Generic External Memory for Switch Data Planes

Daehyeok Kim

Yibo Zhu, Changhoon Kim, Jeongkeun Lee, Srinivasan Seshan

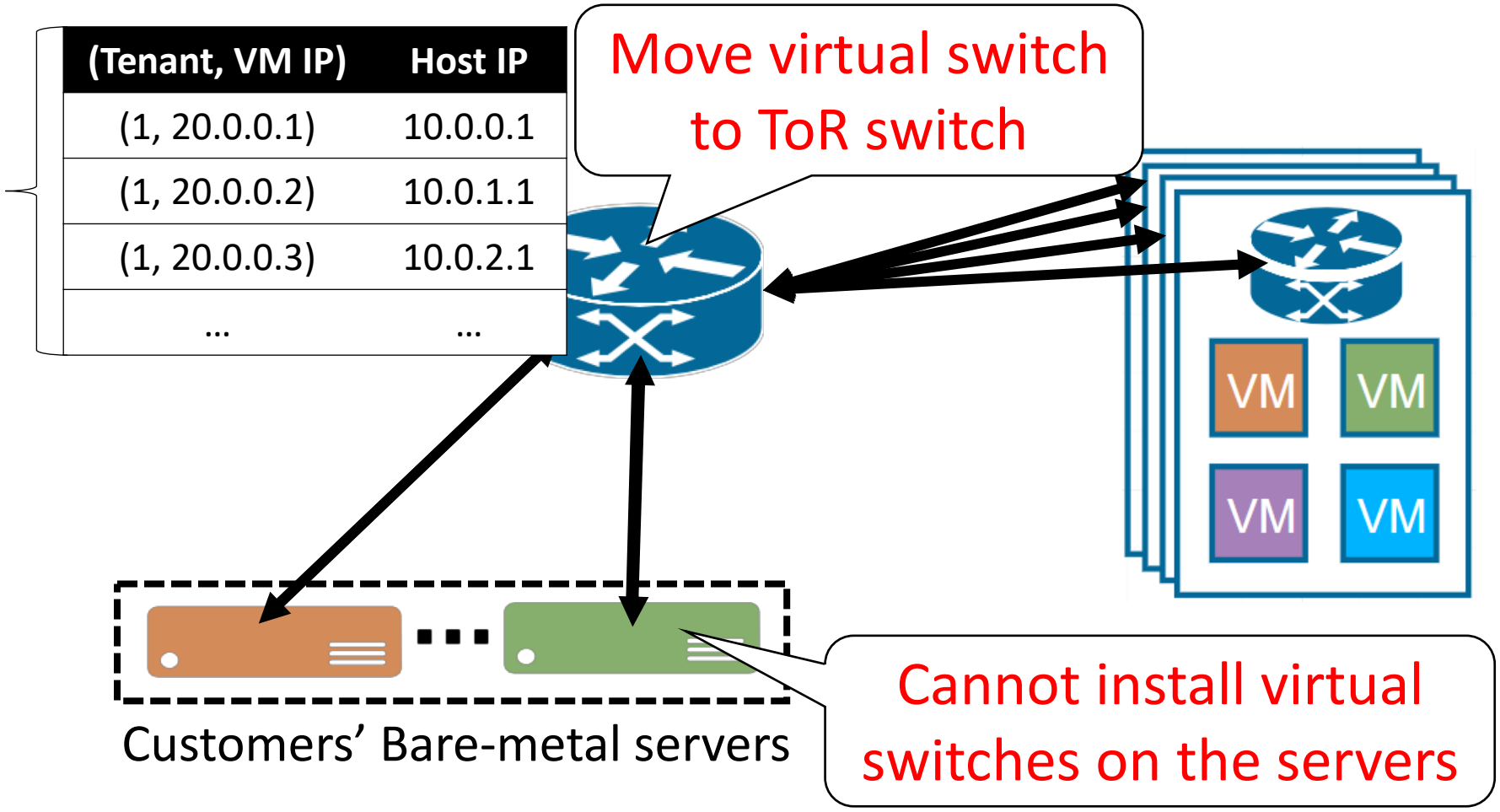
Carnegie Mellon University

Enabling Virtual Switching on ToR Switch

Multi-million Entries

≫ SRAM size!

(Tenant, VM IP)	Host IP
(1, 20.0.0.1)	10.0.0.1
(1, 20.0.0.2)	10.0.1.1
(1, 20.0.0.3)	10.0.2.1
…	…



Limited SRAM space is bottleneck for memory-intensive applications!

Current Trend: Moving Functionality to Switches

SilkRoad: Making Stateful Layer-4 Load Balancing Fast and Cheap Using Switching ASICs

HULA: Scalable Load Balancing Using Programmable Data Planes

Heavy-Hitter Detection Entirely in the Data Plane

Language-Directed Hardware Design for

One Sketch to Rule Them All: Rethinking Network Flow Monitoring with UnivMon

All these applications can benefit from large memory space!

NetCache: Balancing Key-Value Stores with Fast In-Network Caching

Just Say NO to Paxos Overhead: Replacing Consensus with Network Ordering

NetChain: Scale-Free Sub-RTT Coordination

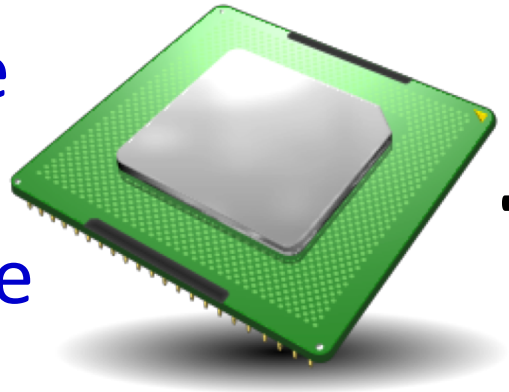
Paxos Made Switch-y

Huynh Tu Dang* Marco Canini† Fernando Pedone* Robert Soulé*

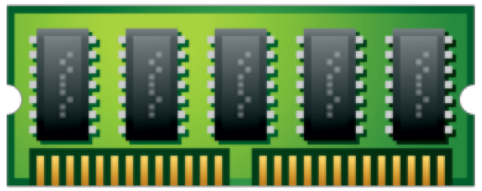
Programmable Switch Chips Need More Memory

- Programmable data plane technology
  - E.g., Protocol-Independent Switch Architecture (PISA) + P4
  - Flexible but only with on-chip SRAM cache

Programmable switch chip w/ SRAM cache



+



DRAM

= Lots of innovative applications!

