Corundum:

An Open-Source FPGA-NIC

Material from FCCM 2020 talk &

Sigcomm 2020 talk (next week)

Alex Forencich (jforenci@eng.ucsd.edu)
George Papen (gpapen@eng.ucsd.edu)
University of California at San Diego

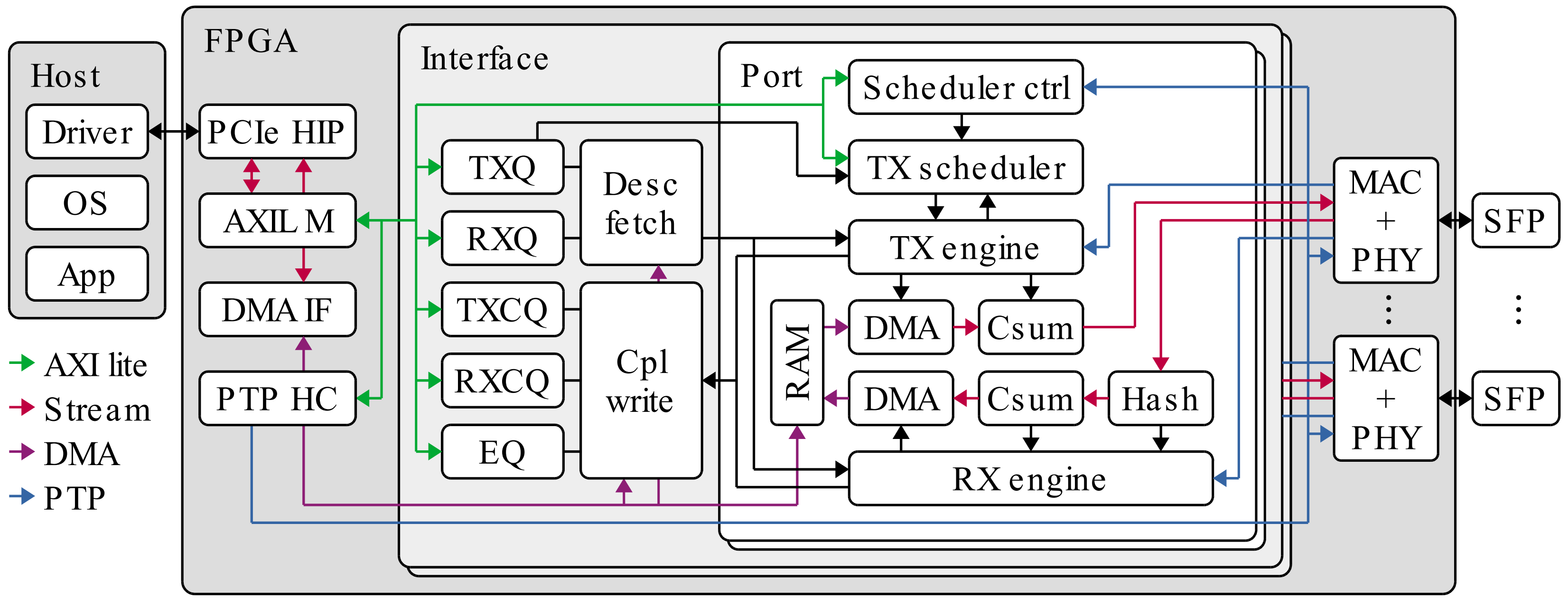
Motivation for FPGA-based NIC

UC San Diego
JACOBS SCHOOL OF ENGINEERING

- Precise control of packet transmission from servers has many applications
  - Rate limiting, flow control, congestion control, TDMA, circuit switching, etc.
- Deterministic sub-microsecond control not feasible in software
- Little hardware support in commercial NICs
  - Limited number of queues, limited control over transmit scheduling
  - Commercial SmartNICs designed for packet processing, not flow control
- Possible to do on an FPGA
  - But there are no extensible high-performance FPGA-based NIC reference designs to use as a foundation

Corundum NIC

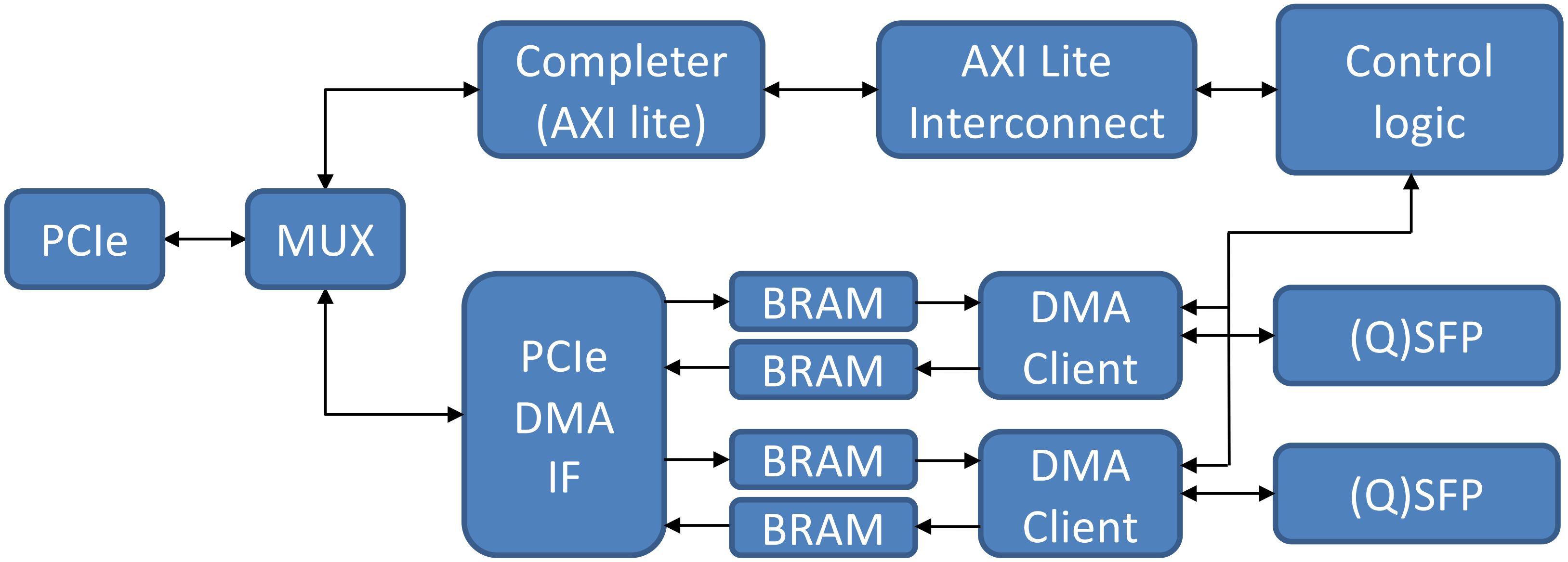
UC San Diego
JACOBS SCHOOL OF ENGINEERING



- Corundum is a reference open-source FPGA-based NIC supporting many FPGA platforms
- Provides a network interface similar in performance to a commercially-available NIC
- Enables the implementation of additional hardware features needed for circuit switching and “in-network” computing.

