NDN-DPDK: NDN Forwarding at 100 Gbps on Commodity Hardware

Junxiao Shi, Davide Pesavento, Lotfi Benmohamed

Advanced Network Technologies Division

National Institute of Standards and Technology

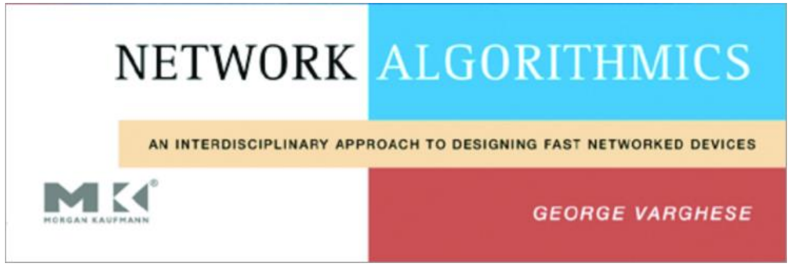
Introduction

- NDN needs a high-speed forwarder:
  - Use case: data intensive science, live video streaming, …
- Goal: line speed on commodity hardware.

How to get there?

➢Adopt better algorithms and data structures.

➢Reduce overhead in library and kernel.



DPDK
DATA PLANE DEVELOPMENT KIT

Data Plane Development Kit (DPDK)

- DPDK: libraries to accelerate packet processing workloads.

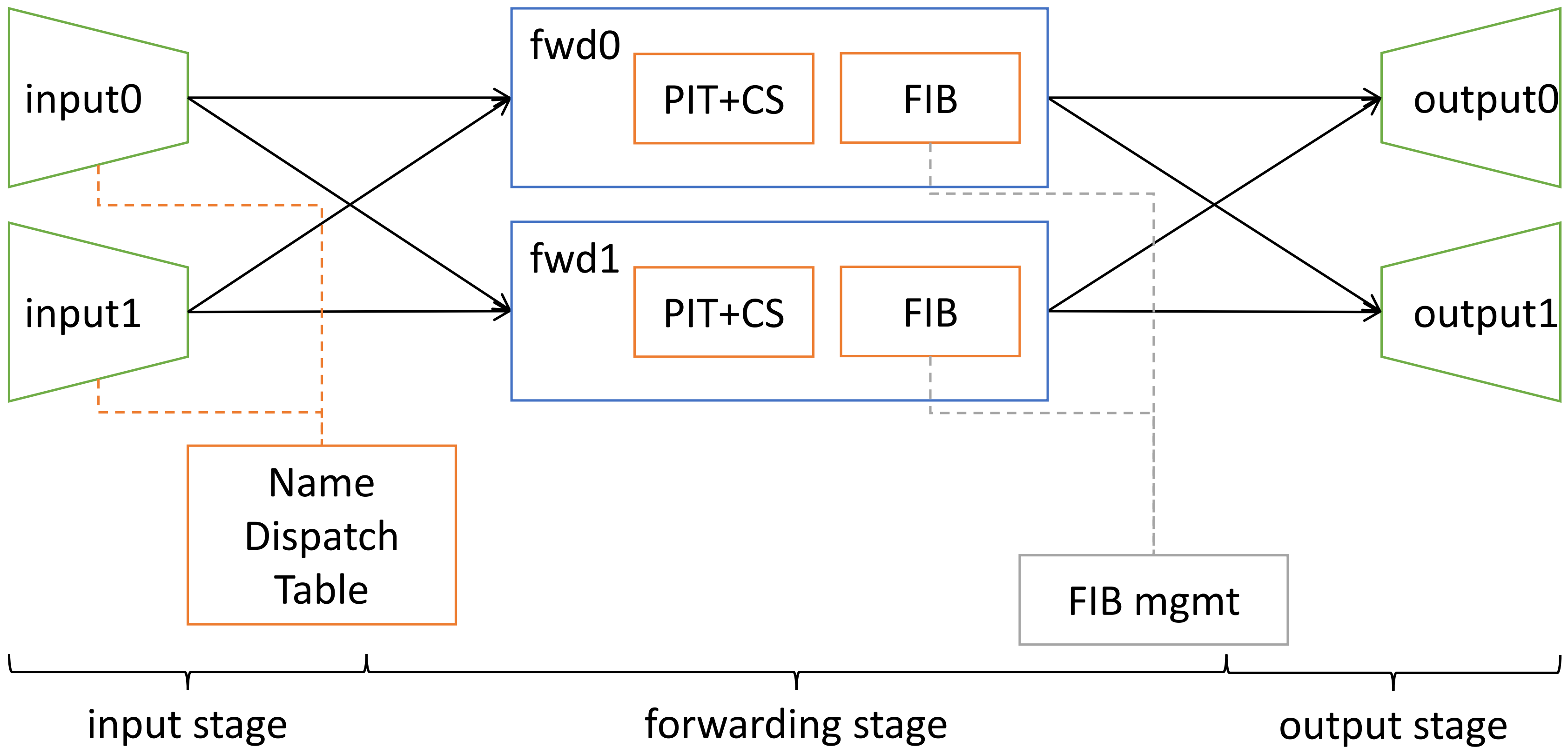
Main DPDK features:

- ➢ Multi-threading: use all available CPU cores.
- ➢ Ring buffer queue: transfer packets between threads.
- ➢ Hugepage-backed memory pools: no malloc() in data path.
- ➢ User-space NIC drivers: bypass the kernel.