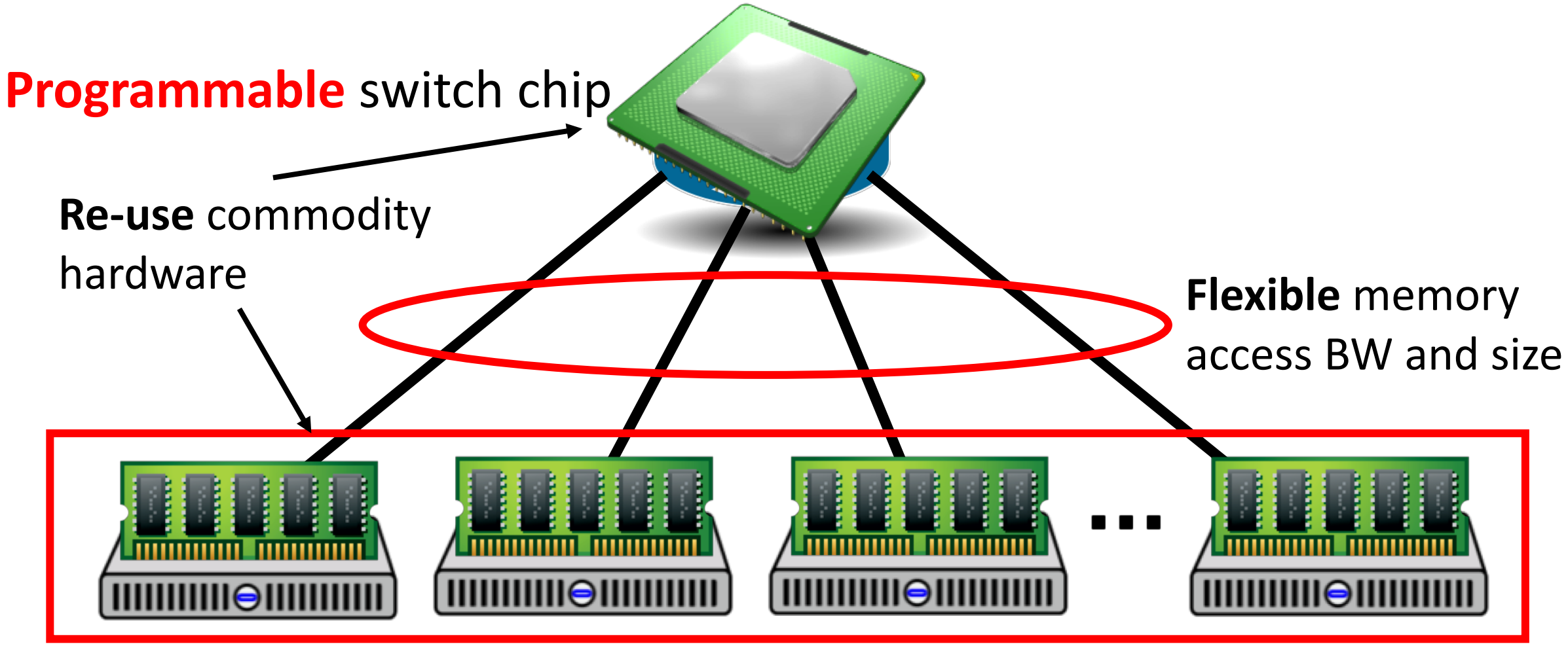
Status quo

- *Fixed-function* switch chips built with *fixed-function* external memory

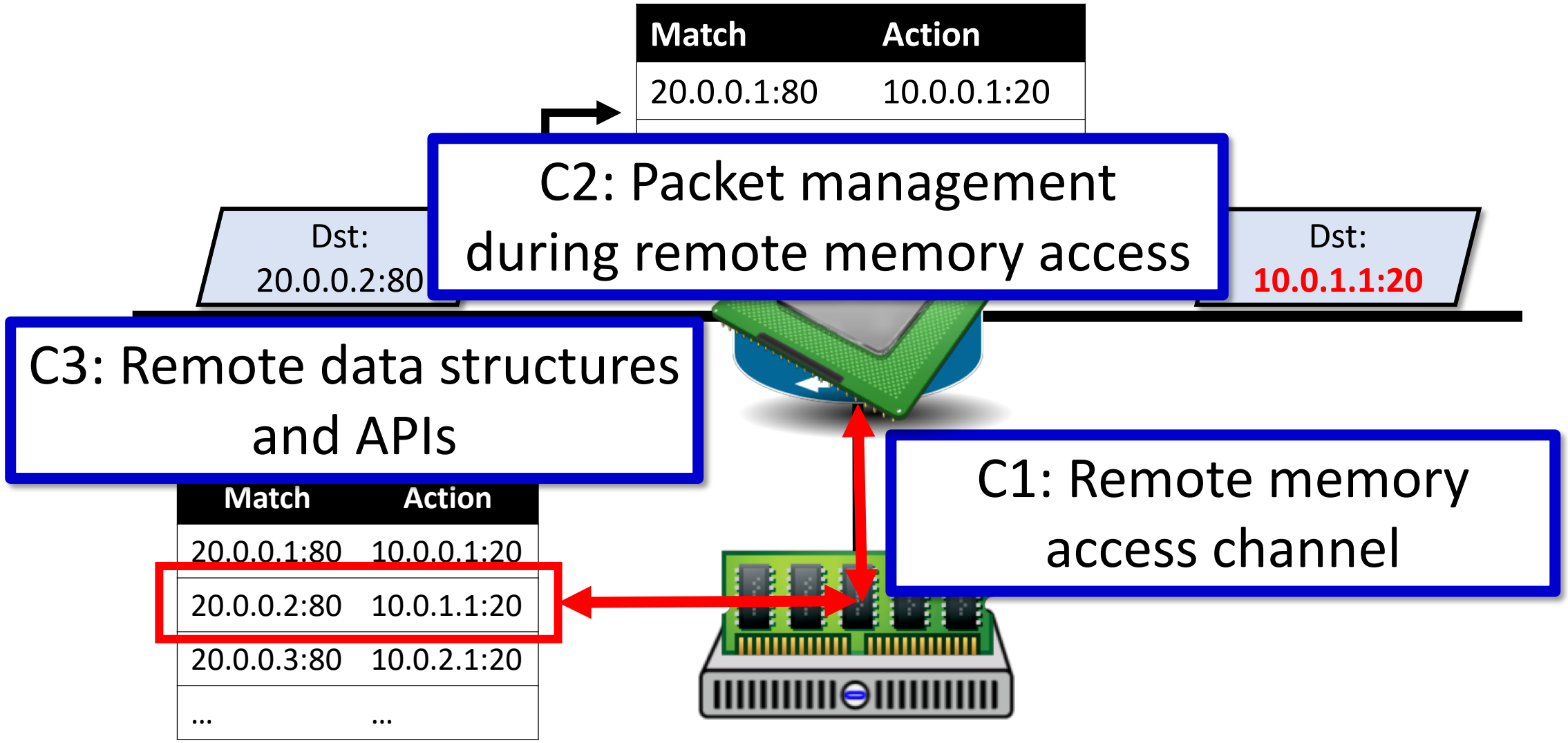
- These aren't very useful
  - Inflexible: Usage fixed at design time
  - Fixed and small scale: Memory size and bandwidth fixed at design time
  - Expensive: Chip getting larger and complex

Is programmable switch chip + general-purpose memory possible?