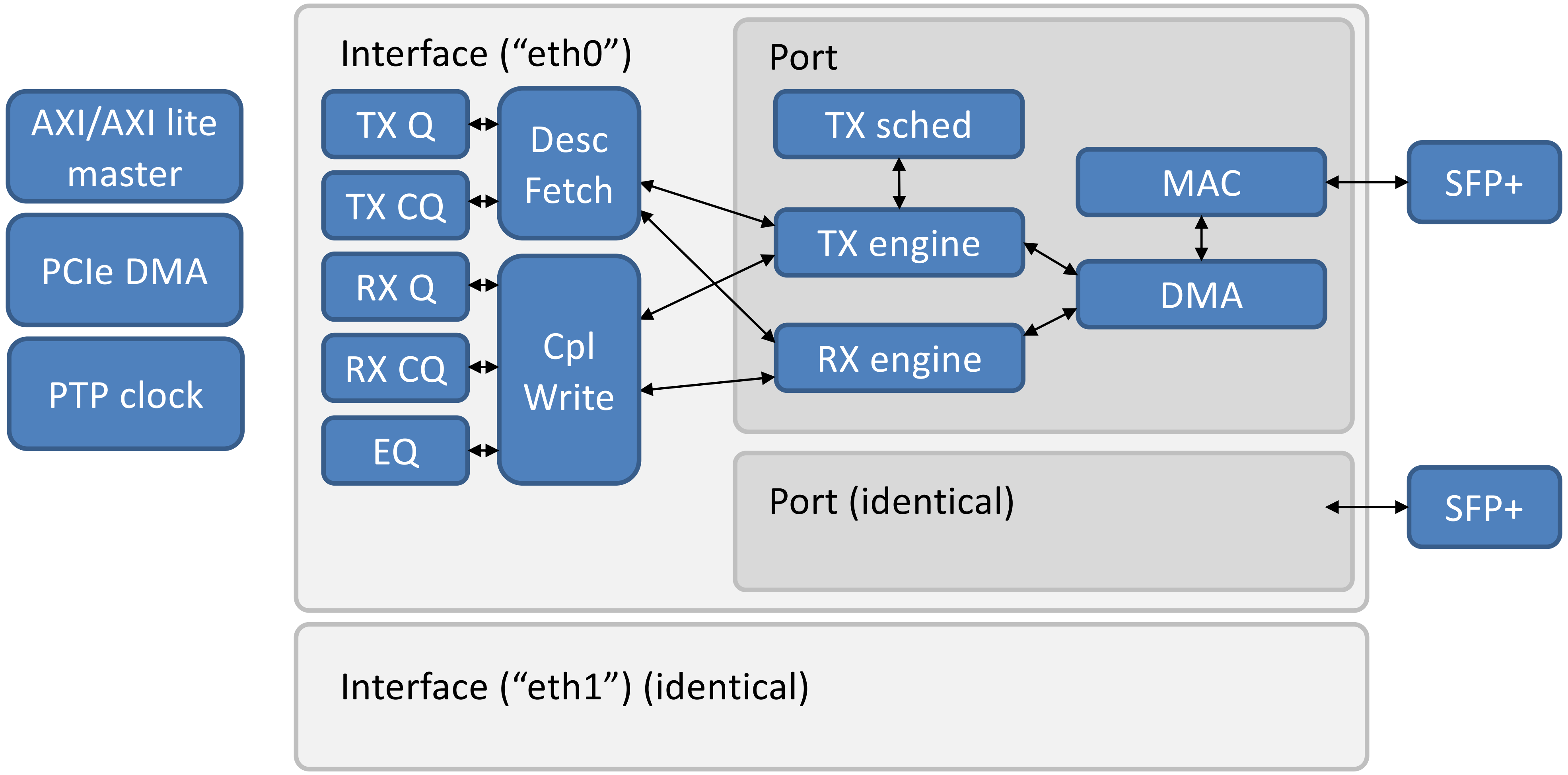
NIC Datapath

UC San Diego
JACOBS SCHOOL OF ENGINEERING



Queue Handling

UC San Diego
JACOBS SCHOOL OF ENGINEERING



Scalable Queue Management

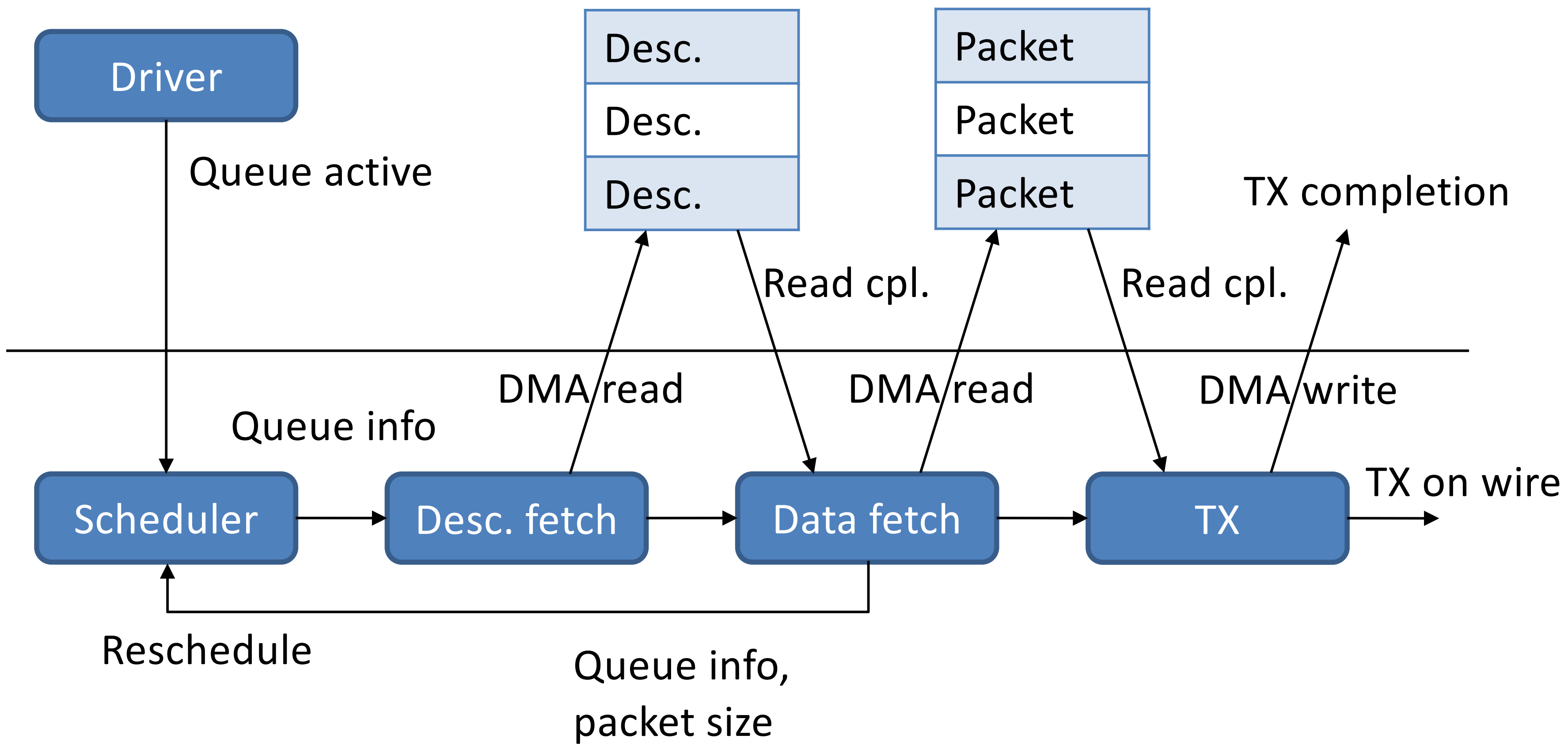
UC San Diego
JACOBS SCHOOL OF ENGINEERING

- Each queue is an independent channel between SW and HW
- Traditional NICs support ~100 queues for load-balancing
  - Each CPU core or virtual machine gets 1 queue
- Queue management logic stores queue state in block RAM
  - 128 bits/queue
  - Scalable to 10,000+ queues
  - Permits fine-grained, per-flow or per-destination control

Transmit scheduler

- Determines which queues to transmit from
- Default scheduler is round-robin
  - Cycles through all active queues, sending one packet at a time
- Possible to extend or replace scheduler
  - Implement rate limiting, WFQ, etc.
  - Ex: SENIC, Carousel, PIEO, Loom
- Possible to enable/disable queues based on events or PTP time
  - Implement TDMA
