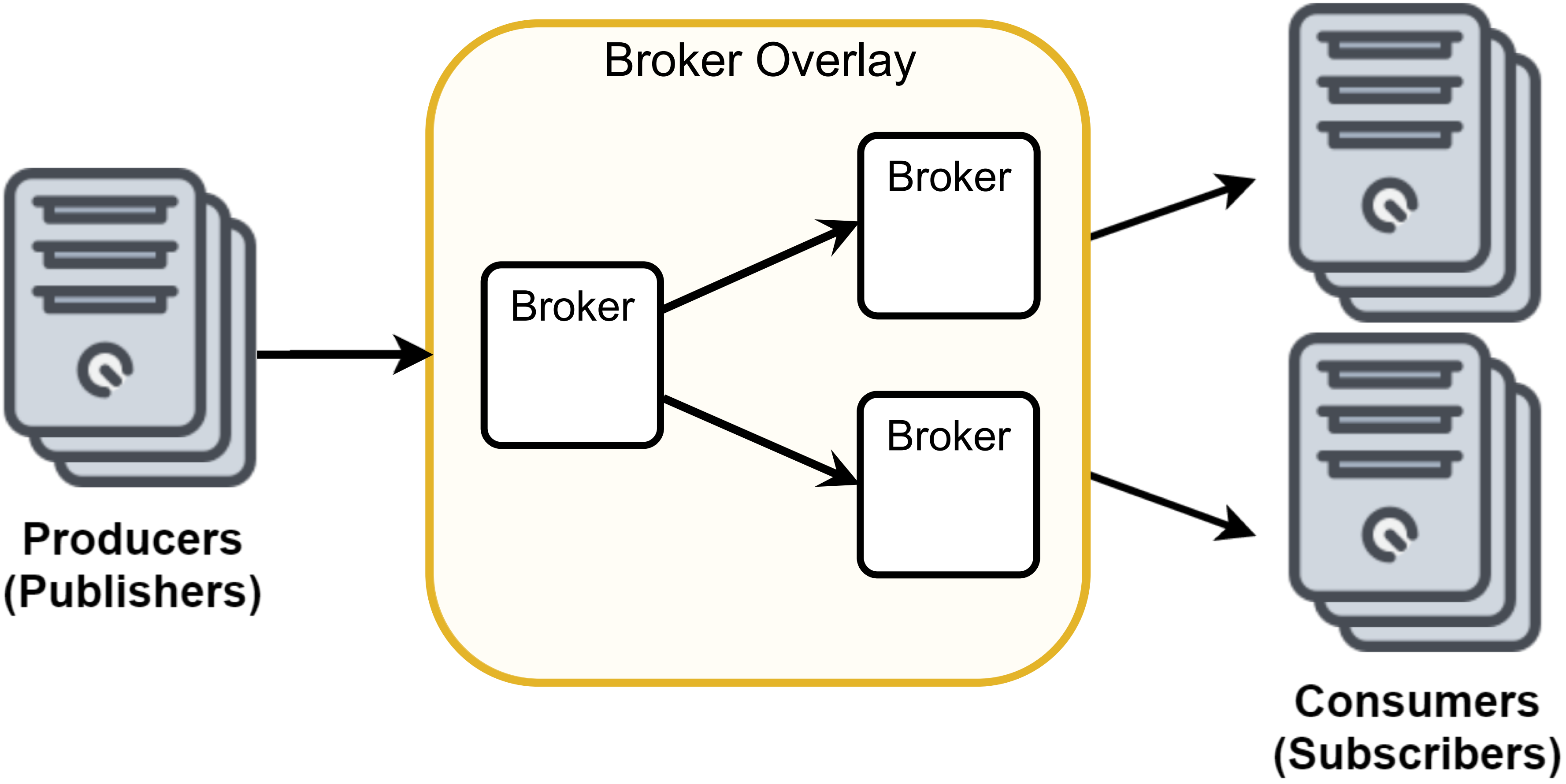
RocketBufs: A Framework for Building Efficient, In-Memory, Message-Oriented Middleware

Huy Hoang, Benjamin Cassell, Tim Brecht,
Samer Al-Kiswany

July 17, 2020

UNIVERSITY OF WATERLOO | DAVID R. CHERITON SCHOOL OF COMPUTER SCIENCE

Message-Oriented Middleware (MOM)



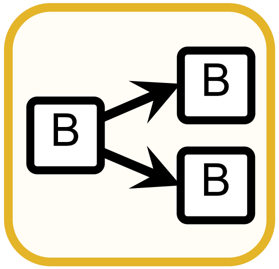
Message-Oriented Middleware

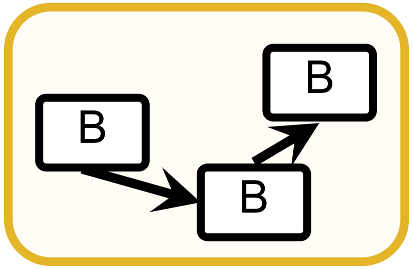
Message-Oriented Middleware

- Lots of MOM systems
- Yet, new systems still being built: 6 new projects in last 5 years
- Not using advanced data center networking technologies
  - RDMA, DPDK, TCPDirect
    - APIs and abstractions different from TCP, hard to program,
    - Difficult to support multiple networking technologies

Goals of RocketBufs

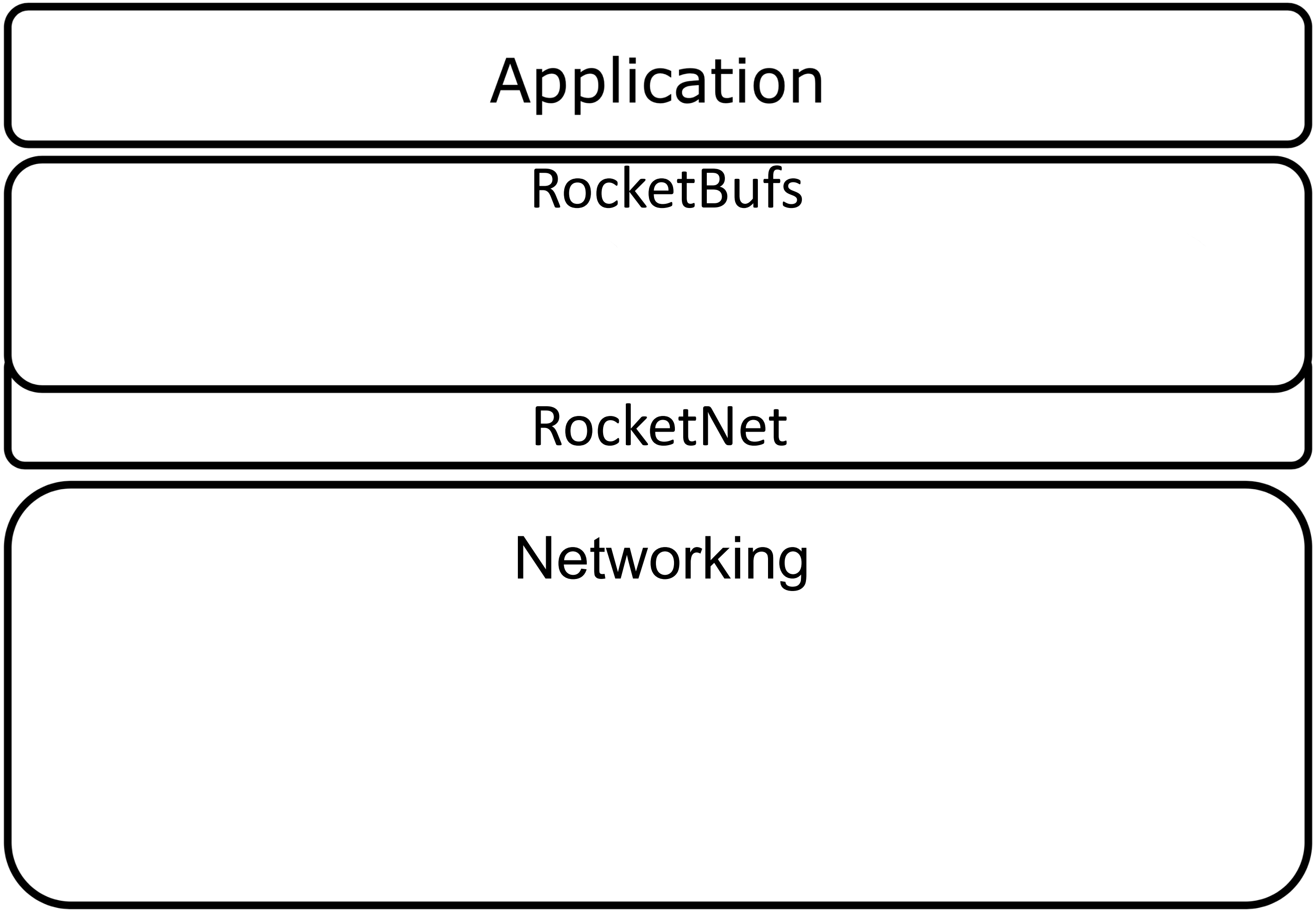
- Natural abstractions and easy-to-use APIs for MOM systems
- Support for different networking technologies / transport protocols
    - Add new technologies to RocketBufs
    - Applications access them with no changes to code
- Support for flexible MOM topologies



