SwiShmem: Distributed Shared State Abstractions for Programmable Switches

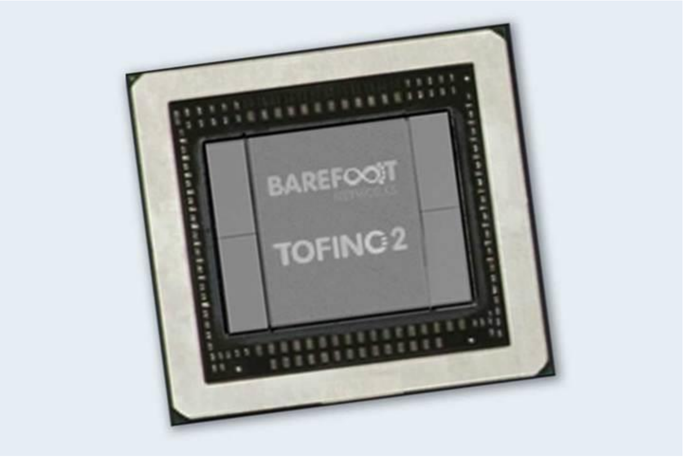
Lior Zeno, Dan R. K. Ports, Jacob Nelson, Mark Silberstein

Technion
Israel Institute
of Technology

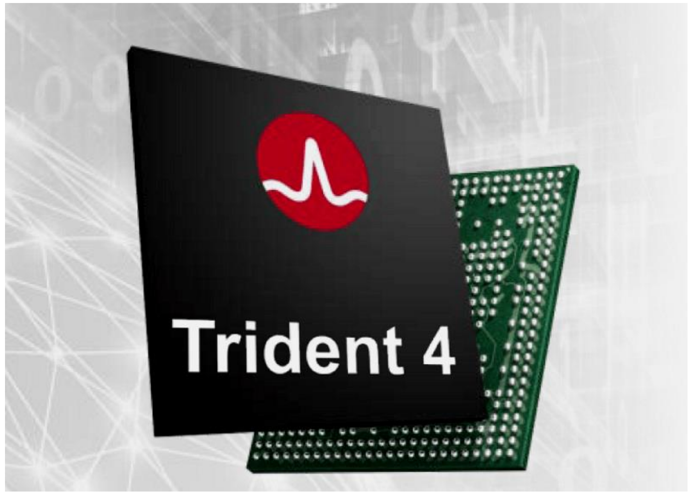
ACSL
Accelerated Computing
Systems Lab

Microsoft

Hardware Trend: PISA



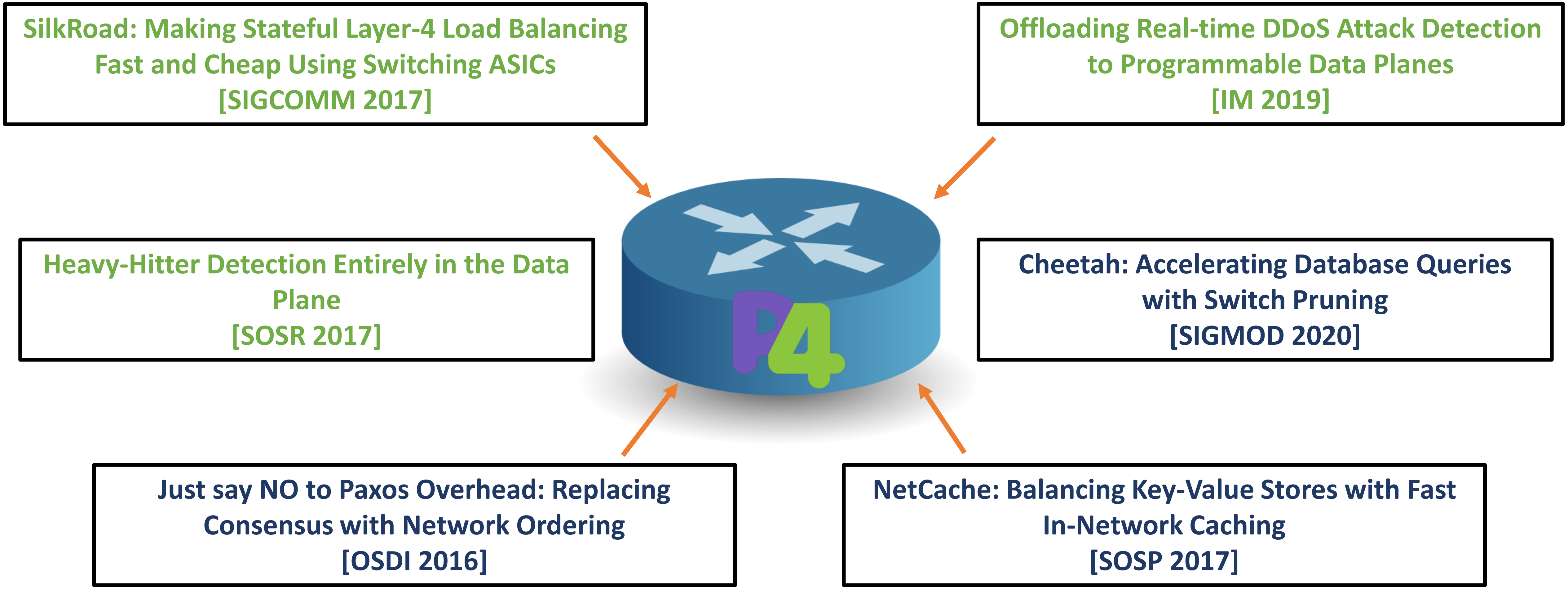
Barefoot Tofino



