Network Measurement for 100Gbps Links Using Multicore Processors

Xiaoban Wu, Dr. Peilong Li, Dr. Yongyi Ran, Prof. Yan Luo
Department of Electrical and Computer Engineering
University of Massachusetts Lowell
http://acanets.uml.edu

Introduction

Network Measurement

- Network measurement at 100 Gbps:
  - Dedicated hardware, e.g. SRAM and TCAM:
    - Inflexible to run-time changes of measurement functions
    - Memory requirement is prohibitively large
  - Programmable box on x86 platform:
    - Run-time programmable
    - Rich memory/compute resources
- Network measurement requirements on x86 platform:
  - High data volume
    - Streaming algorithms such as “Sketches”
  - Low packet delay (< 27 ns per packet):
    - Cache/memory optimization

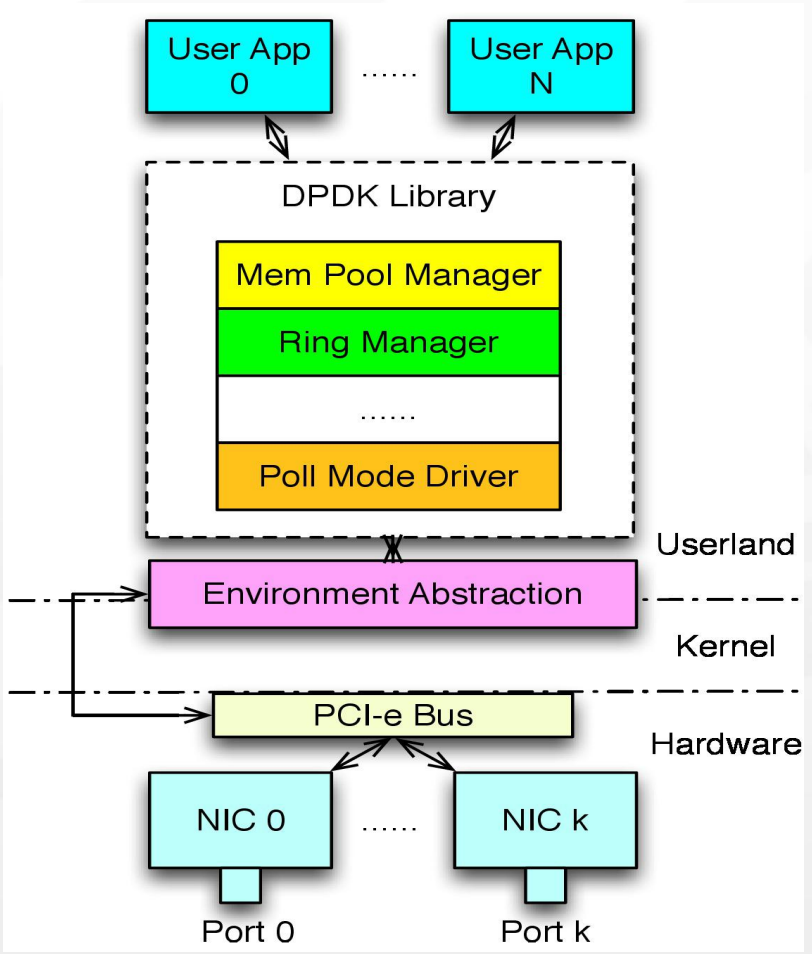
Motivation

- Design a high-performance 100 Gbps measurement platform with scalable performance on multicore x86 machines.
- Design sketch-based streaming measurement functions on a multicore architecture with advanced I/O and memory management techniques.
- Conduct in-depth performance analysis from both system level and micro-architecture level.
- Propose practical implementation solutions for high-speed network measurement on x86 architecture.

Background

The Intel DPDK

- The Intel Data Plane Development Kit (DPDK)
  - Kernel space bypassing
    - Avoids context switching
    - Reduces memory copy
  - Supports multicore framework
  - Employs hugepages
    - Efficient TLB translation
  - Ring buffers
    - Lockless design
  - Poll mode driver (PMD)
    - Avoids interrupts



Background

Streaming Algorithms and Sketches

Sketch	Data Structure	Pros	Cons
Count-Min Sketch (CMS)	(1) Matrix with several rows (2) Each entry of the matrix consists of a counter	Ensure certain accuracy	Can not work alone in distributed measurement
Reversible Sketch (RS)	(1) Matrix with several rows (2) Each entry of the matrix consists of a counter and several sets	(1) Ensure certain accuracy (2) Can work in distributed measurement	Memory consumption is high
Simple Hash Table (SHT)	(1) Matrix with only one row (2) Each entry of the matrix consists of a counter and a 5-tuple	(1) Can work in distributed measurement (2) Low memory consumption	Accuracy is not guaranteed