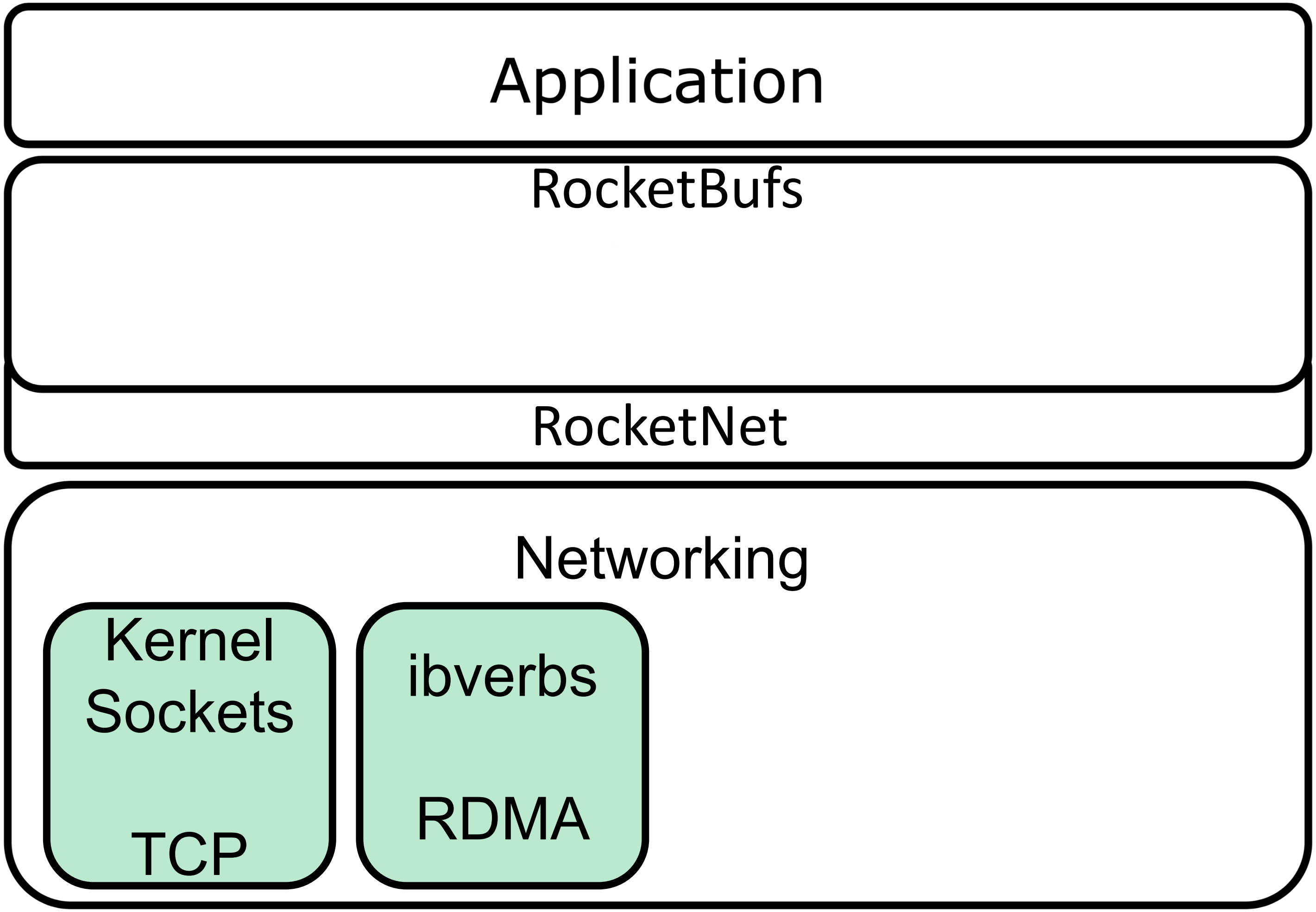


- Resulting applications are efficient and scalable

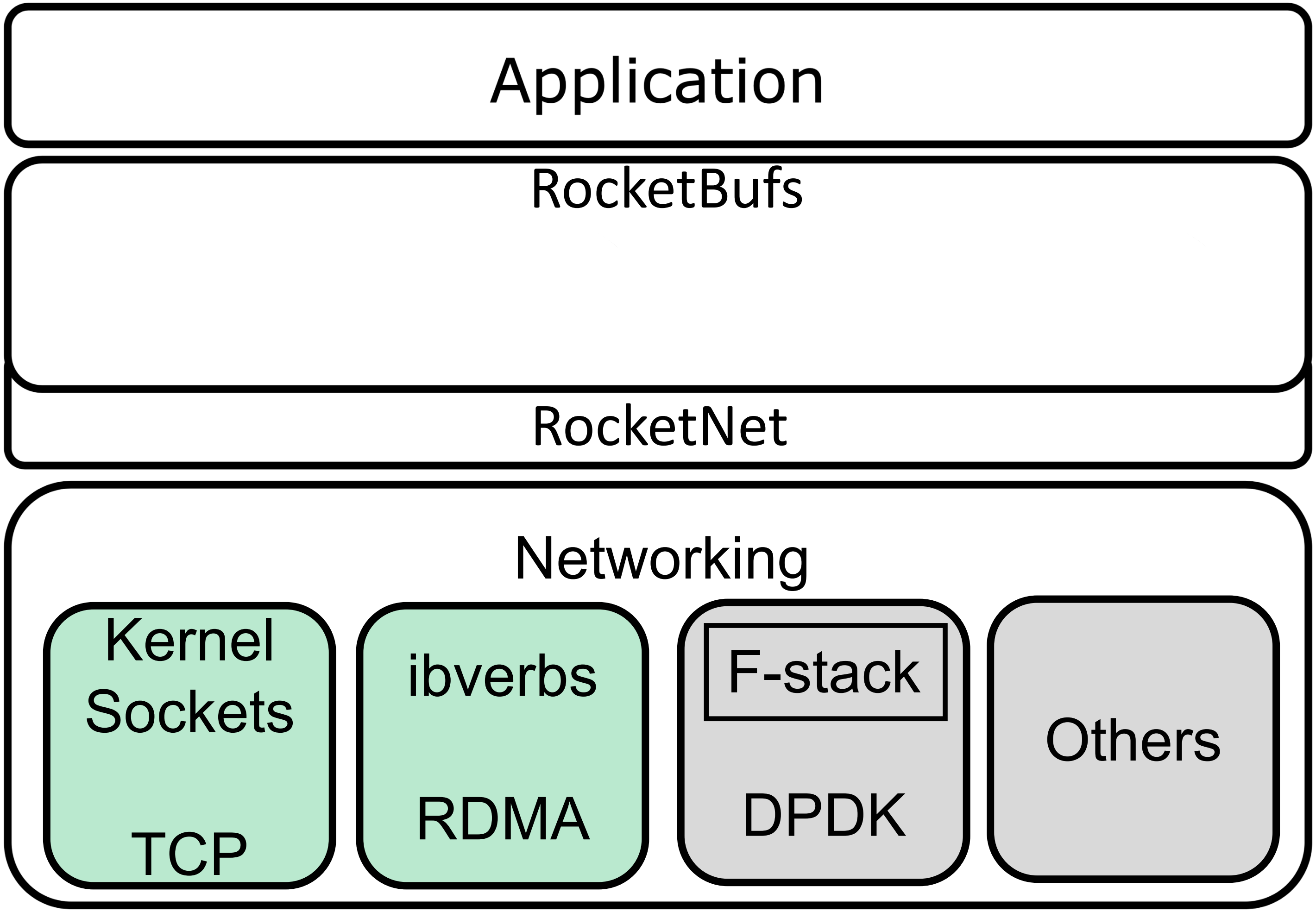
RocketBufs



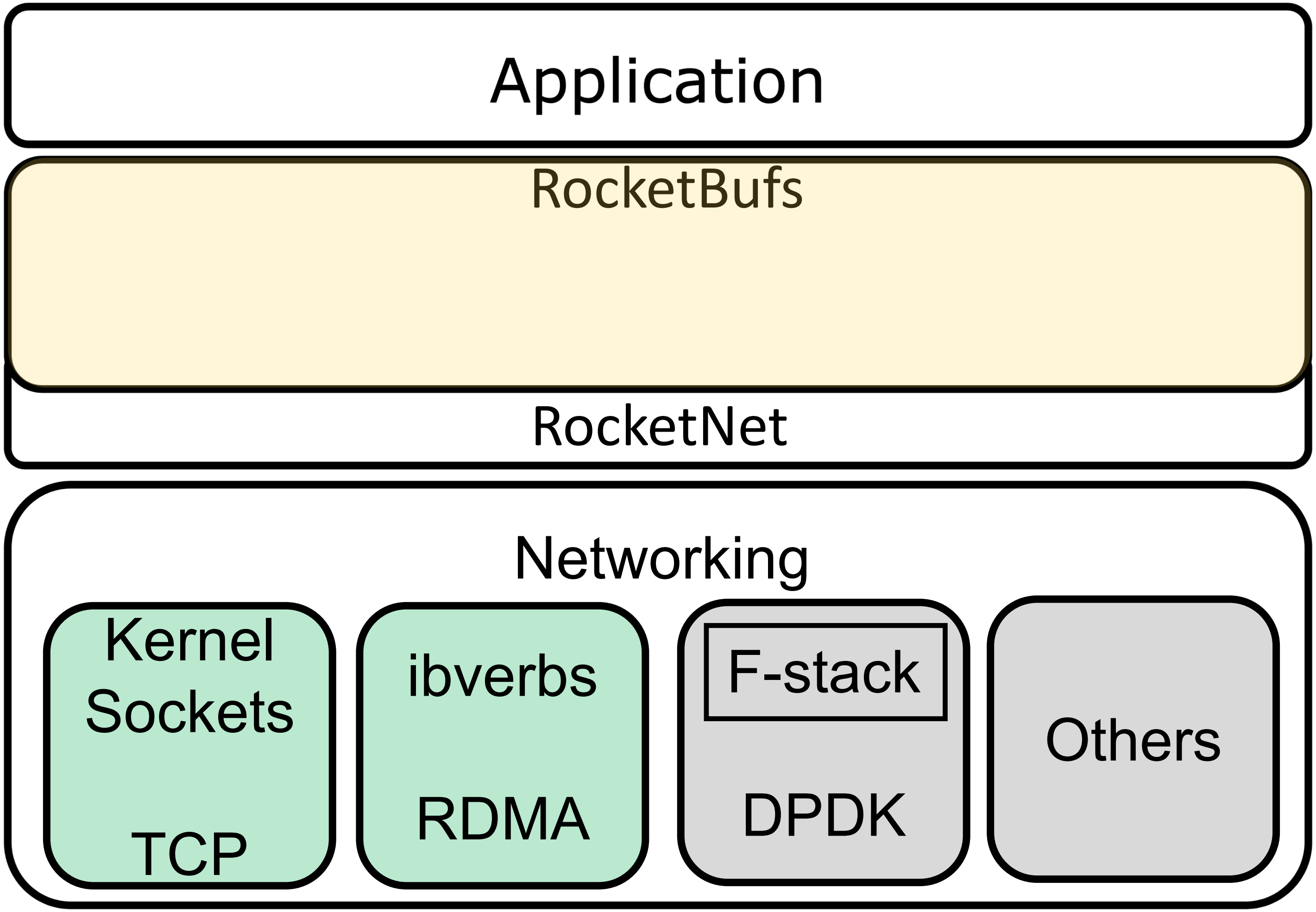
RocketBufs


Application