  - HW congestion control – HPCC, NDP

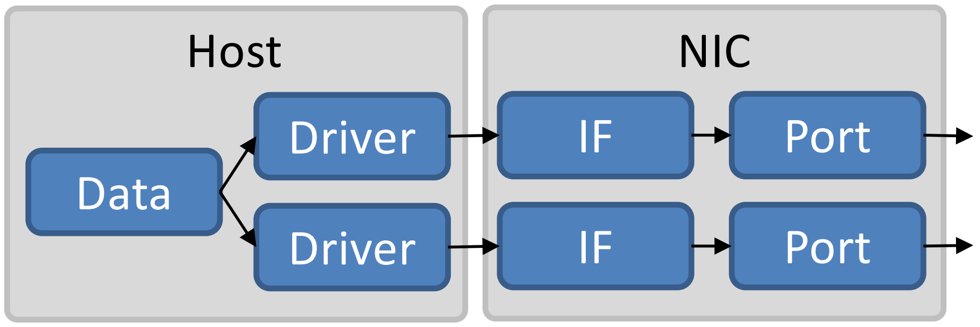
TX Operation



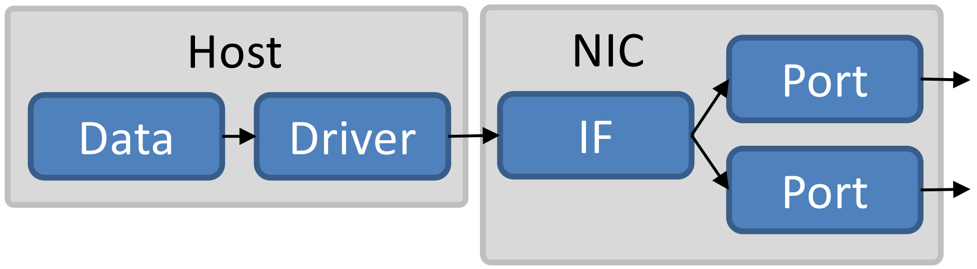
Ports and Interfaces

UC San Diego
JACOBS SCHOOL OF ENGINEERING

- Hardware support for multiple uplinks
- Multiple physical ports appear as single OS-level interface
- Ports have separate schedulers
- Migrate or stripe flows across ports by changing scheduler settings



Traditional NIC: assignment in software



Corundum NIC: assignment in hardware

Host interface

- NIC transfers packet data via direct memory access (DMA) over PCI express
- Custom DMA engine to perform transfers over PCIe
  - Existing cores (i.e. XDMA) could not be easily controlled or simulated
  - DMA engine integral to performance of NIC
- PCIe simulation framework in Python/MyHDL for testing NIC
  - Event-driven, transaction-layer simulation
  - Includes root complex, switches, FPGA PCIe interface core models, etc.
  - Supports MMIO, DMA, P2P DMA, BARs, configuration space, etc.