GEM: Generic External Memory for Programmable Data Planes



Key Components



C1: Remote Memory Access Channel

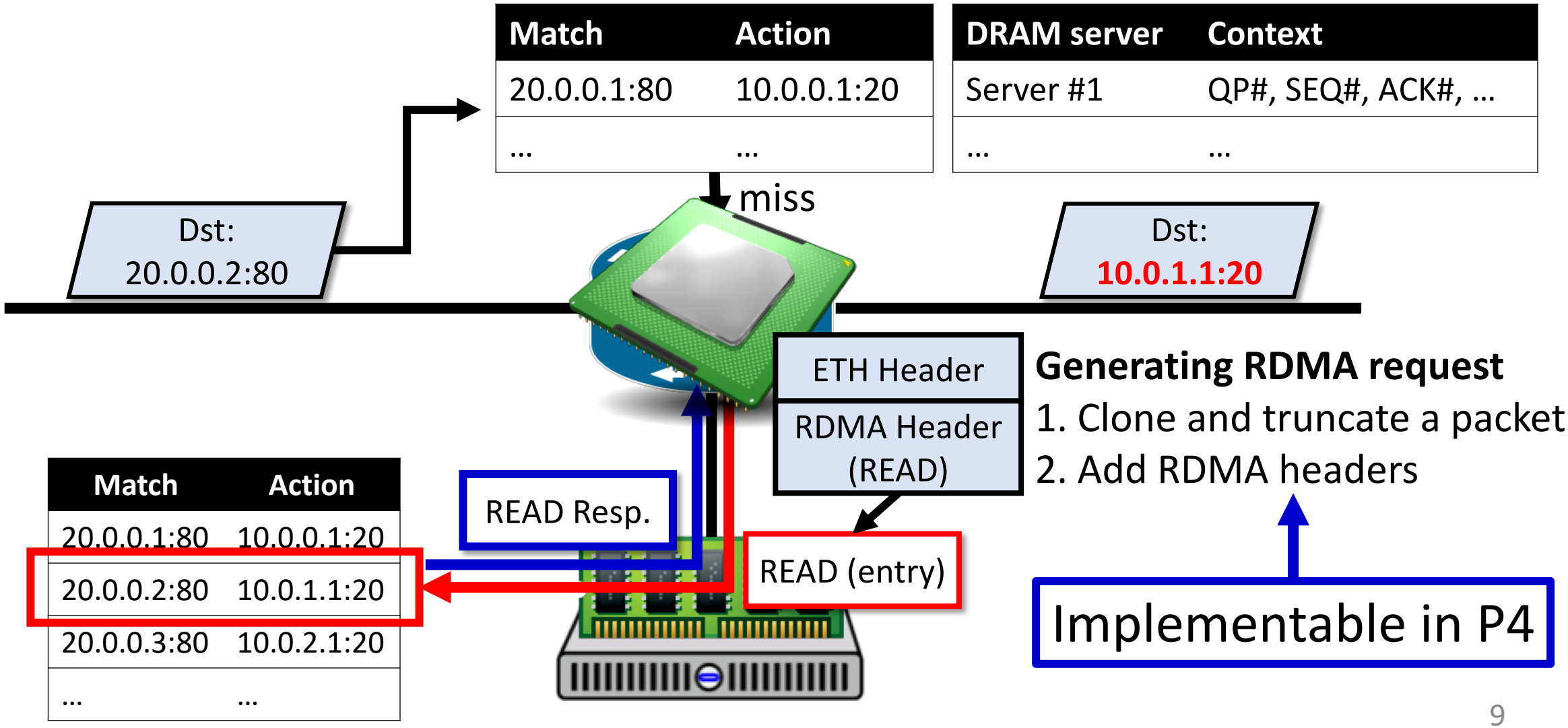
- Goal: Enable programmable switch chip to *directly* access memory
  - *Purely* access DRAM: No impact to the server's existing compute and networking workloads
  - Minimal latency between the chip and memory

Leverage RDMA!

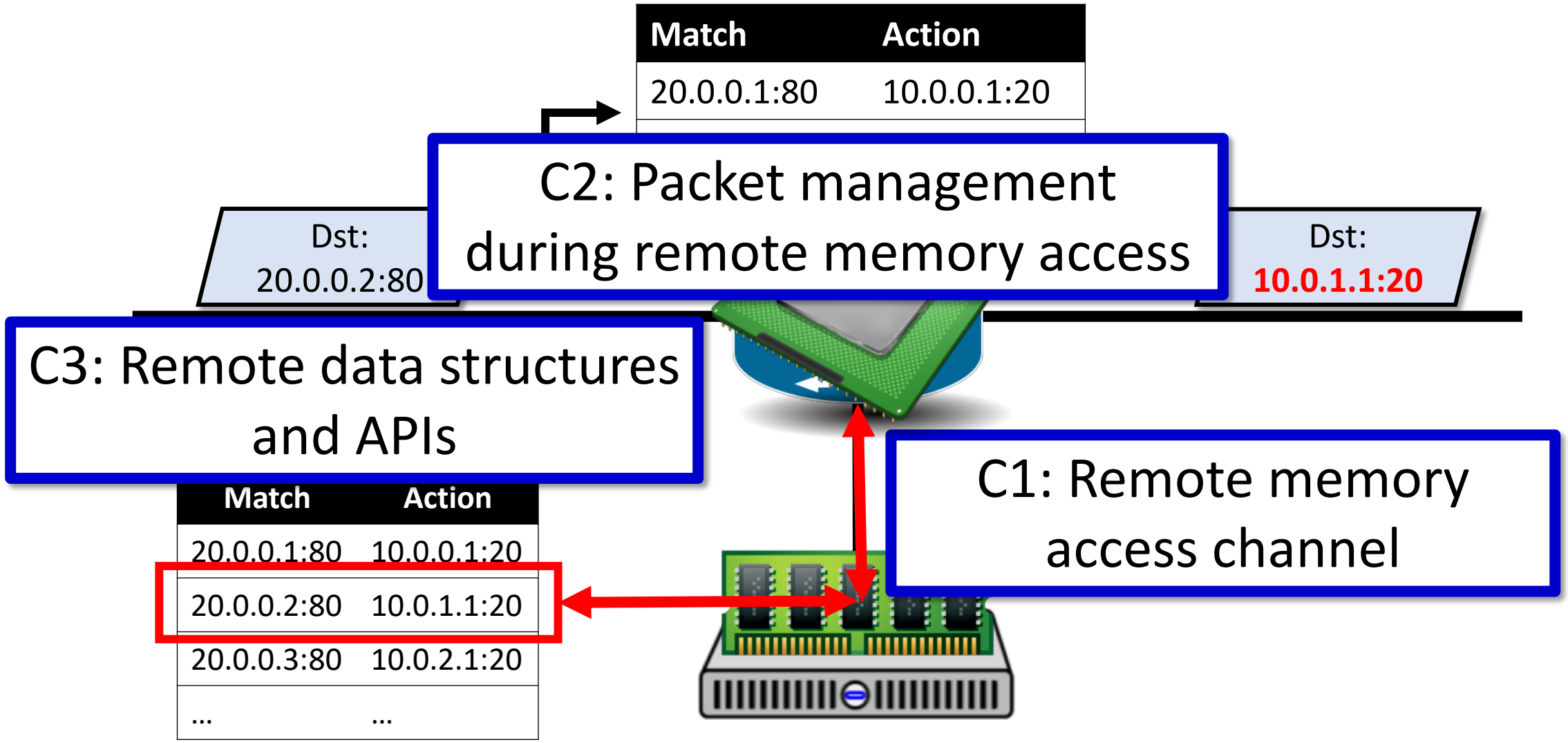
- Challenge: How to generate RDMA requests from the data plane?
  - Programmable switch chip cannot generate arbitrary new packets