RocketBufs
RocketNet
Networking
Kernel Sockets
TCP
ibverbs
RDMA

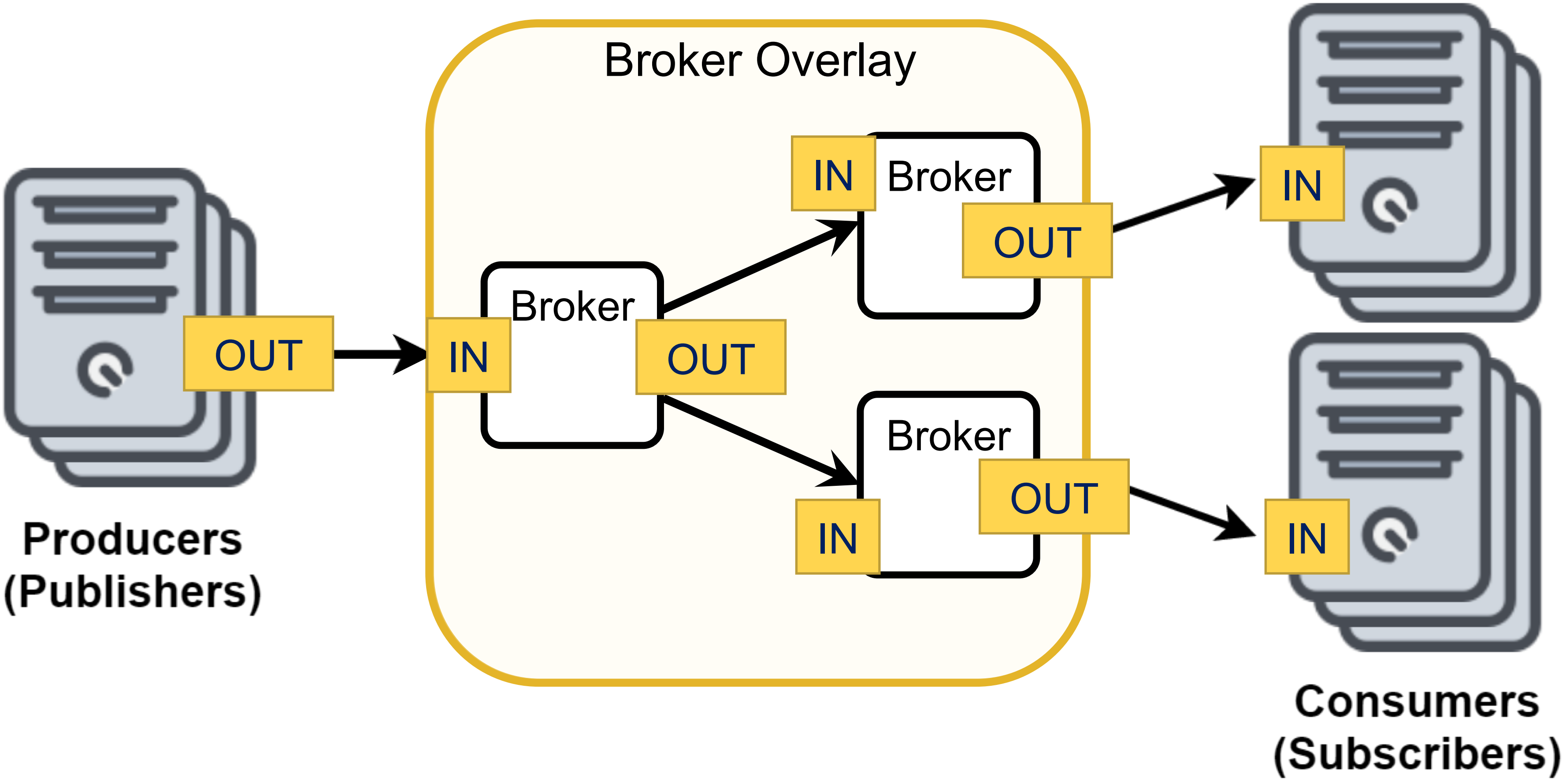
RocketBufs


Application
RocketBufs
RocketNet
Networking
Kernel Sockets
TCP
ibverbs
RDMA
F-stack
DPDK
Others

