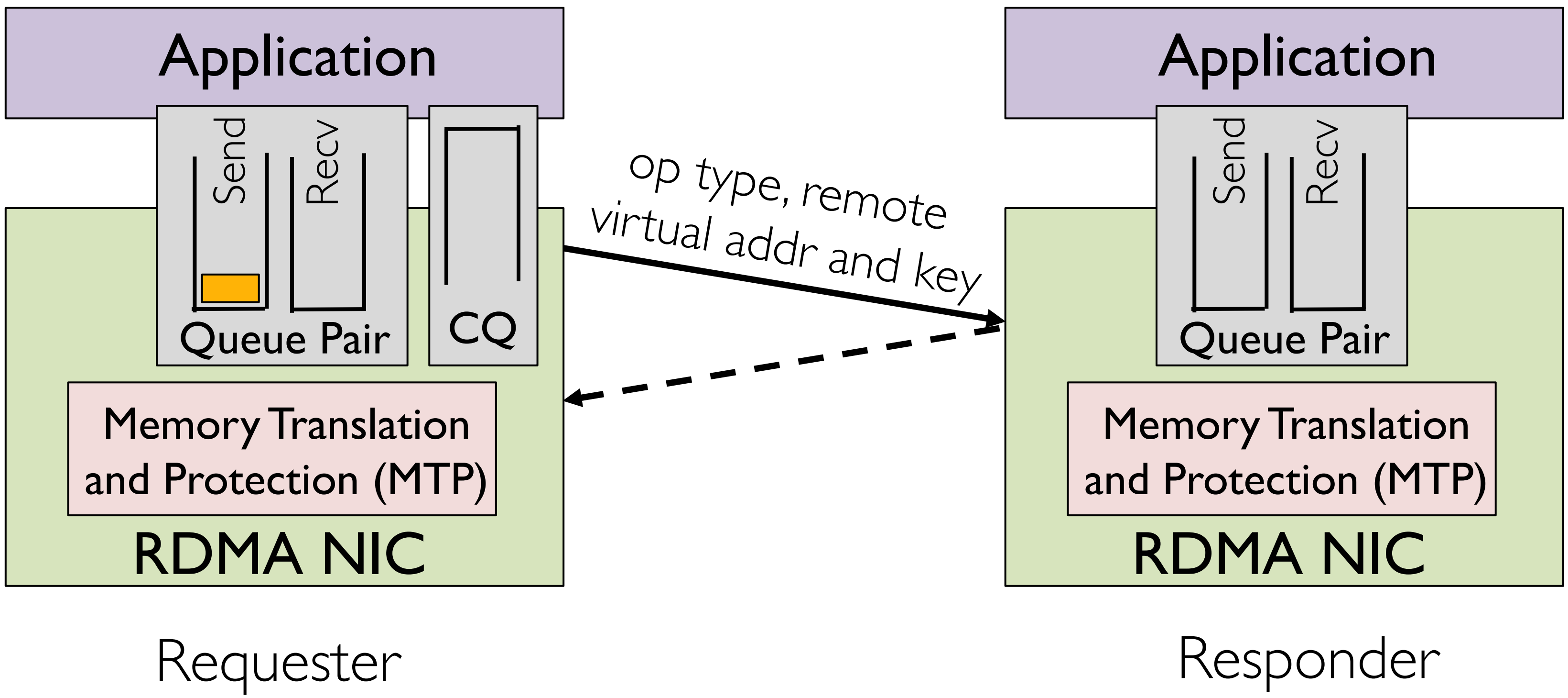
RDMA Read

WR/WQE metadata: *local sink* virtual addr, local key, data length, *remote source* virtual addr, remote key



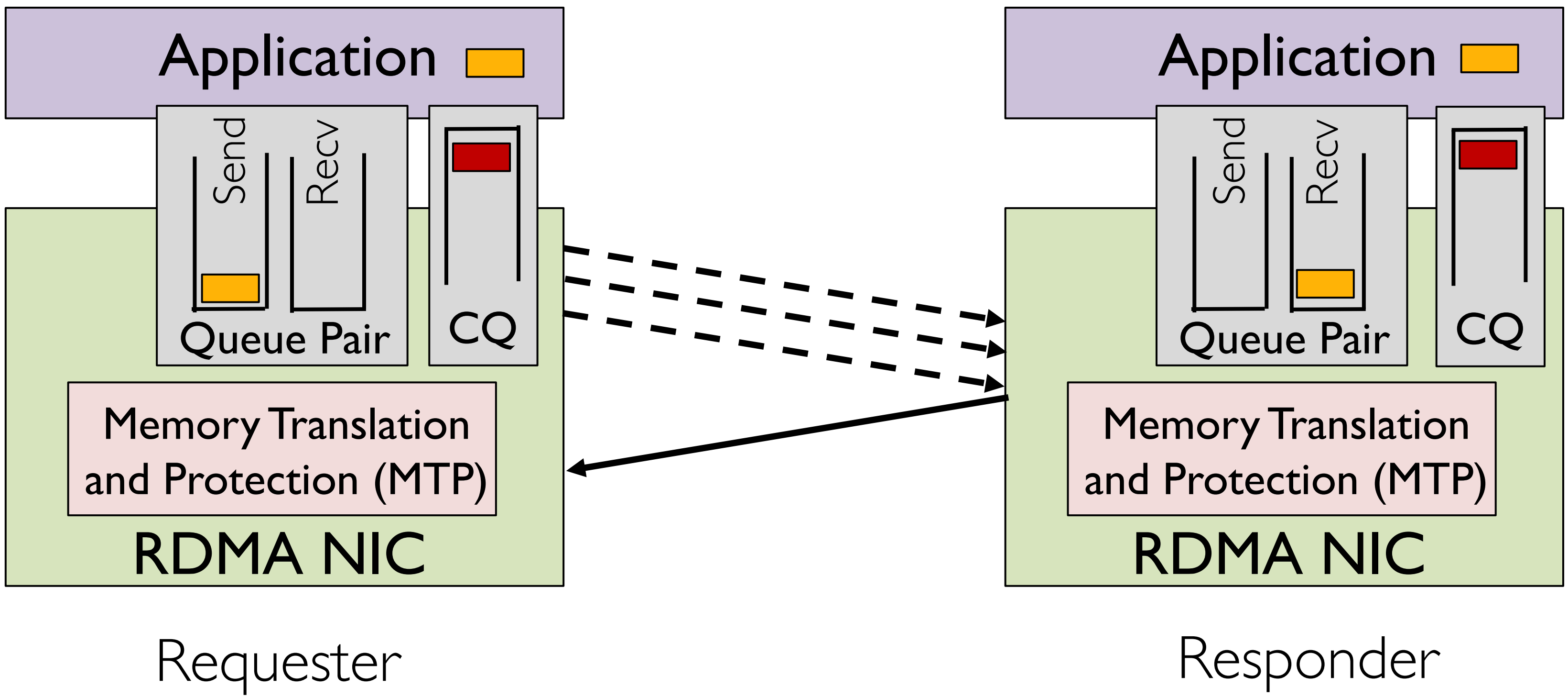
RDMA Atomic

WR/WQE metadata: local sink virtual addr, local key, *atomic operation*, remote source virtual addr, remote key



RDMA Send and Receive

Send WQE metadata: local source virtual addr, local key, data length.
Receive WQE metadata: local sink virtual addr



Four Types of RDMA Operations

- **RDMA Write:** Write data from local node to specified address at remote node.
- **RDMA Read:** Read data from specified address at remote node to local node.
- **RDMA Atomic:** Atomic fetch-add and compare-swap operations at specified location at remote node.
- **Send/Receive:** Send data to a remote node.

Lower layers for RDMA

- Traditionally designed for Infiniband.
  - Own set of networks protocols and addressing.
- RDMA over Converged Ethernet (RoCE)
  - Allows running RDMA over Ethernet.
- RoCEv2
  - Allows running RDMA over IP.

Focus of today's lecture

- Overview of RDMA
- RDMA deployment in today's datacenters

Two papers today

- FaRM: Fast Remote Memory
  - Systems challenges in deploying RDMA
- Revisiting Network Support for RDMA
  - Neworking challenges in deploying RDMA