*RDMA: Remote Direct Memory Access

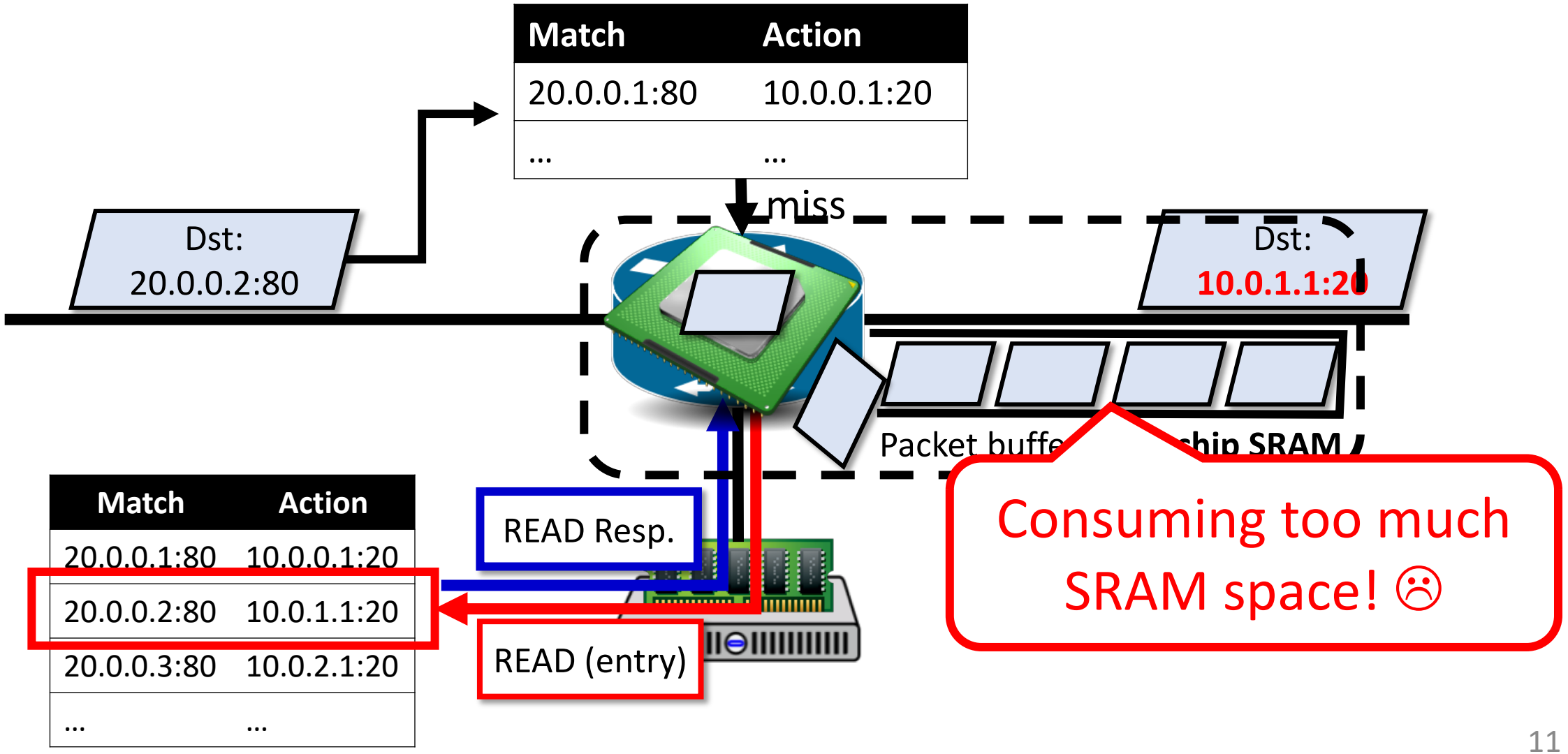
Accessing Remote Memory from Data Plane via RDMA