RocketBufs

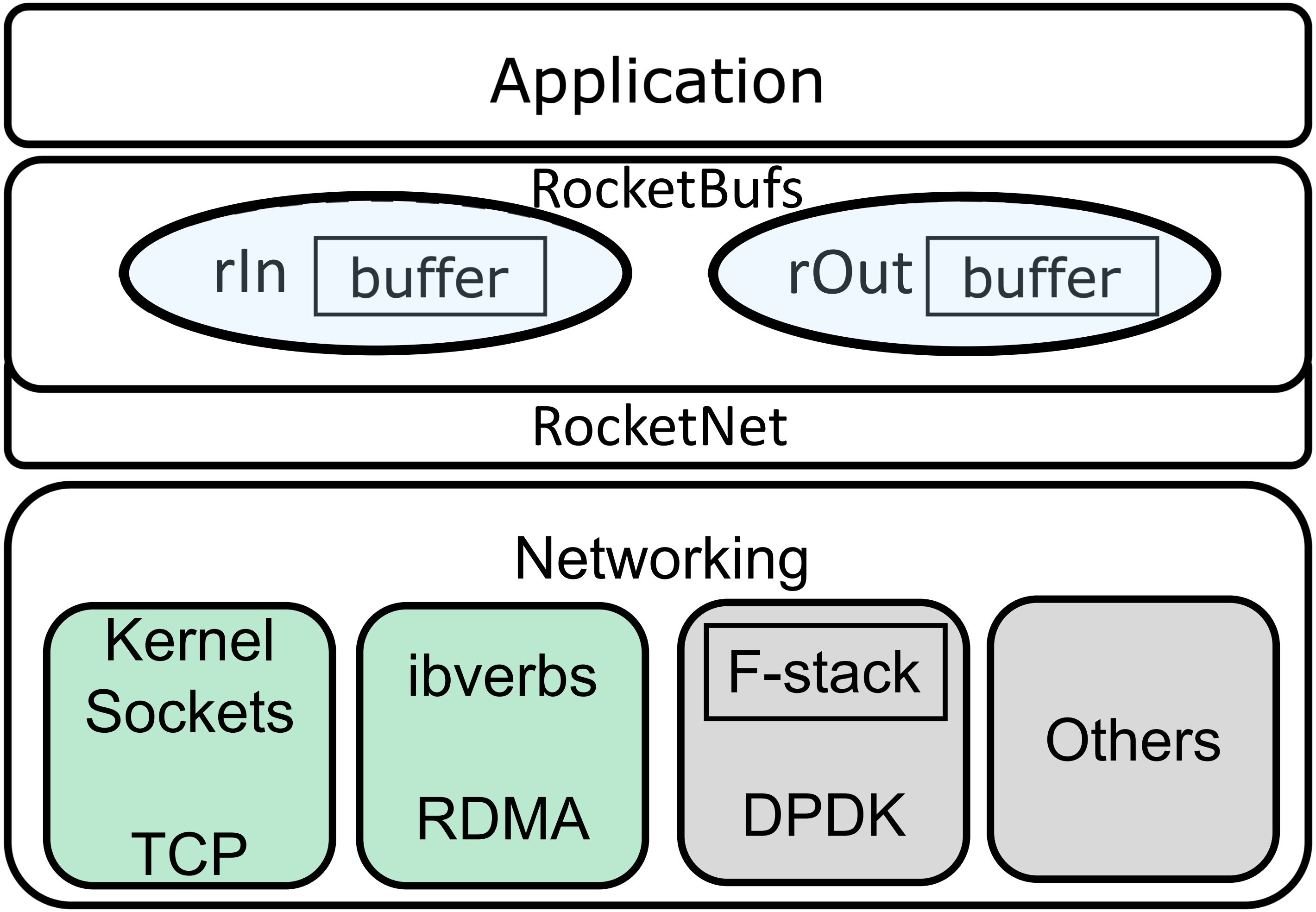

Application
RocketBufs
RocketNet
Networking
Kernel Sockets
TCP
ibverbs
RDMA
F-stack
DPDK
Others

Message-Oriented Middleware (MOM)

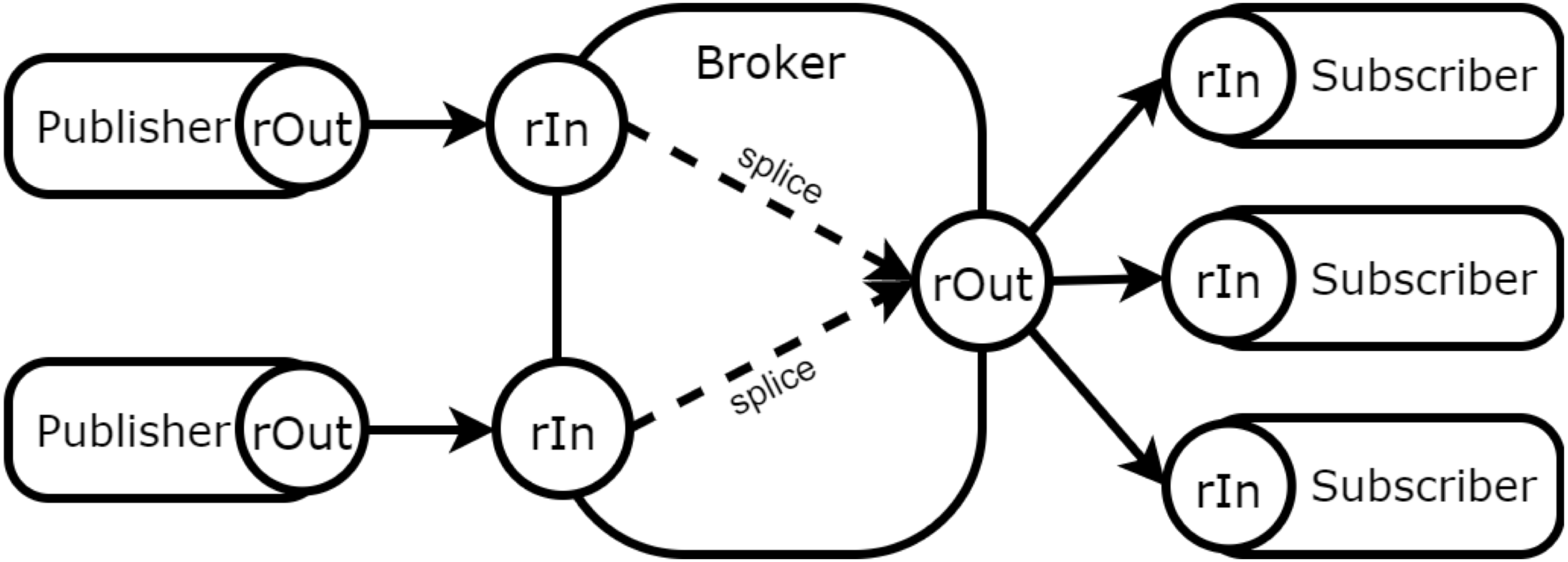


Messages copied from output buffers on one host to input buffers on another

RocketBufs

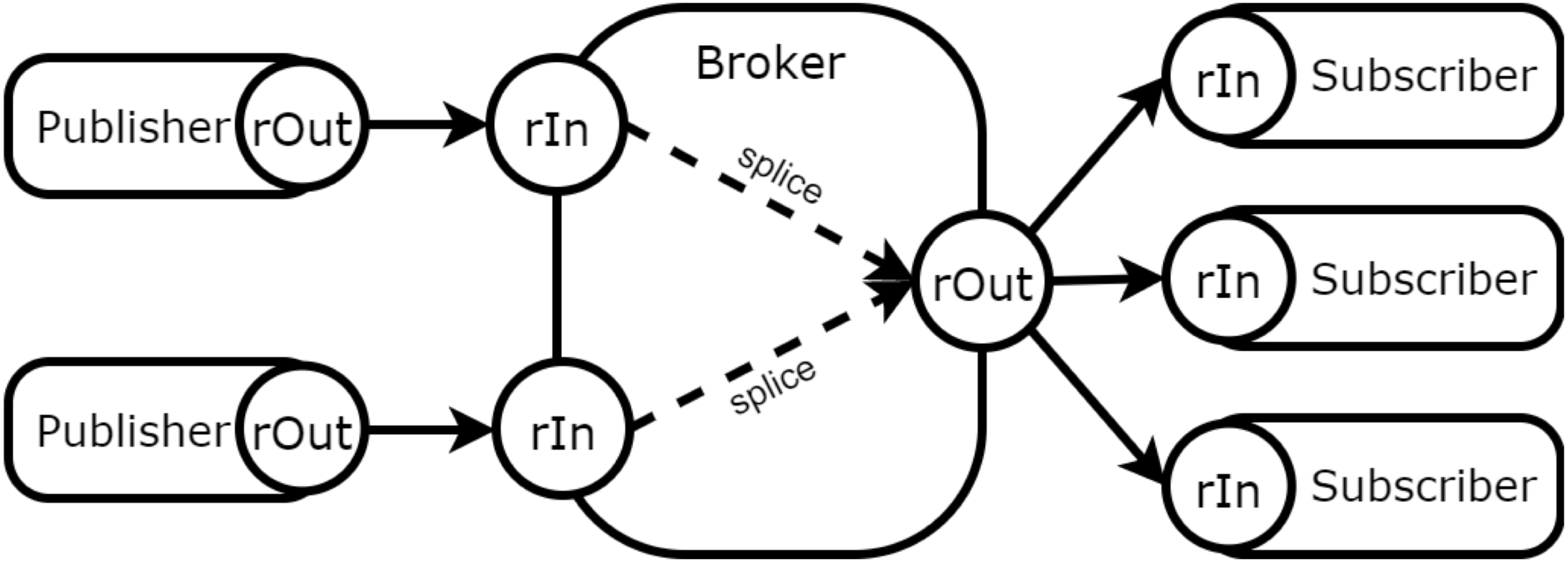

Application
RocketBufs
rIn
buffer
rOut
buffer
RocketNet
Networking
Kernel Sockets
TCP
ibverbs
RDMA
F-stack
DPDK
Others

Use Case Pub/Sub System: RBMQ



- Topics mapped to buffer
- Publishers send messages using `rOut`
- Brokers and subscribers receive messages using `rIn`
- Broker uses buffer splicing to forward messages (w/o copying)