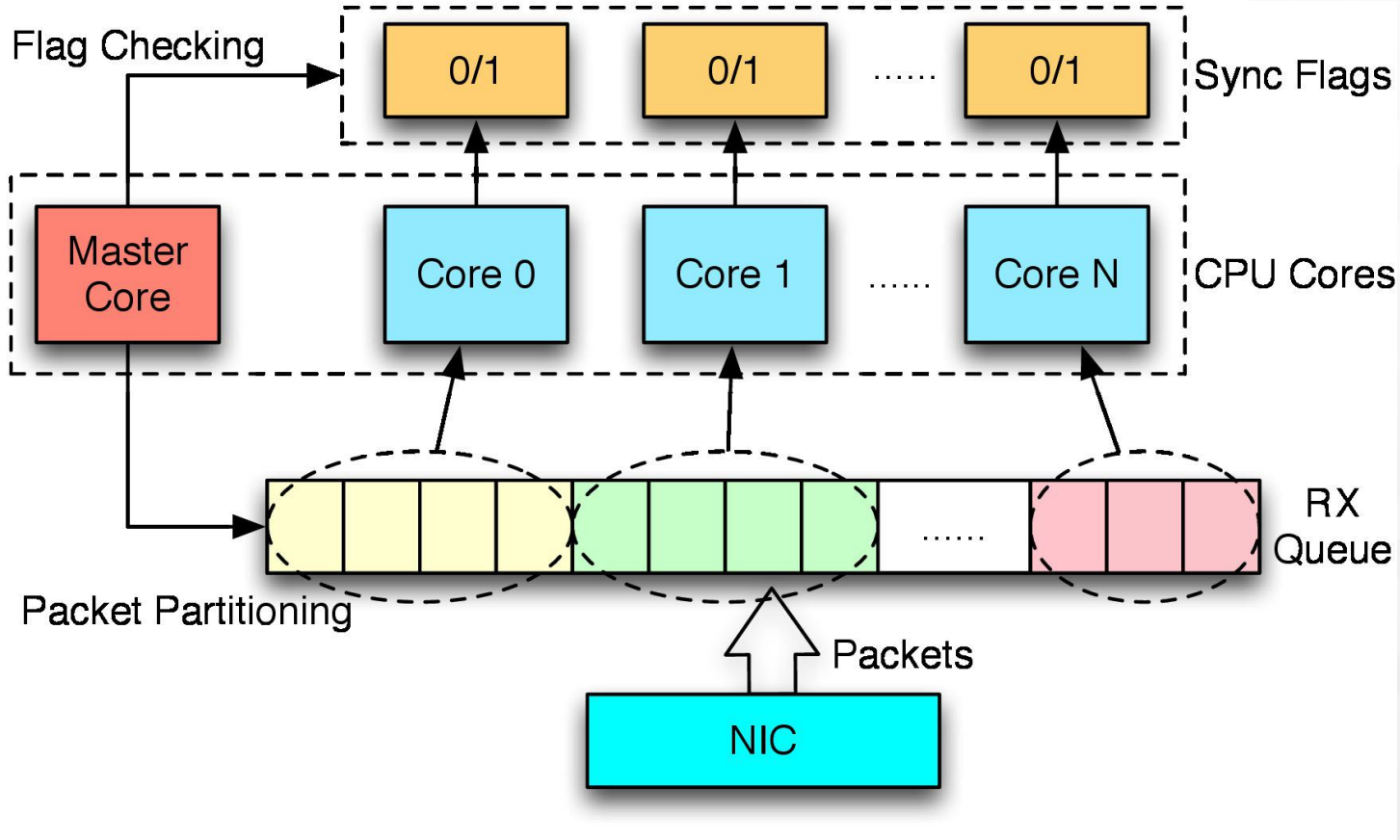
Measurement Design

The Design Options

- Based on the RX modes:
  - Non-RSS design (Master-Slave)
    - **SYNC**: slave nodes work synchronously
    - **AYNC**: slave nodes work asynchronously
  - **RSS** design: requires Receive Side Scaling feature from a NIC
- Based on the Sketch data structure:
  - **Shared**: shares one central data space across all cores
  - **Separate**: each core maintains one data space