Our Contributions

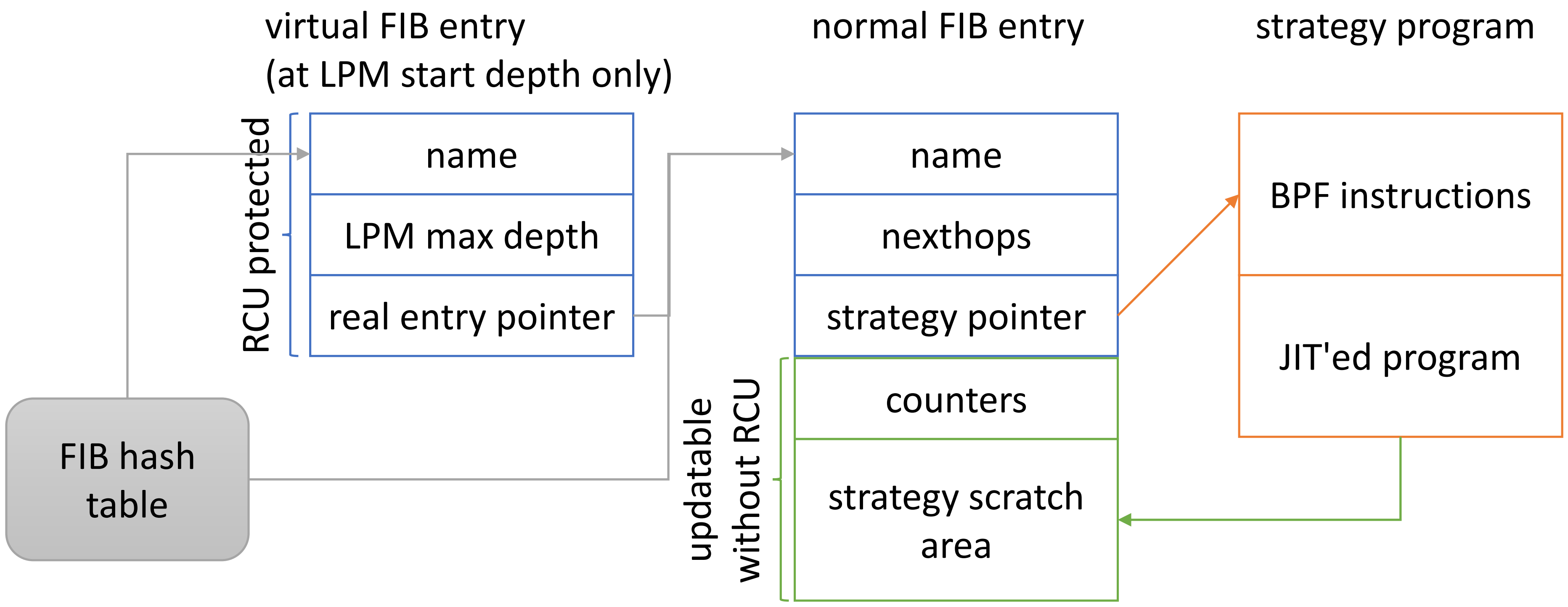
- NDN-DPDK:
  - ✓Complete implementation.
  - ✓Running on real hardware.
  - ✓Support full NDN protocol and name matching semantics.
- Prior works:
  - ❑Focus on a subset of data plane: Mansilha et al (ICN'15), …
  - ❑Rely on simulations: Song et al (ICN'15), …
  - ❑Lack support for Interest-Data prefix match: So et al (ANCS'13), Caesar (ANCS'14), Augustus (ICN'16), …

Forwarder Architecture



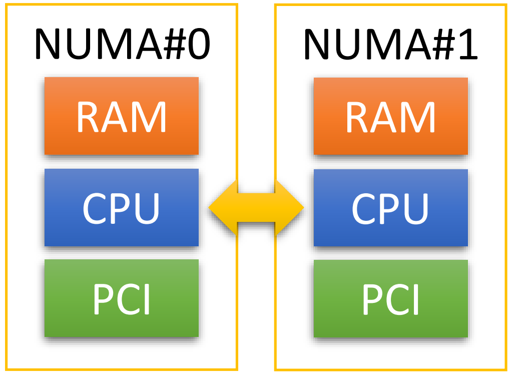
FIB Design

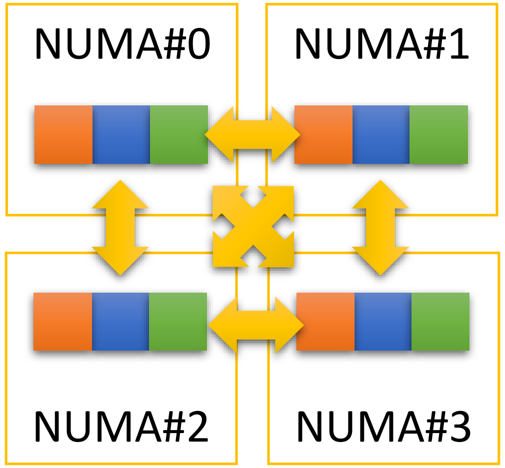
- 2-stage Longest Prefix Match algorithm.
  - So et al, Named data networking on a router: Fast and DoS-resistant forwarding with hash tables (ANCS'13).



FIB Replication on NUMA Sockets

- NUMA: Non-Uniform Memory Access.
  - Hardware in a multi-CPU server is organized in NUMA sockets.





  - Nonlocal memory access incurs higher latency.
- Each NUMA socket has a copy of FIB.
  - Forwarding threads can avoid nonlocal memory access during FIB lookups.

PIT Sharding

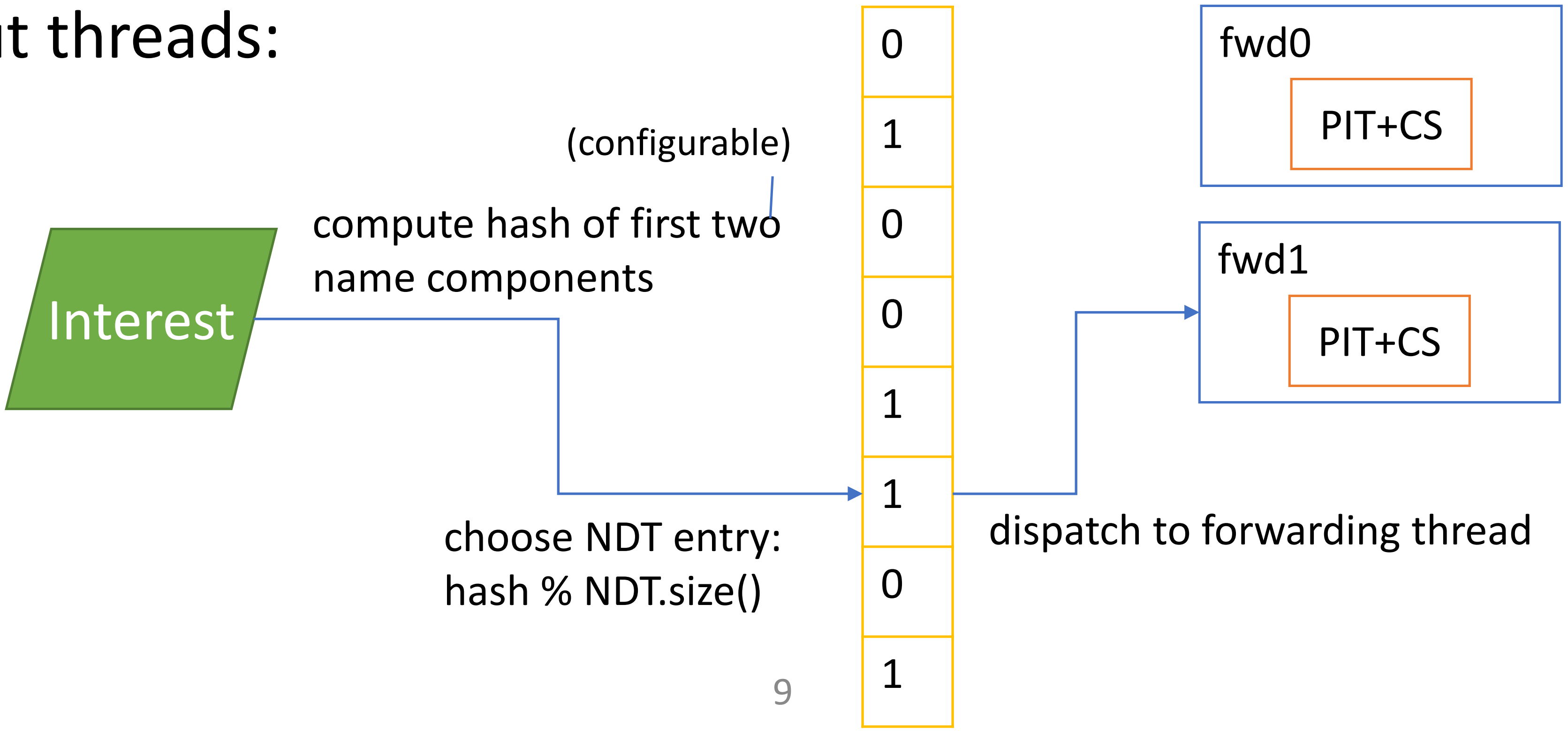
- Each forwarding thread has a private PIT.
  - Non-thread-safe. No RCU.