FaRM: Fast Remote Memory

Aleksandar Dragojevic, Dushyanth Narayanan,
Orion Hodson, Miguel Castro

NSDI'14

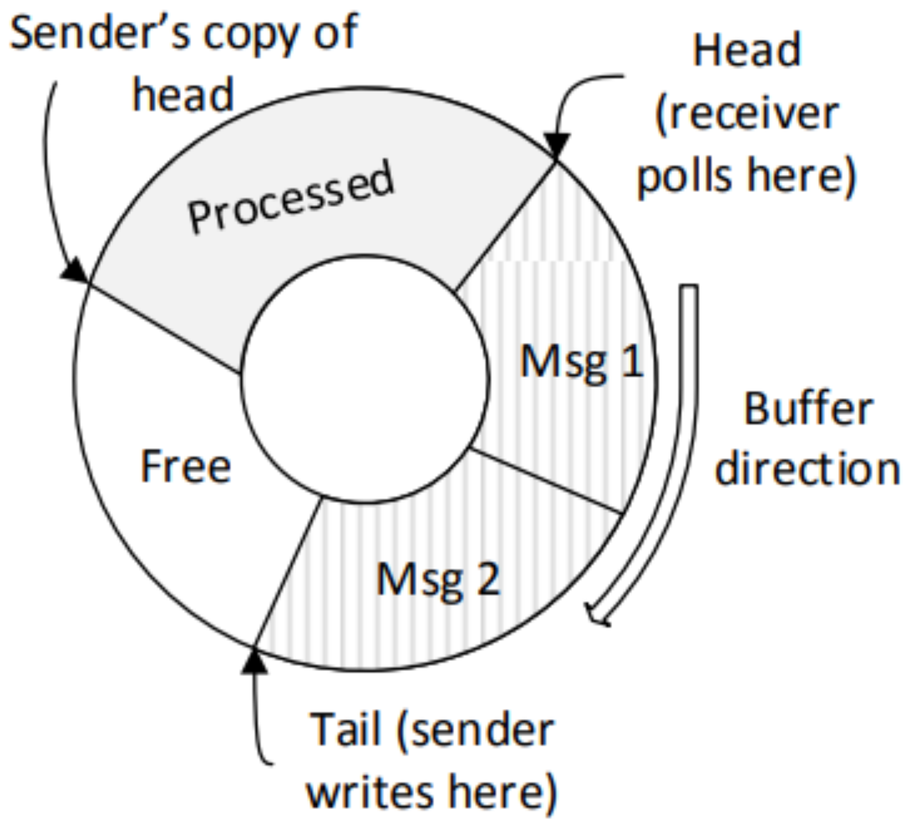
FaRM: Fast Remote Memory

- Distributed computing platform built over RDMA.
  - Distributed key-value store
  - Distributed graph store

Communication Primitives

- RDMA reads to access data directly.
- RDMA writes to implement a fast message passing primitive.

Circular Buffer for RDMA messaging



Performance

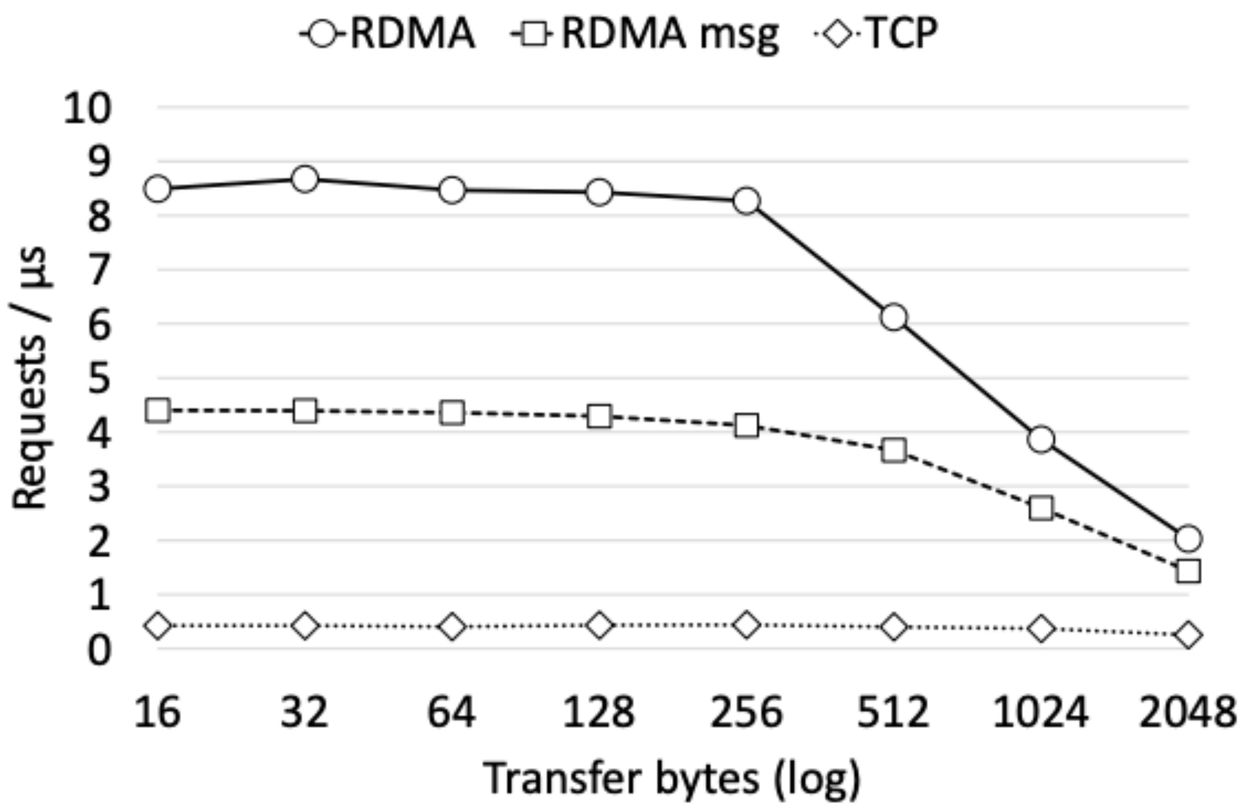


Figure 2: Random reads: request rate per machine

Performance

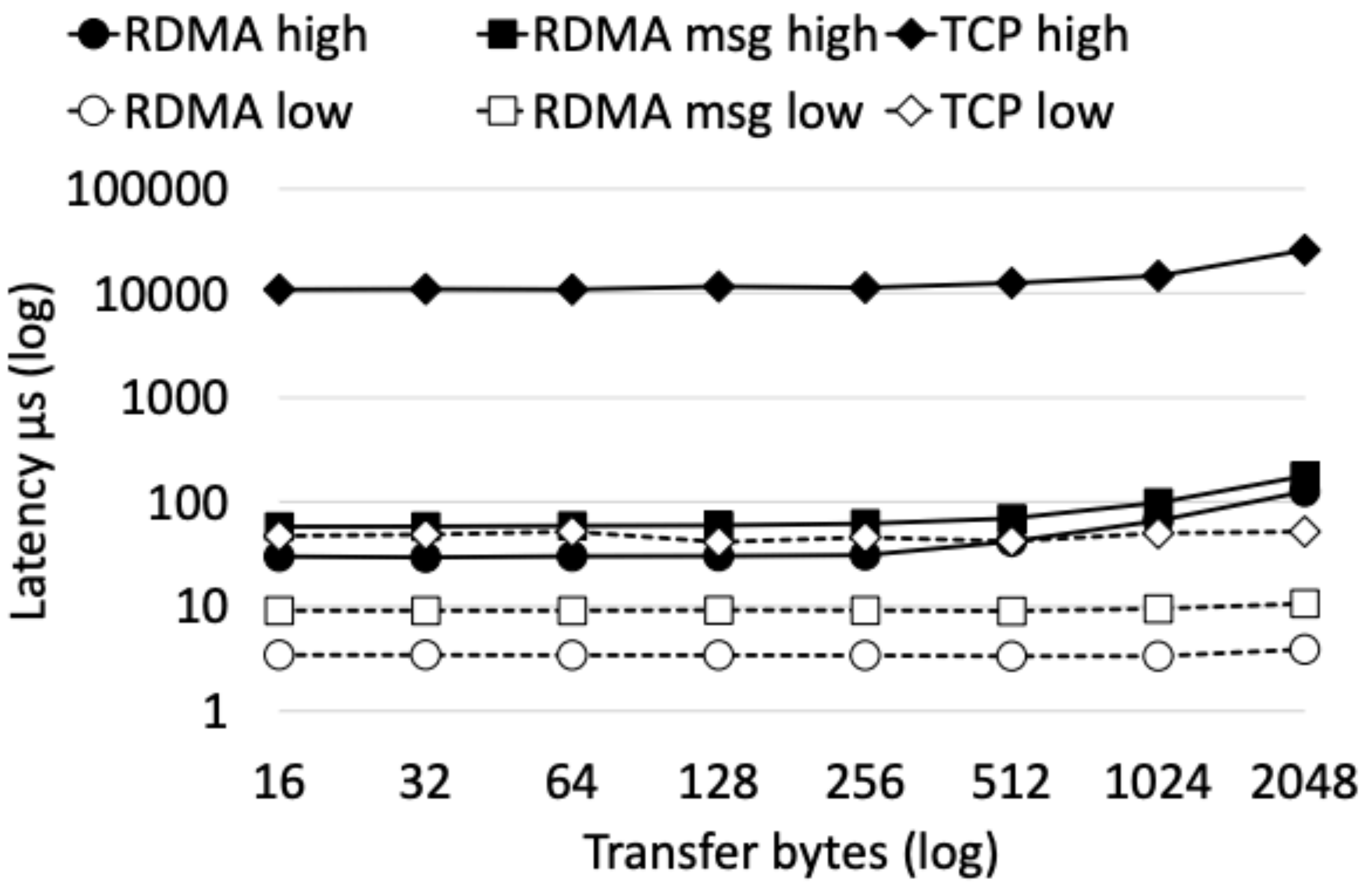


Figure 3: Random reads: latency with high and low load

Performance

"We did not get this performance out of the box. We improved performance by up to a factor of eight with careful tuning and changes to the operating system and the NIC driver."

Issues they needed to address

Issues they needed to address

- Performance decreased with amount of memory registered for remote access
  - More page table entries required.
  - Couldn't fit all entries in NIC cache.

- Solution: use large pages (2GB).
  - Implemented a kernel driver.
  - Unit of address mapping, of recovery, and of registration with memory.