Measurement Design

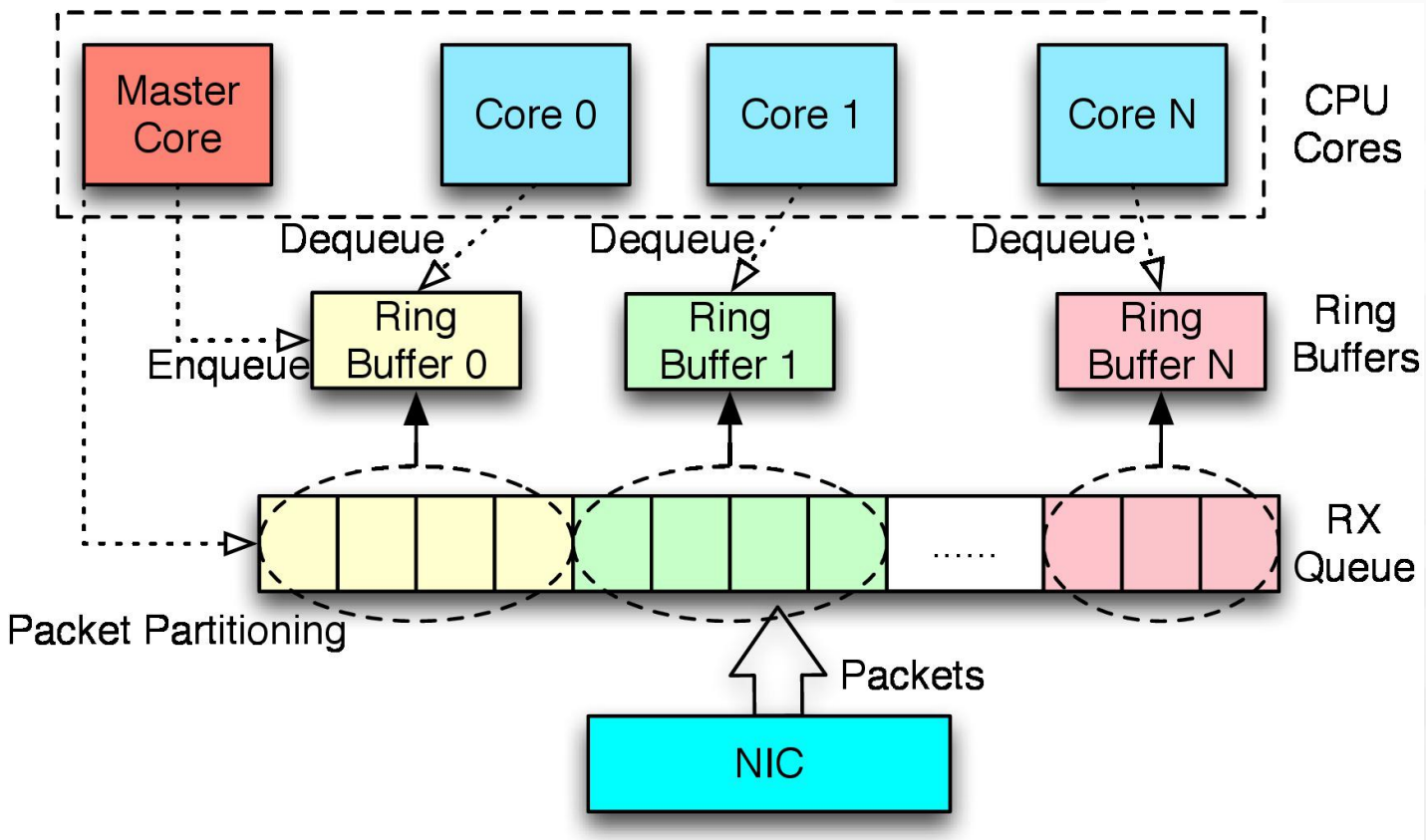
Sync Design



- Master core:
  - Receives a batch of packets;
  - Partitions packets;
  - Assigns slave cores;
  - Wait all slaves to finish.
- Each slave core:
  - Fetches data and processes it;
  - Sets the completion flag

Measurement Design

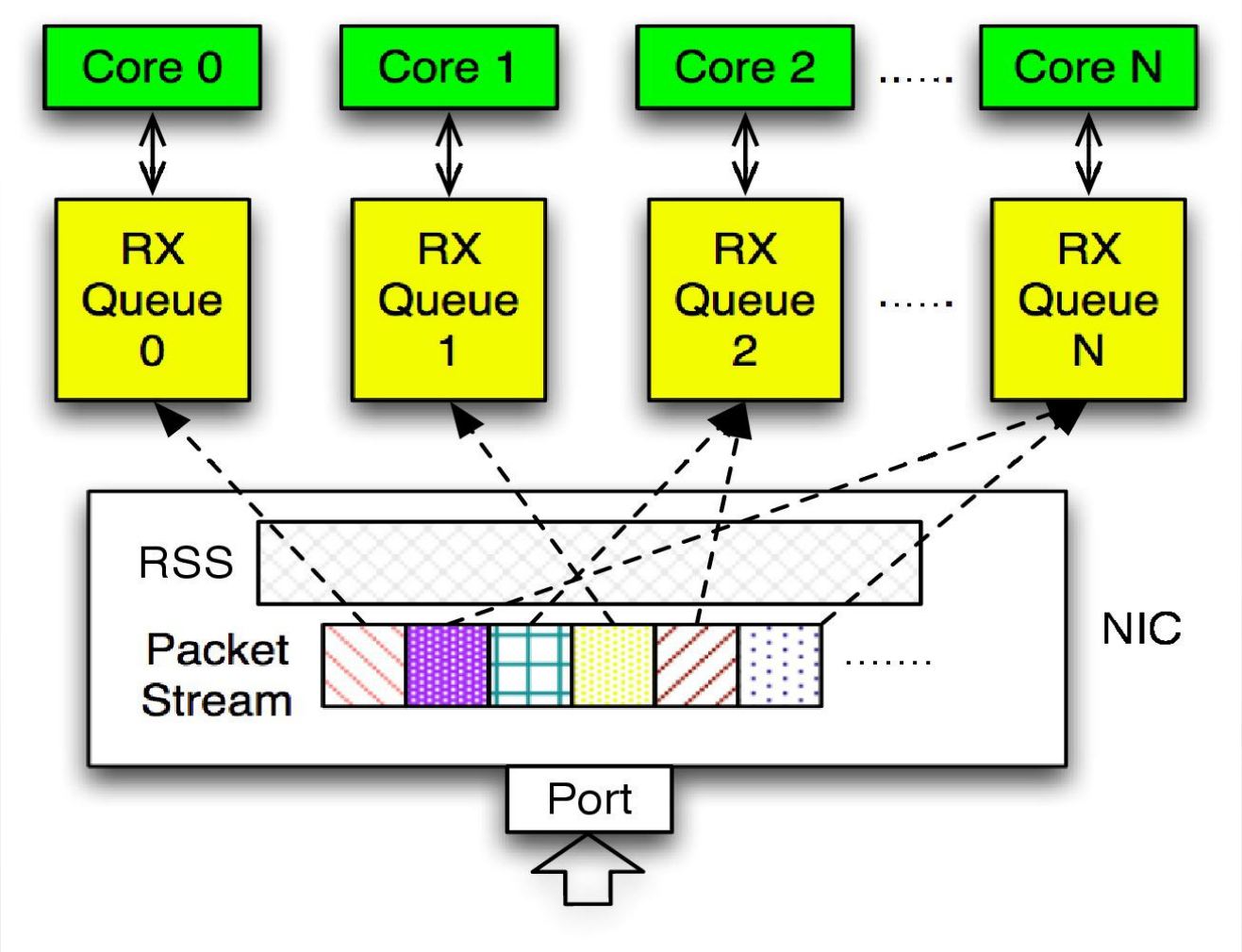
Async Design



- Master core:
  - Receives a batch of packets;
  - Partitions the packets;
  - Sends packets to dedicated ring buffer of each slave core if not empty.
- Each slave core:
  - Pulls its ring buffer
  - Processes the packets if exist.