Requirements on packet dispatching:

1) Same Interest name => same forwarding thread.
   - Required by Interest aggregation and loop prevention.
2) Common Interest prefix => same forwarding thread.
   - Make forwarding strategy effective.
3) Data/Nack => forwarding thread that processed the Interest.
   - So that they can go back to the downstream.

Dispatch Interest by Name

- Name Dispatch Table (NDT)
  - Map: hash of name prefix => forwarding thread ID
  - Thread safe: NDT is an array of atomic_int.
  - Many name prefixes share the same entry.
- In input threads:

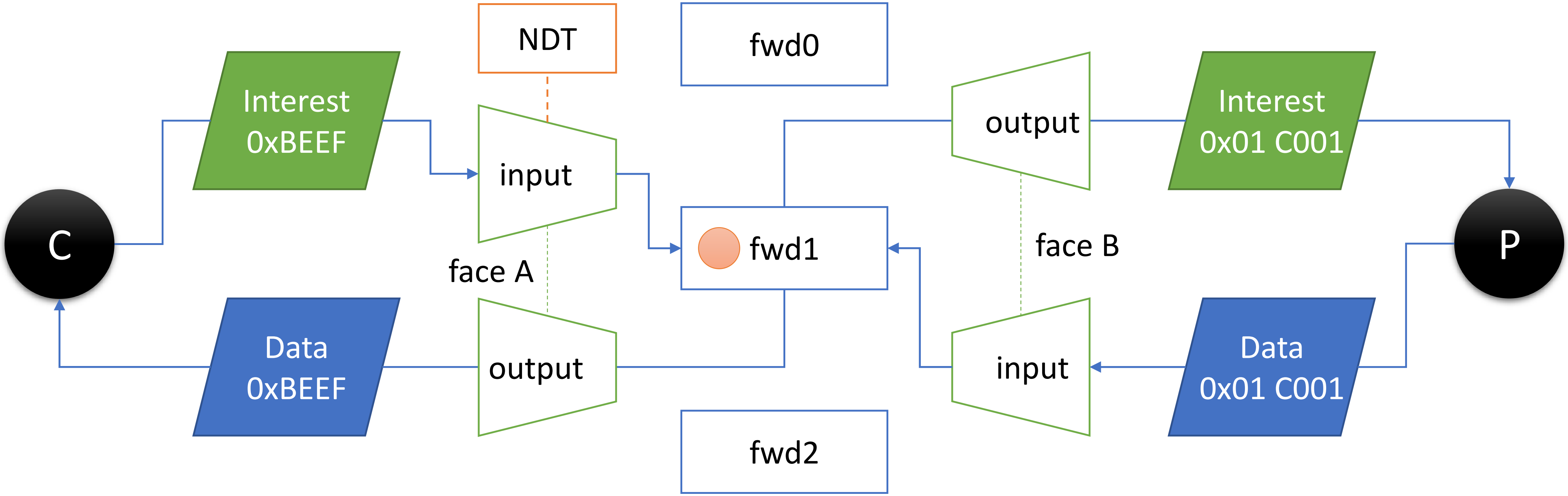


PIT Token

- Data packet: name dispatching works most of the time, except:
  - Interest /A CanBePrefix=1 goes to `NDT[SipHash(/A)]`.
  - Data /A/B/1 goes to `NDT[SipHash(/A/B)]`.
- Solution: use PIT token to associate Interest and Data.
- PIT token is an opaque token carried in a hop-by-hop field.
  - Every outgoing Interest carries a PIT token.
  - Data/Nack must carry the same PIT token.

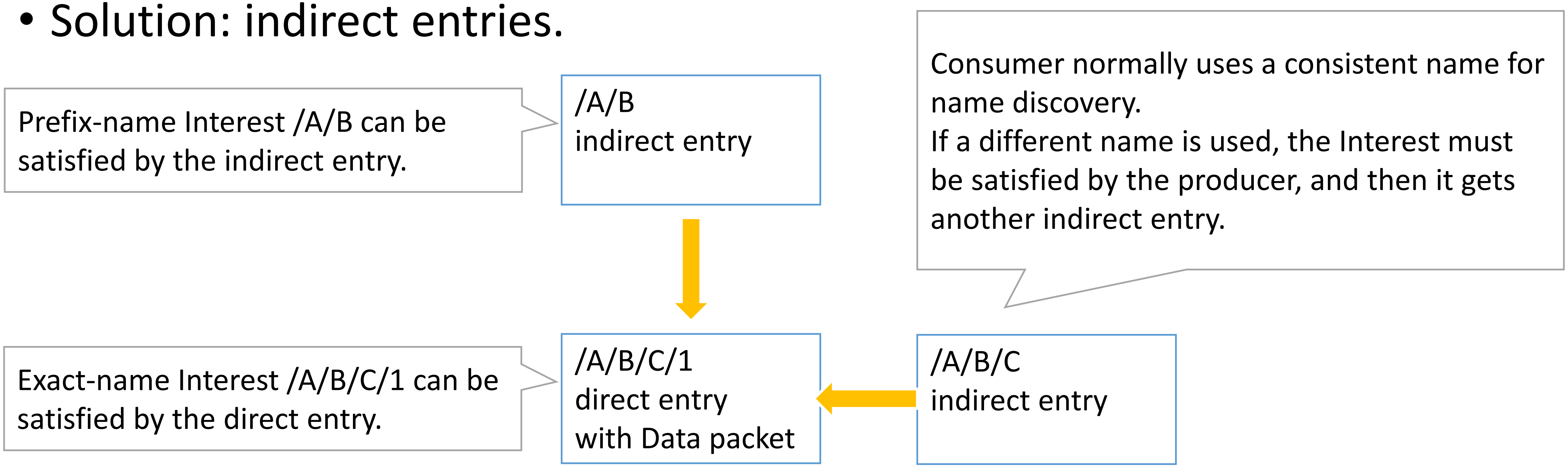
Dispatch Data/Nack by PIT Token

- NDN-DPDK's PIT token contains:
  a) Forwarding thread ID (8 bits), to dispatch Data/Nack correctly.
  b) PIT entry index (48 bits), to accelerate PIT lookups.