Issues they needed to address

- Performance decreased as cluster size increased.
  - Larger number of QPs required.

Issues they needed to address

- Performance decreased as cluster size increased.
  - Larger number of QPs required.
    - Ideally, $2 \times m \times t^2$
    - Reduced to $2 \times m \times t$
    - m = no. of machines, t = no. of threads/machine
  - Couldn't fit all QP context in NIC cache.
- Solution: use fewer QPs ($2mt / q$)
  - Larger 'q', higher sharing overhead.

Issues they needed to address

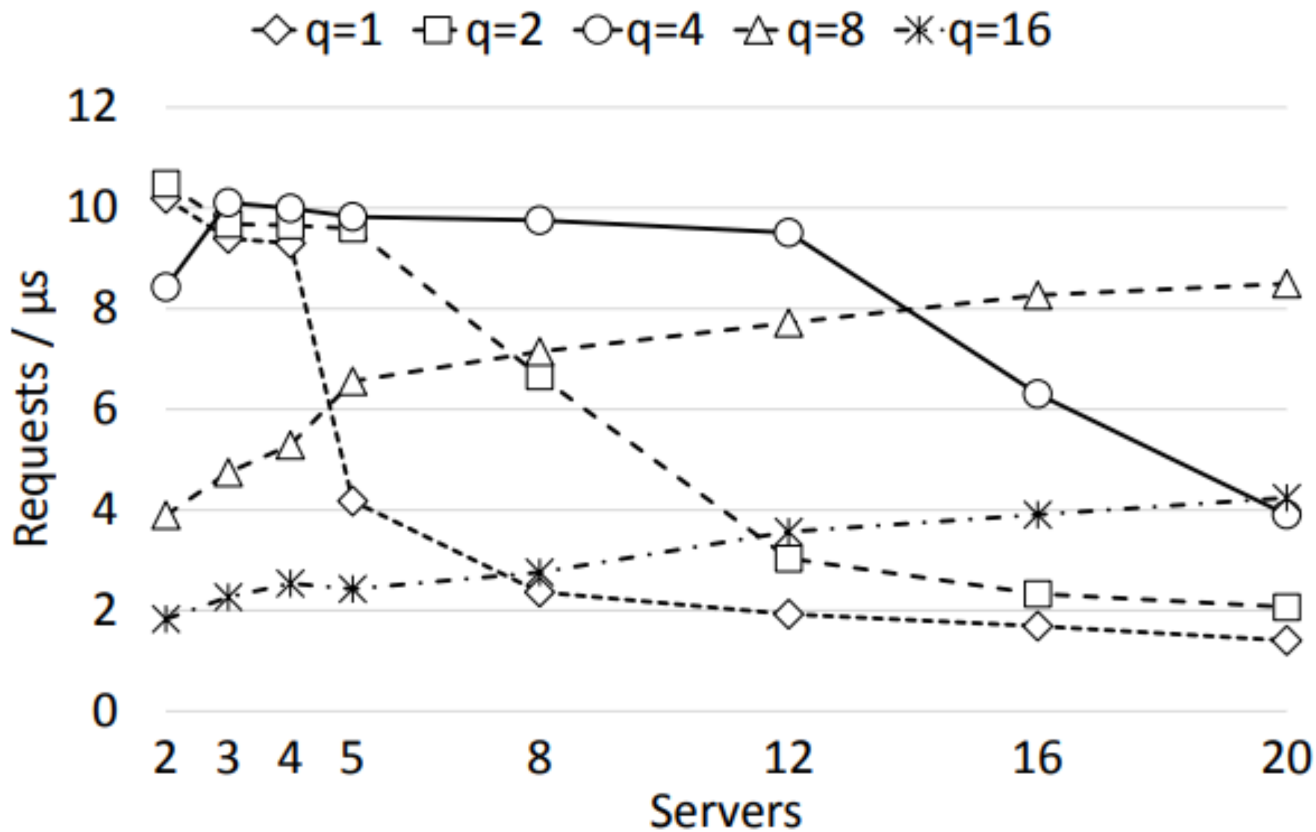


Figure 5: Impact of connection multiplexing

Other aspects

- Distributed memory management
  - Consistent hashing to map region ids to machines.
- Programming model and architecture
  - Favors local object access.
- Transactions use optimistic concurrency control and two-phase commit.
- Lock-free reads
- Hashtable implementation
  - Minimize number and size of required RDMA operations

Your Opinions

- Pros:
  - High performance through using RDMA.
  - Employs many techniques to achieve the high performance.
    - Lock-free reads and support for collocating objects
    - Kernel driver (PhyCo) for large pages.
    - Neat hash table design

Your Opinions

- Cons:
  - Requires application modification
  - No comparison with user-space TCP stacks (e.g. mTCP).
  - Overhead of polling.
  - Drawbacks of large memory regions.

Your Opinions

- Ideas:
  - Comparison with mTCP and IX.
  - Test the system at other configurations (more replicas).
  - Replace two-phase commit with state machine replica.
  - A middleware that allows applications to run with no modifications.

Revisiting Network Support for RDMA

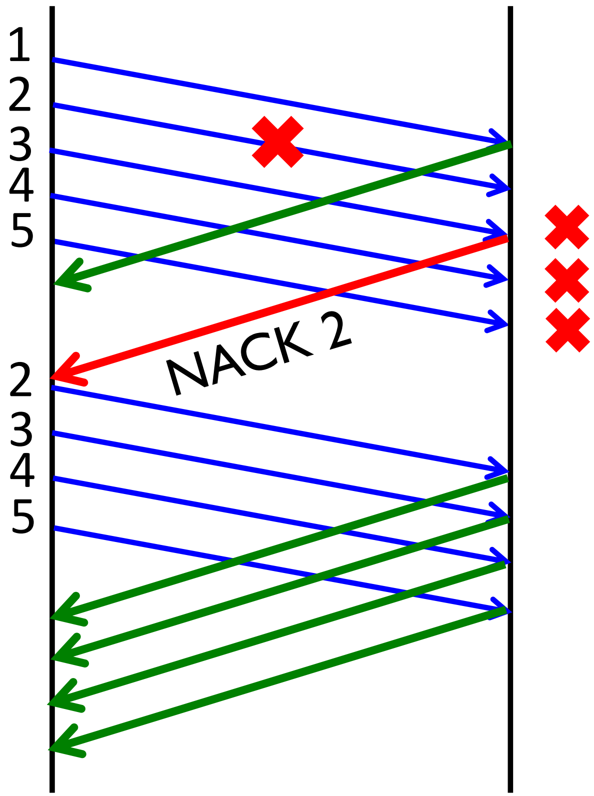
Radhika Mittal, Alex Shpiner, Aurojit Panda,
Eitan Zahavi, Arvind Krishnamurthy,
Sylvia Ratnasamy, Scott Shenker

SIGCOMM'18

Conventional RDMA

- RDMA traditionally used in Infiniband clusters.
  - A different network protocol supporting high bandwidth.
- Infiniband links use credit-based flow control.
  - Losses are rare.
- Transport layer in RDMA NICs not designed to deal with losses efficiently.
  - Receiver discards out-of-order packets.
  - Sender does *go-back-N* on detecting packet loss.

Go-back-N Loss Recovery



Receiver discards all out-of-order packets.

Sender retransmits all packets sent after the last acked packet.

Conventional RDMA

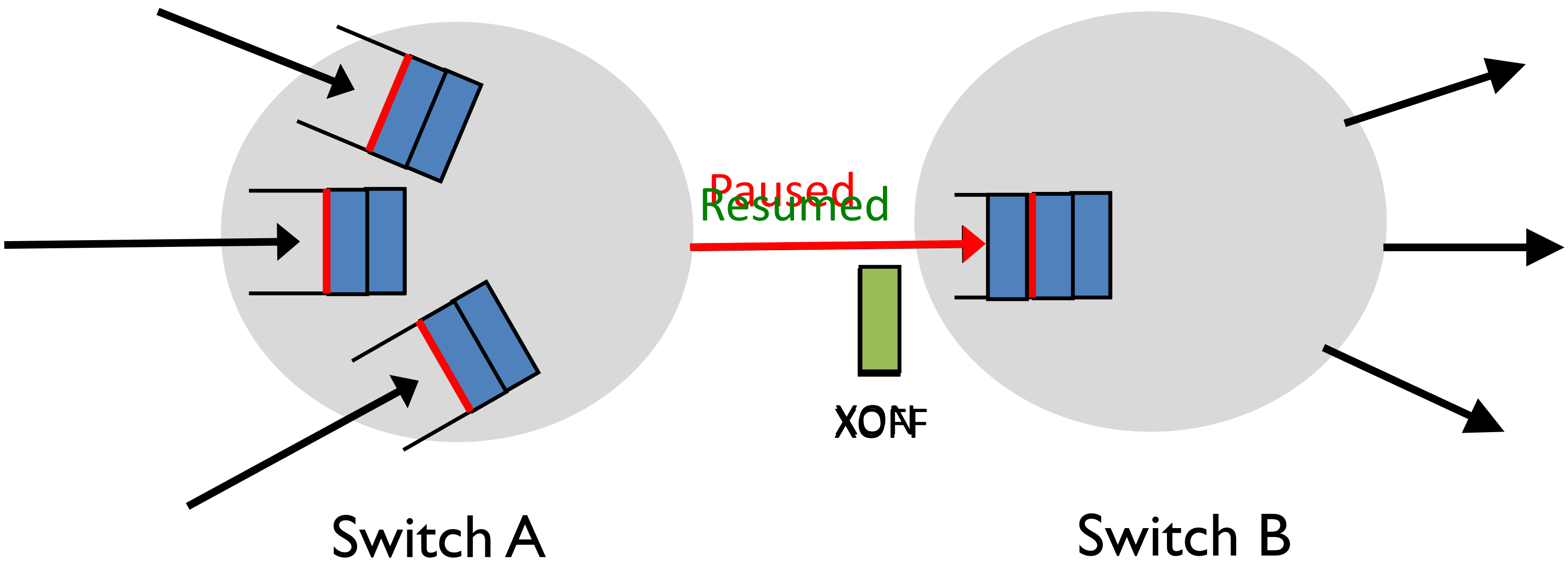
- RDMA traditionally used in Infiniband clusters.
  - A different network protocol supporting high bandwidth.
- Infiniband links use credit-based flow control.
  - Losses are rare.
- Transport layer in RDMA NICs not designed to deal with losses efficiently.
  - Receiver discards out-of-order packets.
  - Sender does *go-back-N* on detecting packet loss.

