RDMA Protocols

- InfiniBand (IB): native RDMA protocol
- iWARP
- RoCE (RoCE v2): RDMA over Commercial Ethernet

InfiniBand

- need to overhaul the datacenter with Infiniband enabled switches and NICs.

Table 3. InfiniBand Link Rates

InfiniBand Link	Signal Count	Signalling Rate	Data Rate	Fully Duplexed Data Rate
1X	4	2.5 Gb/s	2.0 Gb/s	4.0 Gb/s
4X	16	10 Gb/s	8 Gb/s	16.0 Gb/s
12X	48	30 Gb/s	24 Gb/s	48.0 Gb/s

Note: The bandwidth of an InfiniBand 1X link is 2.5 Gb/s. The actual raw data bandwidth is 2.0 Gb/s (data is 8b/10b encoded). Due to the link being bi-directional, the aggregate bandwidth with respect to a bus is 4 Gb/s. Most products are multi-port designs where the aggregate system I/O bandwidth will be additive.

Introduction to InfiniBand (White Paper)

iWARP & RoCE over Ethernet

- iWARP: very heavy weight, lower performance: implements the entire TCP/IP stack in the NIC 👎
  - an “ill-conceived attempt”, 10 Gbps

- RoCE: operating over standard layer 2 and layer 3 Ethernet switches
  - 40 Gbps
  - Microsoft: scale RoCEv2 beyond VLAN

WHITE PAPER: RoCE vs. iWARP Competitive Analysis
RDMA over Commodity Ethernet at Scale, SIGCOMM’ 16

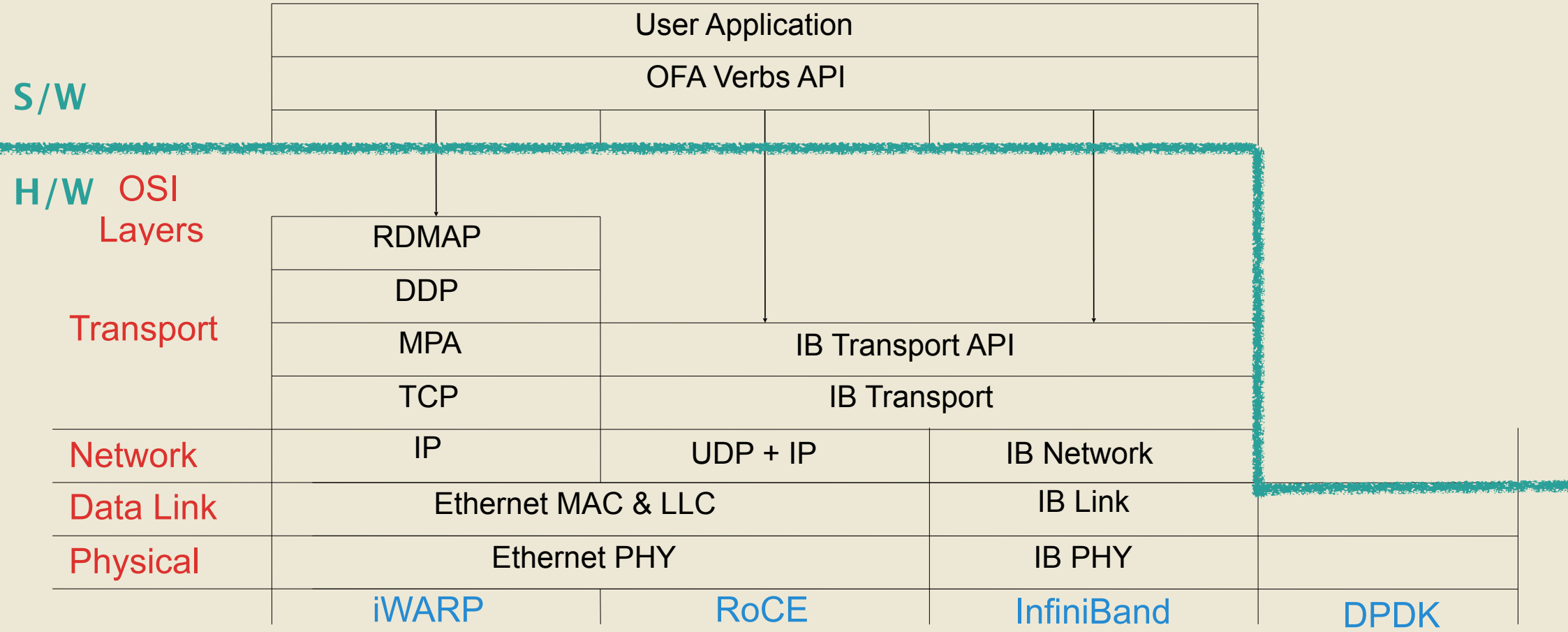
RDMA needs a lossless network!