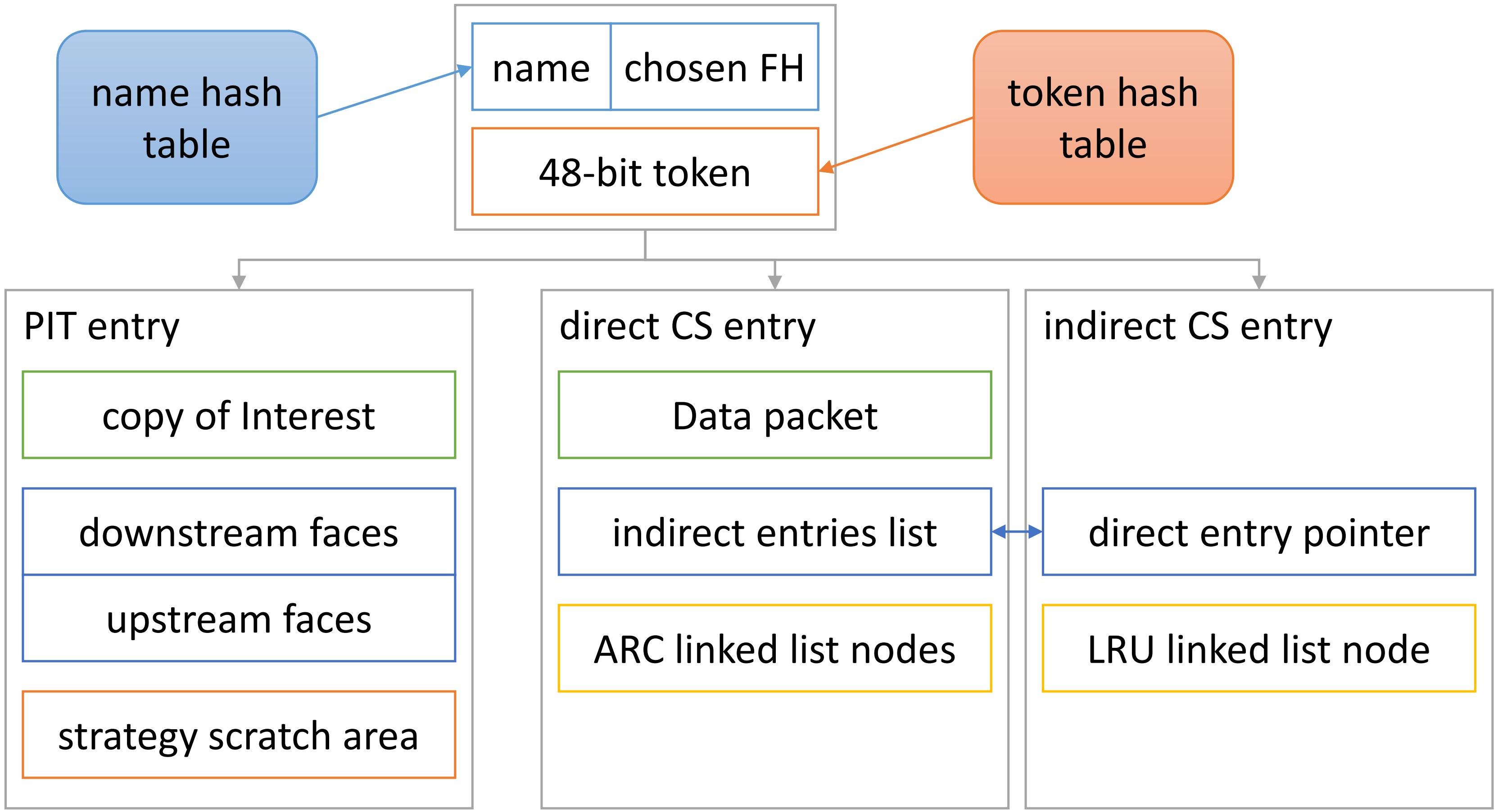


Prefix Match in CS

- In-Network Name Discovery:
  - Interests should be able to use incomplete names to retrieve Data packets.
- CS is a hash table, which only supports exact match.
- Solution: indirect entries.



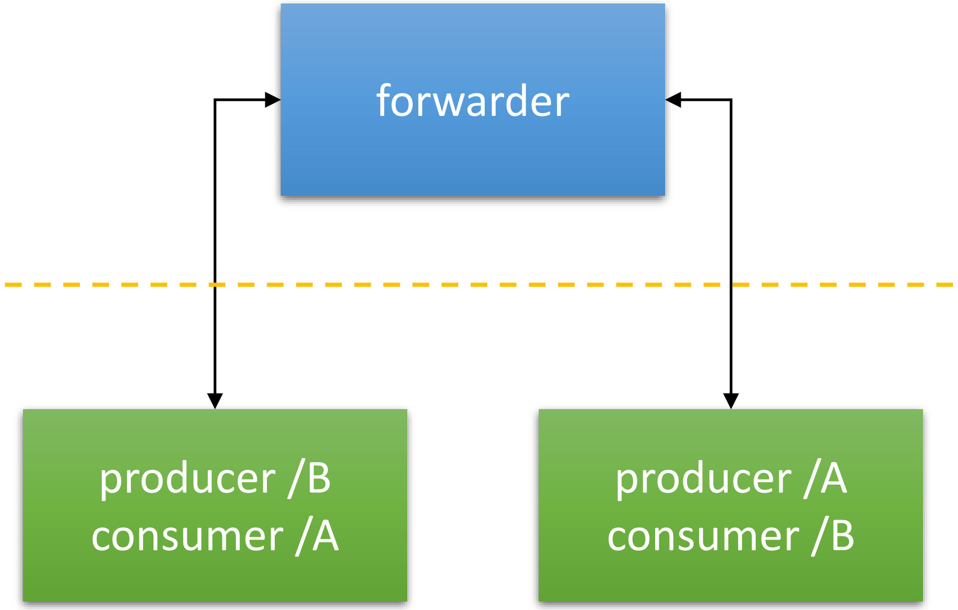
PIT-CS Composite Table (PCCT)