RDMA in datacenters

- Desire to run RDMA over commodity Ethernet.
- RoCE: RDMA over Ethernet fabric.
  - RoCEv2: RDMA over IP-routed networks.
- Infiniband transport was adopted as it is.
  - Go-back-N loss recovery.
  - Needs a lossless network for good performance.

Network made lossless by enabling PFC

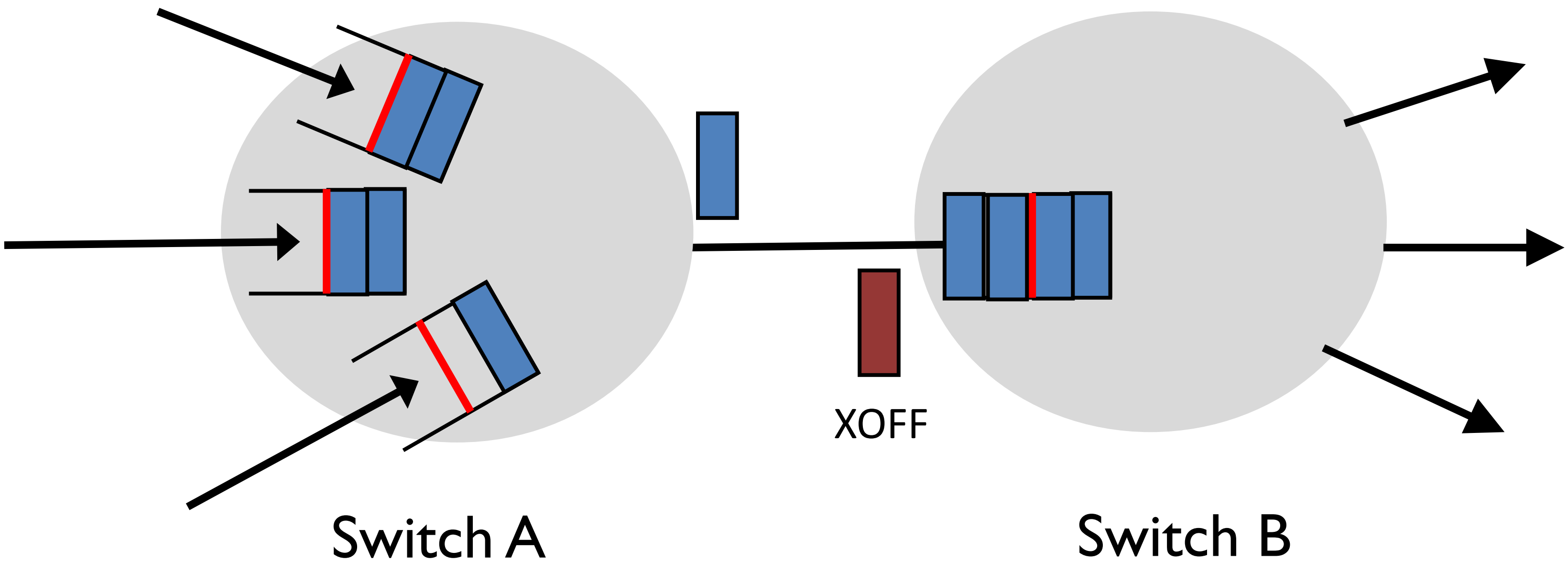
- Priority Flow Control (PFC)
  - Pause transmission when queuing exceeds a certain threshold.



Drawbacks of PFC

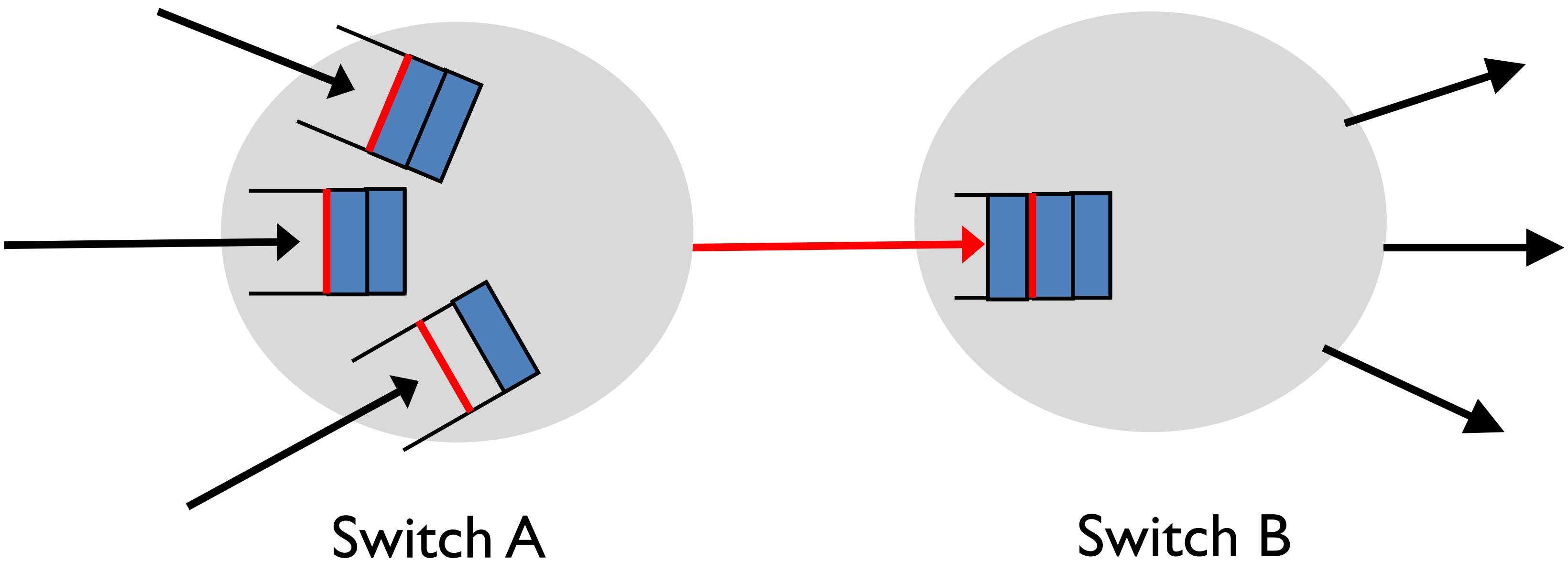
Complicates network management.

PFC threshold requires careful configuration.