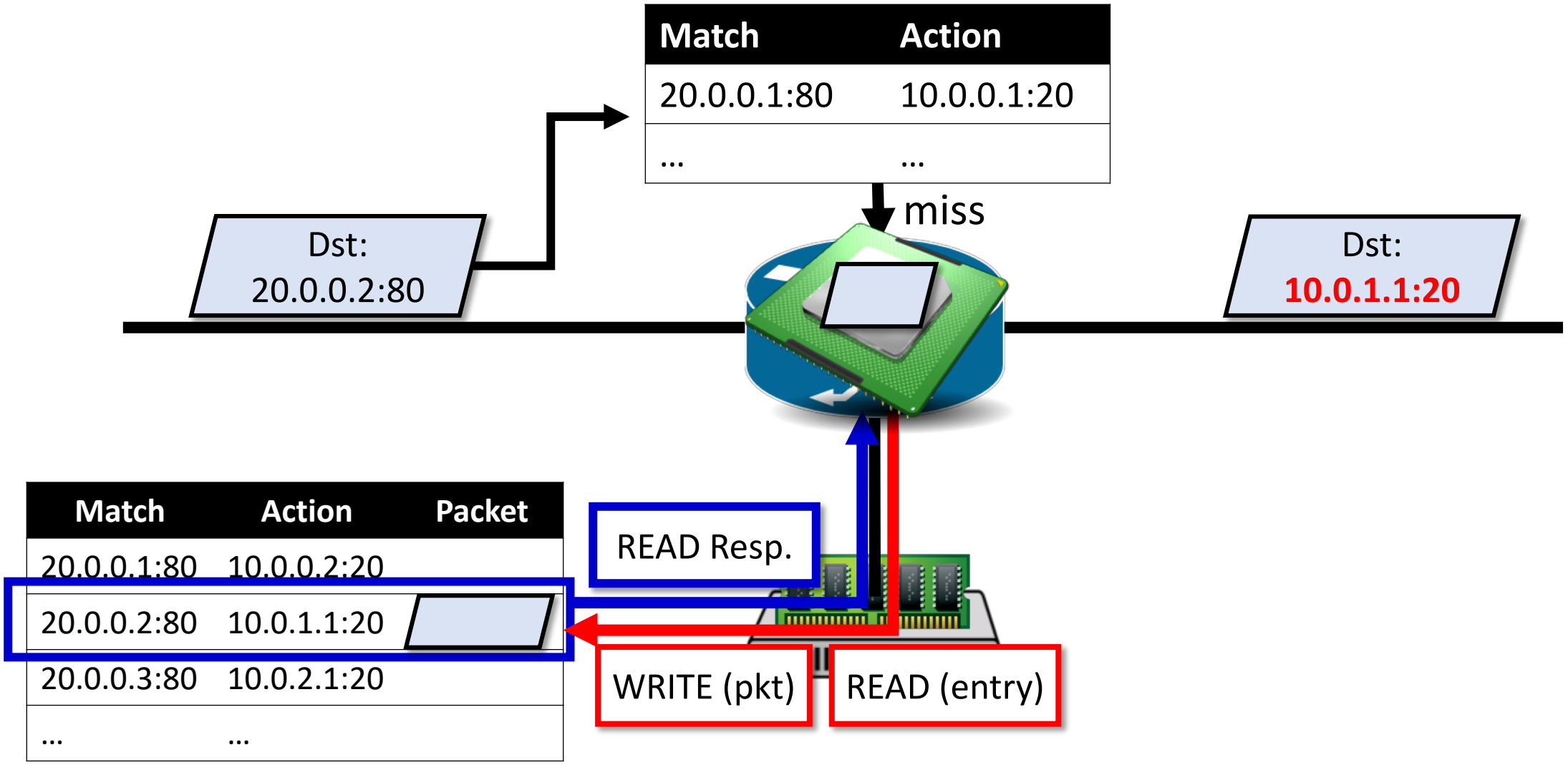
Key Components



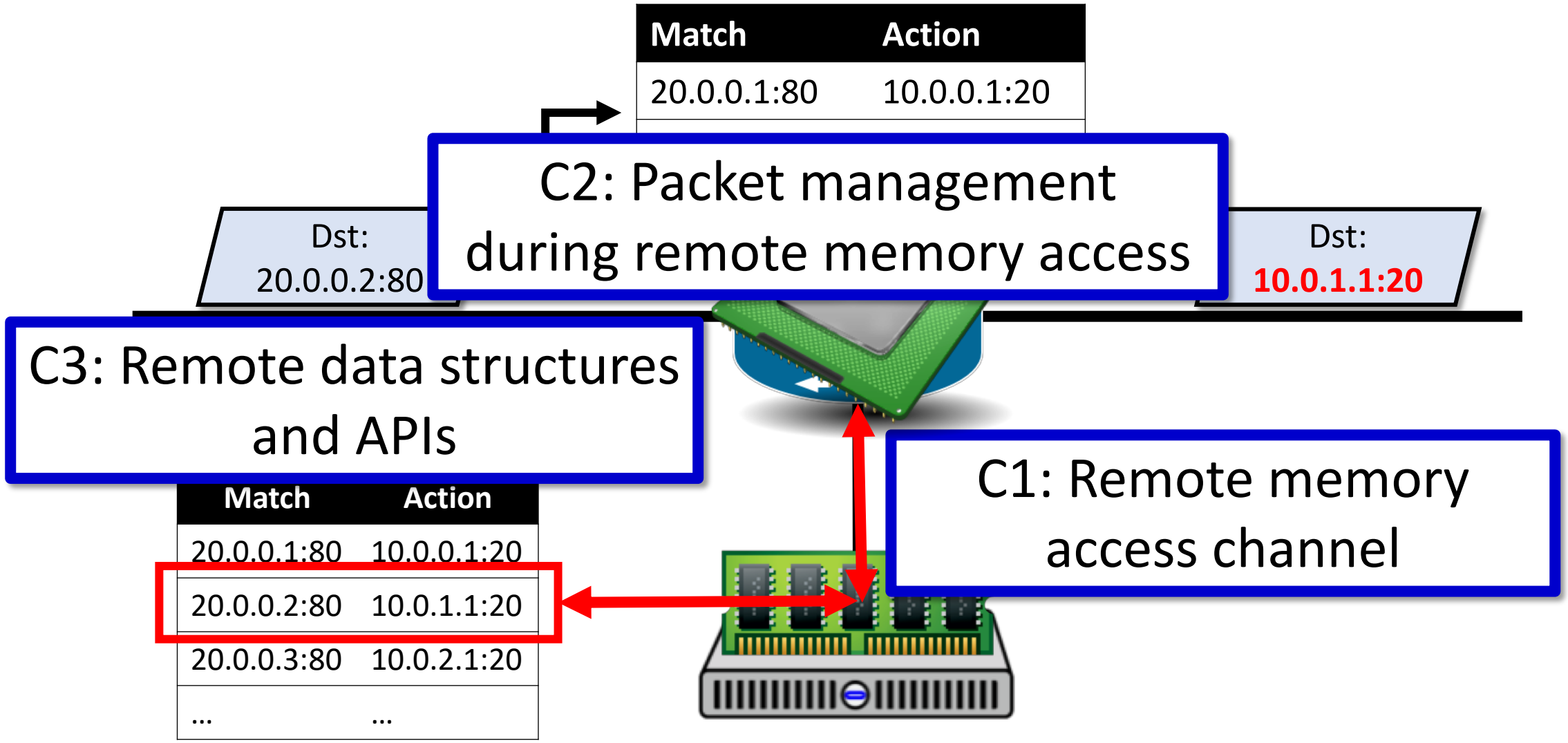
C2: Packet Management during Remote Memory Access