Benchmarks

Spoiler alert: we made it to 100 Gbps

Benchmark Topology

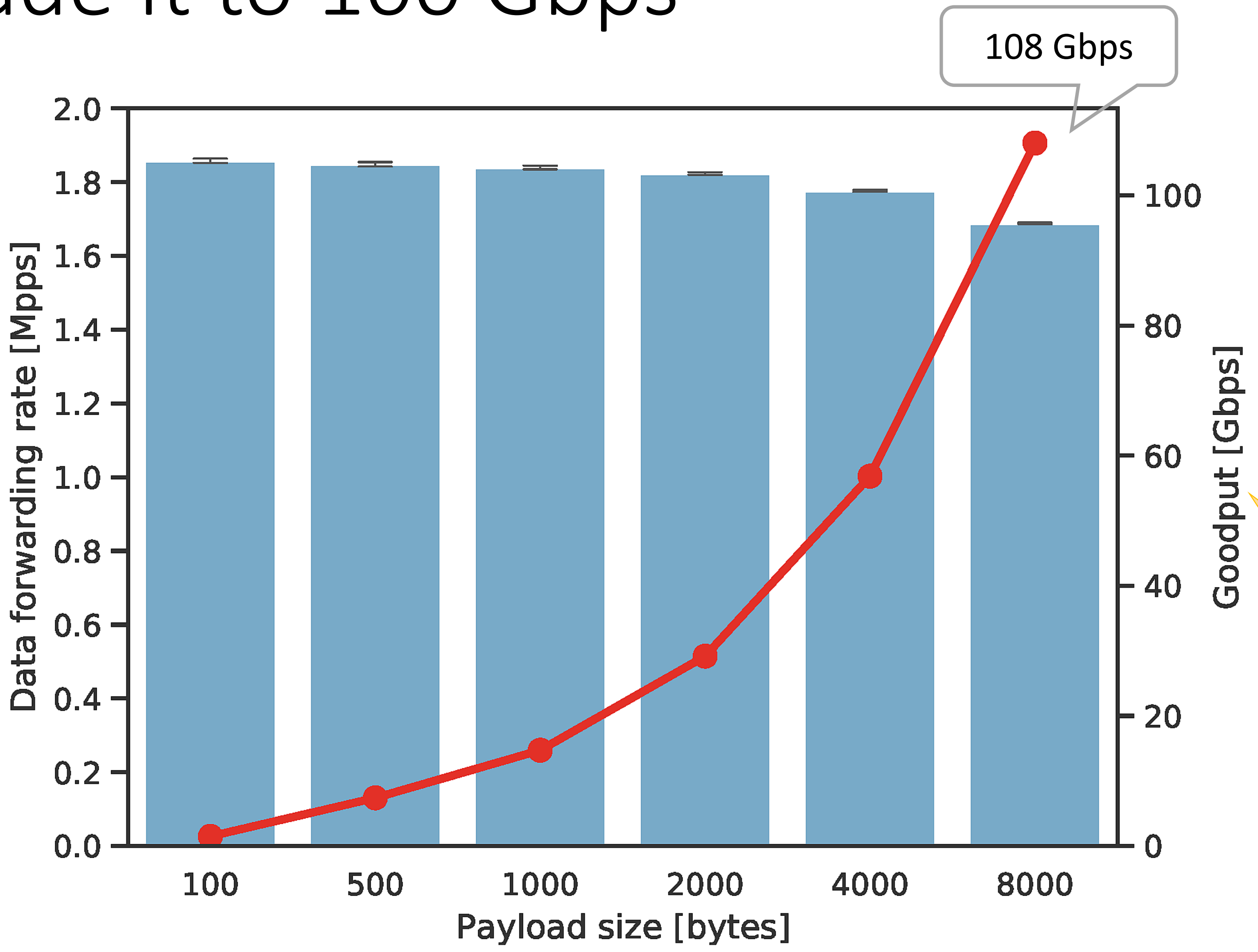


Two physical machines:

- Forwarder.
- Traffic generators: (logically independent)
  - Fetch Data from each other.
  - CUBIC-like congestion control.

- CPU: dual Intel Xeon Gold 6240.
  - 18 cores at 2.60 GHz, Hyper Threading disabled.
- Memory: 256 GB, 2933 MHz, four channels.
  - 64x 1GB hugepages per NUMA socket.
- NIC: Mellanox ConnectX-5 100 Gbps.

We Made It to 100 Gbps

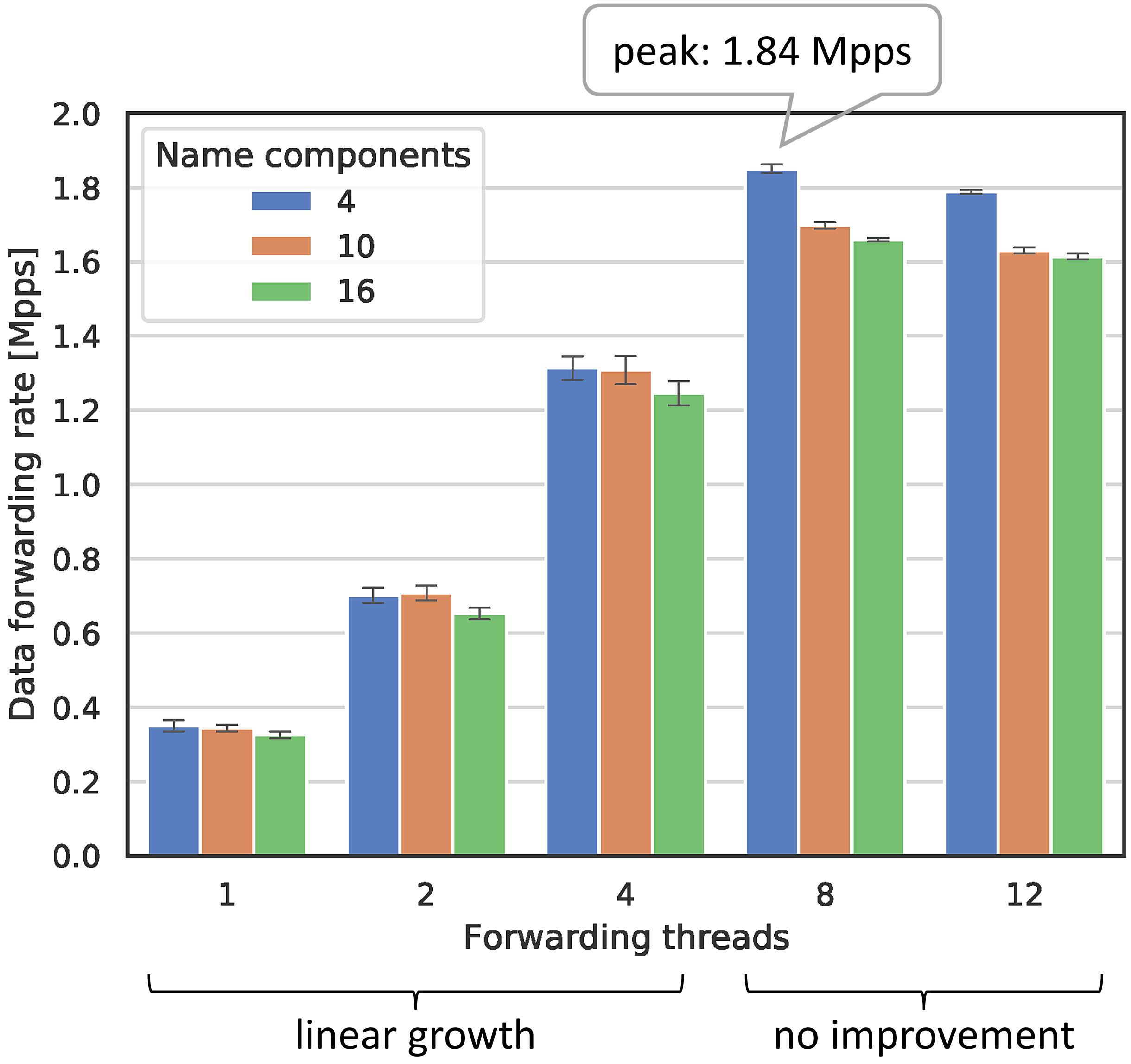


8 forwarding threads.

Data payload only.