Measurement Design

RSS Design



- Maintains N number of DPDK RX queues based on core number;
- RSS in NIC distributes packets to RX queues;
- Each core receives and processes the packets from its RX queue in parallel.

Measurement Design

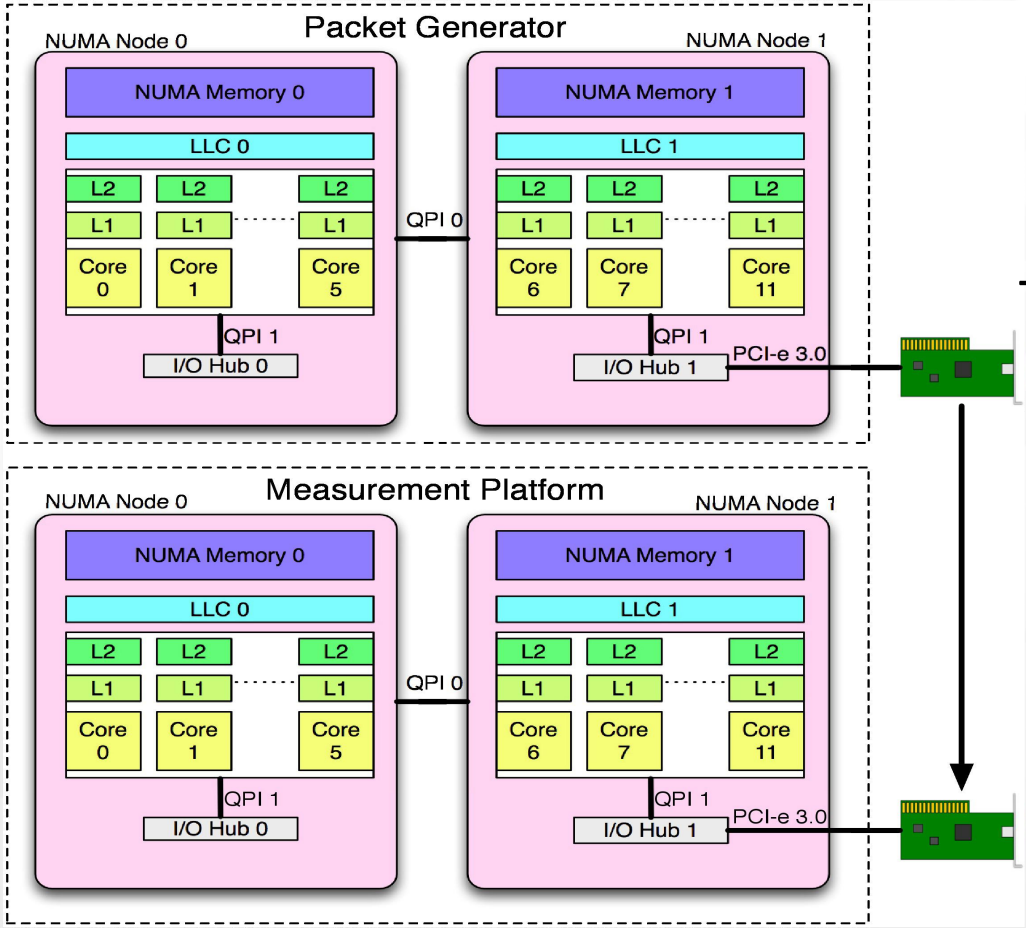
Separate and Shared Design

- Separate design:
  - Each core has its own sketch
- Shared design:
  - Central sketch memory space across cores
- Data structure operations for Sketches (details in paper)
  - Update a counter concurrently
  - Update a set concurrently
  - Update a critical section concurrently