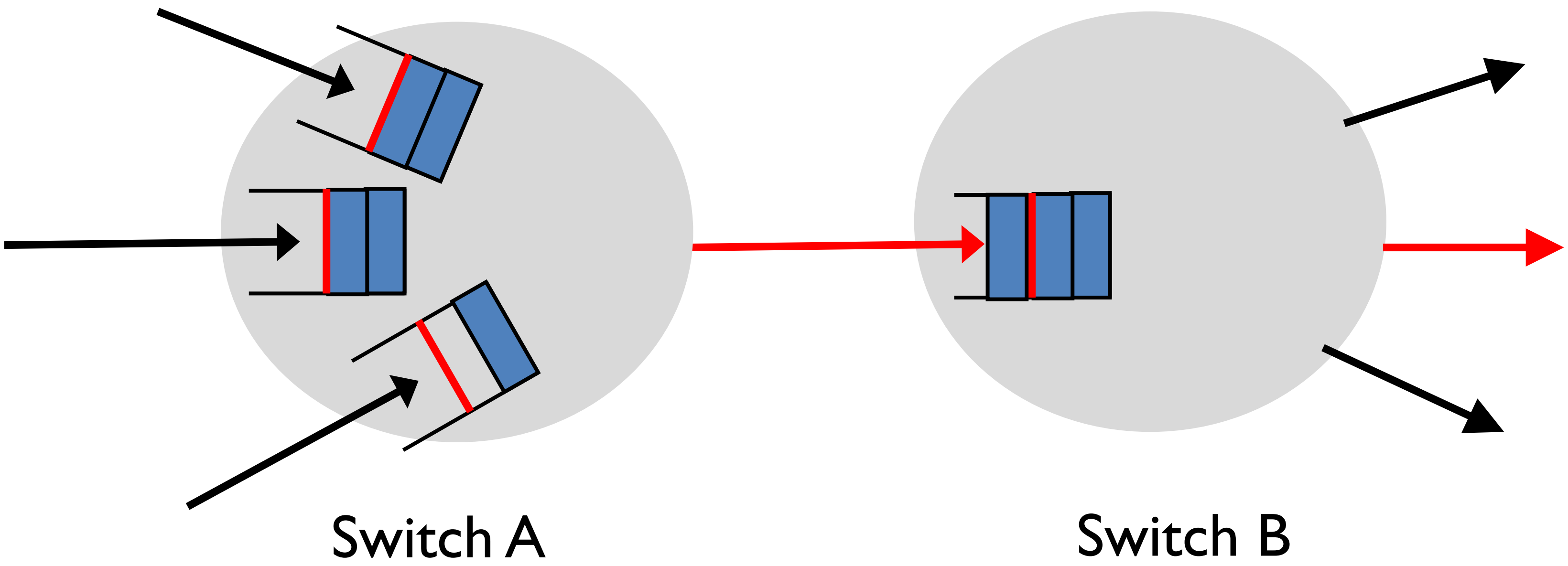
Drawbacks of PFC

Unfairness and Head-of-Line blocking



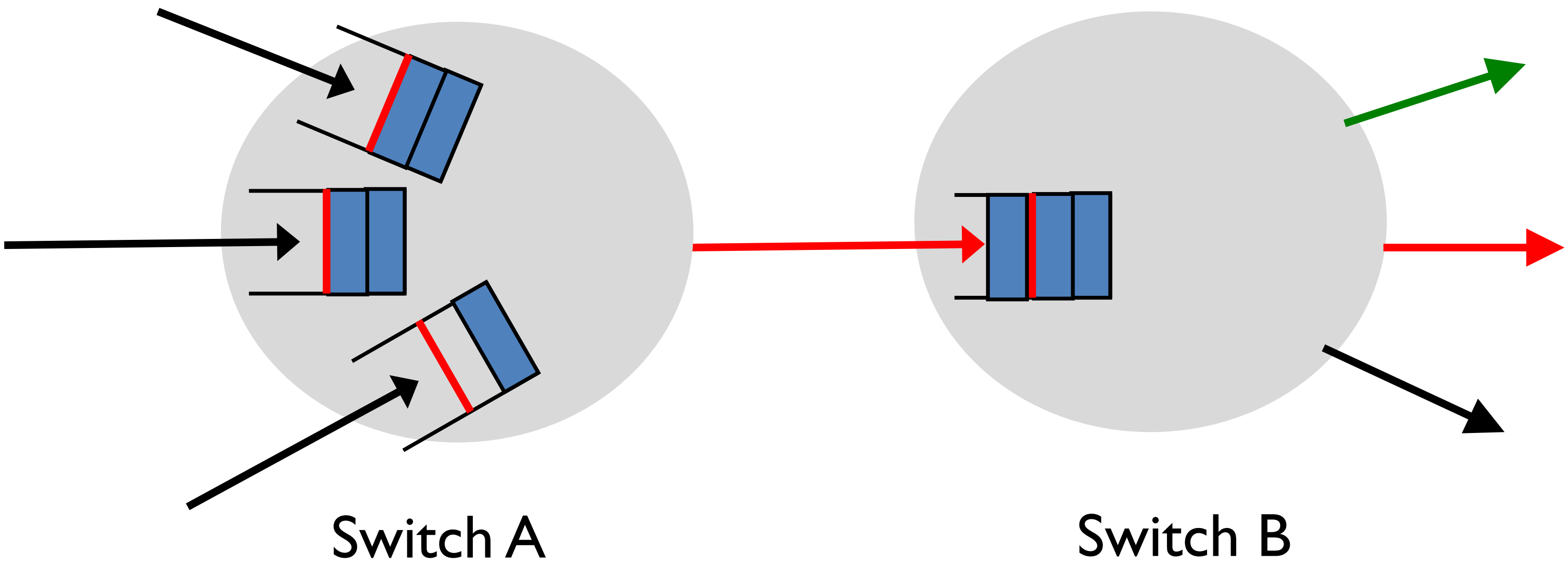
Drawbacks of PFC

Unfairness and Head-of-Line blocking



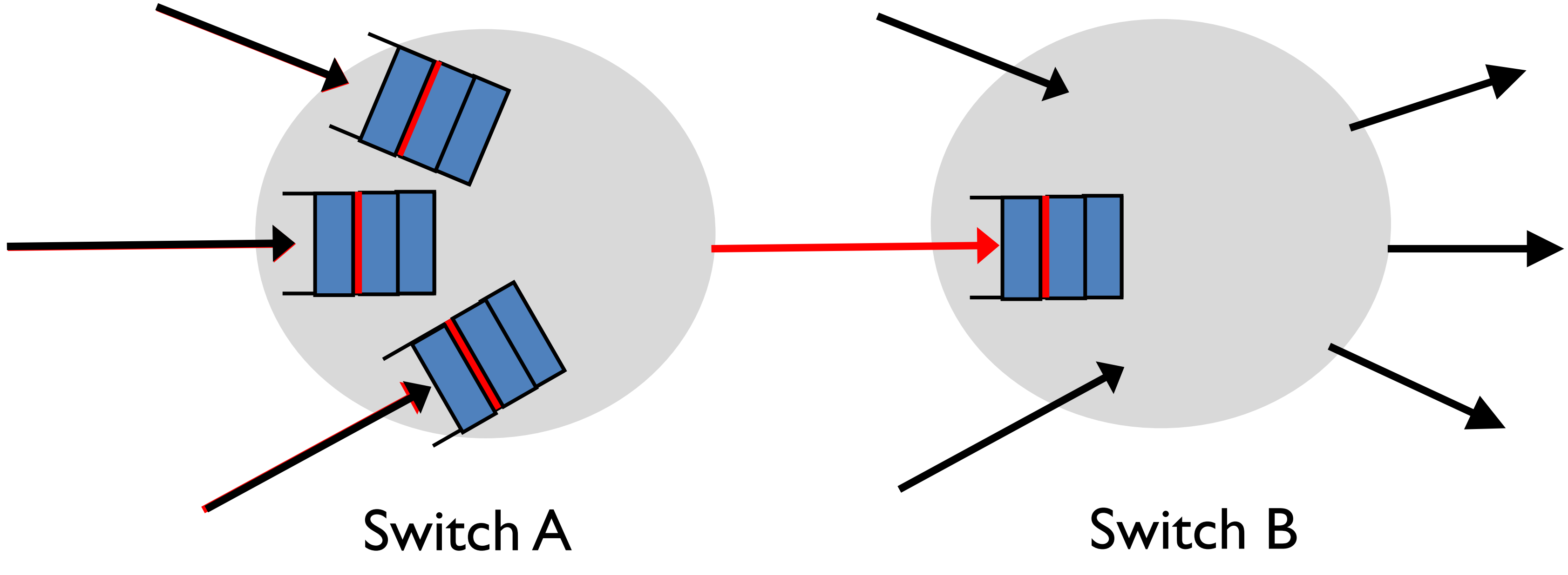
Drawbacks of PFC

Unfairness and Head-of-Line blocking



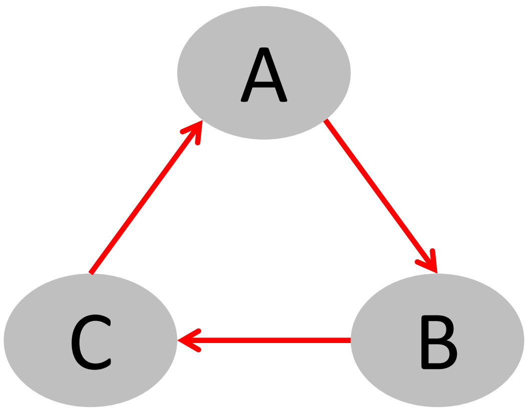
Drawbacks of PFC

Congestion Spreading



Drawbacks of PFC

Deadlocks caused by cyclic buffer dependency



Recent works highlighting PFC issues

- **Congestion control to mitigate PFC issues**
  - TIMELY, *Mittal et al, SIGCOMM'15*
  - DCQCN, *Zhu et al, SIGCOMM'15*
- **Deployment experience**
  - RDMA over commodity Ethernet at scale, *Guo et al, SIGCOMM'16*
- **Deadlock avoidance**
  - Deadlocks in datacenter: why do they form and how to avoid them, *Hu et al, HotNets 2016*
  - Unlocking credit loop deadlock, *Shpiner et al, HotNets 2016*
  - Tagger: Practical PFC deadlock prevention in datacenter networks, *Hu et al, CoNext 2017*

Can we alter the RoCE NIC design such that a lossless network is not required?

Why not iWARP?

- Designed to support RDMA over a fully general network.
  - Implements entire TCP stack in hardware.
  - Needs translation between RDMA and TCP semantics.

