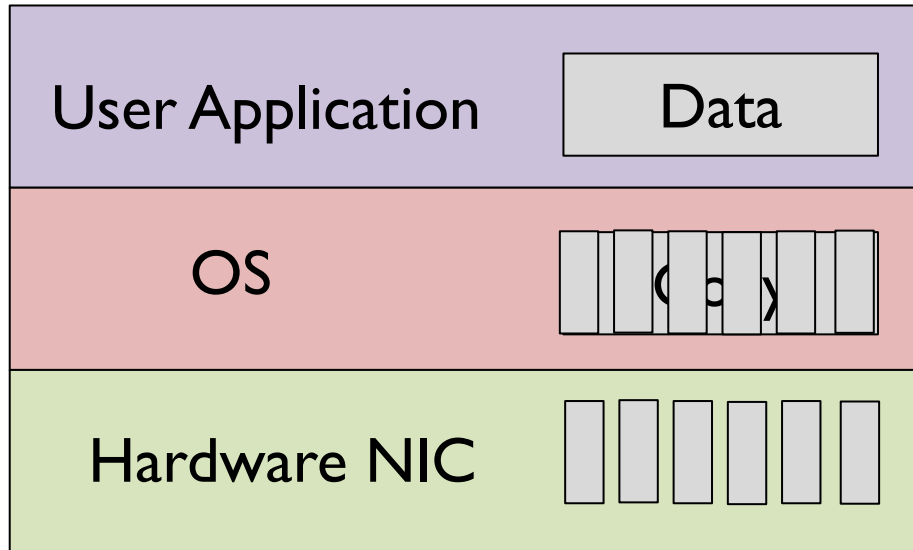


RDMA

ECE/CS598HPN

Radhika Mittal

Traditional Network Stack

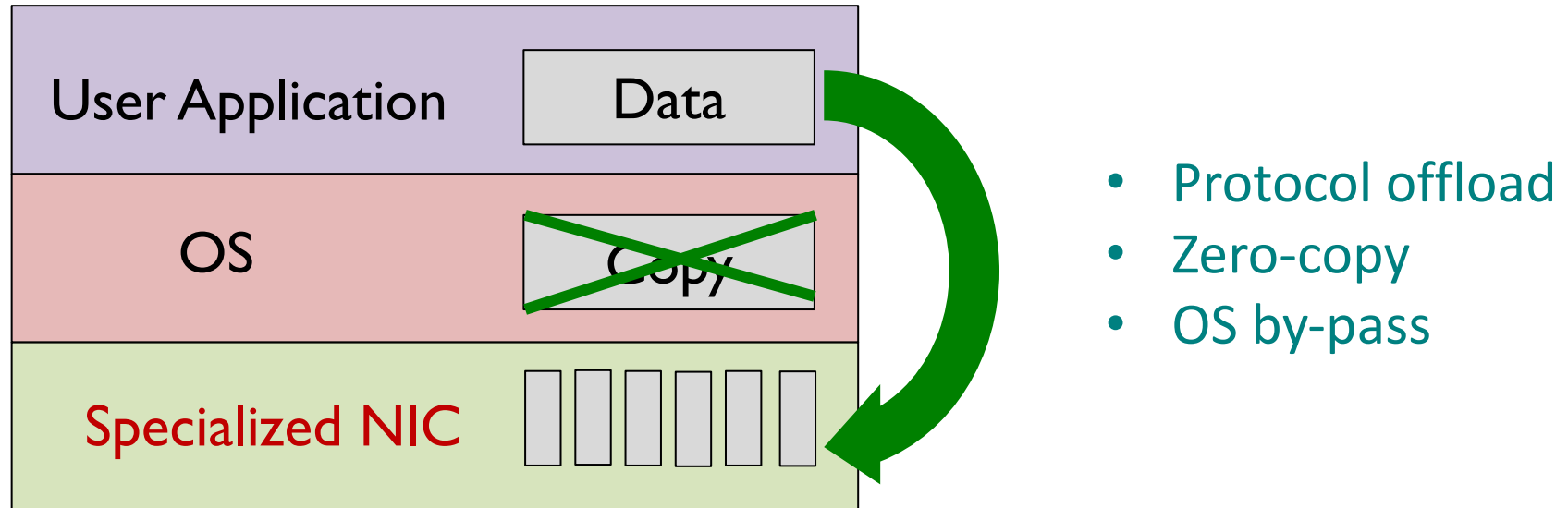


Packet processing in OS adds latency overheads, supports limited throughput (pps), and leads to high CPU utilization.