Evaluation

Experiment Platform and Methodology

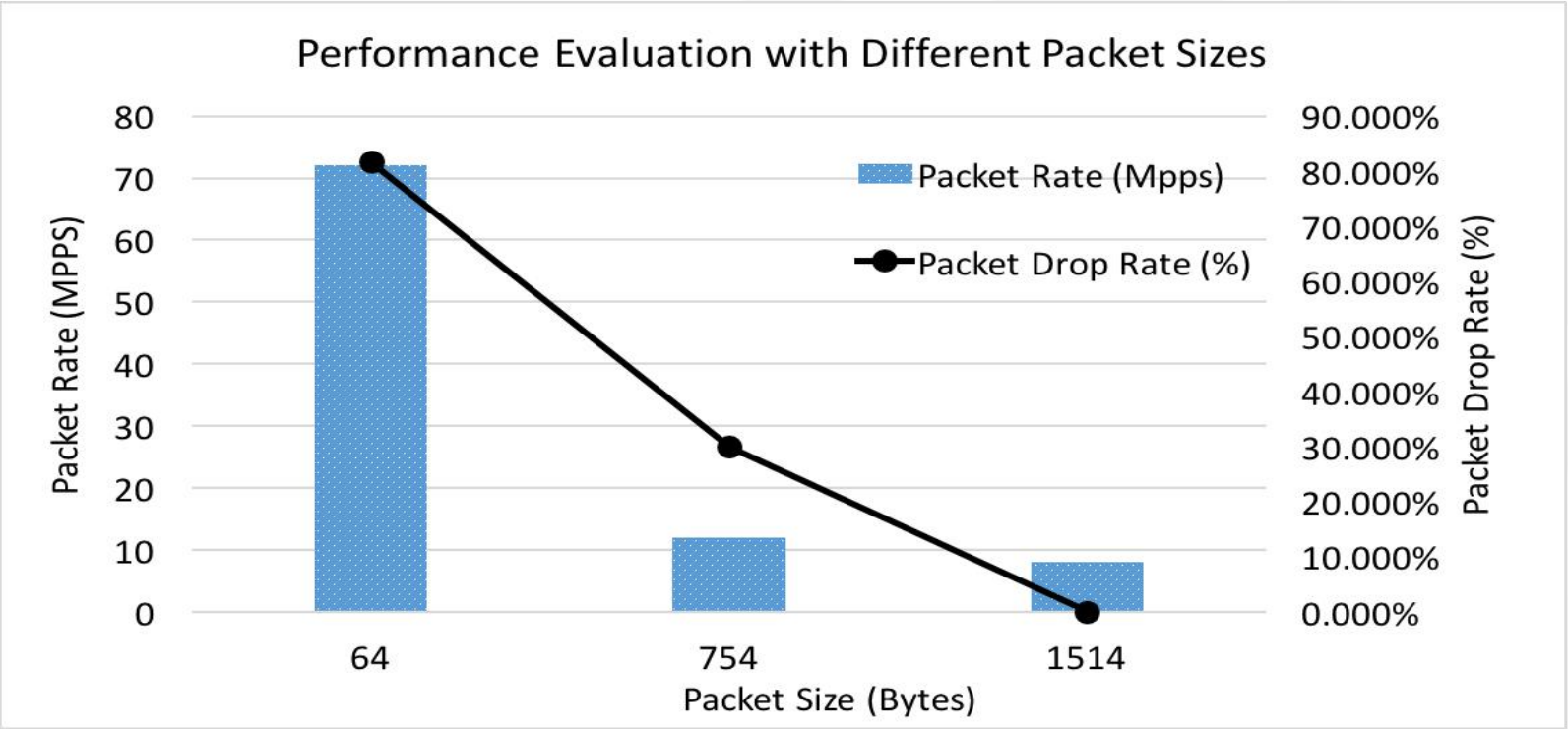
- CPU: Two 6-core Xeon E5-2643 on each server @ 3.4 GHz
- L1 Cache: i-$ 32KB, d-$ 32 KB
- L2 Cache: 256 KB
- LLC: 20 MB
- Memory: 16 GB
- NIC: Mellanox ConnectX-4 EDR 100GbE
- NIC is on socket #1
- Host OS: Ubuntu 16.04
- DPDK: Version 16.04



Evaluation

Packet Generator and Packet Size Study

- Packet generator: modified pktgen-dpdk with random L2/L3/L4 headers and random sized payload.