RBMQ Evaluation

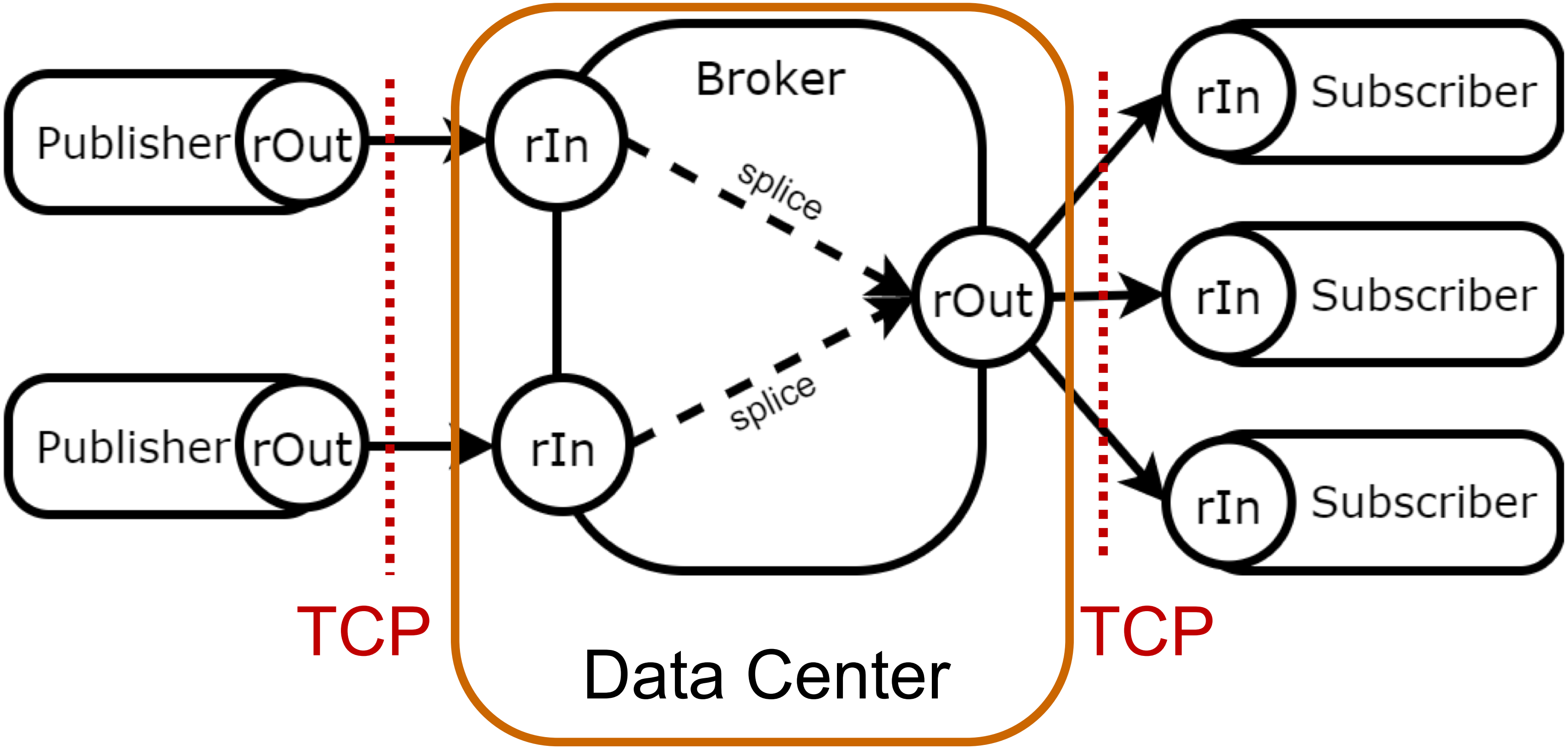


2.0 GHz Intel Xeon,
8 cores,
1x 40 Gbps NIC

2.6 GHz Intel Xeon,
10 cores
4x 40 Gbps NICs

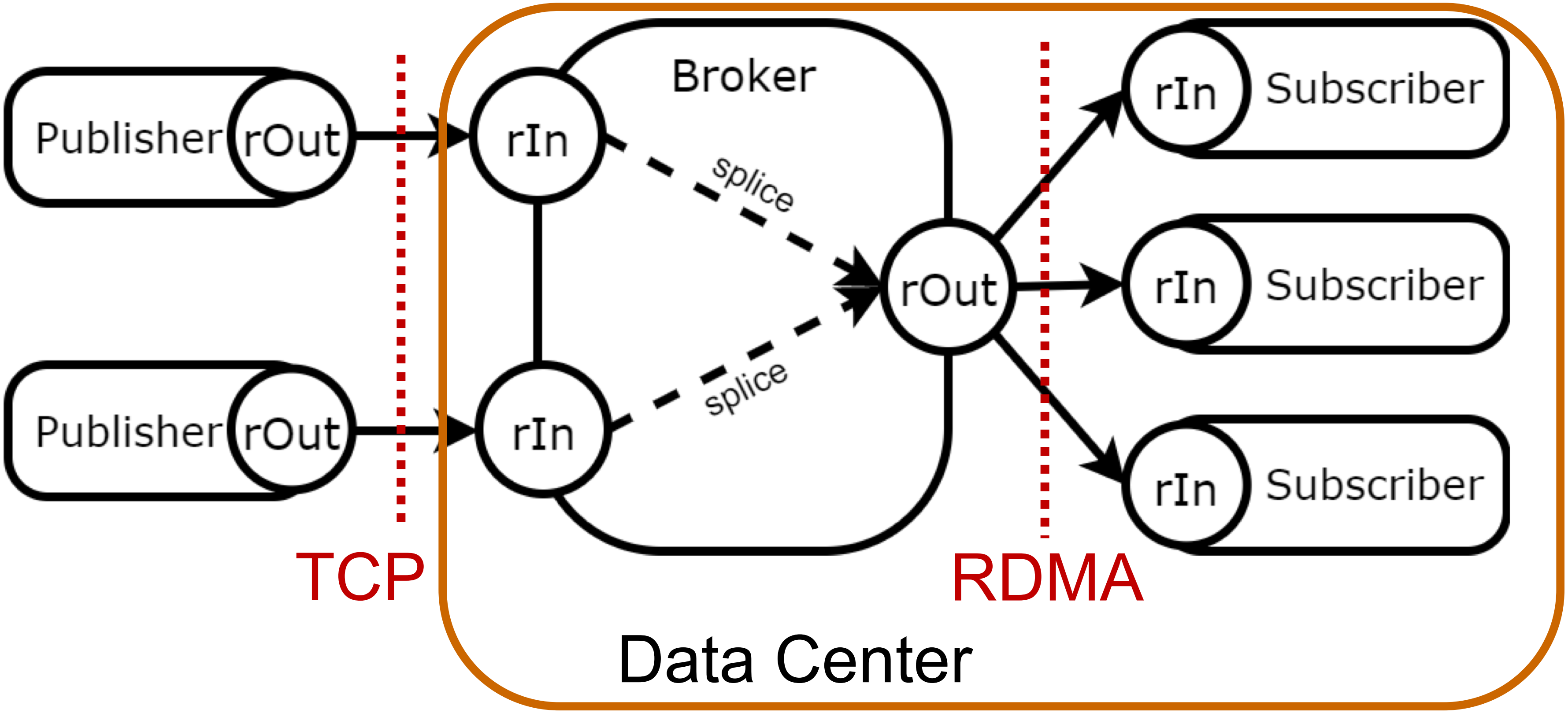
2.0 GHz Intel Xeon,
8 cores,
1x 40 Gbps NIC