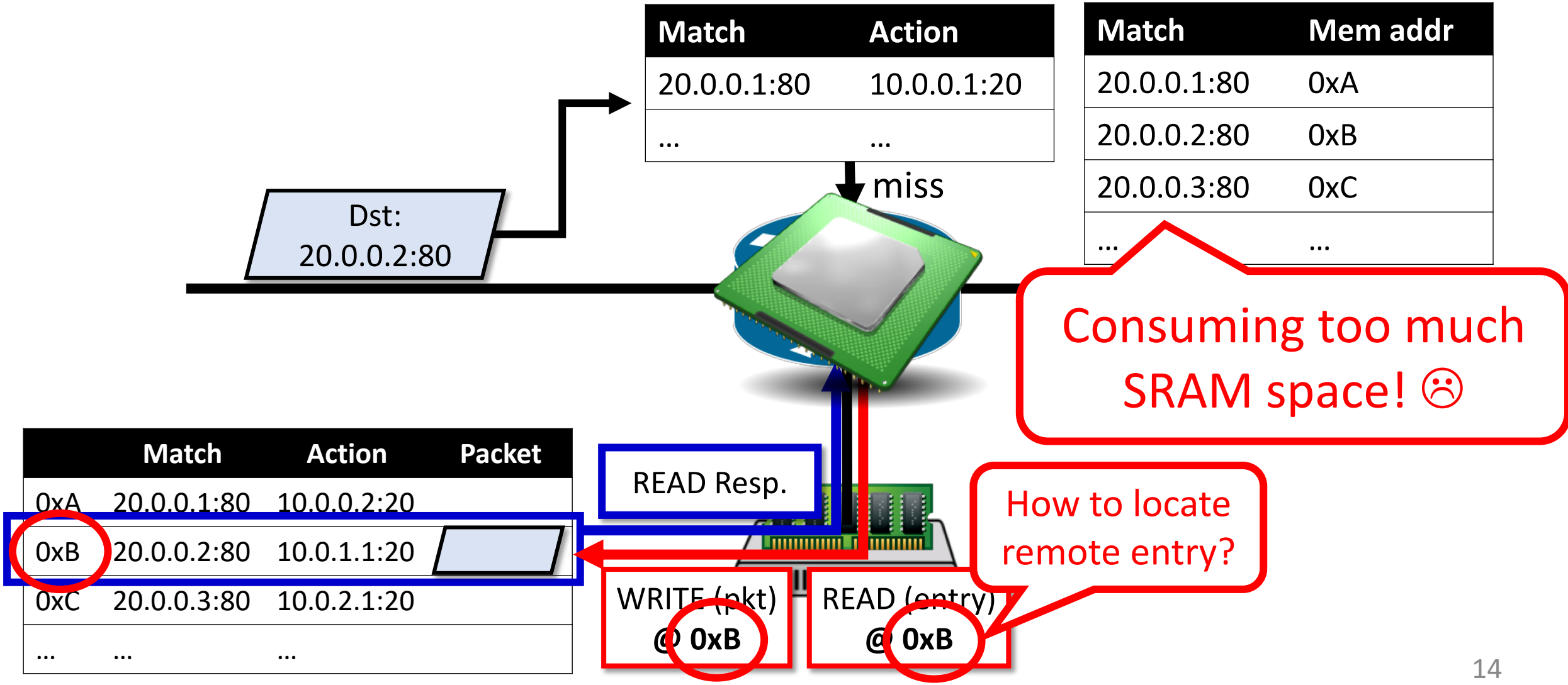
Depositing Packets on Remote Buffer



Key Components



C3: Remote Data Structures and APIs

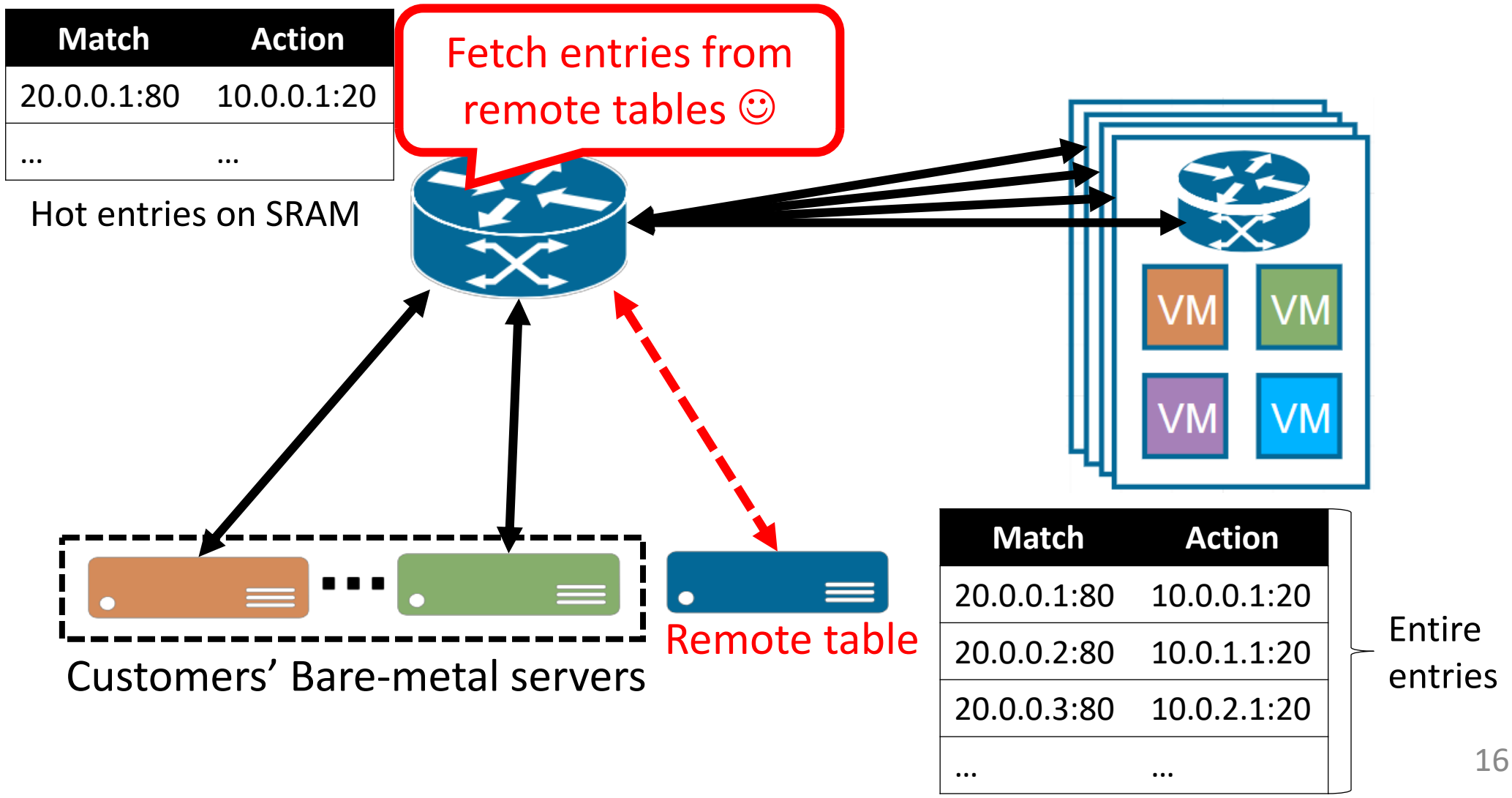


General Data Structures and APIs?