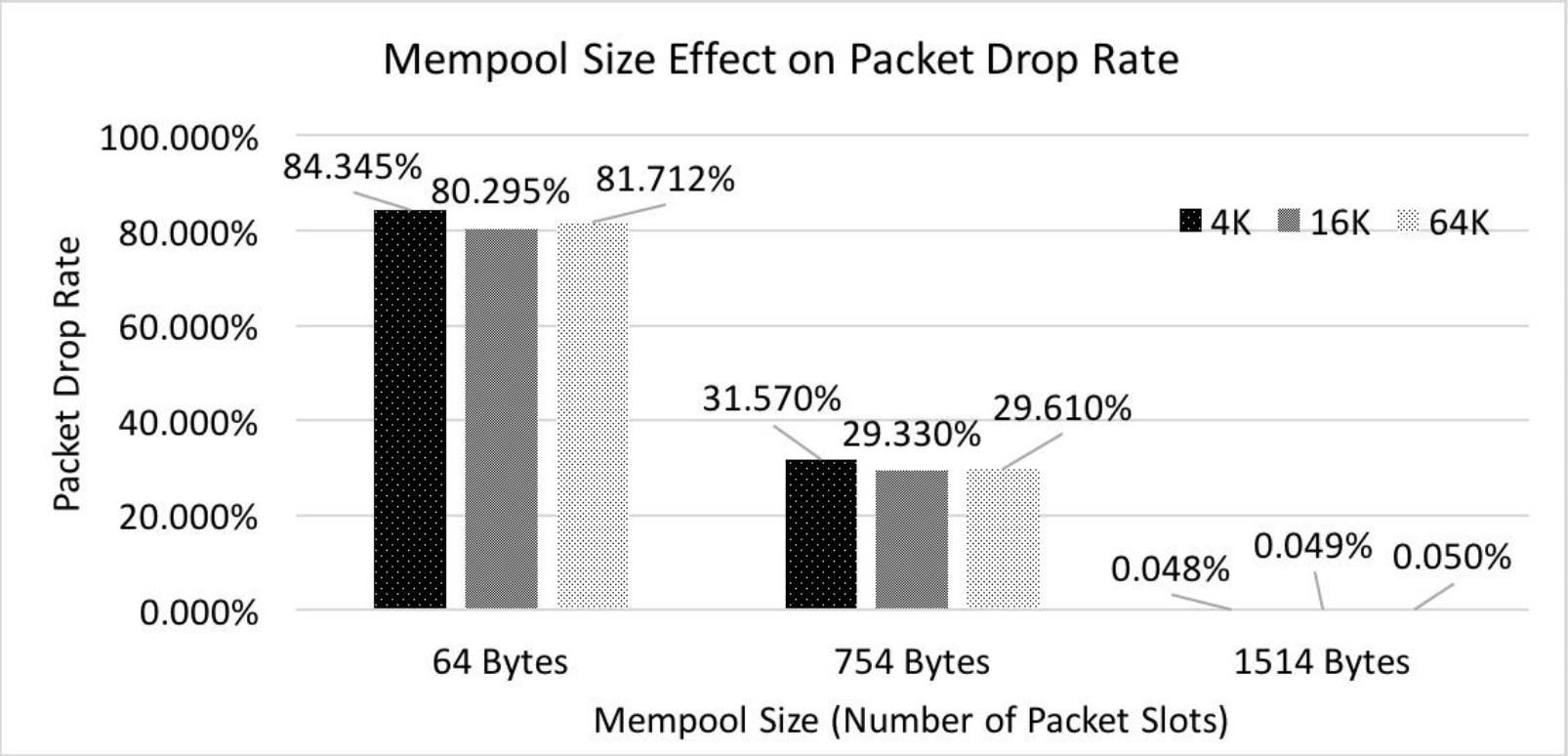


The 64-byte packet has the highest packet rate and packet drop rate.

Evaluation

DPDK Mempool Size

- How to decide the DPDK mempool size?

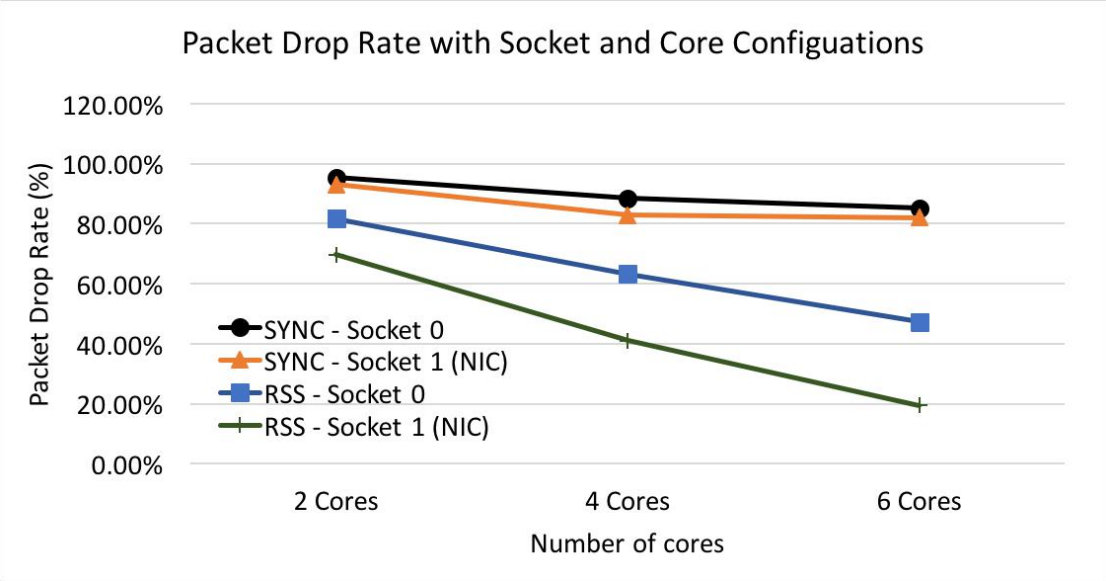


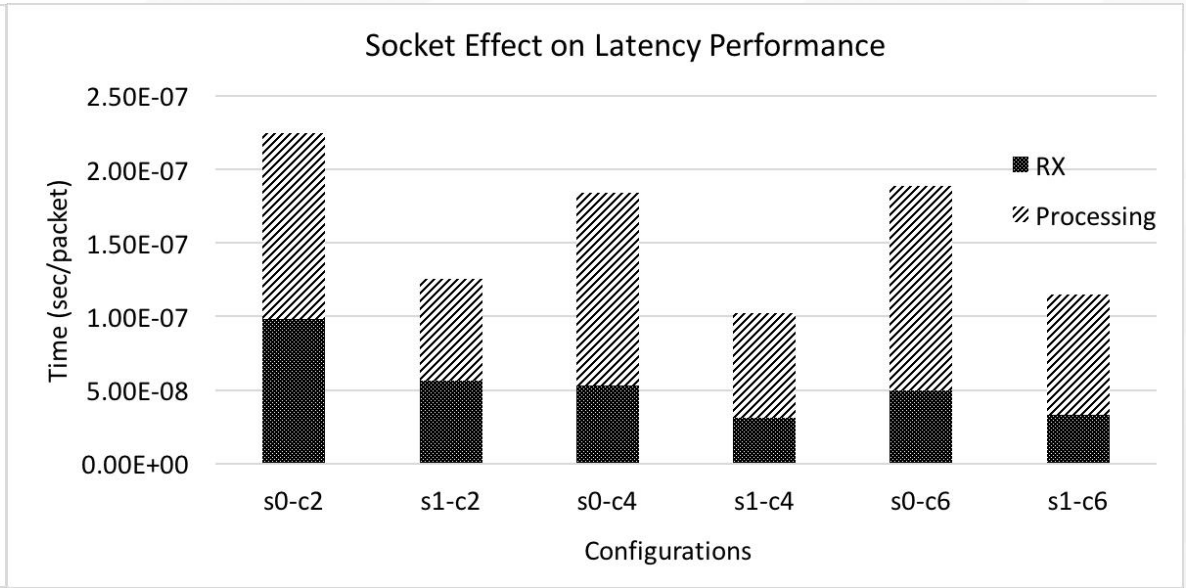
16K mempool size renders optimized performance.