Broadcom Trident

NVIDIA Networking Spectrum

Current Trend: In-Switch Acceleration



SilkRoad: Making Stateful Layer-4 Load Balancing Fast and Cheap Using Switching ASICs
[SIGCOMM 2017]

Offloading Real-time DDoS Attack Detection to Programmable Data Planes
[IM 2019]

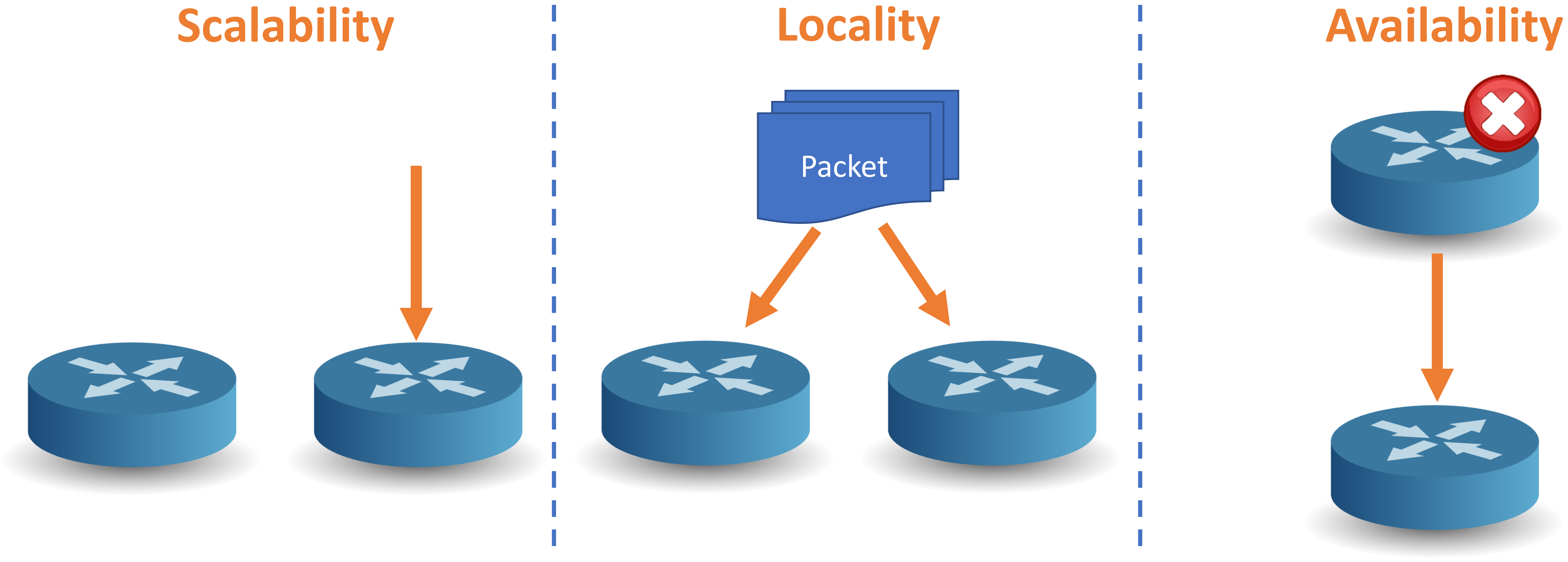
Heavy-Hitter Detection Entirely in the Data Plane
[SOSR 2017]