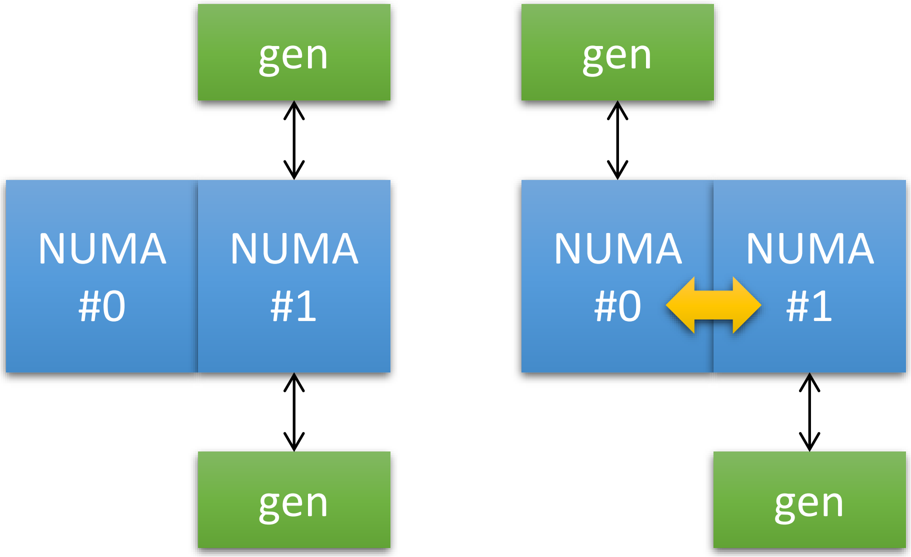
Measured from consumers.
Data packets only.
Not counting retransmissions.

Input Thread Bottleneck



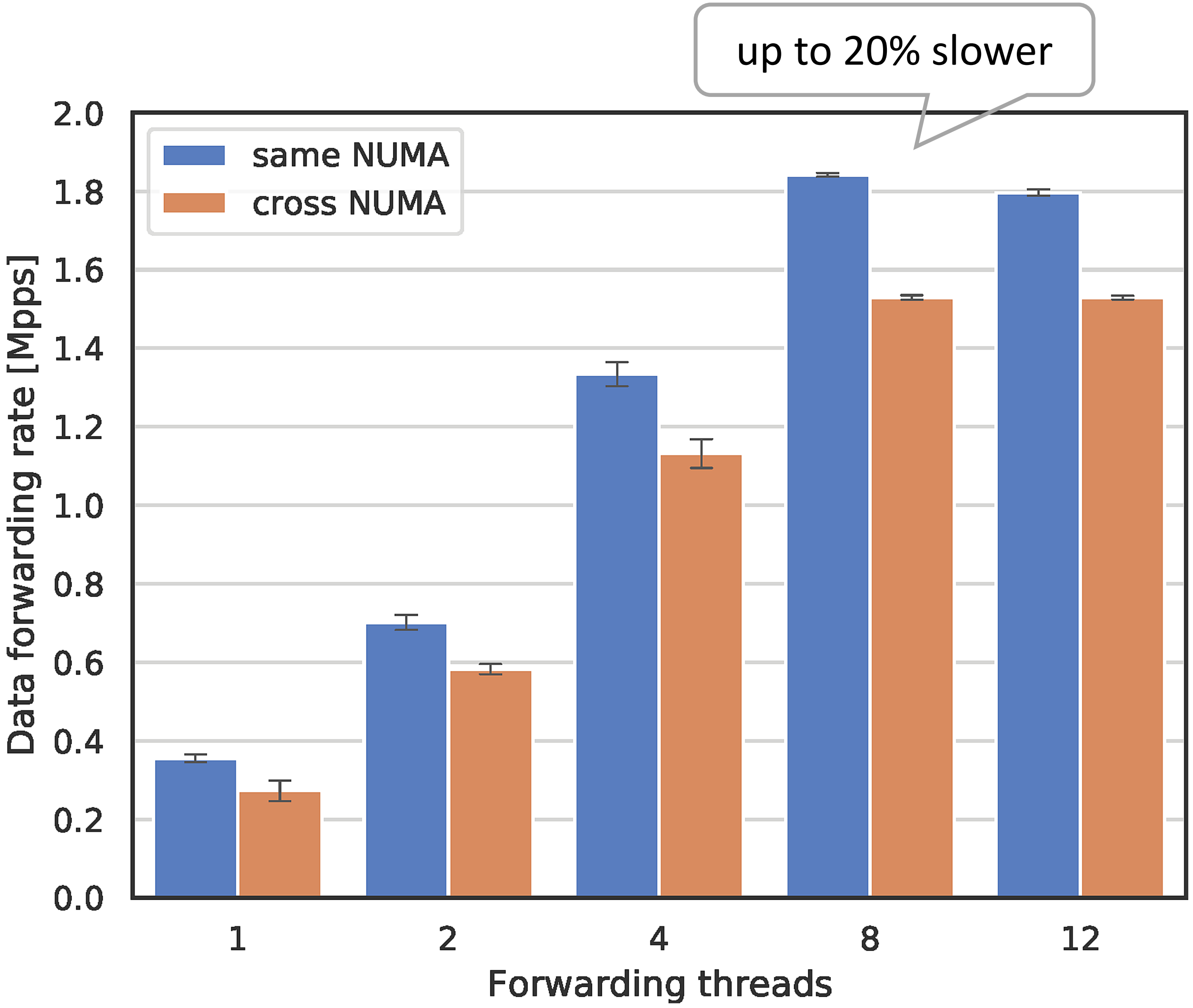
- Expectation:
  - ↑ # forwarding threads
  - ↑ Data forwarding rate (pps)
- Reality:
  - Data forwarding rate plateaus at 8 forwarding threads.
- Bottleneck: input thread.
  - Current architecture only allows one input thread per face.