Evaluation

NUMA and Cores

- Study the # of cores and and NUMA effect:



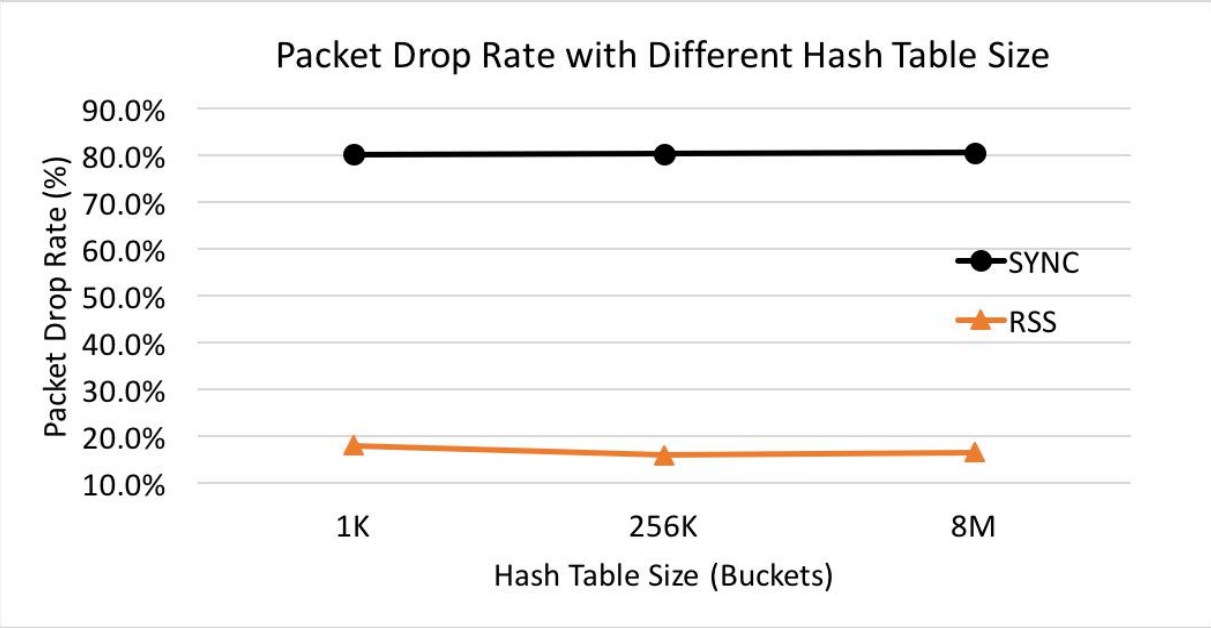


- Packet drop rate: ↓ up to 50% (2 cores to 6 cores).
- NUMA effect: up to 50% difference on the packet latency.

Evaluation

Memory and Cache

- How does the size of Sketches effect cache:



- The over-sized separate data structure maintained by CPU core cache can hardly affect the performance.
- Two reasons:
  - Cache accesses are almost overwhelmed by the size of mempool instead of sketches
  - Processing time only contributes around 40% of the overall packet delay

Evaluation

Shared/Separate Design

	Packet Drop Rate (PDR)	Packet Processing Time (PPT)
RSS-Separate-SHT	1.66E-01	3.80E-08
RSS-Shared-SHT	2.02E-01	4.95E-08
RSS-Separate-CMS	3.86E-01	8.29E-08
RSS-Shared-CMS	9.44E-01	1.48E-06
RSS-Separate-RS	9.61E-01	2.14E-06
RSS-Shared-RS	9.91E-01	9.15E-06

- Separate design: PDR is always lower (up to 60% lower for the CMS case).
- Separate design: Packet processing time is also lower (17.8x speedup for the CMS case).