- Ongoing work: designing general data structures for remote memory

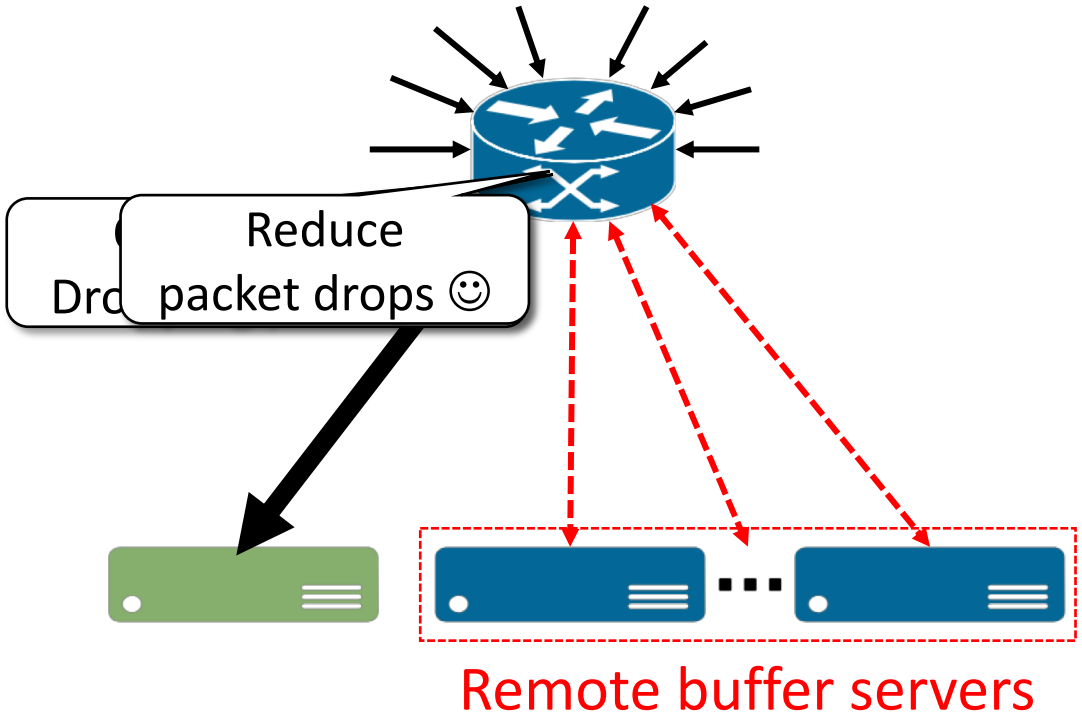
- Proof-of-concept use cases for specific applications
  - Lookup table extension for extending virtual switch table
  - Packet buffer extension for mitigating packet drops due to incast
  - State store extension for network telemetry

Use Case: Extending Lookup Table

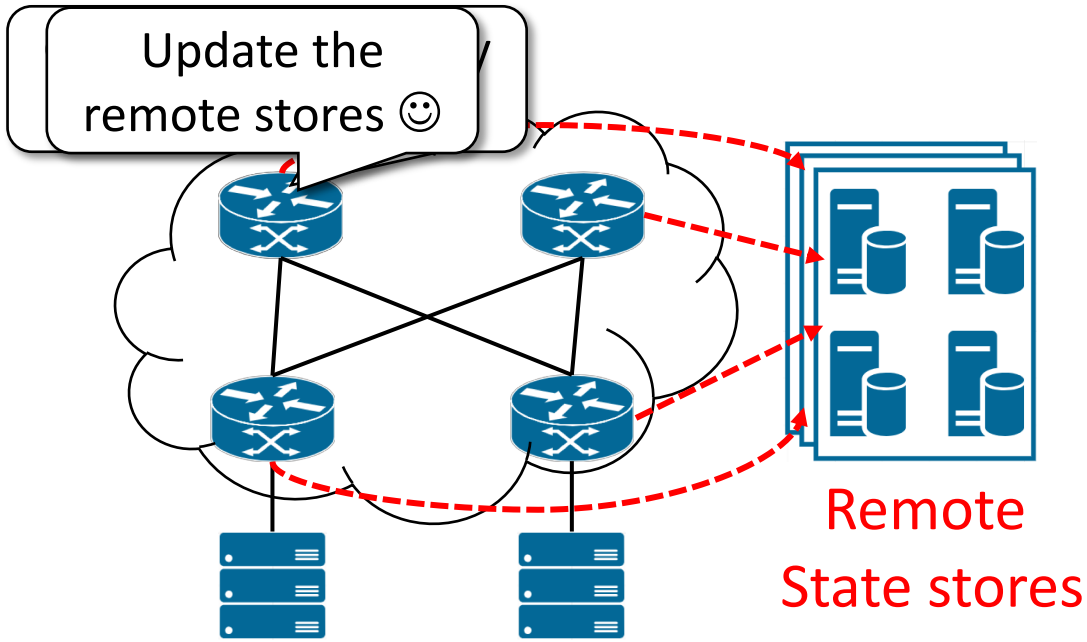


Other Use Cases

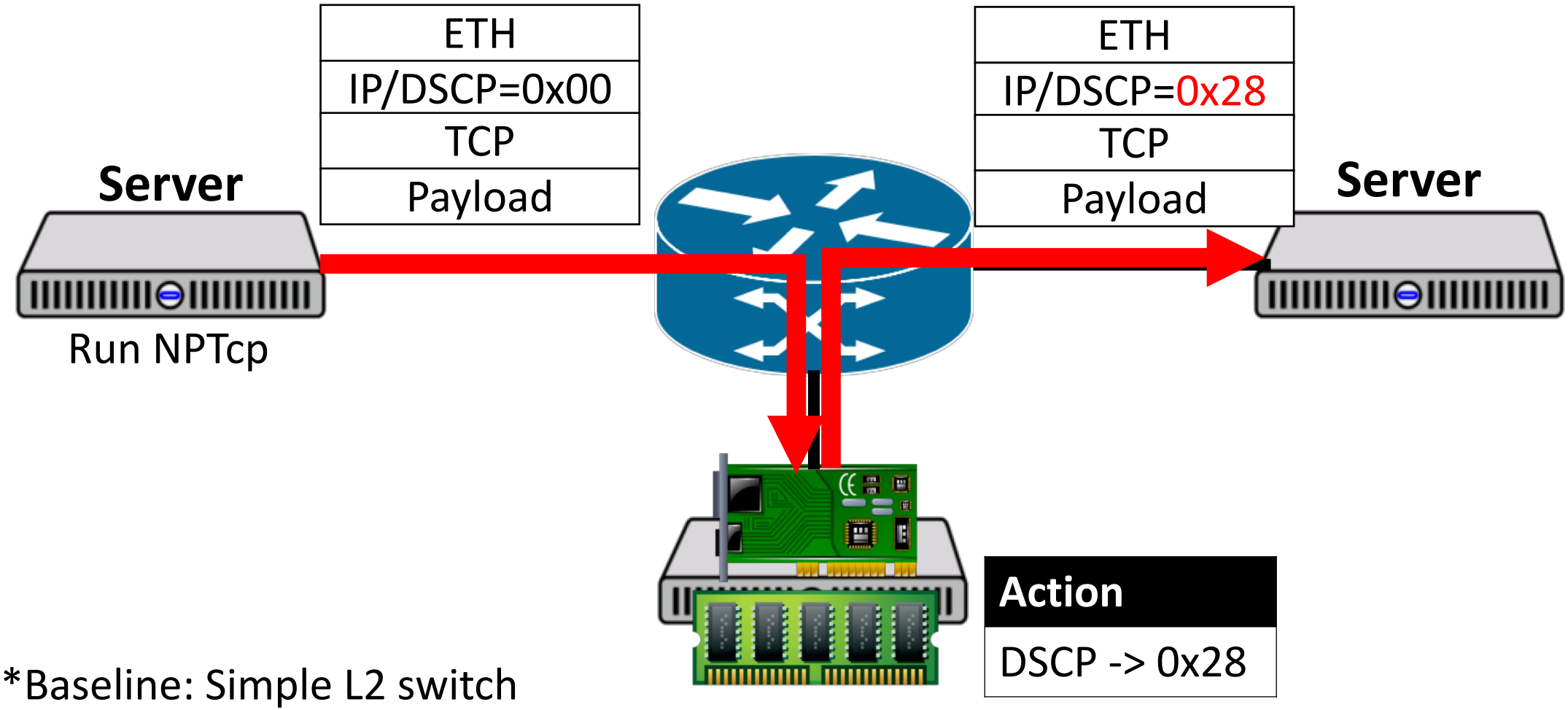
- **Packet buffer extension** for mitigating packet drops