Cheetah: Accelerating Database Queries with Switch Pruning
[SIGMOD 2020]

Just say NO to Paxos Overhead: Replacing Consensus with Network Ordering
[OSDI 2016]

NetCache: Balancing Key-Value Stores with Fast In-Network Caching
[SOSP 2017]

Unrealized Potential



Scalability

Locality

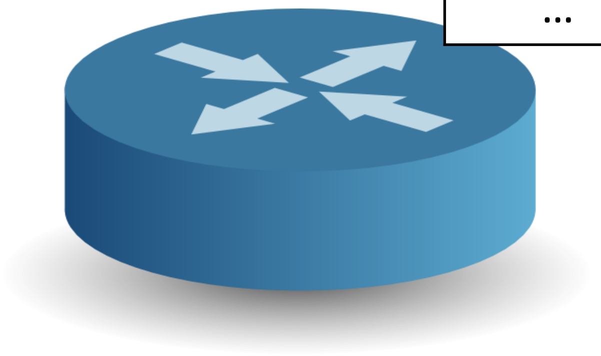
Availability

Problem: Stateful + Distributed is a challenging combination

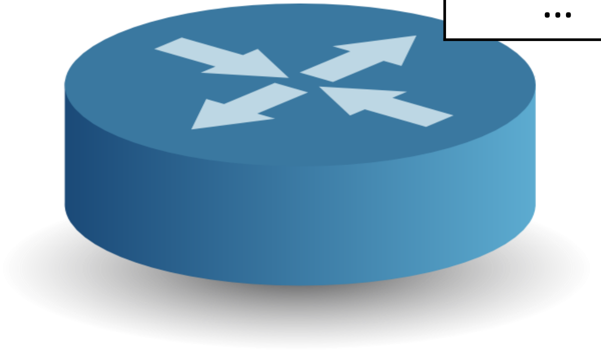
Network-Wide Heavy-Hitters Detection

Match	Counter
flow$_0$	128
flow$_1$	64
flow$_2$	1024
…	…

Data-plane updateable registers

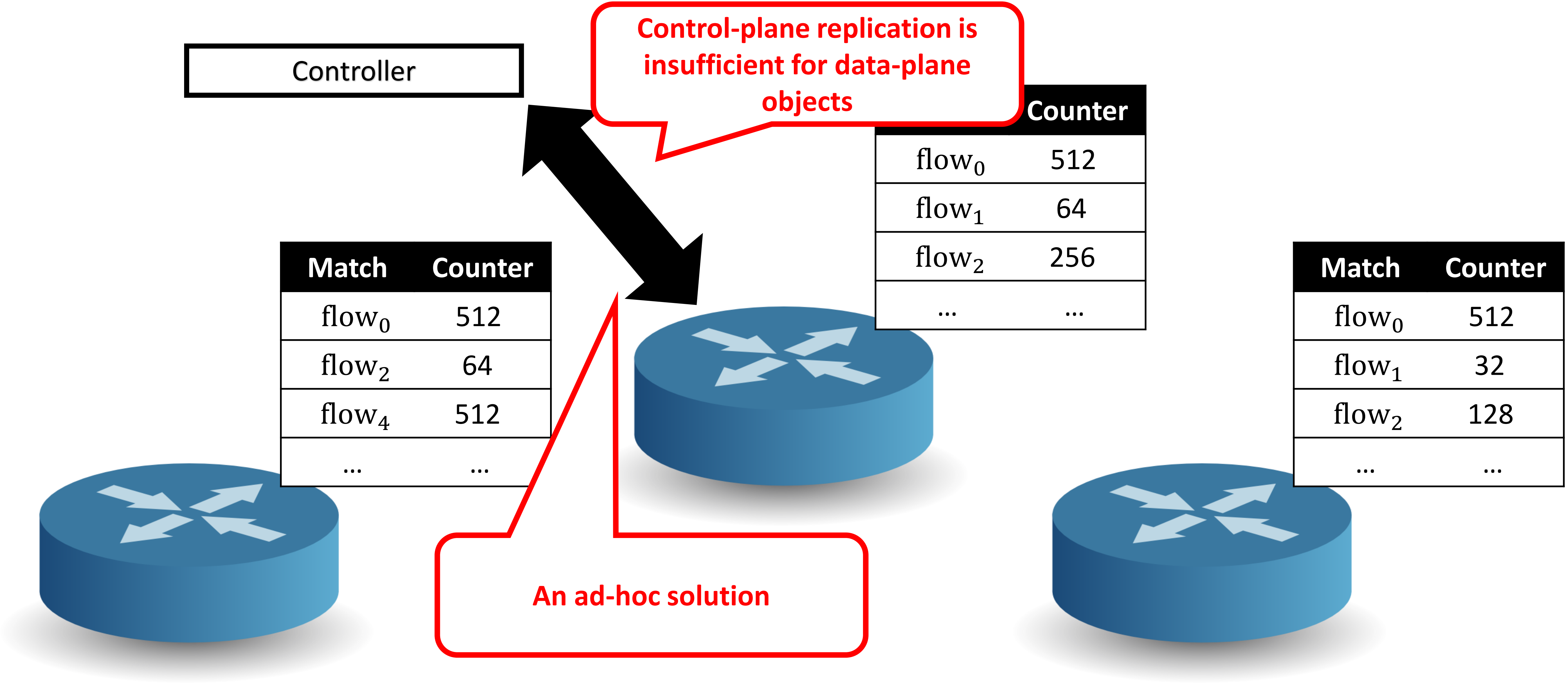