Evaluation

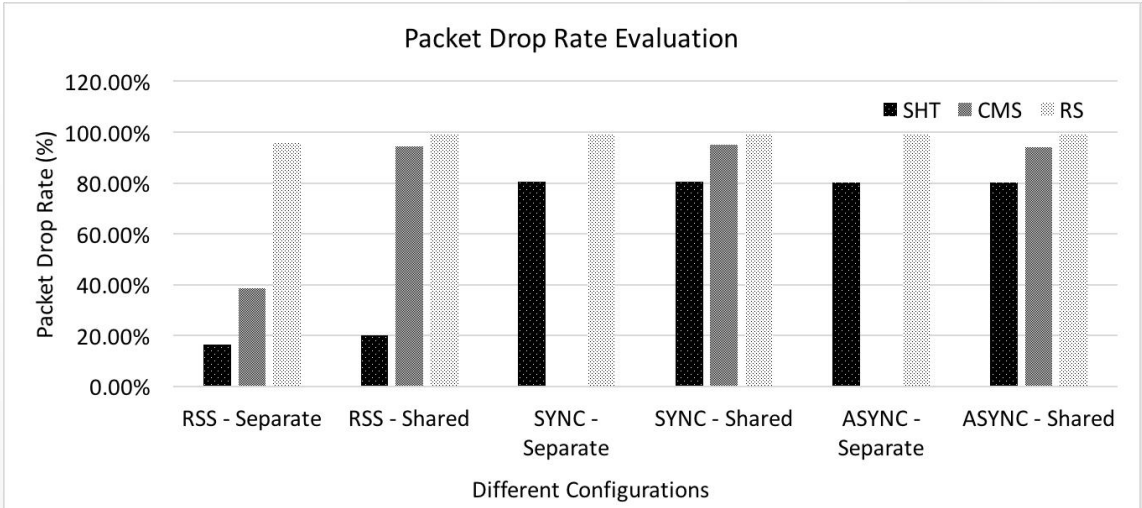
Shared/Separate Design

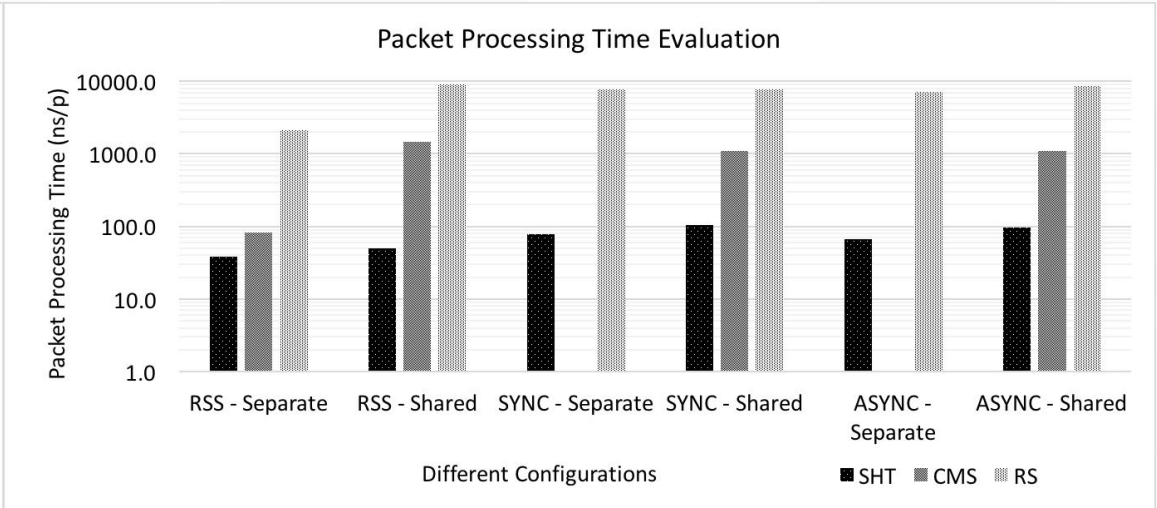
	Packet Drop Rate (PDR)	Packet Processing Time (PPT)
SYNC-Separate-SHT	8.06E-01	7.90E-08
SYNC-Shared-SHT	8.04E-01	1.04E-07
ASYNC-Separate-SHT	8.03E-01	6.66E-08
ASYNC-Shared-SHT	8.02E-01	9.72E-08
SYNC-Separate-RS	9.93E-01	7.83E-06
SYNC-Shared-RS	9.92E-01	7.70E-06
ASYNC-Separate-RS	9.93E-01	7.19E-06
ASYNC-Shared-RS	9.92E-01	8.54E-06

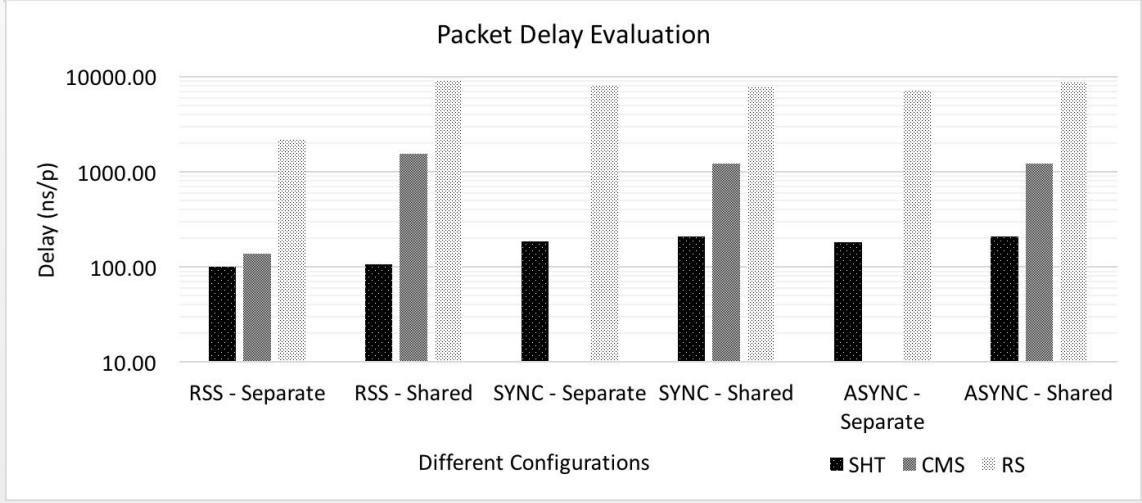
- Shared/separate designs do not impact too much on the PDR.
- On the contrary, PPT is almost always better if applying the separate design (only exception: synchronous RS).

Evaluation

Which Design to Choose?







- Compare three performance metrics:
  - Packet Drop Rate
  - Packet Processing Time
  - Packet Delay

Evaluation

Which Sketch to Choose?

- Which sketch data structure?
  - No accuracy boundary guarantee: choose SHT.
  - Accuracy boundary guarantee:
    - CMS (low PDR and delay)
    - RS (high measurement accuracy and space efficiency).
- Which parallel design?
  - RSS-enabled: RSS (esp. with separate data structure)
  - For non-RSS NICs, we should opt for the ASYNC design.
- Shared or separate?
  - Separate data structure renders better performance.

Conclusion

- We explore various software/hardware techniques of packet receiving and processing for network measurement.
- We propose parallel designs of measurement function by using sketch-based streaming techniques.
- We conduct in-depth performance analysis of the proposed design options.
- We provide insights on the optimal practical implementation within 100 Gbps network measurement environment.

That's It

Questions, comments, thoughts?