RBMQ Configurations



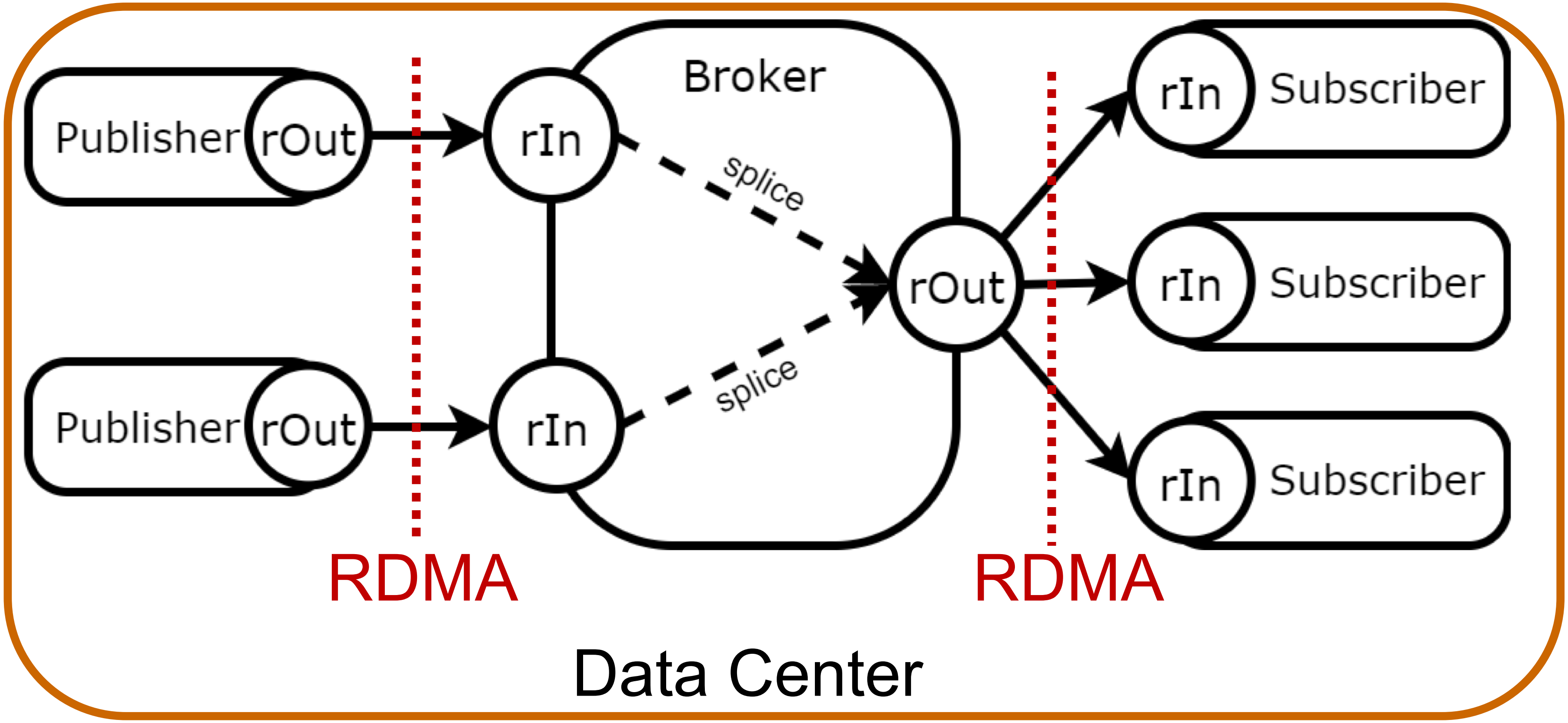
RBMQ-tcp-tcp
RBMQ-tcp-no-fc

RBMQ Configurations



RBMQ-tcp-rdma

RBMQ Configurations



RBMQ-rdma-rdma

RBMQ Comparison

Redis (pub/sub mode)

- Used by Twitter, Github, StackOverflow
- Brokers, subscribers, publishers implemented in C

RabbitMQ

- Used by Reddit, VMWare, Mozilla
- Brokers implemented in Erlang (RabbitMQ is in Erlang)
- Publishers, subscribers implemented in C

RBMQ Comparison

Redis (pub/sub mode)

- Used by Twitter, Github, StackOverflow
- Brokers, subscribers, publishers implemented in C

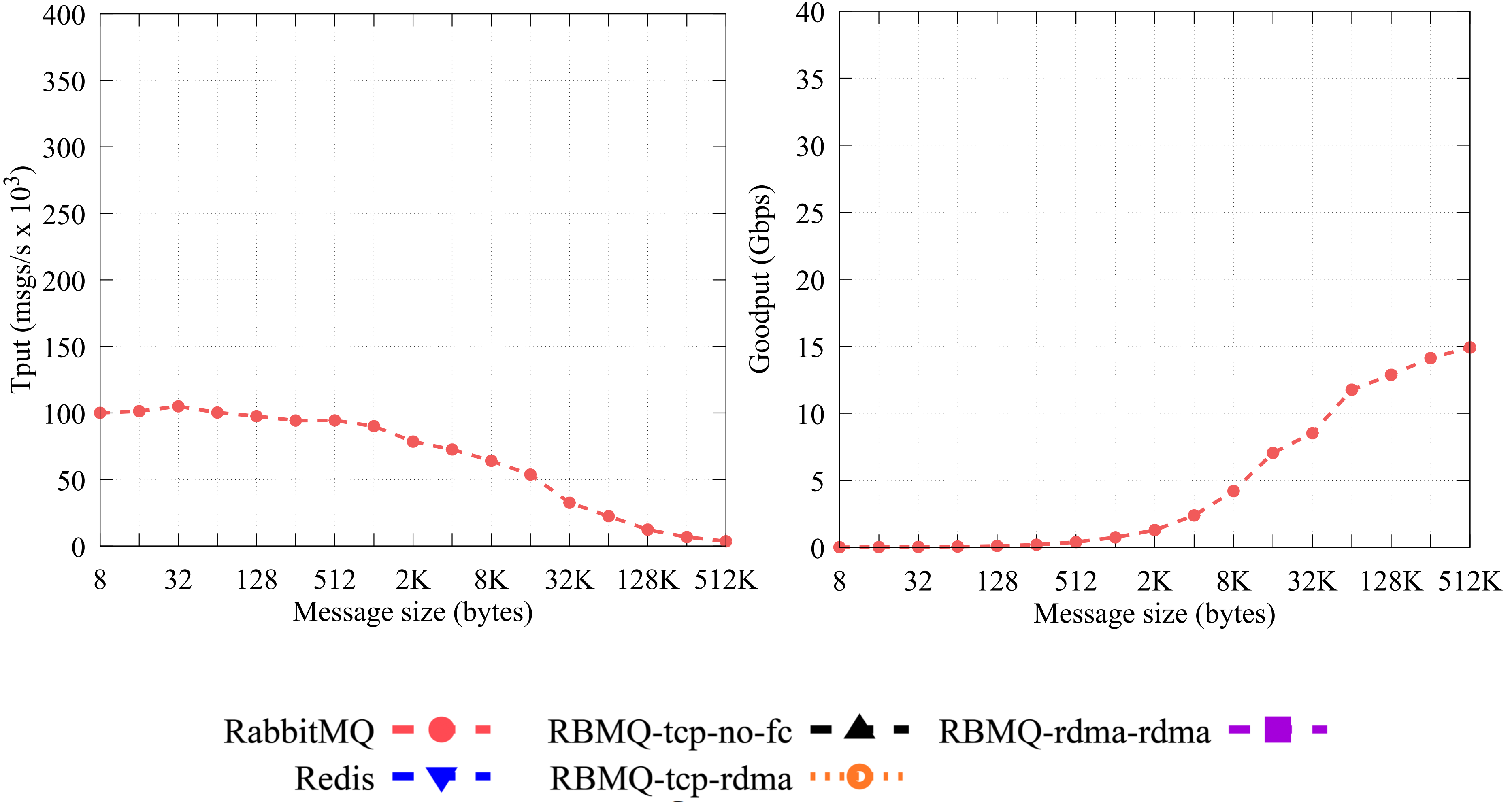
RabbitMQ

- Used by Reddit, VMWare, Mozilla
- Brokers implemented in Erlang (RabbitMQ is in Erlang)
- Publishers, subscribers implemented in C

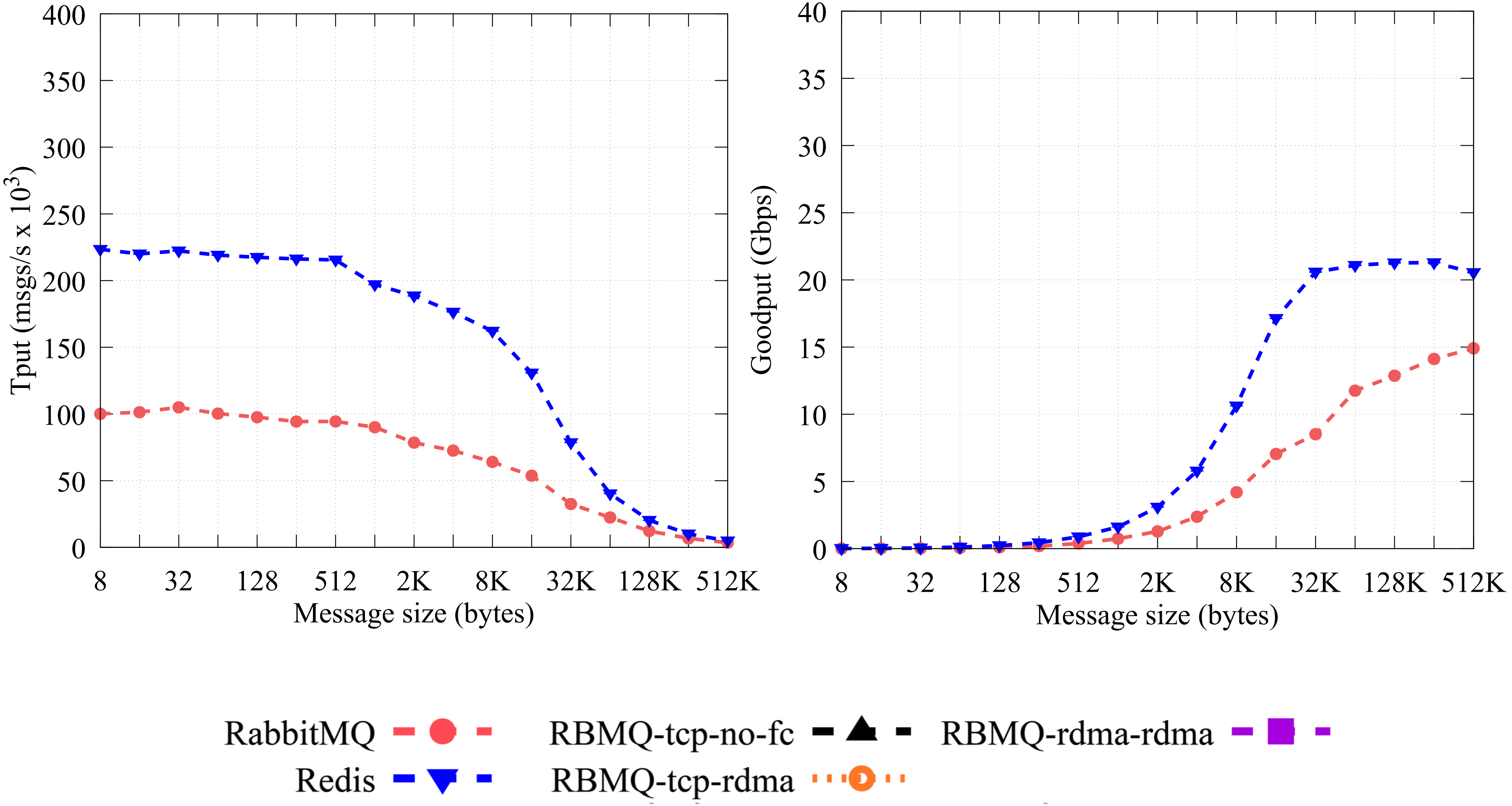
Broker disseminates all messages to all subscribers