Effect of Nonlocal Memory Access



same NUMA

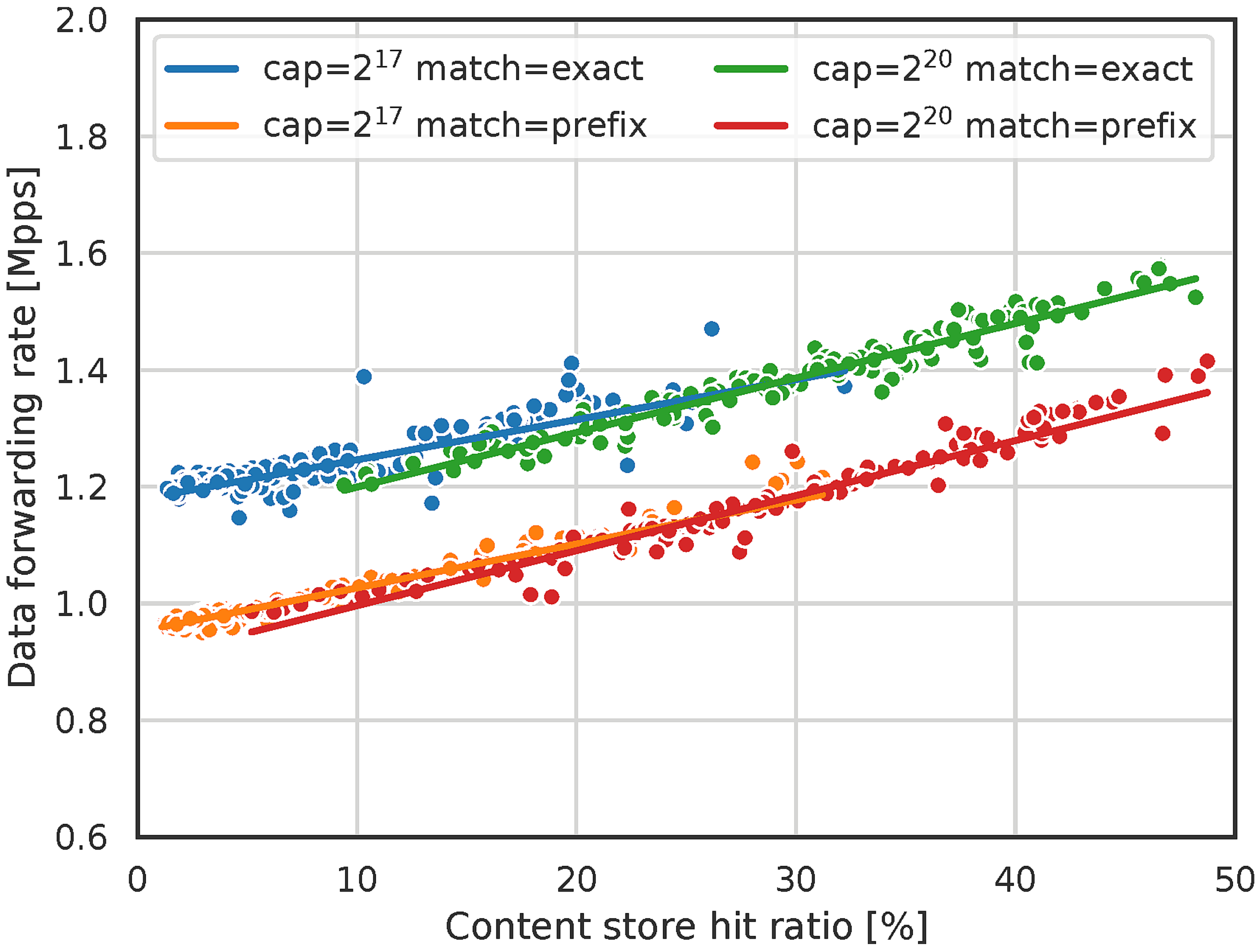
cross NUMA: higher memory access latency

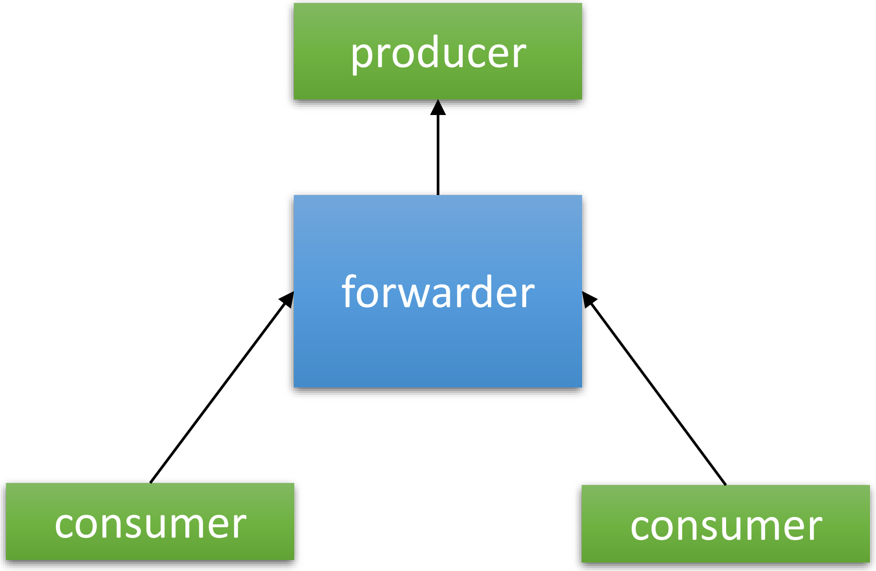


Performance with Large FIB

FIB entries	forwarding rate (kpps)		Interest latency (μs)	
	mean	stdev	median	95th percentile
10^4	1840	5.59	90	227
10^5	1835	4.92	92	234
10^6	1839	4.42	97	249