Network-Wide Heavy-Hitters Detection



Match	Counter
flow_0	128
flow_1	64
flow_2	1024
…	…

Network-Wide Heavy-Hitters Detection



Control-plane replication is insufficient for data-plane objects

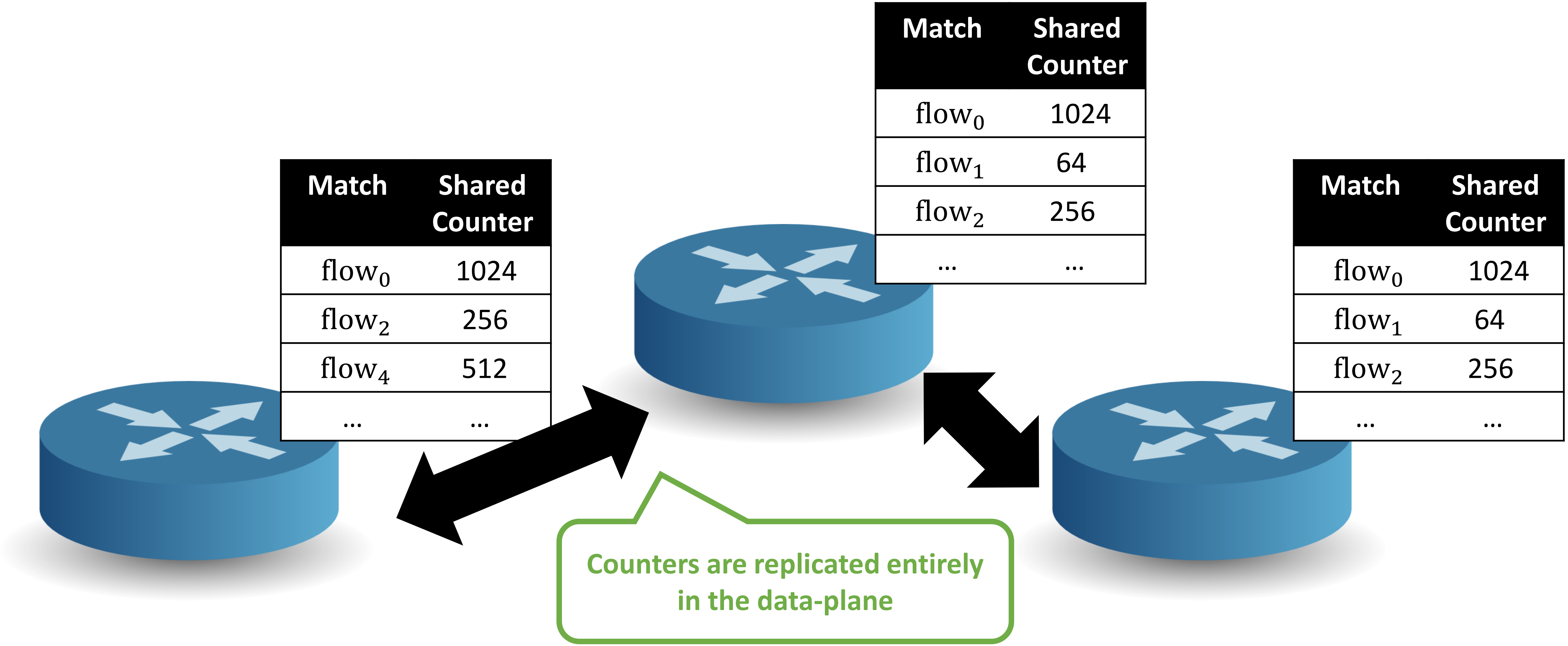
	Counter
flow_0	512
flow_1	64
flow_2	256
…	…