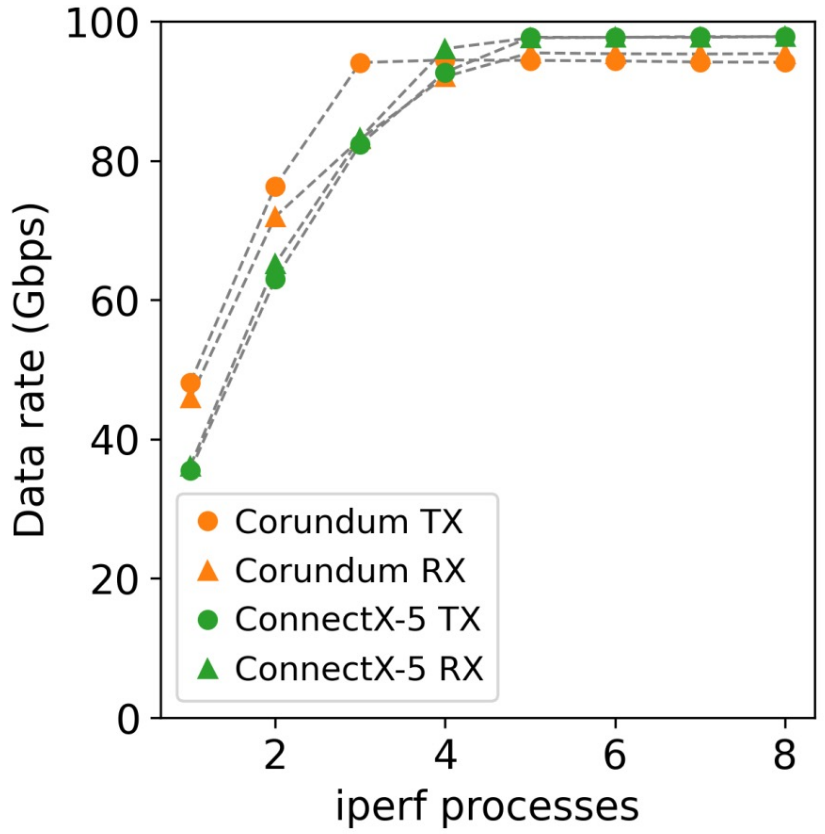
Performance evaluation

UC San Diego
JACOBS SCHOOL OF ENGINEERING

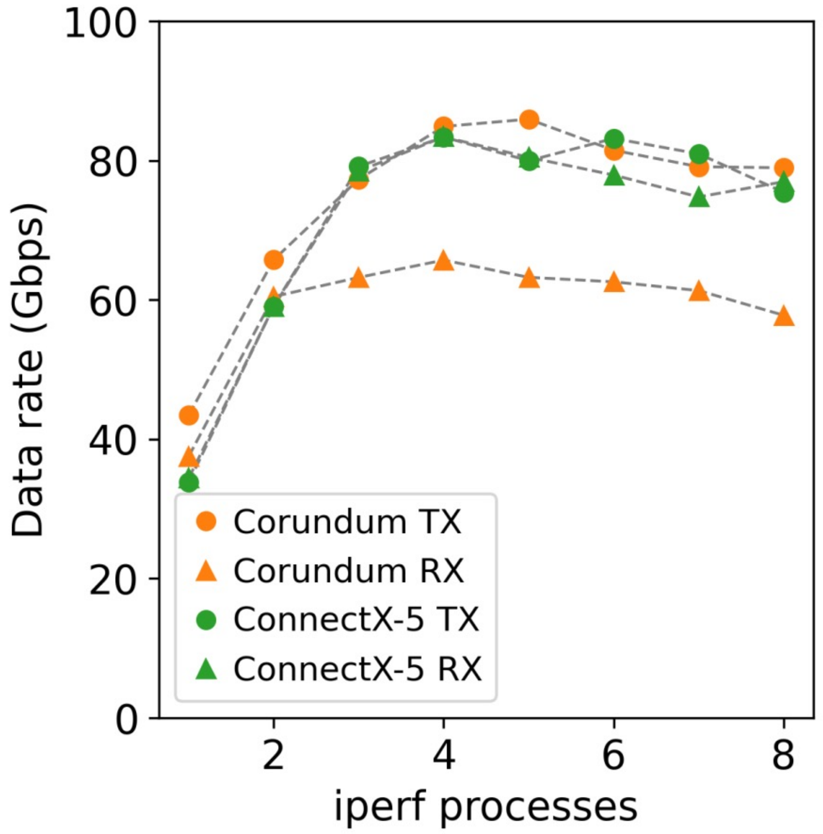
- Installed 100G Corundum NIC in Dell R540 server
  - Alpha Data ADM-PCIE-9V3
  - Dual socket Intel Xeon Gold 6138 (20 cores)
- Direct connection to Mellanox ConnectX-5 NIC
- Loaded link with multiple instances of iperf3
- Tested MTU 1500 B and 9000 B
- Compared against pair of Mellanox ConnectX-5 NICs
  - Data is somewhat old.