- InfiniBand: credit-based algorithm
- RoCEv2: PFC (Priority-based Flow Control)
  - IRN: eliminate the need for PFC (SIGCOMM' 18)
- iWARP: TCP

Revisiting Network Support for RDMA, SIGCOMM' 18

DPDK

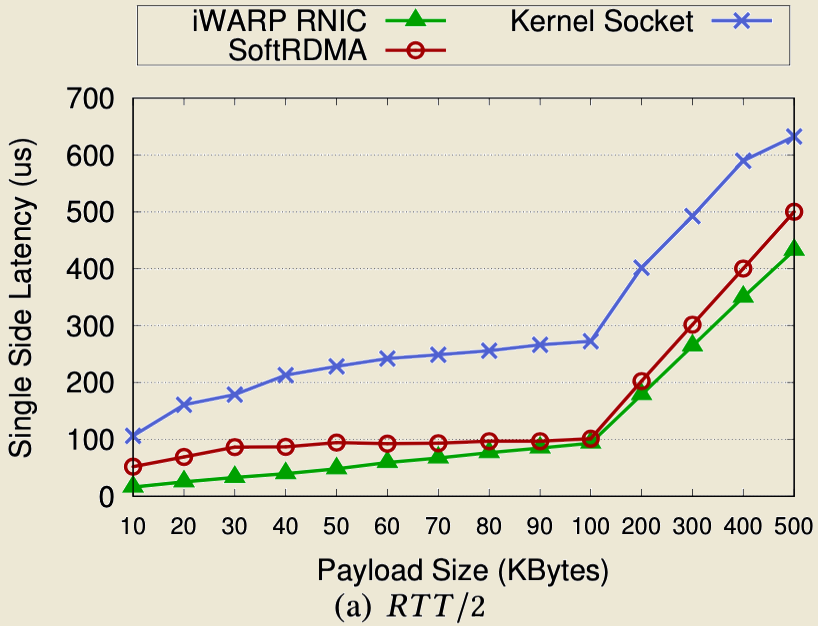
- provide
  - kernel bypassing (zero copy)
- does not provide
  - IP forwarding
  - firewalls
  - TCP or UDP



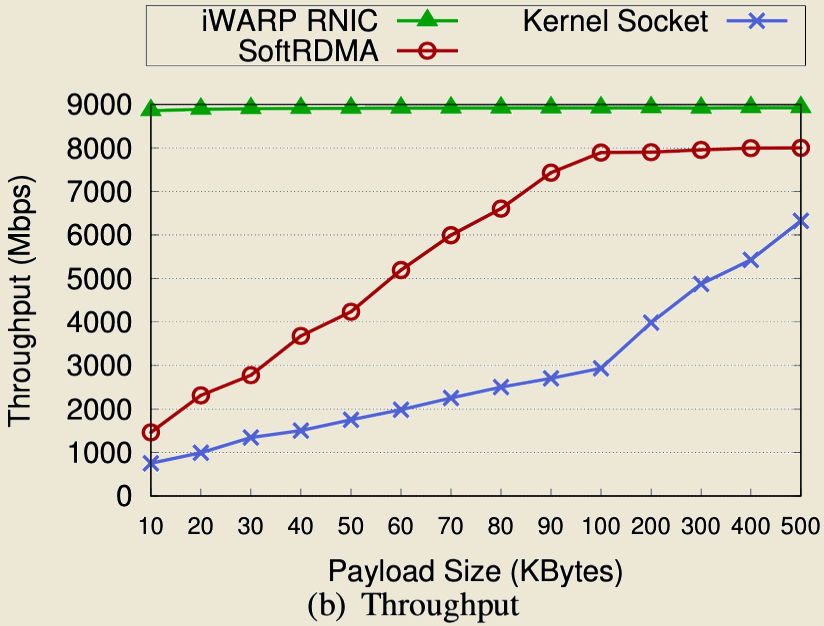
Introduction to RDMA Programming
WHITE PAPER: RoCE vs. iWARP Competitive Analysis
WHITE PAPER: RoCE in the Data Center

SoftRDMA

- implement a entirely user-level iWARP stack (DPDK, LwIP, iWARP)
- One copy: NIC is unaware about the application-appointed place