- **State store extension** for network telemetry



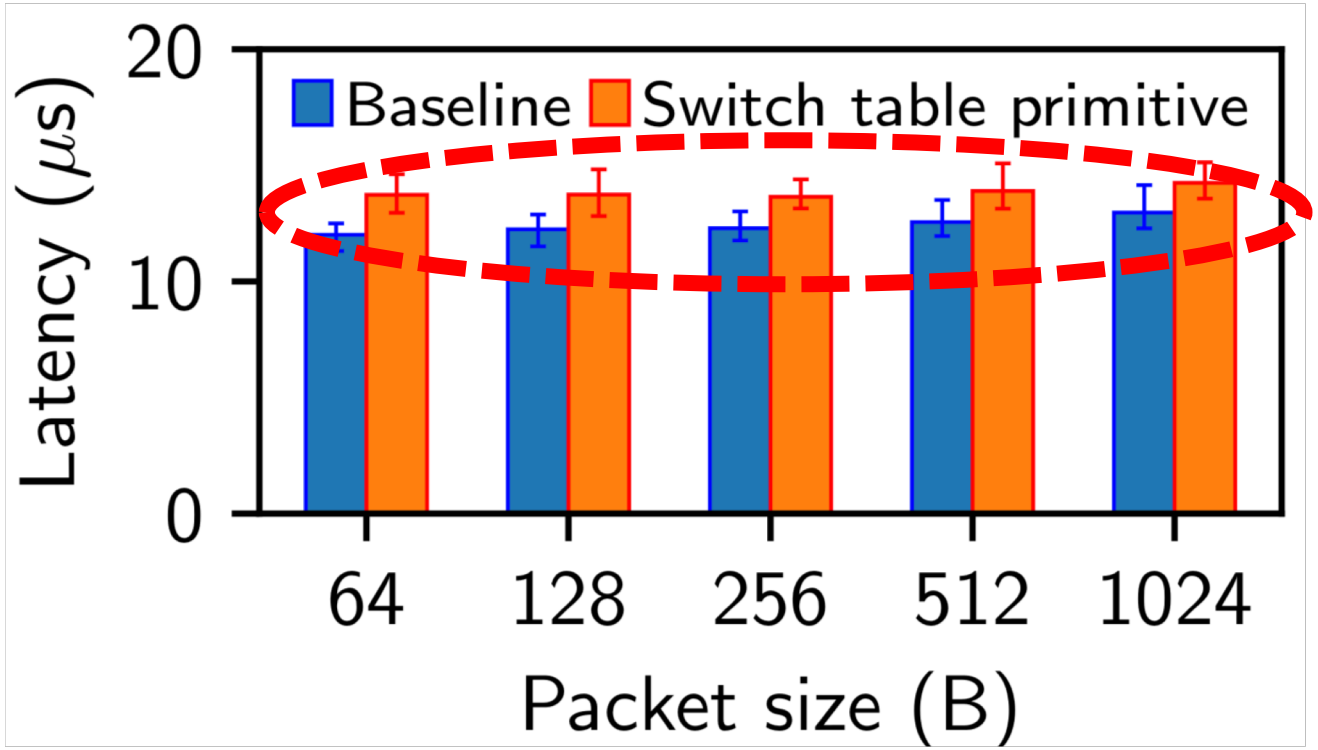
Experiment Setup



*Baseline: Simple L2 switch

Results

- End-to-end latency

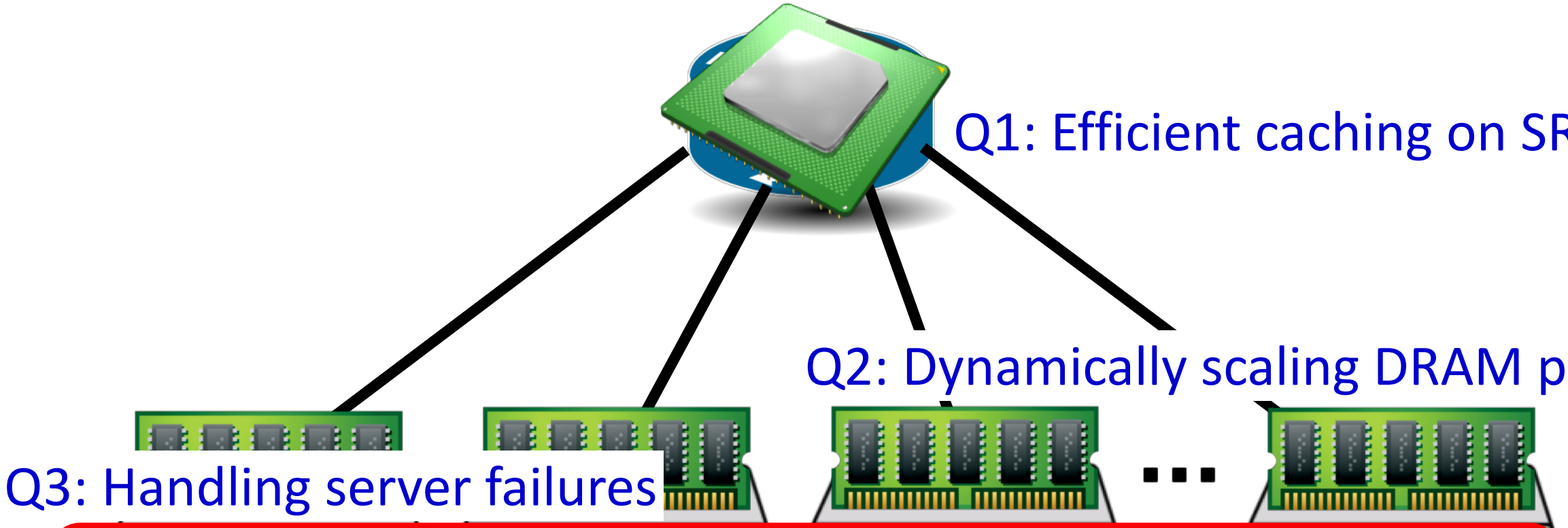


1 - 2 μs additional latency

- Packet store / load throughput: close to the line rate (≈ 37.5 Gbps)

Summary

Vision: **G**eneric **E**xternal **M**emory for Programmable Data Plane



Q1: Efficient caching on SRAM

Q2: Dynamically scaling DRAM pool

Q3: Handling server failures

GEM will be a key enabler
for innovations in networking and computational networking!