Datacenter requirements:

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

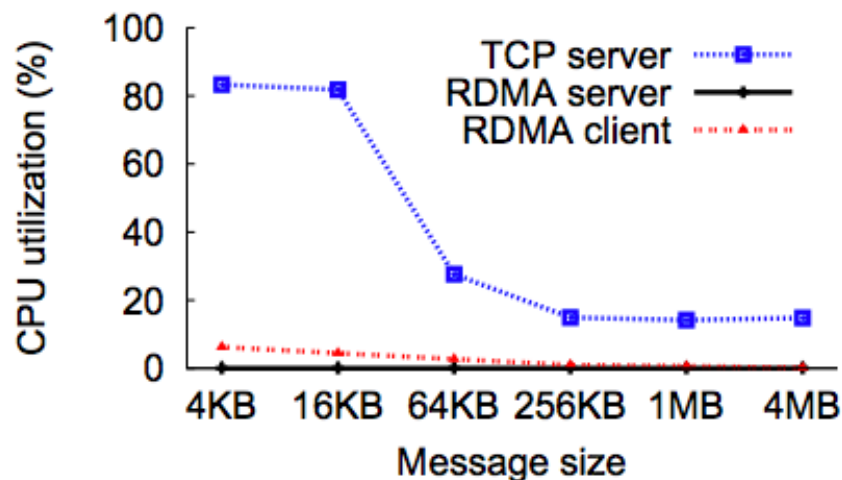
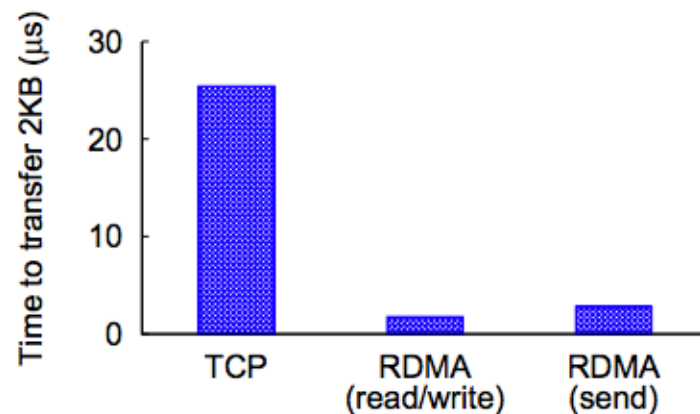
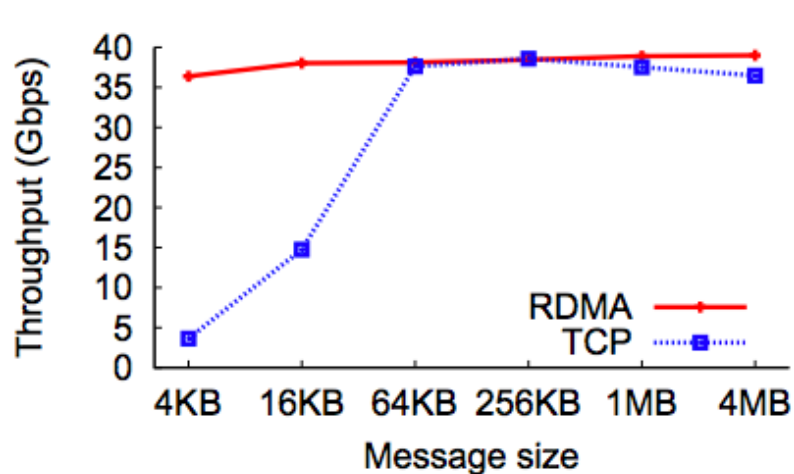
Remote Direct Memory Access



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

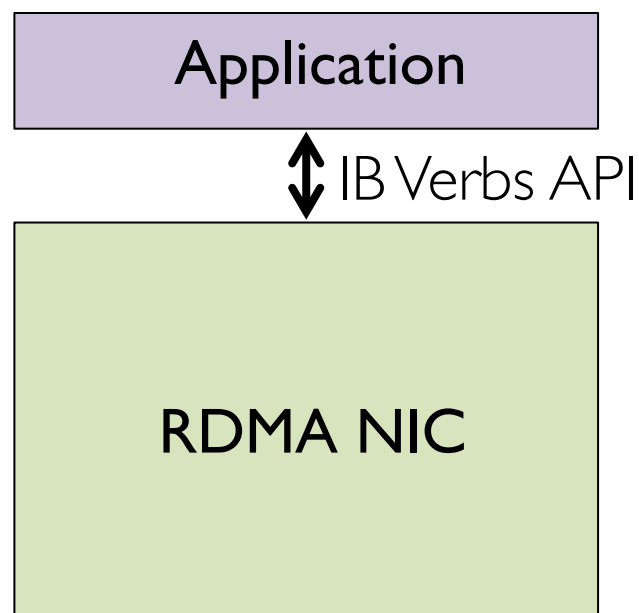
RDMA usecases in datacenters

- 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 (e.g. search queries)
- Other proposals
 - Resource disaggregation (OSDI'16), Remote swapping (NSDI'17), ...
 - use RDMA interface to support arbitrary computation without involving server CPU (NSDI'22)

RDMA usecases in datacenters

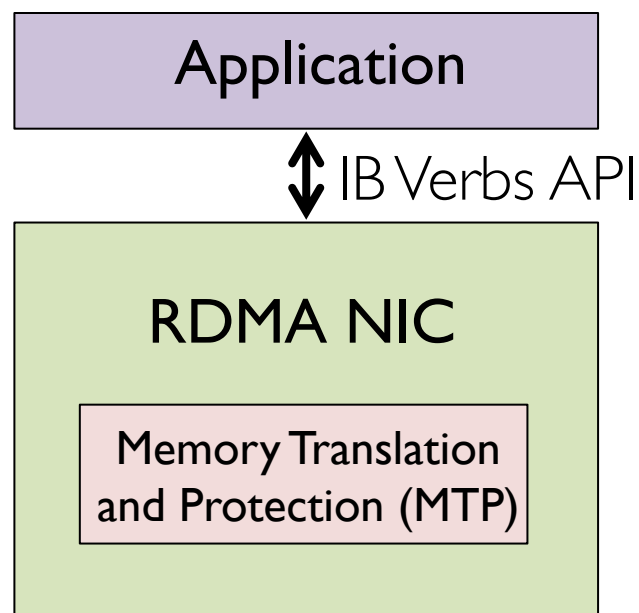
- Microsoft one of the early adopters and proponents of RDMA / NIC hardware based solutions:
 - Key motivation: save CPU cycles to sell to customers.
- Non-trivial to deploy RDMA (hardware transport) at scale.
- Use of RDMA / hardware transport remained debatable.
- **Killer usecase: GPU Direct communication**
 - by-pass CPU for ML training.
 - more on this next week.