(a) $RTT/2$



(b) Throughput

SoftRDMA: Rekindling High Performance Software RDMA over Commodity Ethernet, APNet'17

mTCP

- focuses on small message transactions on multicore systems
  - Implementing TCP in the user level, One copy
  - batch of packet-level and socket-level events (reduce IPC overhead)
  - CPU Cache Locality

- 25x faster than latest Linux TCP stack

mTCP: A Highly Scalable User-level TCP Stack for Multicore Systems

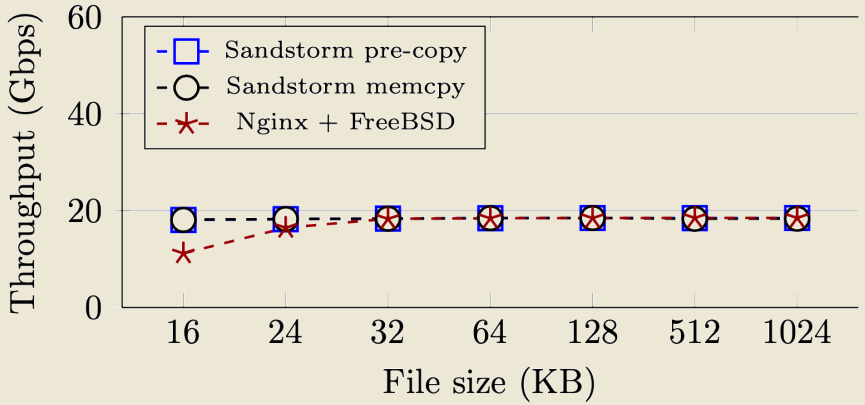
Sandstorm

- a specialized userspace stack for serving static content
- implement a complete zero-copy stack (netmap)
- under high load, the same packet may need to be sent more than once: pre-copy stack

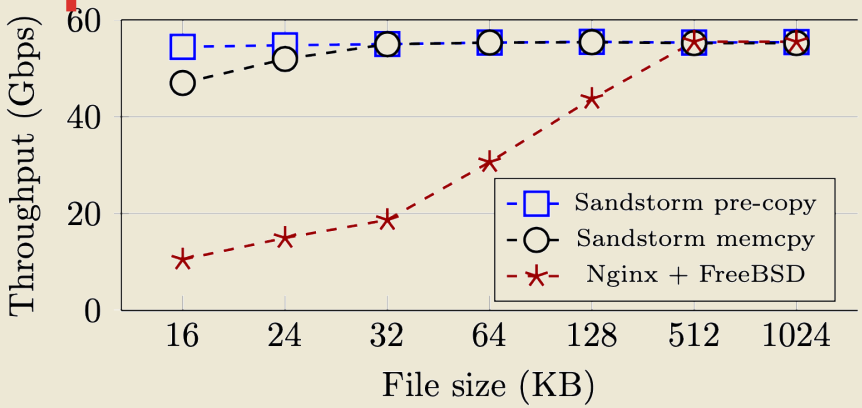
Network Stack Specialization for Performance, SIGCOMM' 14