Measure: Throughput, CPU utilization, latency (see paper)

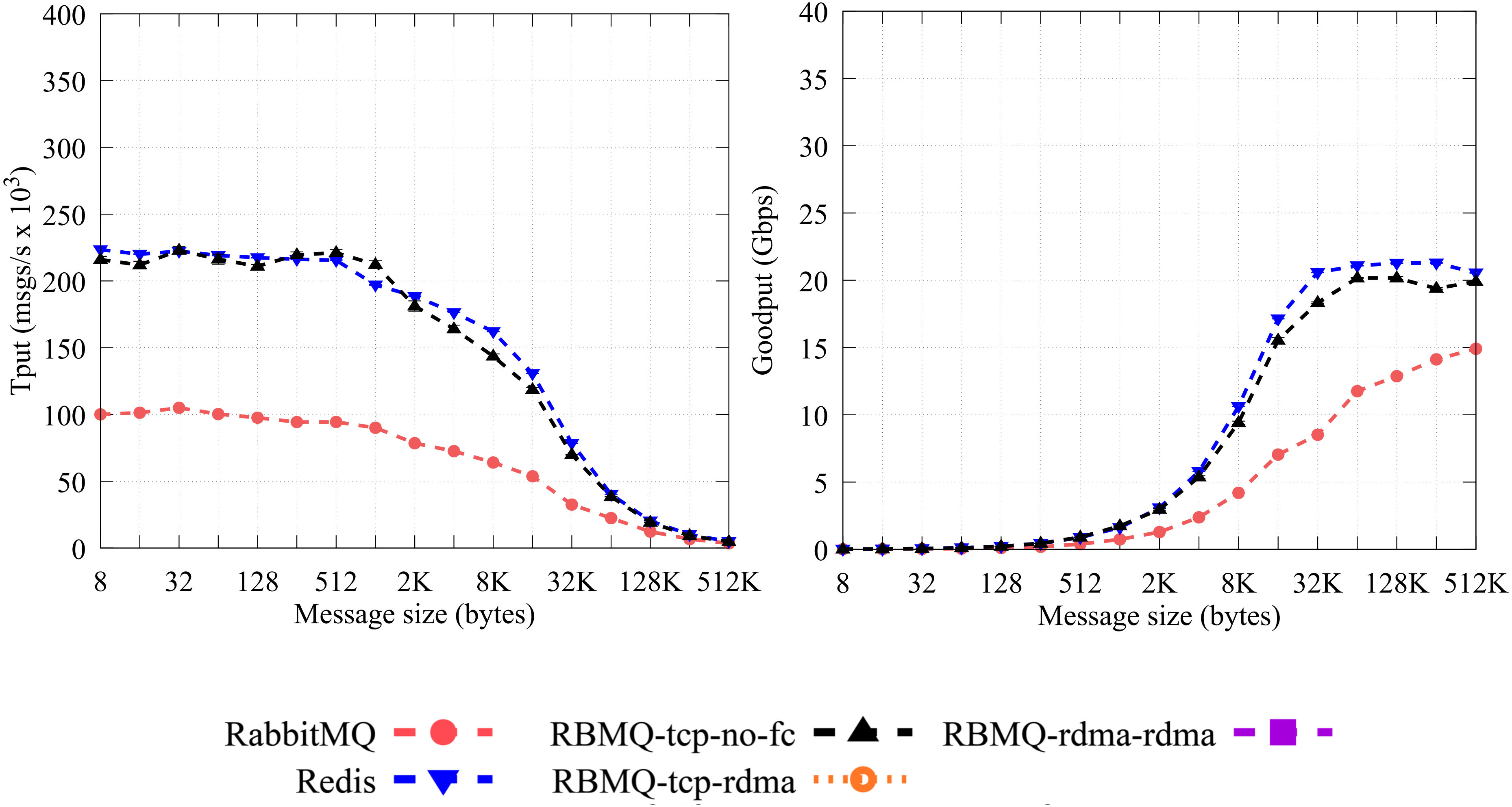
System Throughput (4 subscribers)



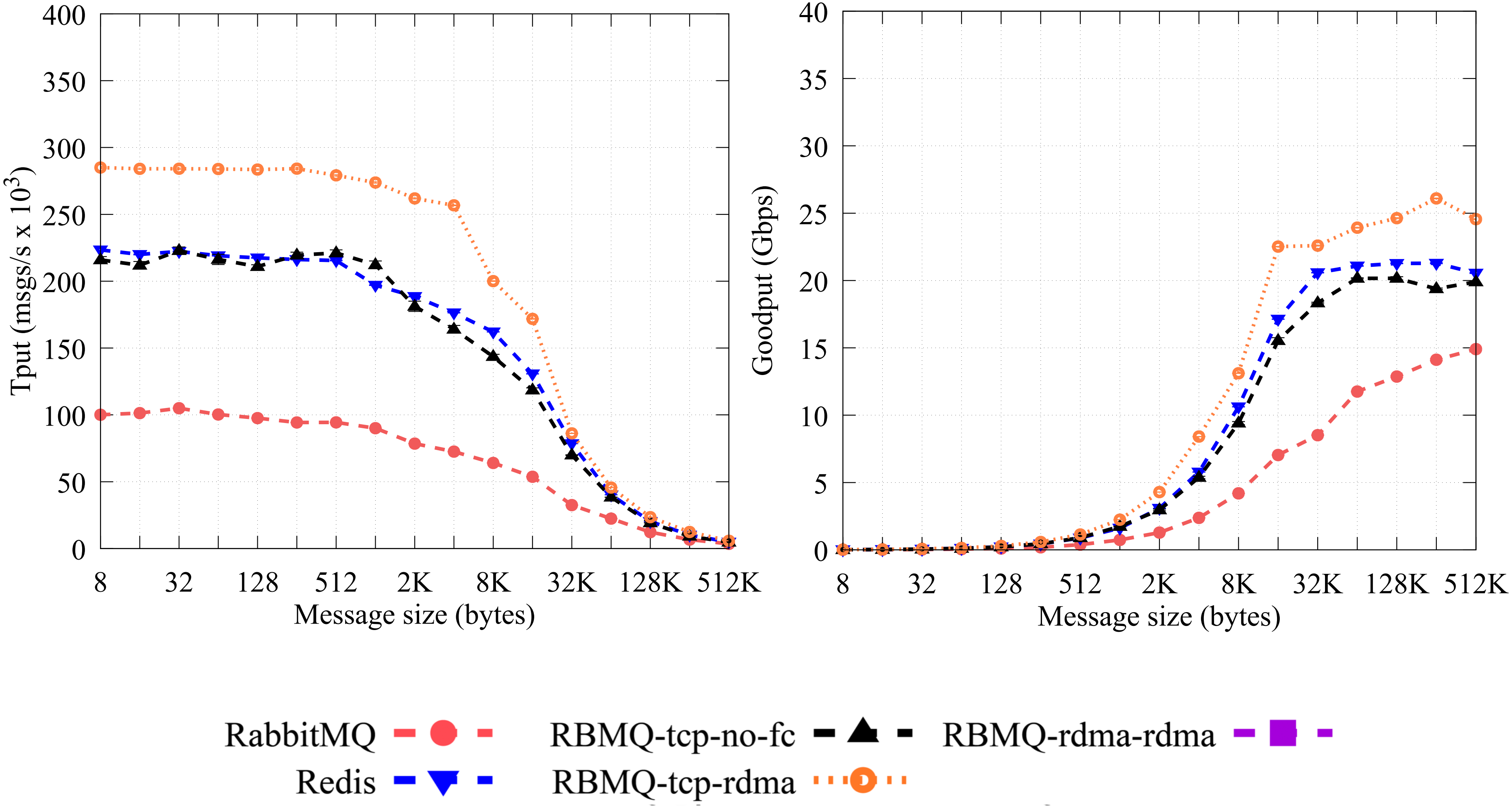
System Throughput (4 subscribers)