Performance: 9 KB MTU

UC San Diego
JACOBS SCHOOL OF ENGINEERING



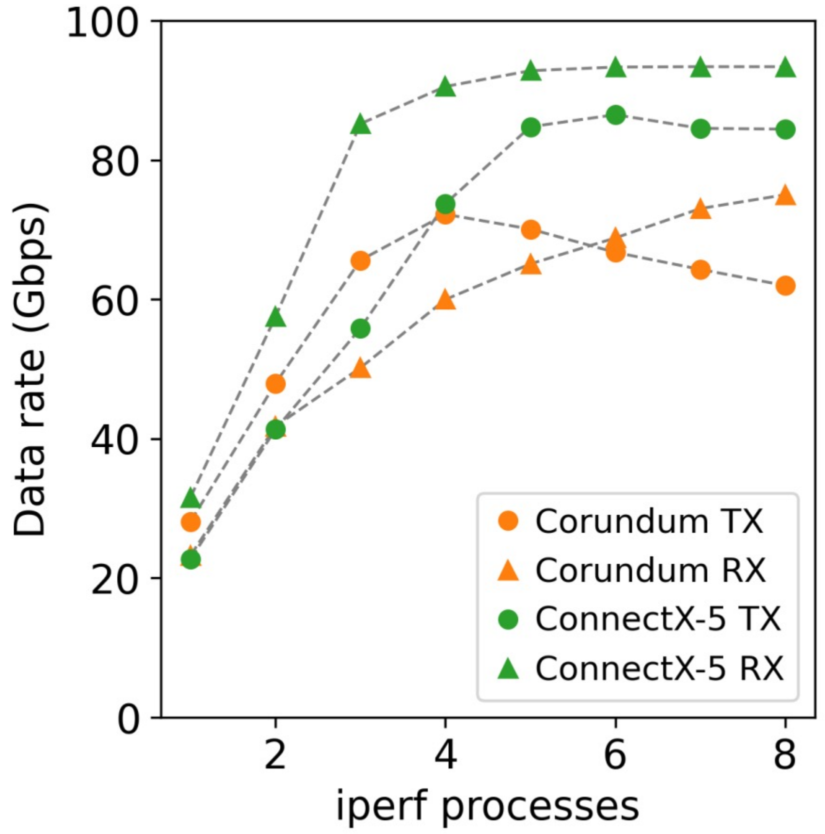
Intel host, 9KB MTU, separate



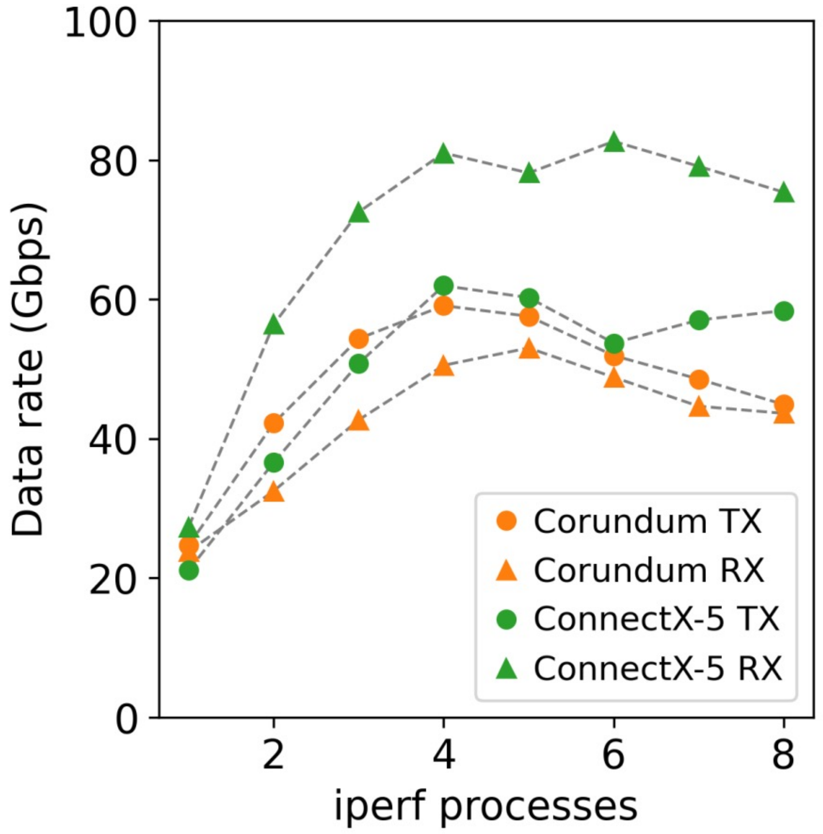
Intel host, 9KB MTU, simultaneous

Performance: 1.5 KB MTU

UC San Diego
JACOBS SCHOOL OF ENGINEERING



Intel host, 1.5KB MTU, separate



Intel host, 1.5KB MTU, simultaneous

Corundum Ecosystem

UC San Diego JACOBS SCHOOL OF ENGINEERING

- Github repository – public since SIGCOMM 2019 (August)
- Google group
- Hit front page of Hacker News in January (~10k views on GH)
- Published in FCCM 2020
- Interest from many groups including Microsoft Research, NetFPGA team in Cambridge, IBM research, UIC, UW-Madison, University of Fribourg, NUDT
- Working with Xilinx Research Labs to include Corundum in OpenNIC/NetFPGA 2020

Corundum Development Roadmap

- Driver development
  - Improve kernel driver performance
  - Investigate supporting DPDK/MPI/libfabric
- Gateware improvements
  - Variable length descriptors, metadata, SR-IOV, etc.
  - Firmware updates, transceiver access, etc.
- A suite of hardware shims for Corundum that can enable innovative networking applications
- End goal: Open “full-stack” optical-networking ecosystem
  - Corundum + DPDK driver + userspace network stack

Applications

- Datapath for novel transmit schedulers
- Instrument Corundum for performance measurements
- Direct transceiver access permits physical-layer measurements and development of new wireline protocols
- Corundum can be used in isolation as a NIC, or as part of a larger system as a packet DMA engine
- Discuss two applications:
  - TDMA for microsecond circuit switching
  - PHY layer BER measurement for link characterization

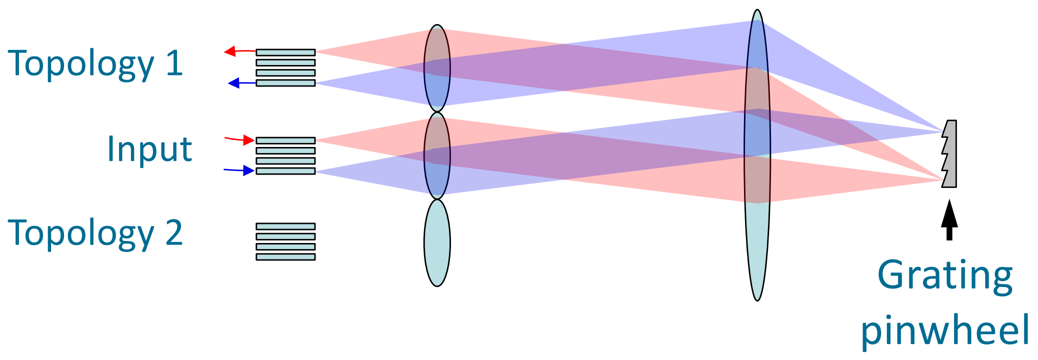
Corundum for optical networking

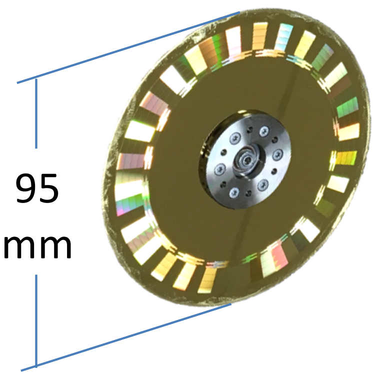
- Features specific for circuit switching:
  - Microsecond precision time synchronziation
  - Microsecond precision TDMA
  - Numerous hardware queues for per-destination control
  - PHY layer link characterization
  - PHY layer access for custom line protocols
  - Reconfigurable hardware for implementing custom routing protocols

