

# Lightweight Data Center Packet Processing with RMT TCP

Antoine Kaufmann, Naveen Kr. Sharma  
Thomas Anderson, Arvind Krishnamurthy  
*University of Washington*

**Timothy Stamler**, Simon Peter  
*The University of Texas at Austin*

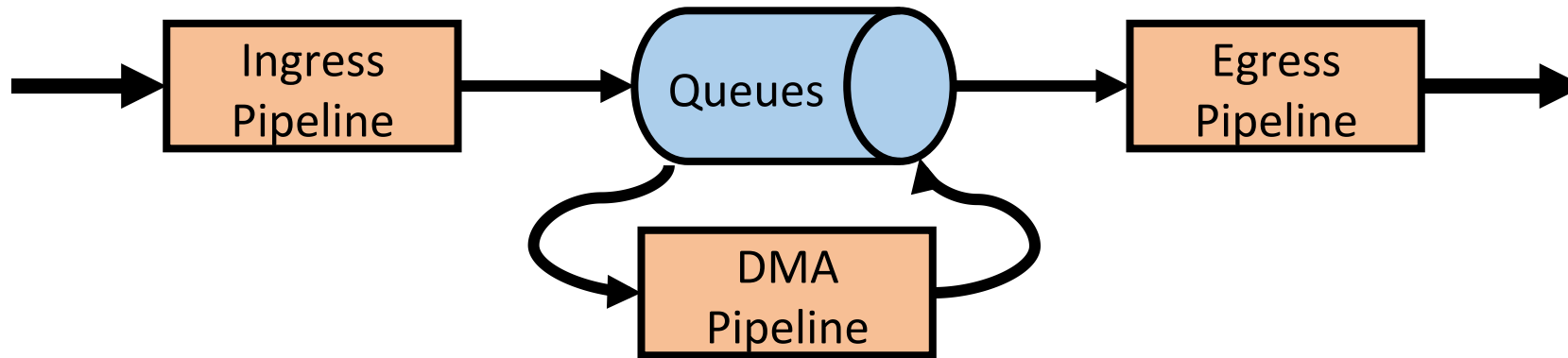
# The Problem

- Packet processing within data center applications is *slow*
  - Many cloud apps are dominated by packet processing
  - Key-value store, real-time analytics, intrusion detection
- Some solutions exist...
  - TCP segmentation/checksum offload
  - Kernel bypass architectures
  - RDMA
- ...but for data center applications, they just aren't fast enough
- NICs are poorly integrated with software
  - Wasted cycles, poor cache locality, extra synchronization

# FlexNIC:

## A Model for Integrated NIC/SW Processing

[ASPLOS'16]

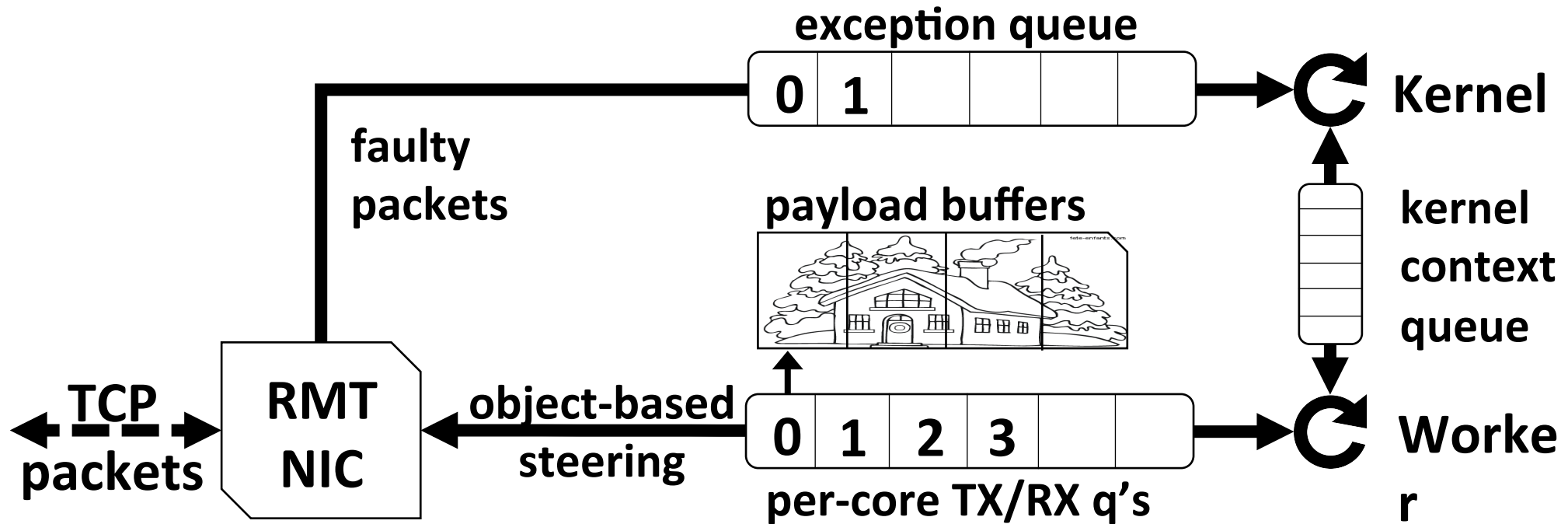


- Efficient application level processing in the NIC
  - Done via Match and Action pipelines (programmed in P4)
  - Improve locality by steering to cores based on app criteria
  - Transform packets for efficient processing in SW
  - DMA directly into and out of application data structures
  - Send acknowledgements on NIC

# FlexTCP

- Full TCP processing is too complex for FlexNIC
  - Large connection state required, tricky edge cases, complicated congestion control
- But the common case is simpler: it can be offloaded
- **Main idea:** Split packet processing into common case and rest
  - Common case: no drops, no re-ordering -> **processed in NIC**
  - Rest: in kernel or background thread
- Safety critical processing like filtering, validation, and rate limits on NIC
- Handle all edge cases like drops, re-orderings, and timeouts in software
- Requires small per-flow state
  - Rate-limit, and ~8 registers (SEQ/ACK, queues,...)

# FlexTCP Overview



# My Contributions

- Existing evaluation had been done through an emulator
- I worked on taking FlexTCP and implementing it on Netronome NIC
  - Worked with 2 Netronome reprogrammable network cards
  - Will evaluate end-to-end performance of this implementation
- The FlexTCP NIC responsibilities were implemented in P4 and micro-C
  - Identifying flows and steering packets to the correct queues/cores
  - Generating and sending ACK packets
  - Forwarding exception packets to the kernel
  - Allowing for new connections to be established and properly handled