Sandstorm

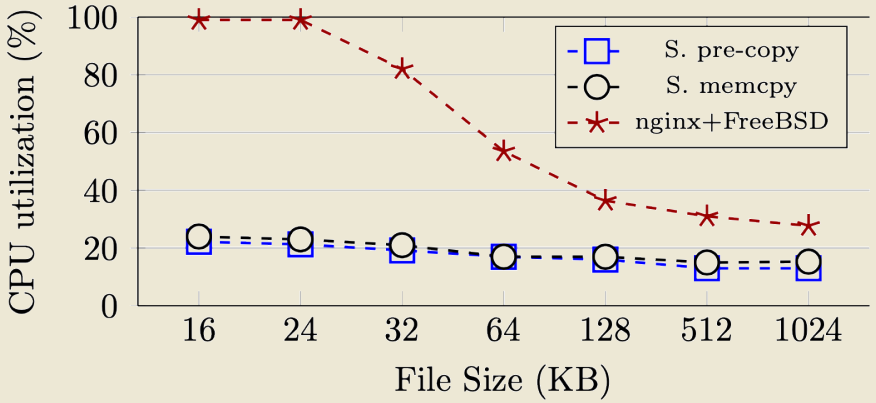
CPU limits the performance



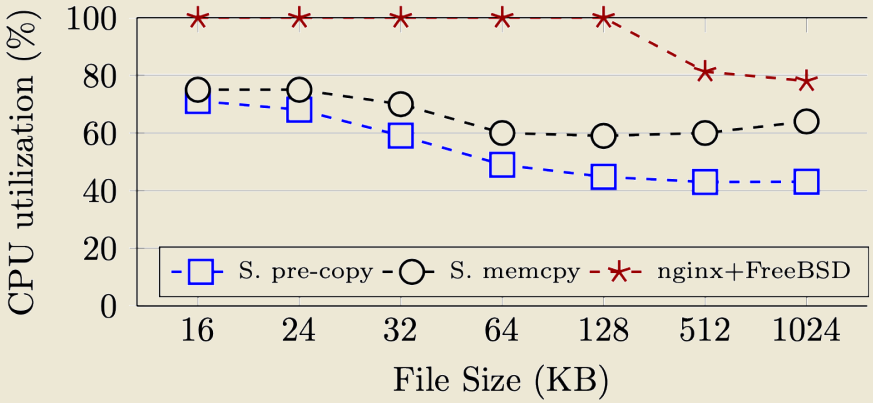
(a) Network throughput, 2 NICs



(b) Network throughput, 6 NICs



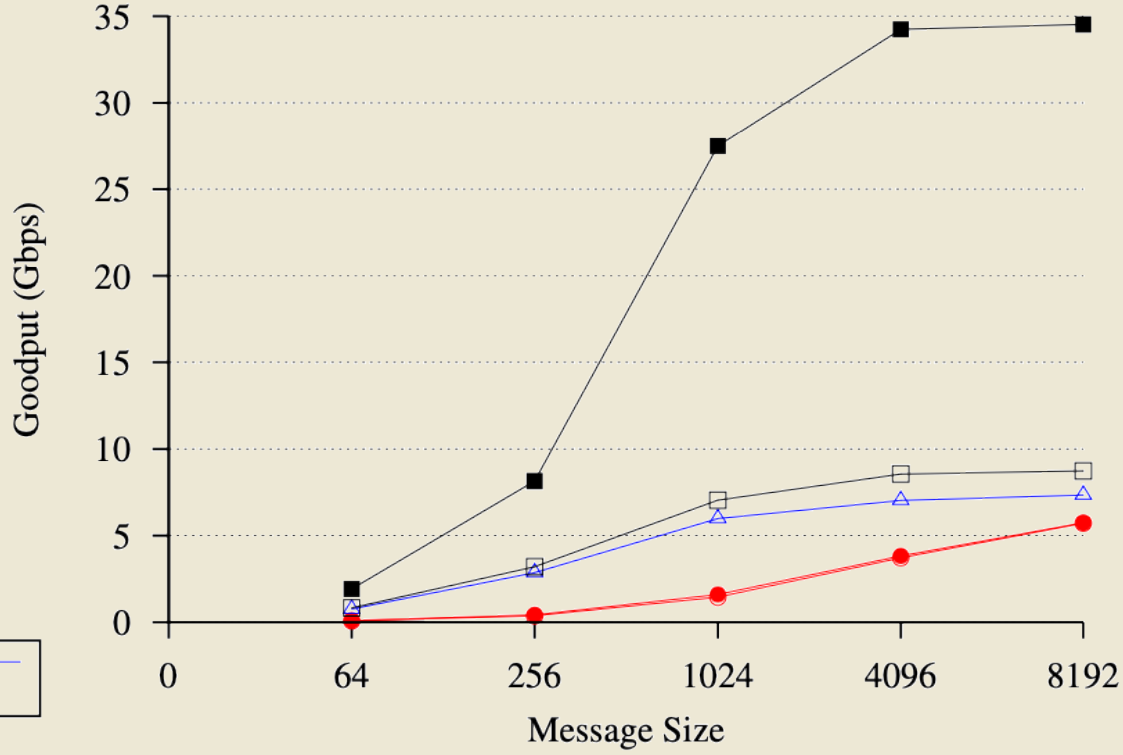
(c) CPU utilization, 2 NICs



(d) CPU utilization, 6 NICs

IX: A Protected Dataplane Operating System

- Dune, DPDK, LwIP
- a networking stack can be implemented in a protected OS kernel
- Zero-copy



(c) Different message sizes s (n=1)

IX: A Protected Dataplane Operating System for High Throughput and Low Latency, nsdi' 14
Dune: Safe User-level Access to Privileged CPU Features
Arrakis: The Operating System Is the Control Plane

Lessons

- loss rate on wireless path and wired path?
- starting point: DPDK, lwIP
- the best batch size? adaptive batching strategy?
- at least one copy?