- General consensus:
  - iWARP is more complex, more expensive, and has worse performance.

iWARP vs RoCE

NIC	Cost in Dec 2016	Throughput	Latency
iWARP: *Chelsio T-580-CR*	$760	3.24Mpps	2.89us
ROCE: *Mellanox MCX 416A-BCAT*	$420	14.7Mpps	0.94us

**Could be due to a number of reasons besides transport design: different profit margin, engineering effort, supported features etc.*

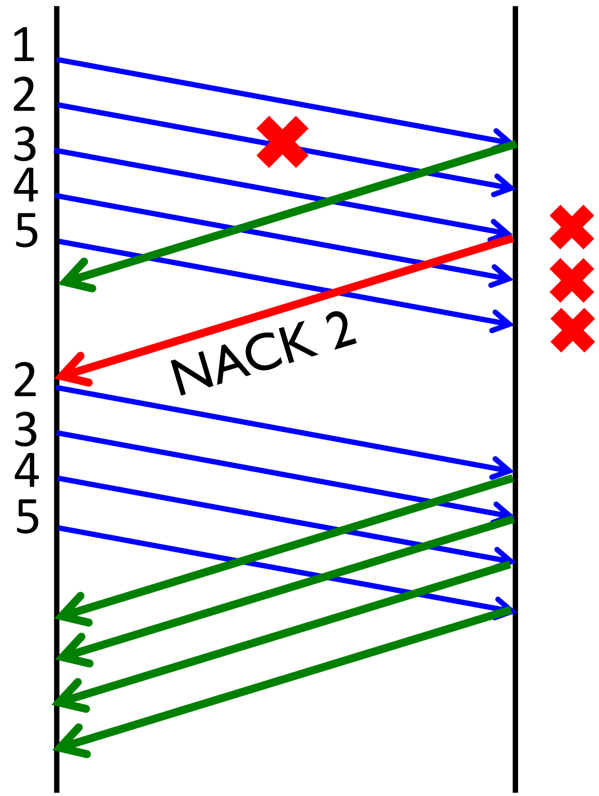
Our work shows that

- iWARP had the right philosophy.
  - NICs should efficiently deal with packet losses.
  - Performs better than having a lossless network.
- But we can have a design much closer RoCE.
  - No need to support the entire TCP stack.
  - Identify incremental changes for better loss recovery.
  - Less complex and more performant than iWARP.

Improved RoCE NIC (IRN)

1. Better loss recovery.

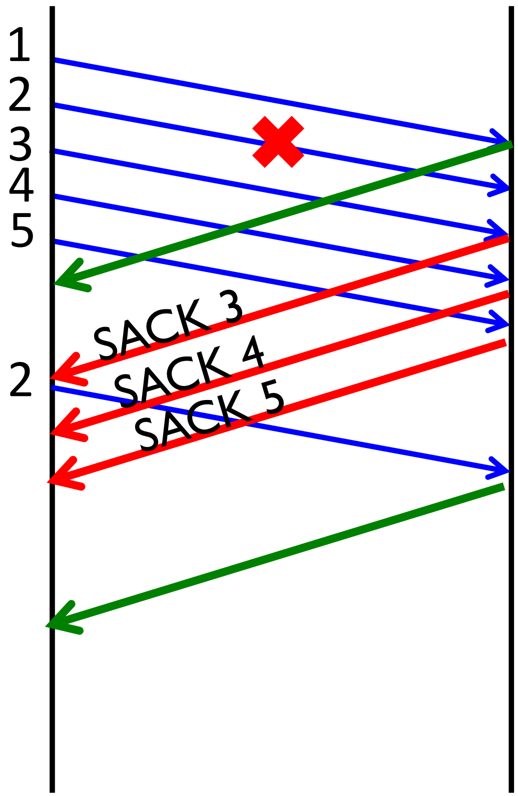
Instead of go-back-N loss recovery…



Receiver discards all out-of-order packets.

Sender retransmits all packets sent after the last acked packet.

...use selective retransmission



Receiver does not discard out-of-order packets and *selectively acknowledges* them.

Sender retransmits only the lost packets.

Use bitmaps to track lost packets.

0	1	1	1	0	0

↑ Seq. No. = 2

Handling timeouts

- Very small timeout value
  - Spurious retransmissions.
- Very large timeout value
  - High tail latency for short messages.

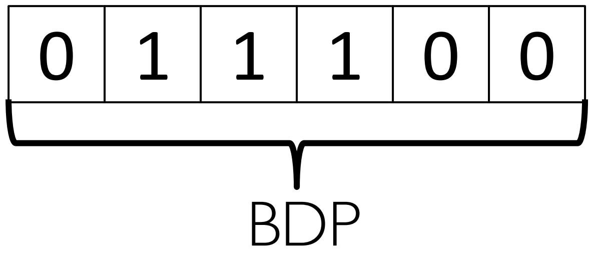
- IRN uses two timeout values
  - RTO_{low}: Less than N packets in flight.
  - RTO_{high}: Otherwise.

Improved RoCE NIC (IRN)

1. Better loss recovery.
   - Selective retransmission instead of go-back-N.
     - Inspired from traditional TCP, but simpler.
   - Two timeout values instead of one.
2. BDP-FC: BDP based flow control.

BDP-FC

- Bound the number of in-flight packets by the bandwidth-delay product (BDP) of the network.
- Reduces unnecessary queuing.
- Strictly upper-bounds the amount of required state.



Improved RoCE NIC (IRN)

1. Better loss recovery.
   - Selective retransmission instead of go-back-N.
     - Inspired from traditional TCP, but simpler.
   - Two timeout values instead of one.

2. BDP-FC: BDP based flow control.
   - Bound the number of in-flight packets by the bandwidth-delay product (BDP) of the network.

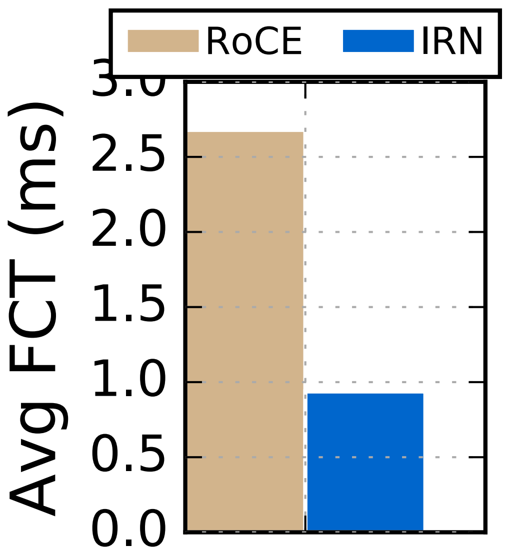
Can IRN eliminate the need for a lossless network?

Default evaluation setup

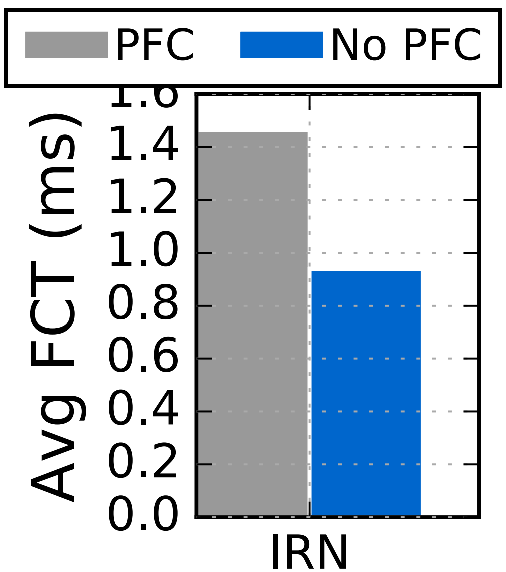
- Mellanox simulator modeling ConnectX4 NICs.
  - Extended from Omnet/Inet.
- Three layered fat-tree topology.
- Links with capacity 40Gbps and delay 2us.
- Heavy-tailed distribution at 70% utilization.
- Per-port buffer of 2 x (bandwidth-delay product).

Key results

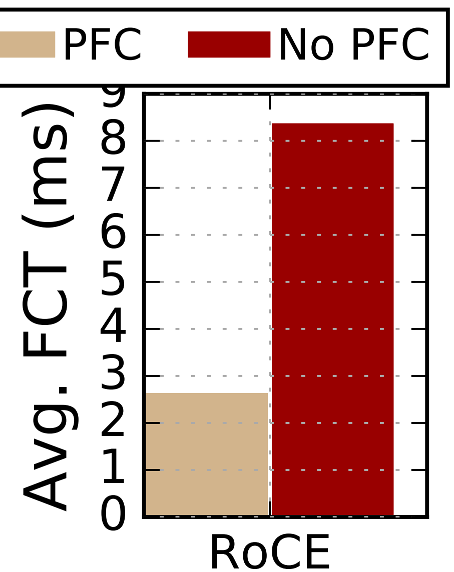
IRN *without PFC* performs better than RoCE *with PFC.*



IRN does not require PFC.

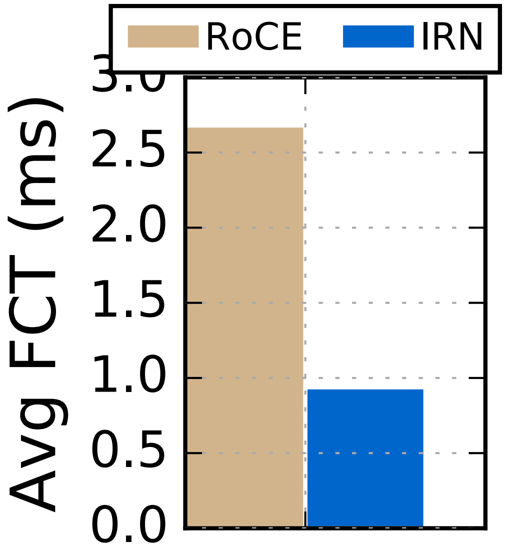


RoCE requires PFC.