RDMA Overview and Components



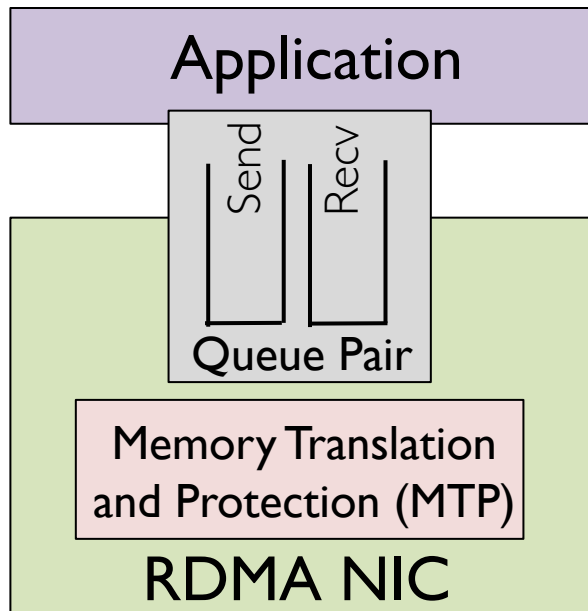
Applications bypass the kernel and interact directly with the RDMA NIC using the **IB verbs** API provided by the NIC driver.

Memory Translation and Protection



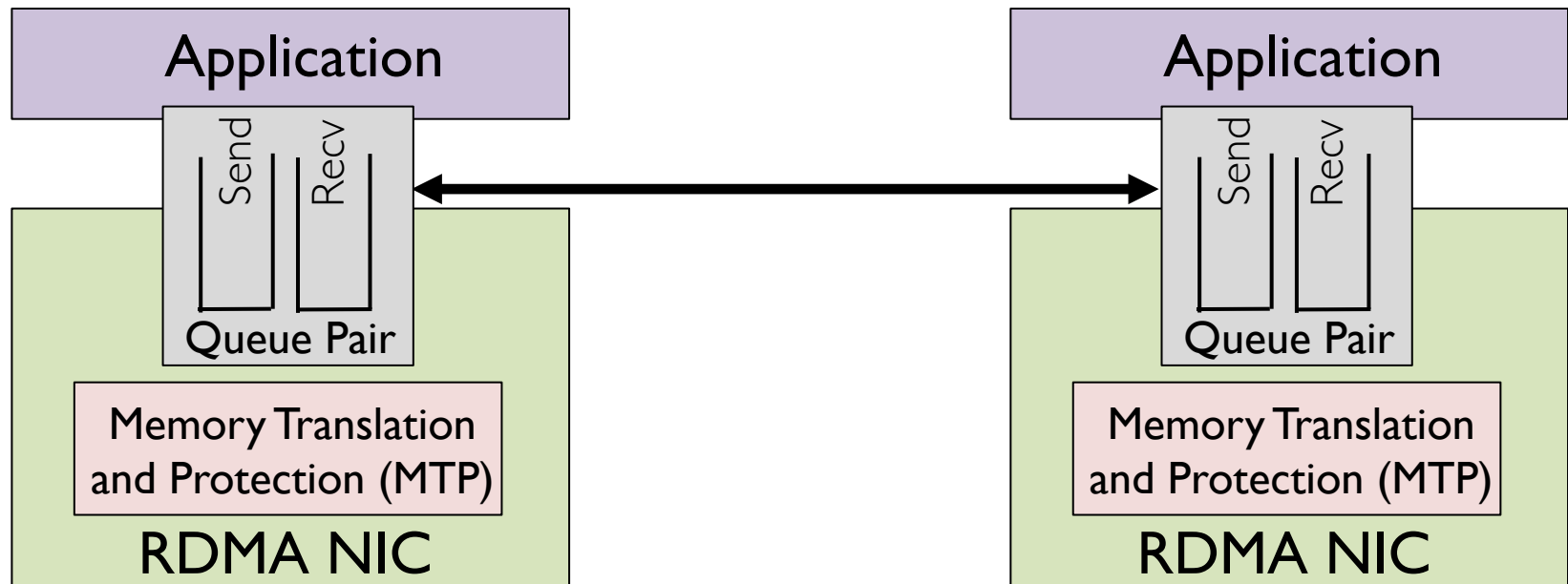
- Applications register *memory regions* with the NIC.
- **Translation:** MTP maintains *virtual address* to *physical address* mapping.
- **Protection:** MTP assigns local and remote access keys to memory region.