Forwarding Rate with Large CS

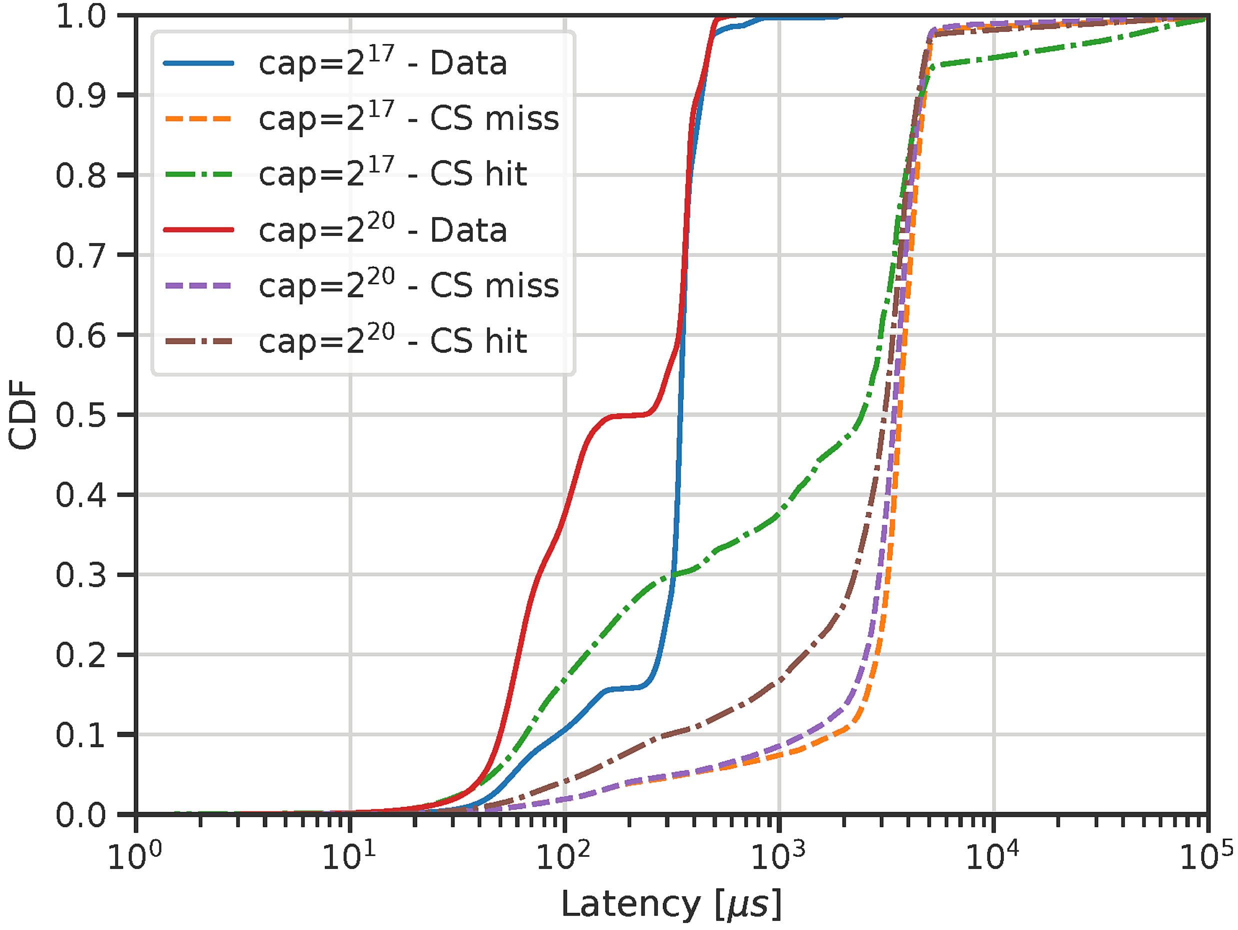




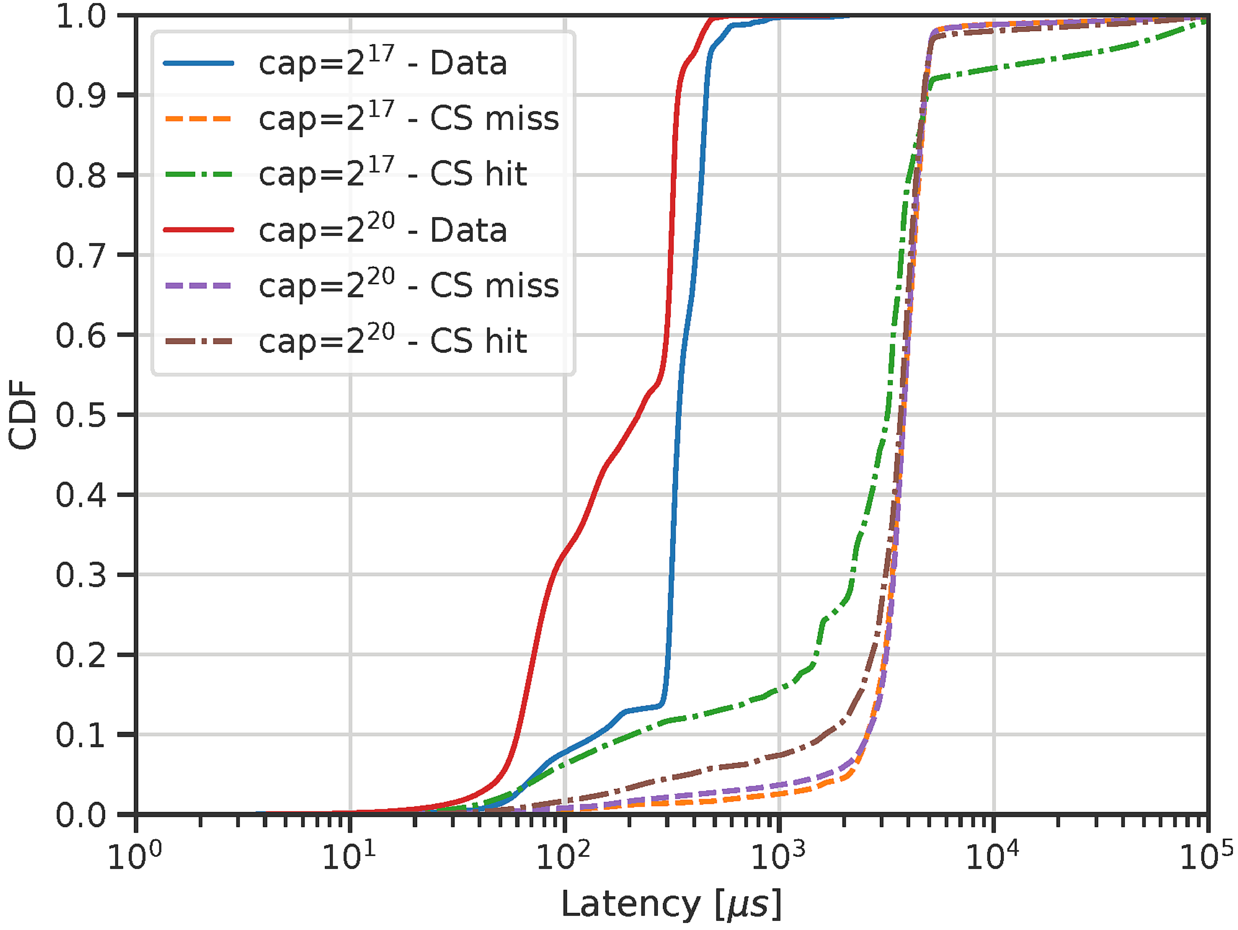
Latency with Large CS

	90th percentile
Interest	4875 µs
Data	456 µs

Exact Match



Prefix Match

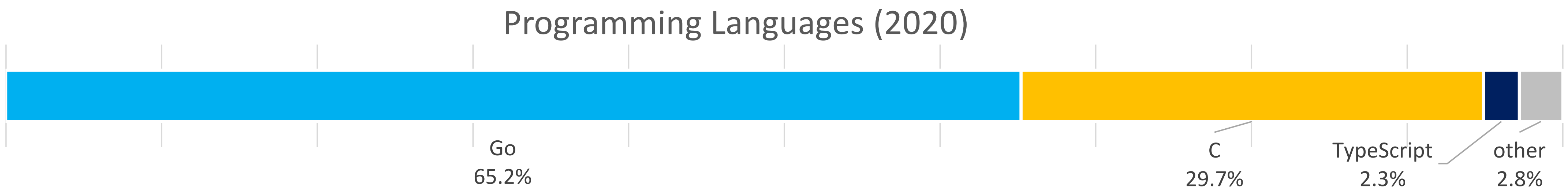


Future Work

- Remove the input thread bottleneck:
  - Design changes to allow multiple input threads per face.
  - Dispatch Data/Nack via Receive Size Scaling (RSS).
  - Dispatch Interest (NDT) using eBPF or FPGA hardware.
- Expand Content Store to NVMe disk storage.
- Load balancing by adjusting NDT entries.
- Performance profiling and improvement toward 200 Gbps.

NDN-DPDK Codebase

- https://github.com/usnistgov/ndn-dpdk
  - Forwarder
  - Traffic Generator
  - GraphQL-based management tools
  - NDNgo library for application development
- Dedicated to public domain



Thank You

Junxiao Shi, Davide Pesavento, Lotfi Benmohamed

NDN-DPDK: NDN Forwarding at 100 Gbps on Commodity Hardware

7th ACM Conference on Information-Centric Networking (ICN 2020)