Match	Counter
flow_0	512
flow_2	64
flow_4	512
…	…

Match	Counter
flow_0	512
flow_1	32
flow_2	128
…	…

An ad-hoc solution

Network-Wide Heavy-Hitters Detection



Match	Shared Counter
flow$_0$	1024
flow$_2$	256
flow$_4$	512
…	…

Match	Shared Counter
flow$_0$	1024
flow$_1$	64
flow$_2$	256
…	…

Match	Shared Counter
flow$_0$	1024
flow$_1$	64
flow$_2$	256
…	…

Counters are replicated entirely in the data-plane

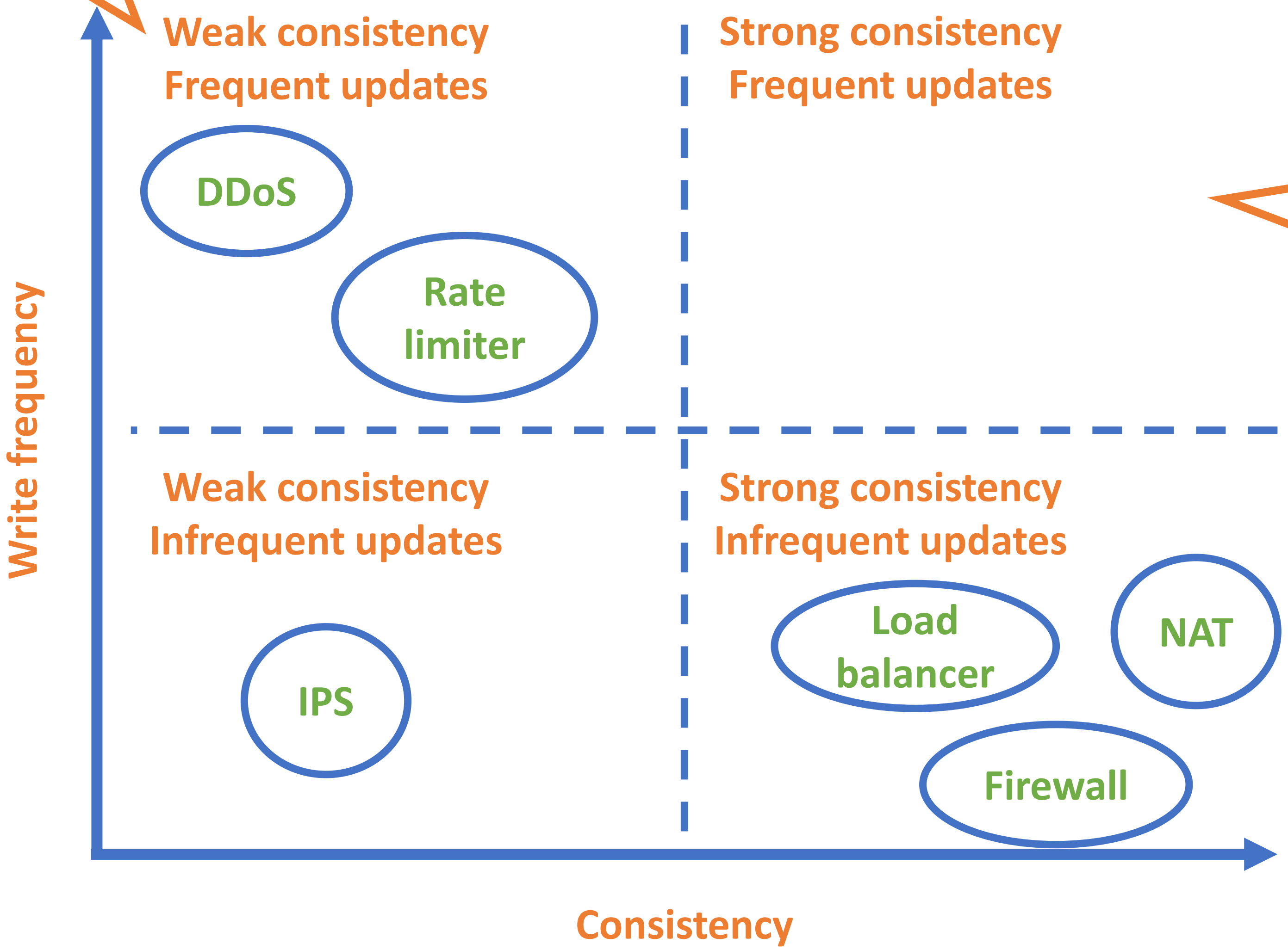
A Principled Approach

Do not reinvent the wheel

Map proven and tested replication protocols

Applications with frequent updates can tolerate weak consistency

State Access in NFs



Weak consistency
Frequent updates

DDoS

Rate limiter

Strong consistency