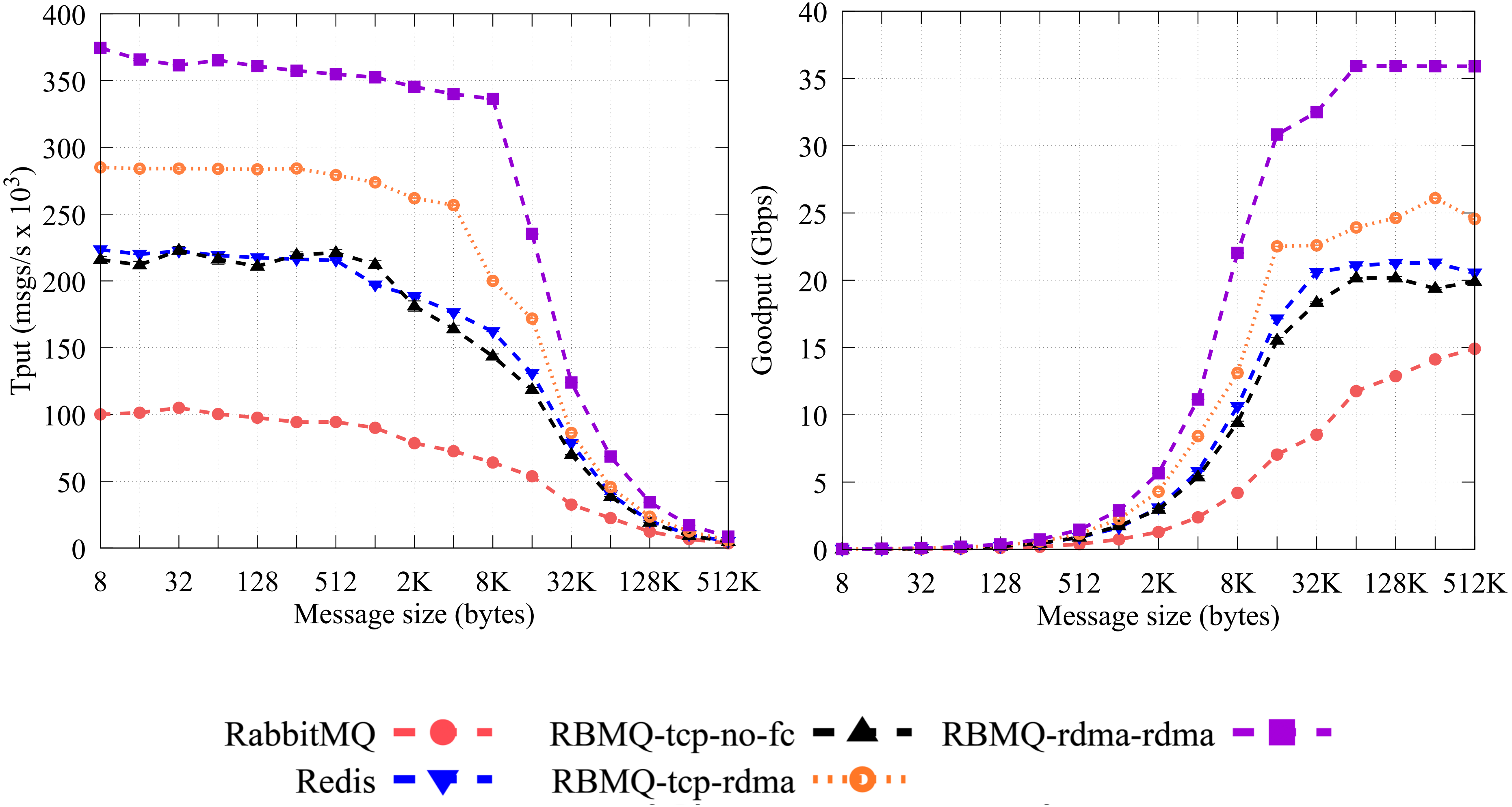
System Throughput (4 subscribers)



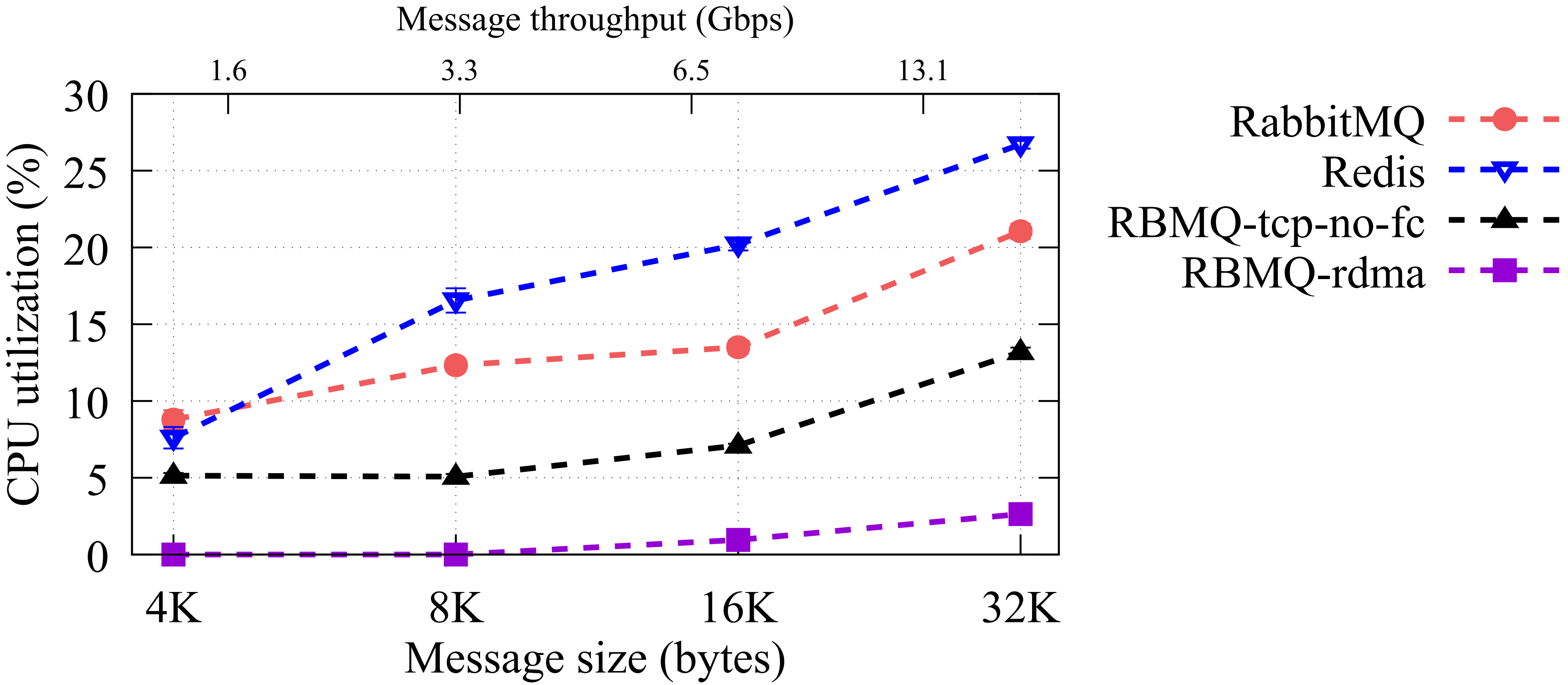
System Throughput (4 subscribers)



System Throughput (4 subscribers)



Subscriber CPU utilization



Conclusions

RocketBufs: framework for high-performance MOM systems

- Memory-based buffer abstractions
- APIs for producing, disseminating, and subscribing to data
- Supports buffer splicing and flexible MOM topologies
- Supports multiple transport protocols (no application code changes)

Performance:

- Good TCP performance compared to production systems
- RDMA produces significant performance gains

Discussion Points (stuff not in the talk)

See the paper and Huy Hoang's M.Math Thesis for:

- API details / code snippets
- Latency results
- Live streaming video application and performance results

Possible Future Work:

- Ports to other transports (e.g., DPDK, TLS, etc.)
- Modify an existing MOM to use RocketBufs (e.g., Redis)
- Fault tolerance and persistence