Queue Pairs (QP)



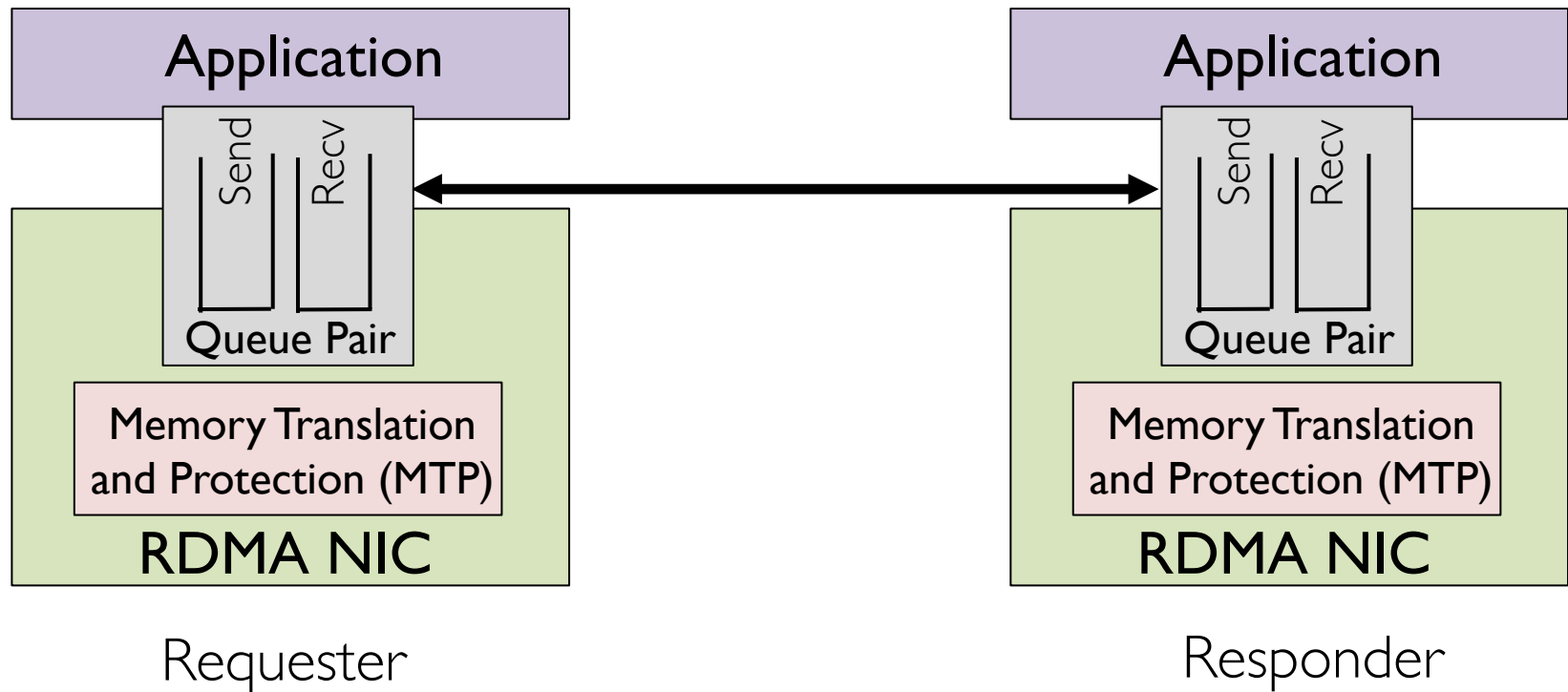
- QPs are interfaces between the application and the NIC.
- Different types:
 - Connection-oriented vs Datagram
 - Reliable vs unreliable.
- Reliable Connected (RC) QPs
 - Analogous to a TCP connection.
 - Support all types of operations.

Connection Establishment

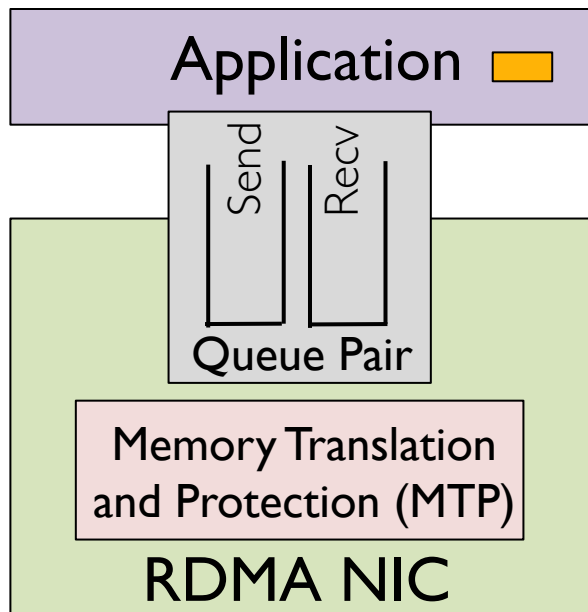


Connection establishment requires out-of-band exchange of node identifiers, QP id, and remote keys.