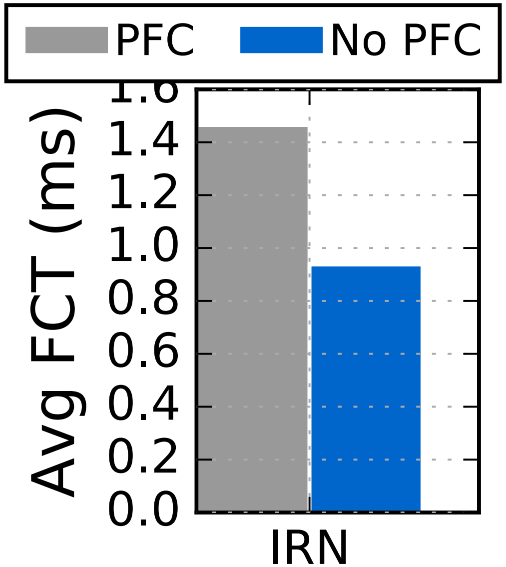


Average flow completion times

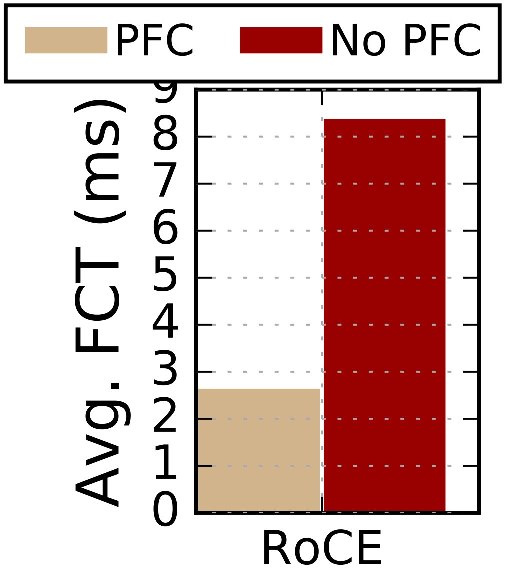
IRN *without PFC* performs better than RoCE *with PFC.*



IRN does not require PFC.

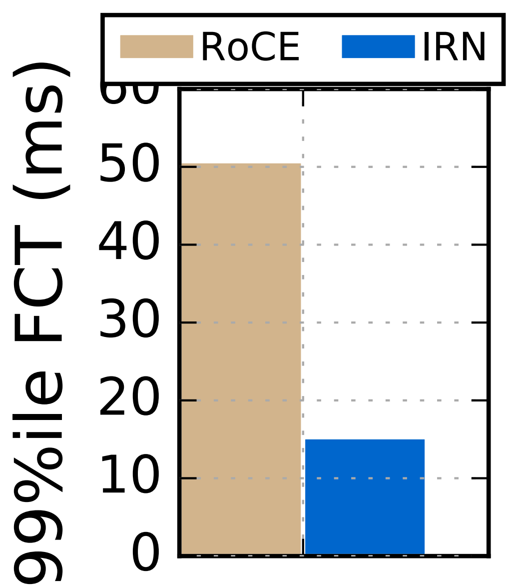


RoCE requires PFC.

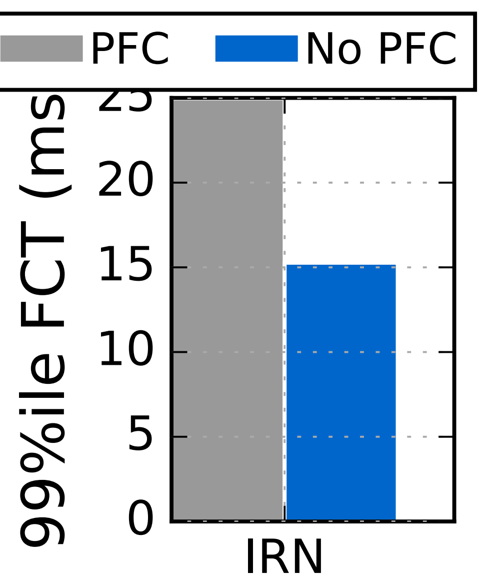


Tail flow completion times

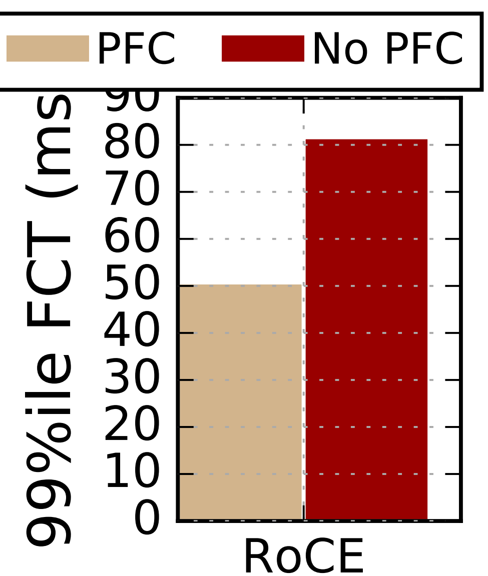
IRN *without PFC* performs better than RoCE *with PFC.*



IRN does not require PFC.

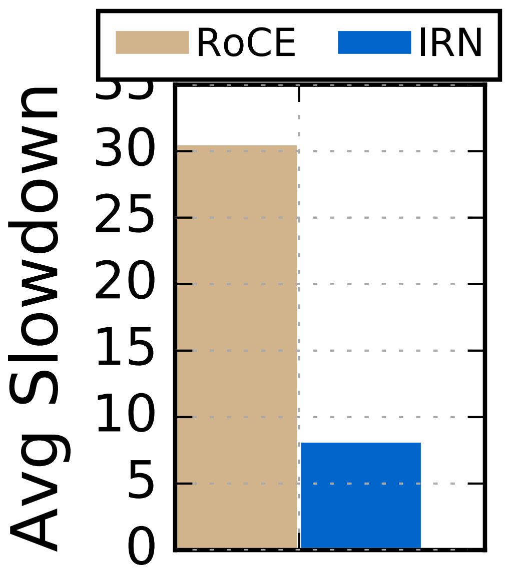


RoCE requires PFC.

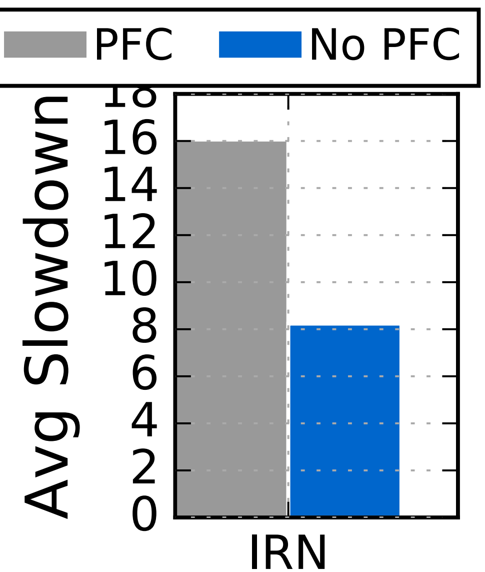


Average slowdown

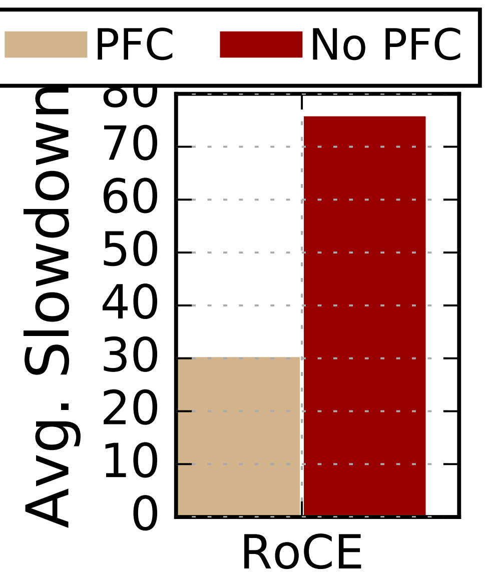
IRN *without PFC* performs better than RoCE *with PFC.*



IRN does not require PFC.

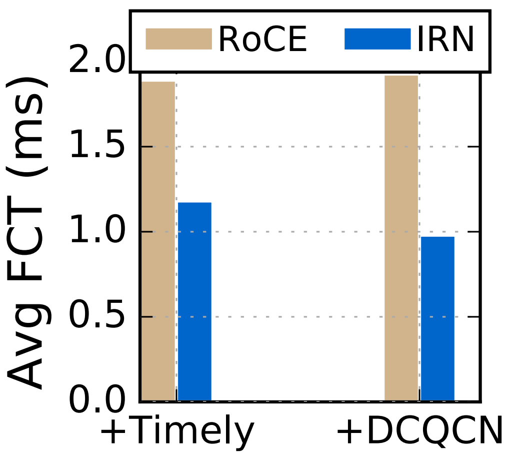


RoCE requires PFC.