Rotor switch prototype

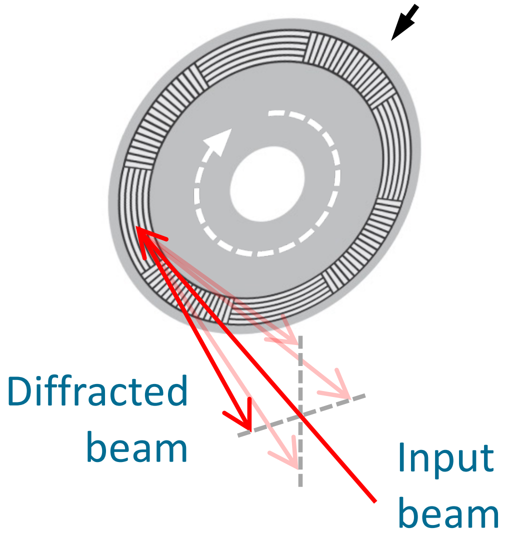
UC San Diego
JACOBS SCHOOL OF ENGINEERING

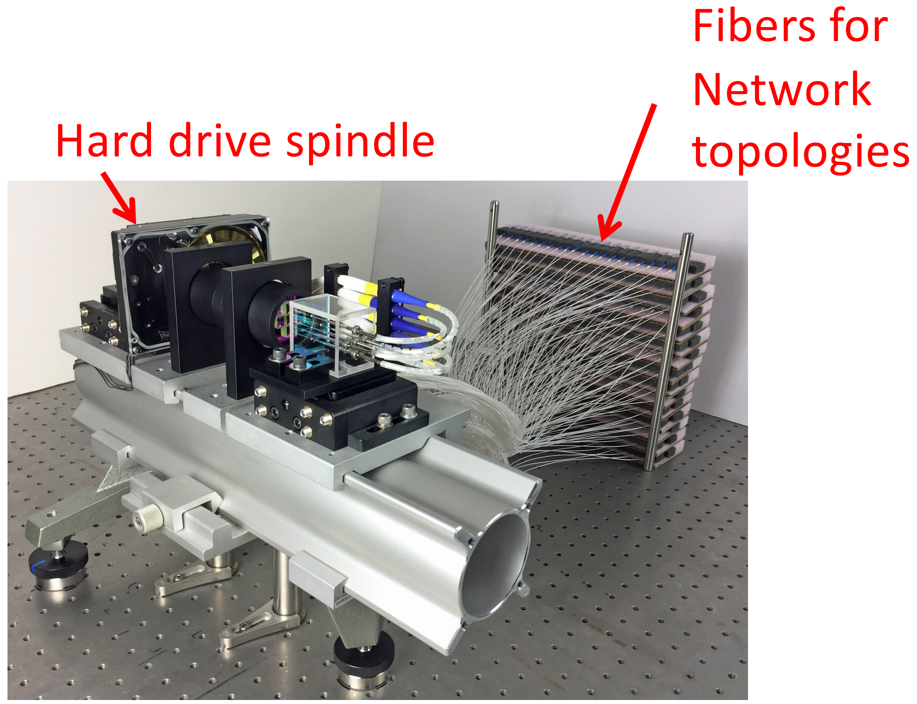
Optical layout:





Laser-written grating pinwheel



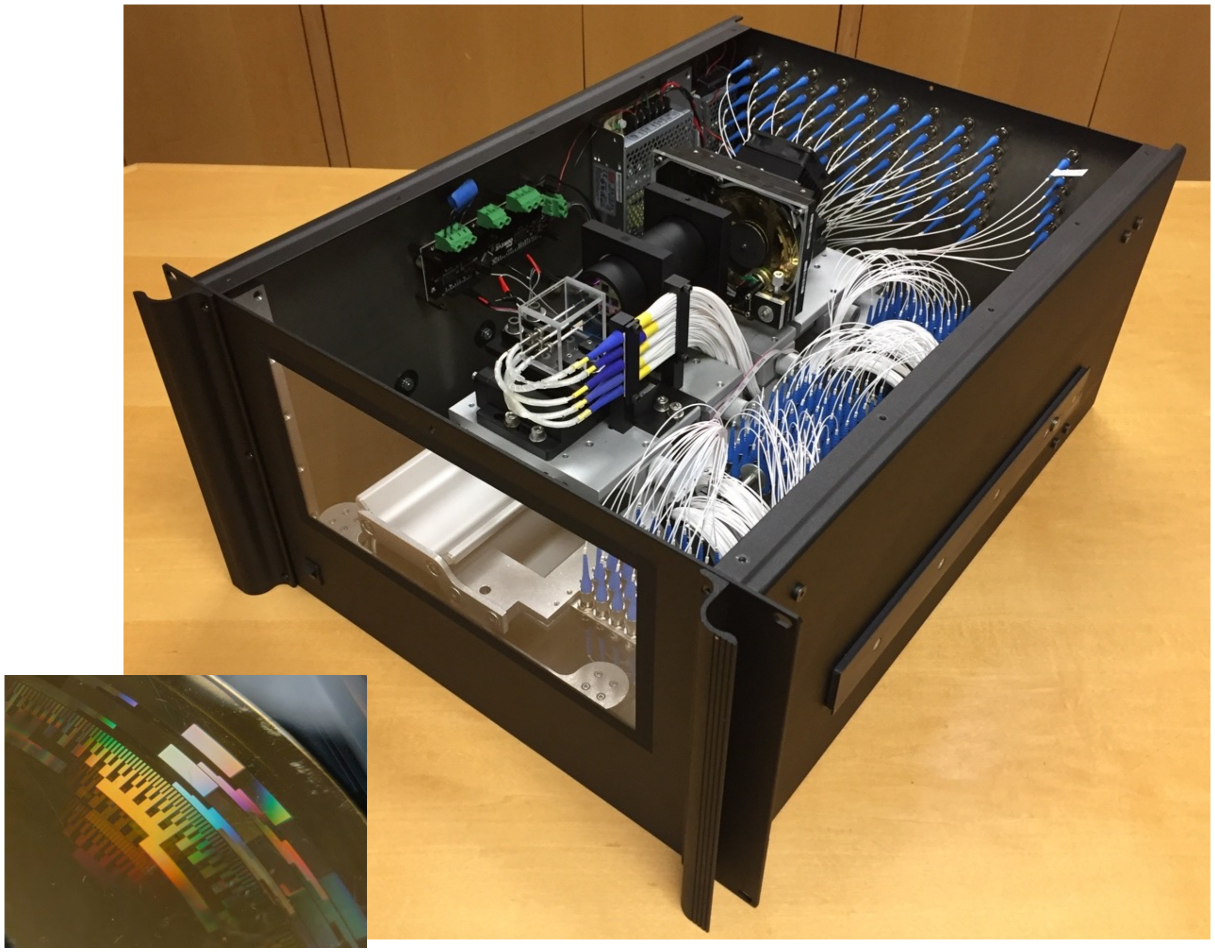


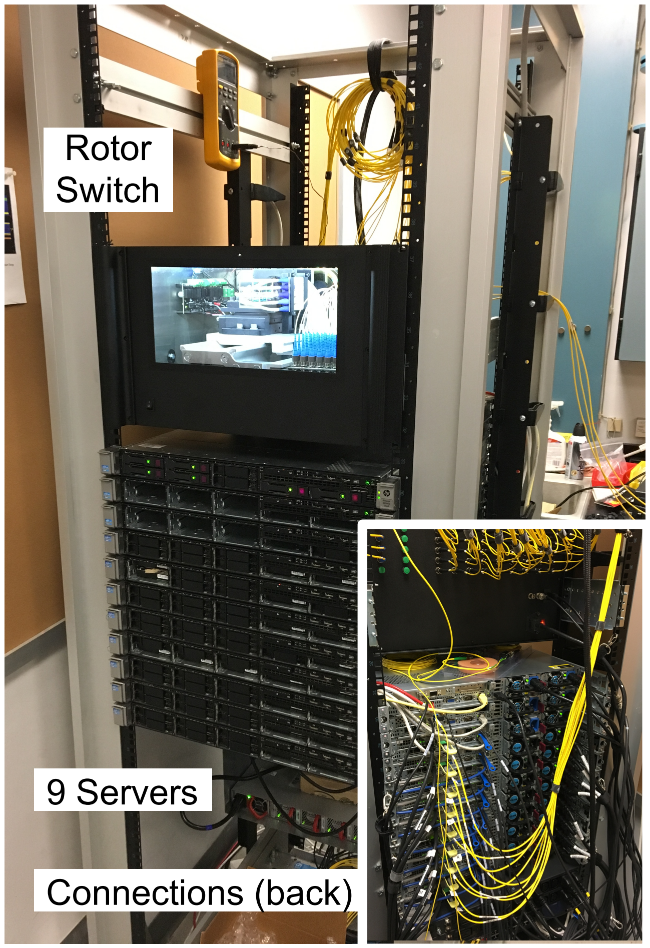
Crosstalk:	< 30 dB
Operating spectrum:	> 120 nm
2-pass insertion loss:	5 – 8 dB*

(*7 improved with better grating)

Corundum in use: Testing a new Rotor switch prototype

UC San Diego
JACOBS SCHOOL OF ENGINEERING

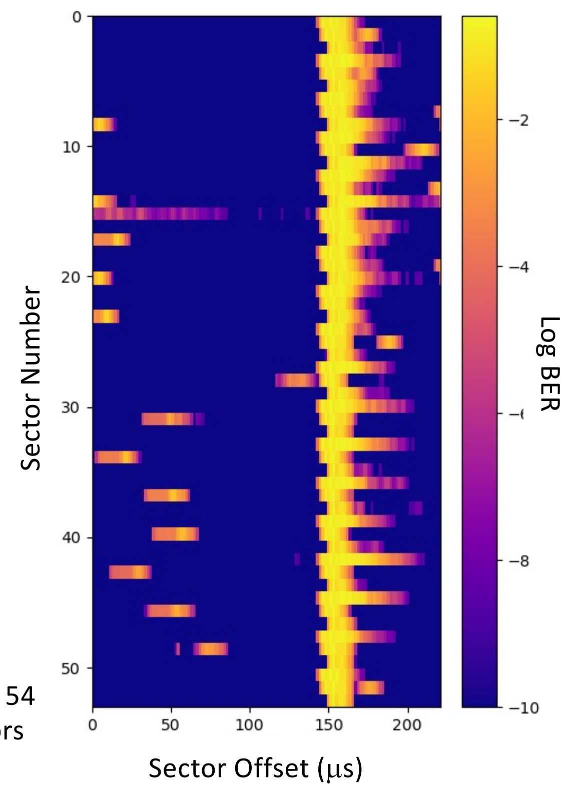




Switch system-level testing using Corundum

UC San Diego
JACOBS SCHOOL OF ENGINEERING

Measured BER "heat map"



Total 54 sectors

Switching time ~ 22 μs

Each heat map shows BER vs time for one input connection, through 54 sectors (3 configurations repeated 18 times) of one full disk rotation.

System-level switching time includes:

- Physical switching time (~22 us)
- AGC and CDR lock time (~10 us)
- Disk synchronization (~10 us)

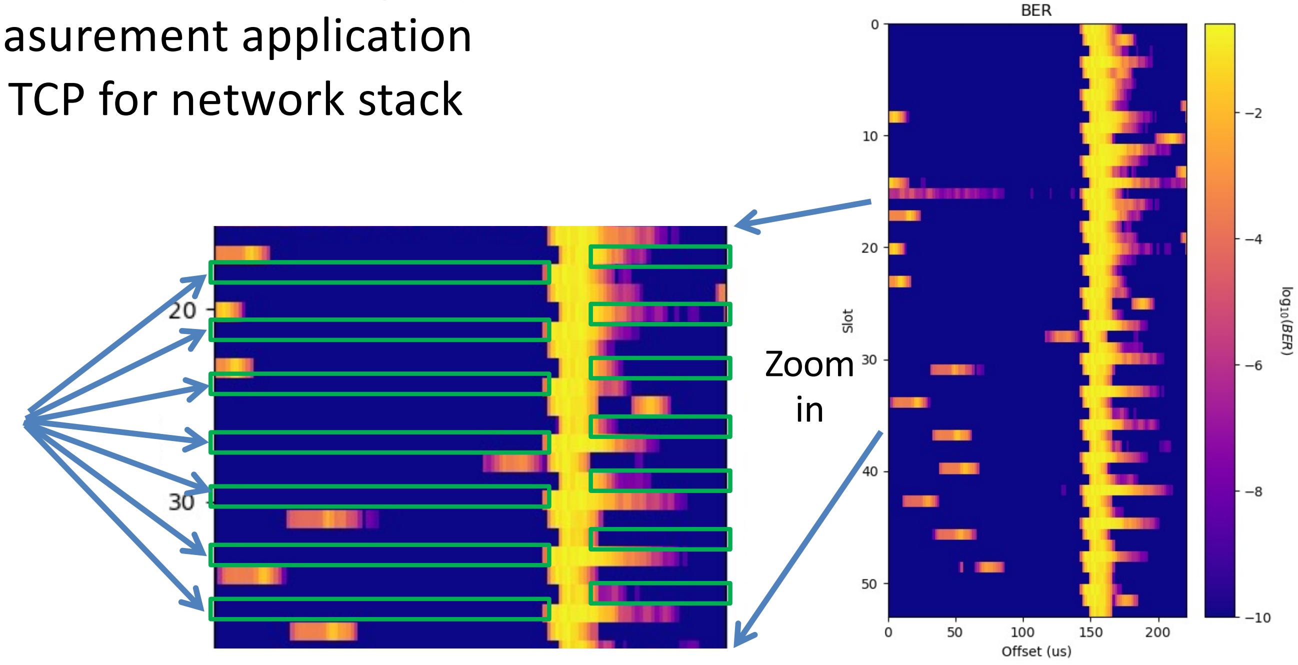
Does not include:

- Ethernet 64b/66b frame sync
- NIC transmit timing accuracy

Full stack optically-switched unmodified Linux app (iperf)

- Structure in BER showed some paths through the rotor switch are usable (with few errors from pinwheel fab errors /power offset)
- Ran app “iperf” on a “clean” single path
  - Network measurement application
  - **Unmodified** TCP for network stack

Every third sector has low errors - sufficient to run app.



Optical networks built on top of Corundum

UC San Diego
JACOBS SCHOOL OF ENGINEERING

- We are building additional logic – called shims – on top of the baseline design to support new networking research
- Shims are logic that sits between the core Corundum datapath and the Ethernet MACs
- Corundum can be used by other groups as a foundation for other networking research by building different shims

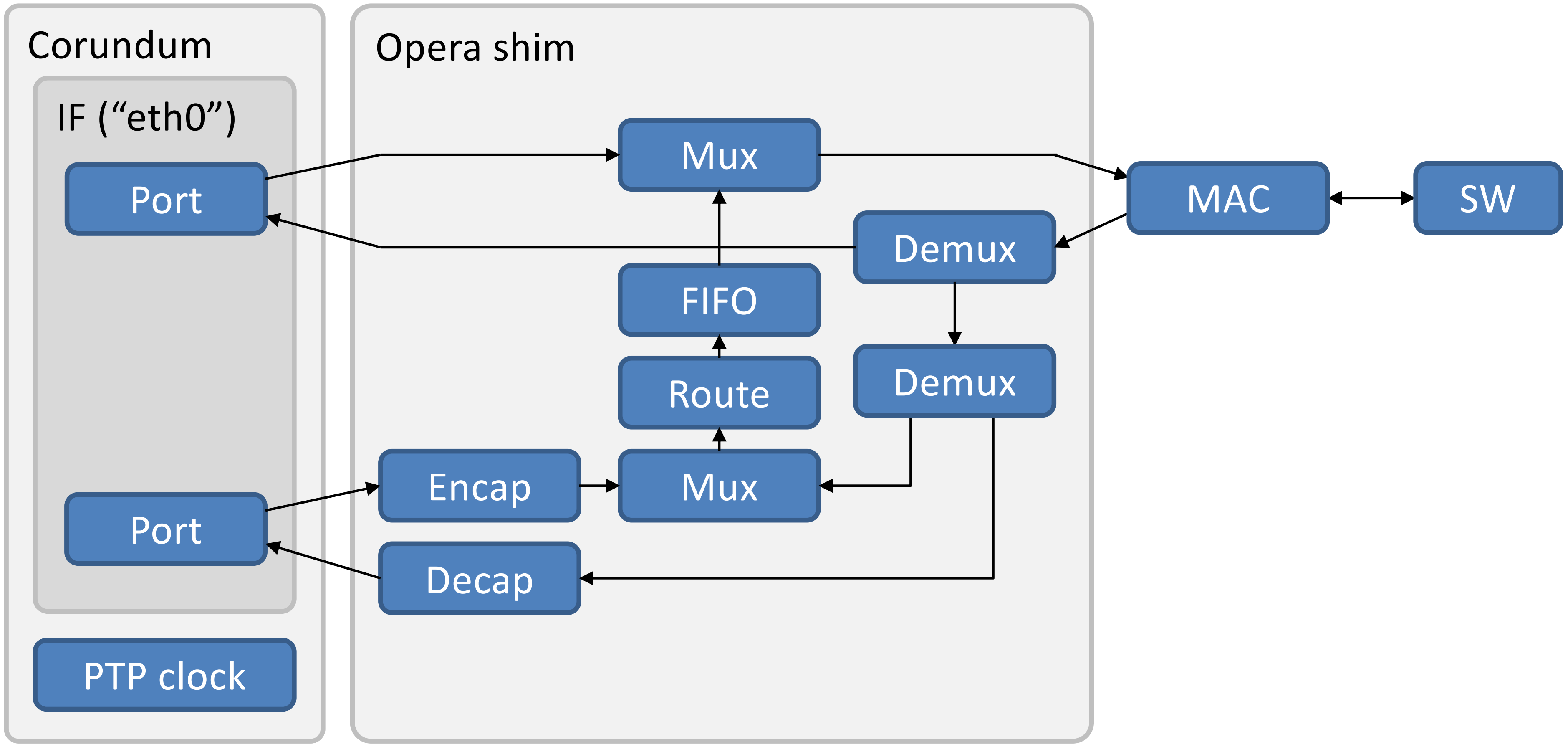
Example shim: Opera protocol

UC San Diego
JACOBS SCHOOL OF ENGINEERING

- Opera protocol (NSDI 2020) is designed to support low-latency traffic over RotorNet
- Separate port/interface to handle low-latency traffic
- Encapsulate opera traffic
  - can add extra metadata in header if needed
- Hop-by-hop routing determined by PTP time
- Merge incoming indirect traffic with new indirect traffic
- Current effort is at server level – can be adapted to ToR as in paper

Example shim: Opera protocol

UC San Diego
JACOBS SCHOOL OF ENGINEERING



Research Directions: Line protocol development

- Ethernet line protocols are not designed for circuit switching
  - Slow frame sync (~10 us)
  - Slow lane bonding/alignment marker lock (~200 us for 100G CAUI-4)
  - Slow FEC block lock (~ms for 100G CAUI-4 RS-FEC)
- New line protocols required for efficient circuit switching
  - Burst-mode receivers, fast frame sync
  - Switch-aware lane bonding and FEC techniques
- Corundum provides a platform for evaluating line protocols in a datacenter environment