Work Requests



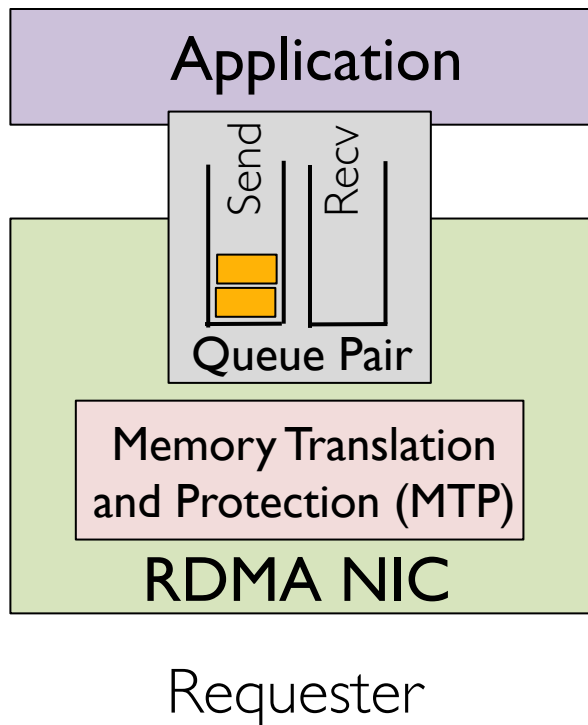
Work Requests



Requester

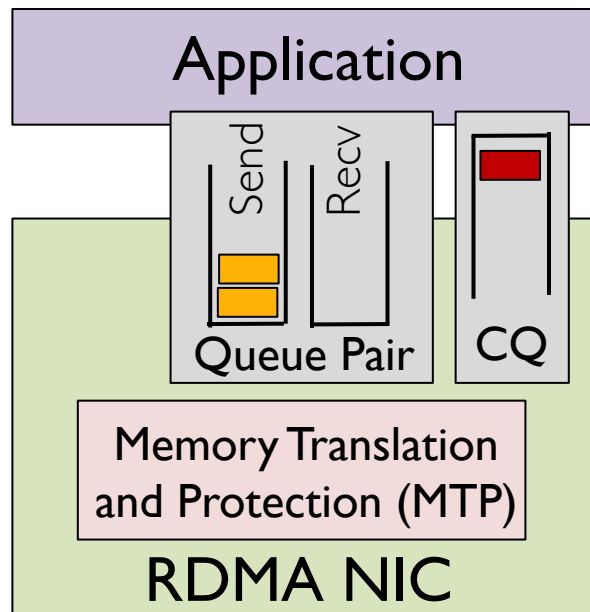
- Application issues a *work request* (WR) for a QP.
- WR contains all the metadata associated with a message transfer.

Work Queue Element (WQE)



- 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

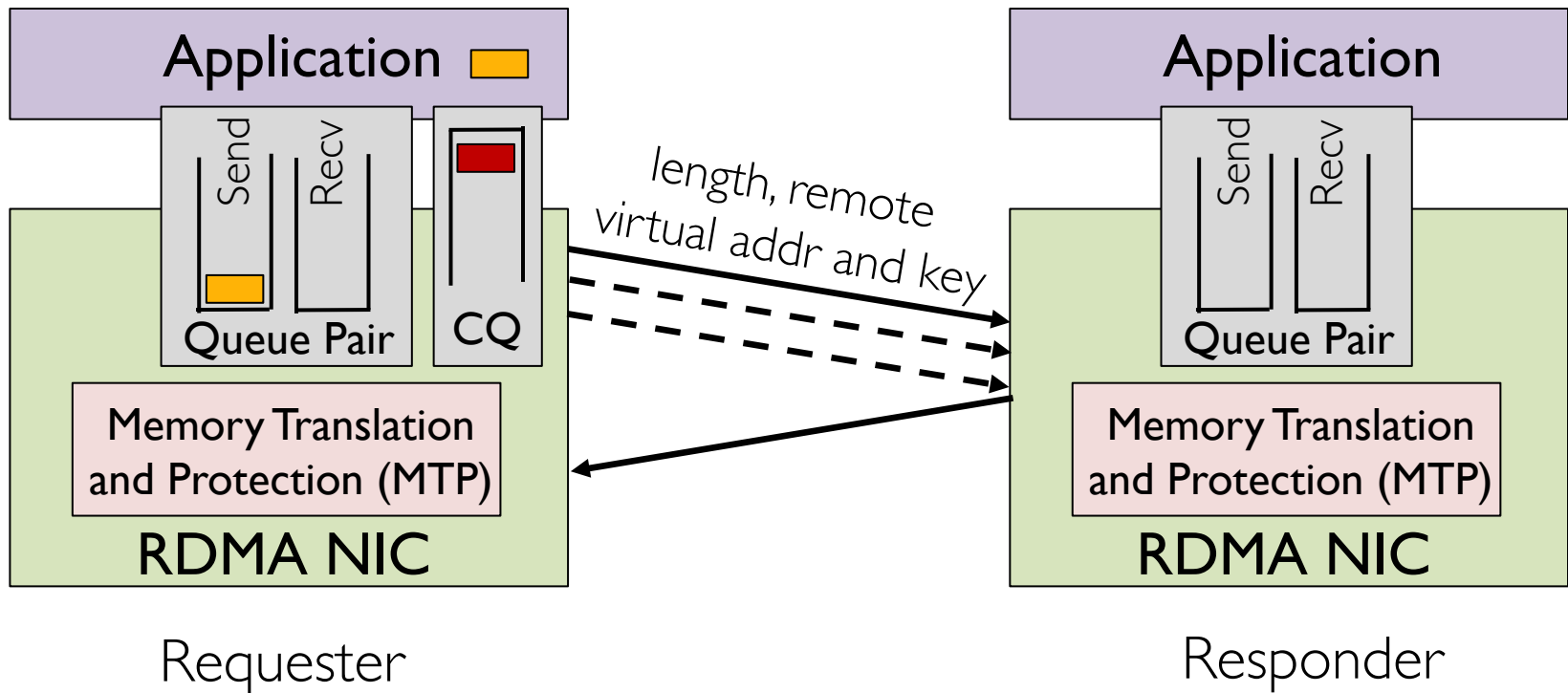
- Each QP is associated with a completion queue (CQ).
- Upon request completion,
 - The WQE expires.
 - CQE is created.
- CQE notifies request completion to application.