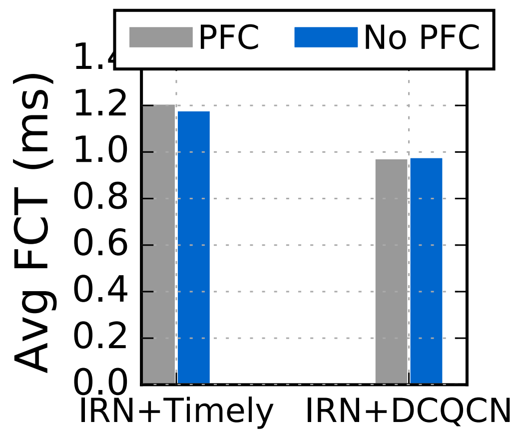


With explicit congestion control

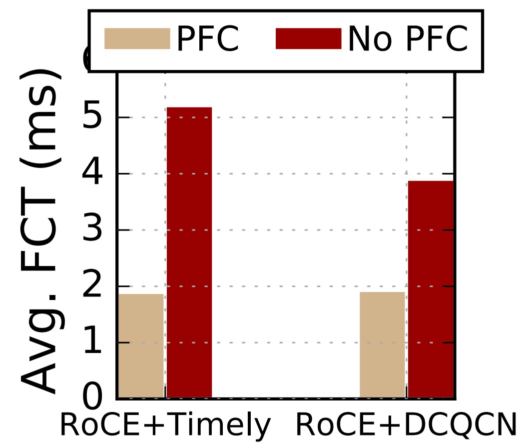
IRN *without PFC* performs better than RoCE *with PFC.*



IRN does not require PFC.

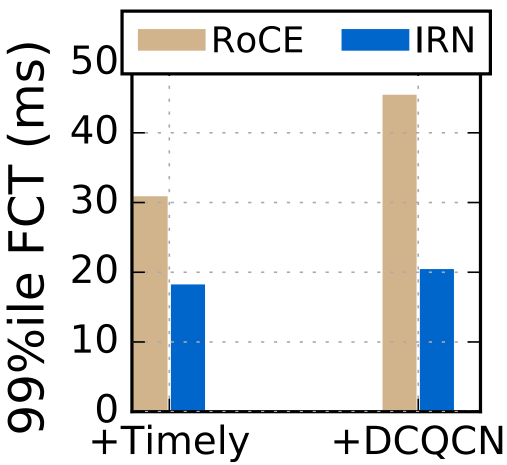


RoCE requires PFC.

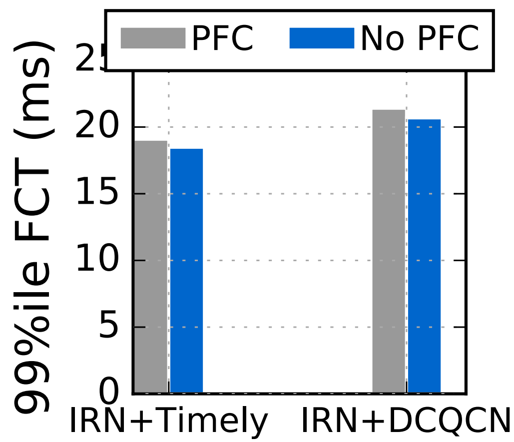


With explicit congestion control

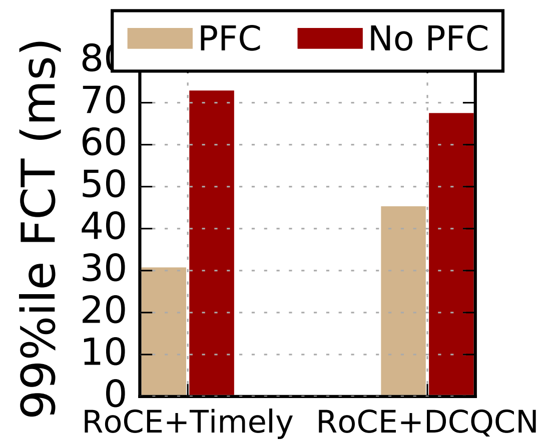
IRN *without PFC* performs better than RoCE *with PFC.*



IRN does not require PFC.

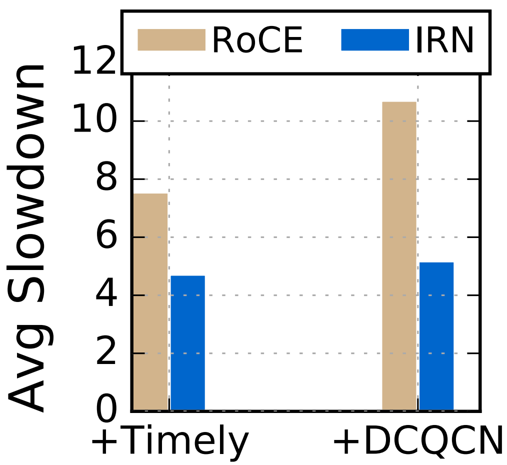


RoCE requires PFC.

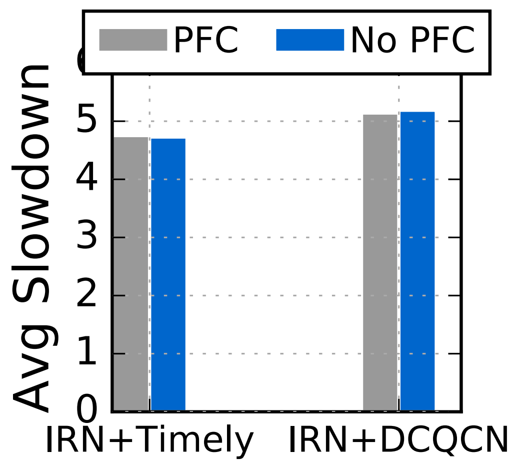


With explicit congestion control

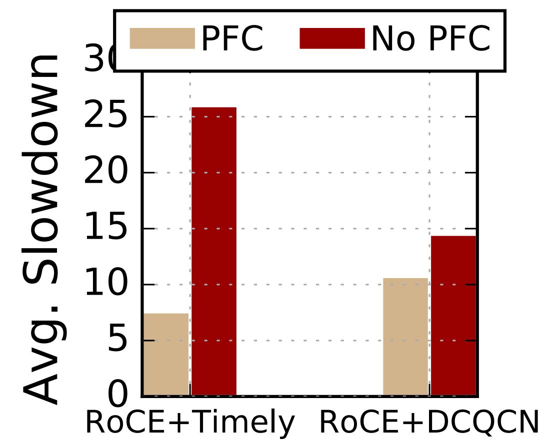
IRN *without PFC* performs better than RoCE *with PFC.*



IRN does not require PFC.



RoCE requires PFC.



Robustness of results

- Tested a wide range of experimental scenarios:
  - Varying link bandwidth.
  - Varying workload.
  - Varying scale of the topology.
  - Varying link utilization.
  - Varying buffer size.
  - …

- Our key takeaways hold across all of these scenarios.

Can IRN eliminate the need for a lossless network? Yes.

Can IRN be implemented easily?

Implementation challenges

- Need to deal with out-of-order packet arrivals.
  - Crucial information in first packet of the message.
    - Replicate in other packets.

Implementation challenges

- Need to deal with out-of-order packet arrivals.
    - Crucial information in first packet of the message.
        - Replicate in other packets.
    - Crucial information in last packet of the message.
        - Store it at the end-points.

Implementation challenges

- Need to deal with out-of-order packet arrivals.
  - Crucial information in first packet of the message.
    - Replicate in other packets.
  - Crucial information in last packet of the message.
    - Store it at the end-points.
  - Implicit matching between packet and work queue element (WQE).
    - Explicitly carry WQE sequence in packets.

Implementation challenges

- Need to deal with out-of-order packet arrivals.
  - Crucial information in first packet of the message.
    - Replicate in other packets.
  - Crucial information in last packet of the message.
    - Store it at the end-points.
  - Implicit matching between packet and work queue element (WQE).
    - Explicitly carry WQE sequence in packets.

- Need to explicitly send Read Acks.

Implementation overheads

- New packet types and header extensions.
  - Upto 16 bytes.

- Total memory overhead of 3-10%.

- FPGA synthesis targeting the device on an RDMA NIC.
  - Less than 4% resource usage.
  - 45.45Mpps throughput (without pipelining).

Can IRN eliminate the need for a lossless network? Yes.

Can IRN be implemented easily? Yes.

IRN Summary

- IRN makes incremental updates to the RoCE NIC design to handle packet losses better.
- IRN performs better than RoCE without requiring a lossless network.
- The changes required by IRN introduce minor overheads.

Your Opinions

- Pros:
  - Questions the requirement of PFC.
  - Comprehensive experiments and analysis.

Your Opinions

- Cons:
  - Changing NIC design is non-trivial.
    - Additional overhead and state.
  - Fairness as another metric for evaluation.

Your Opinions

- Ideas:
  - Other loss recovery mechanism?
  - Better congestion control to mitigate PFC issues.
  - Dynamically estimate BDP?

Challenges of deploying RDMA in DCs

- Need for a lossless network
  - Better loss recovery in the NIC (IRN, SIGCOMM'18)
  - Large buffers (eRPC, NSDI'19)
- Limited NIC cache:
  - Use bigger pages for memory translation (FaRM, NSDI'14).
  - Optimizing number of QPs (FaRM, NSDI'14; FASST, OSDI'16).
- Limited resource sharing and isolation
  - Kernel re-direction (LITE, SOSP'17)
- Limited flexibility
  - FPGA-based implementation
- ….

Is RDMA the right choice for datacenters?

What will a clean slate approach look like?