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.

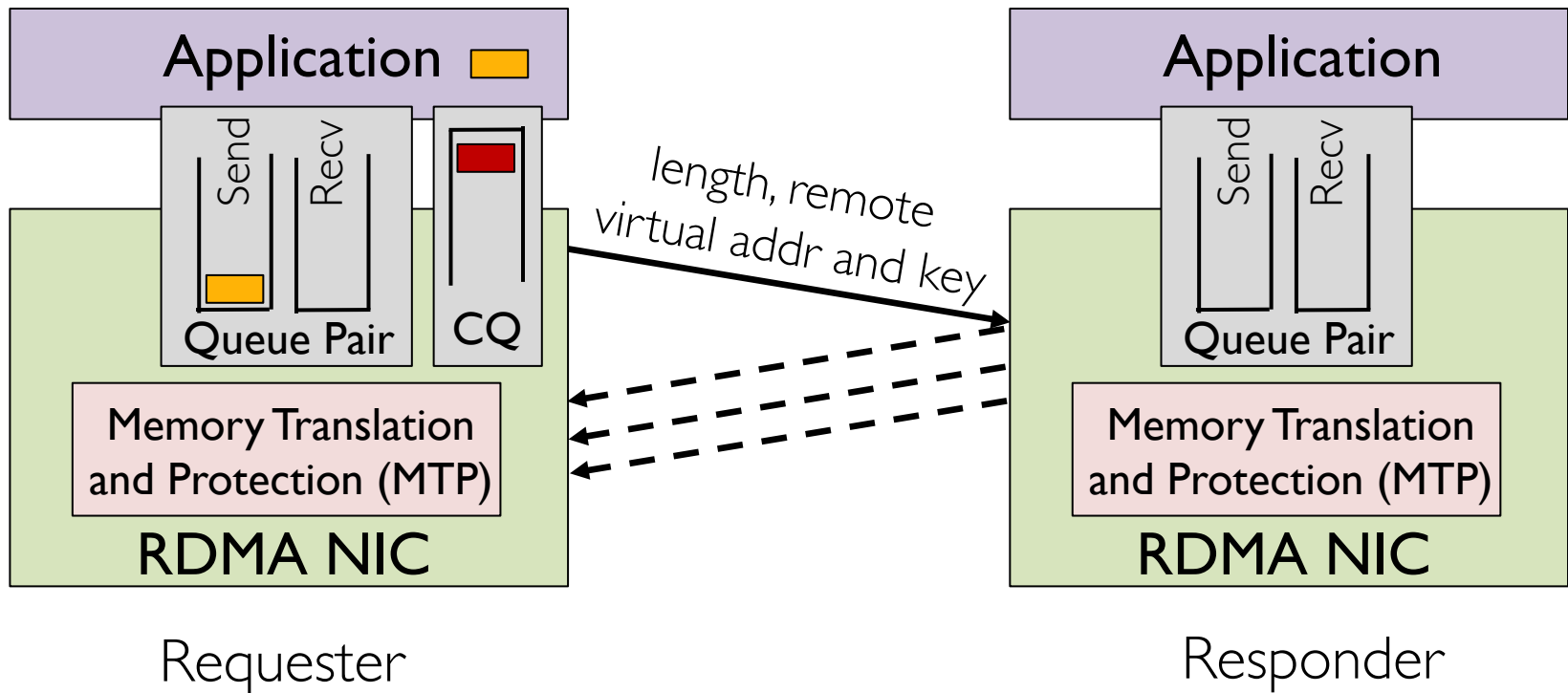
RDMA Write

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



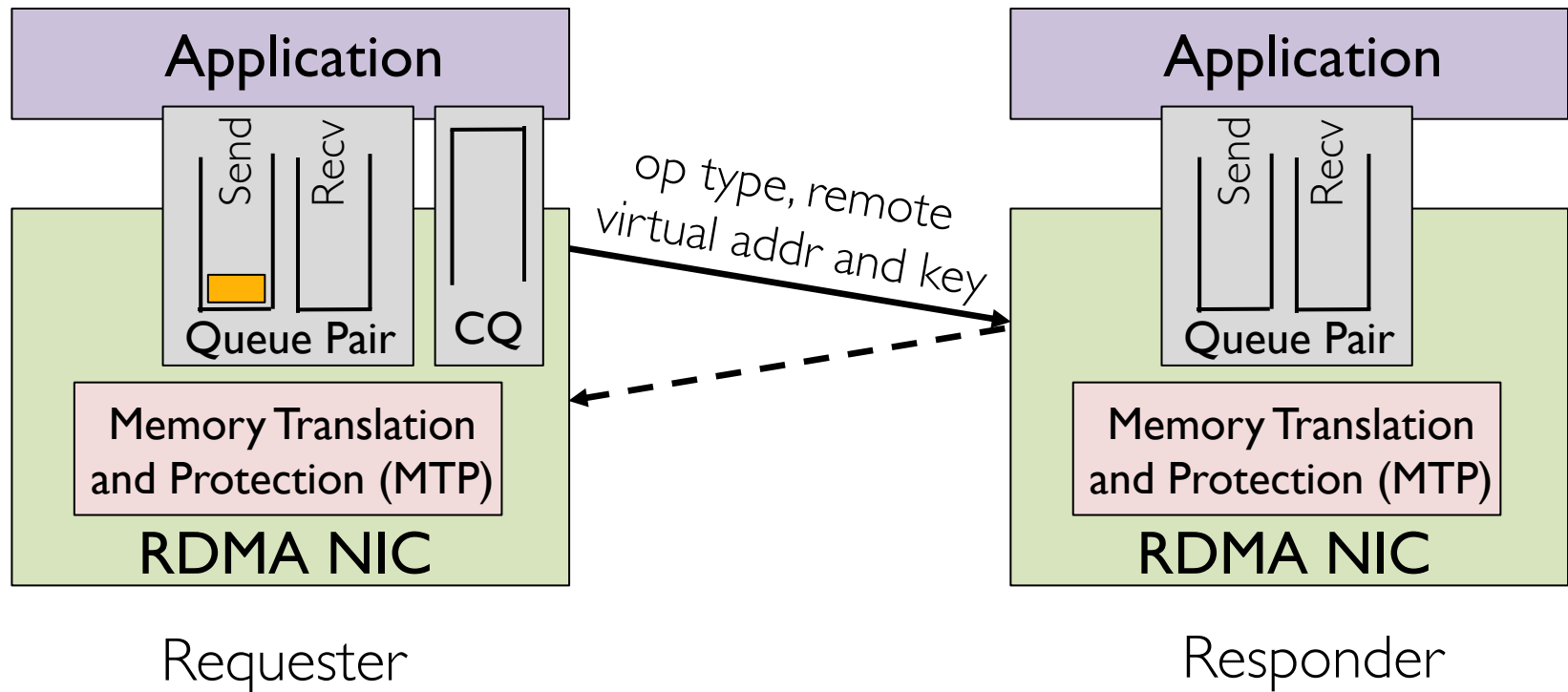
RDMA Read

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



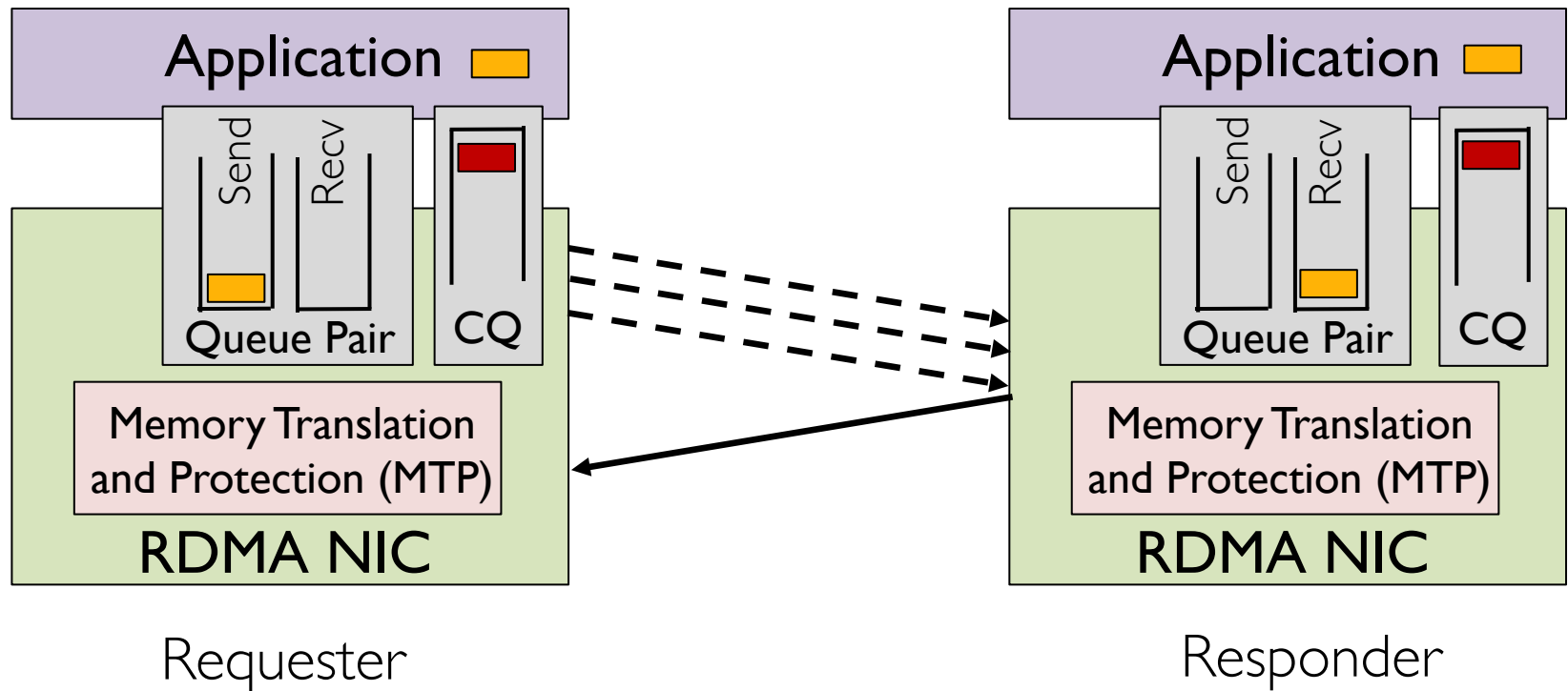
RDMA Atomic

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

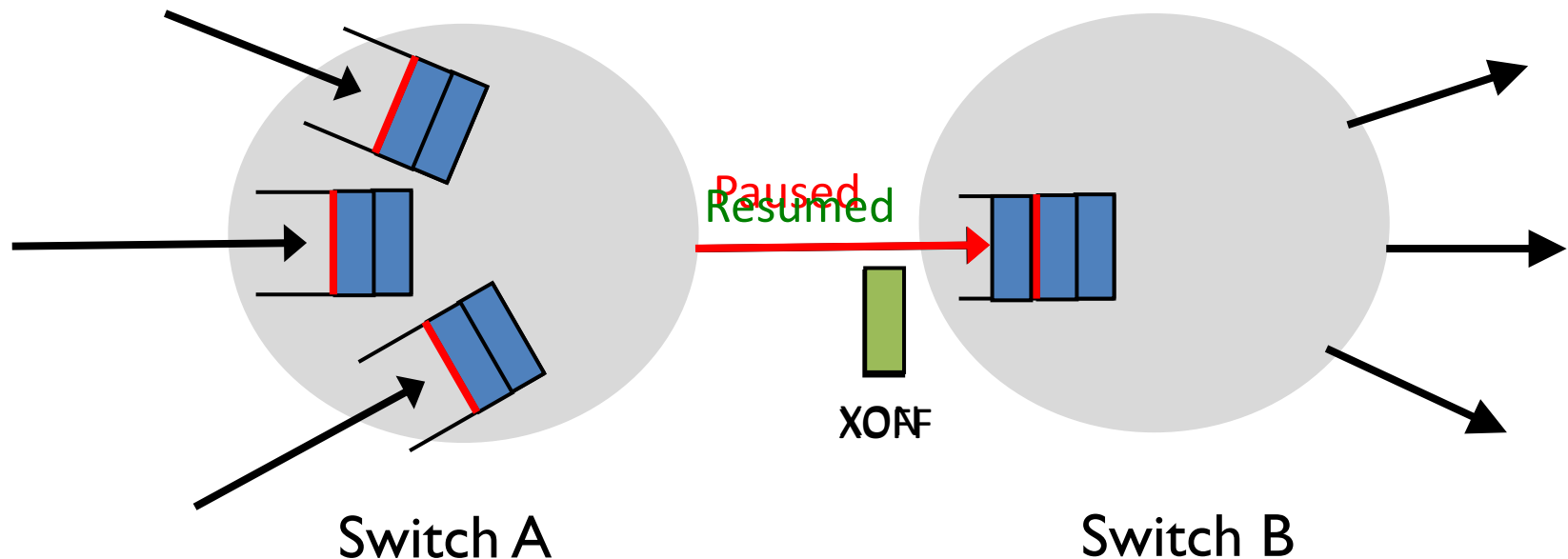
- **RDMA Write:** Write data from local node to specified address at remote node.
- **RDMA Read:** Read data from specified address at remote node to local node.
- **RDMA Atomic:** Atomic fetch-add and compare-swap operations at specified location at remote node.
- **Send/Receive:** Send data to a remote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

Challenges: RDMA at scale

- Required a lossless network.
 - Network made lossless by enabling Priority Flow Control (PFC).
 - Pause transmission when queuing exceeds a certain threshold.



Challenges: RDMA at scale

- Required a lossless network.
 - Network made lossless by enabling Priority Flow Control (PFC).
 - Pause transmission when queuing exceeds a certain threshold.
 - Led to many issues: deadlocks, head-of-line blocking, increased complexity.
 - Congestion control added to mitigate these issues:
 - DCQCN (Microsoft, Zhu et. al., SIGCOMM'15)
 - TIMELY (Google, Mittal et. al., SIGCOMM'15)
 - More on this in the next class!

Challenges: RDMA at scale

- Required a lossless network.
 - Network made lossless by enabling Priority Flow Control (PFC).
 - Pause transmission when queuing exceeds a certain threshold.
 - Led to many issues: deadlocks, head-of-line blocking, increased complexity.
- Limited NIC memory:
 - for storing QP context and page tables at scale.
- Limited flexibility

Today's reading

- Empowering Azure Storage with RDMA (NSDI'23)
- What was the primary usecase?
- What are the other challenges that arise?
- What do you all think of using RDMA for this usecase?