Frequent updates

Weak consistency
Infrequent updates

IPS

Strong consistency
Infrequent updates

Load balancer

NAT

Firewall

Write frequency

Consistency

This combination is challenging

Strong consistency is often paired with infrequent updates

SwiShmem Registers

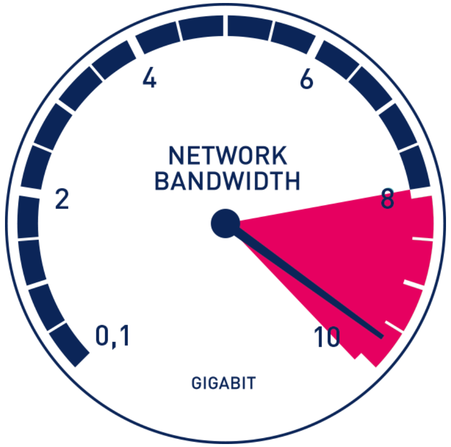
- Eventual Write-Optimized (EWO)
  - Eventual consistency (low read/write latency)

In the paper

- Strong Read-Optimized (SRO)
  - Linearizability
- Eventual Read-Optimized (ERO)
  - Eventual consistency (lower read latency)

Design Principles

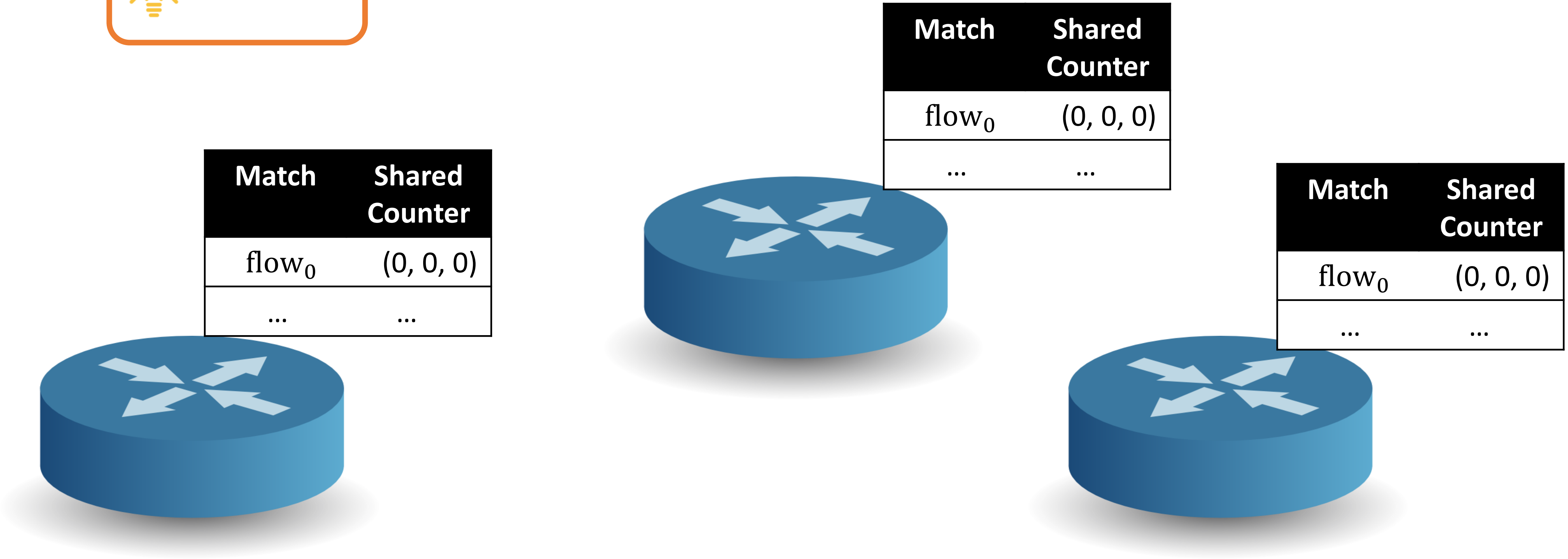
Memory is scarce ($O(10\ MB)$ SRAM)



Communication is cheap ($O(5\ Tbps)$)

Eventual Write-Optimized: HH Detection

CRDTs

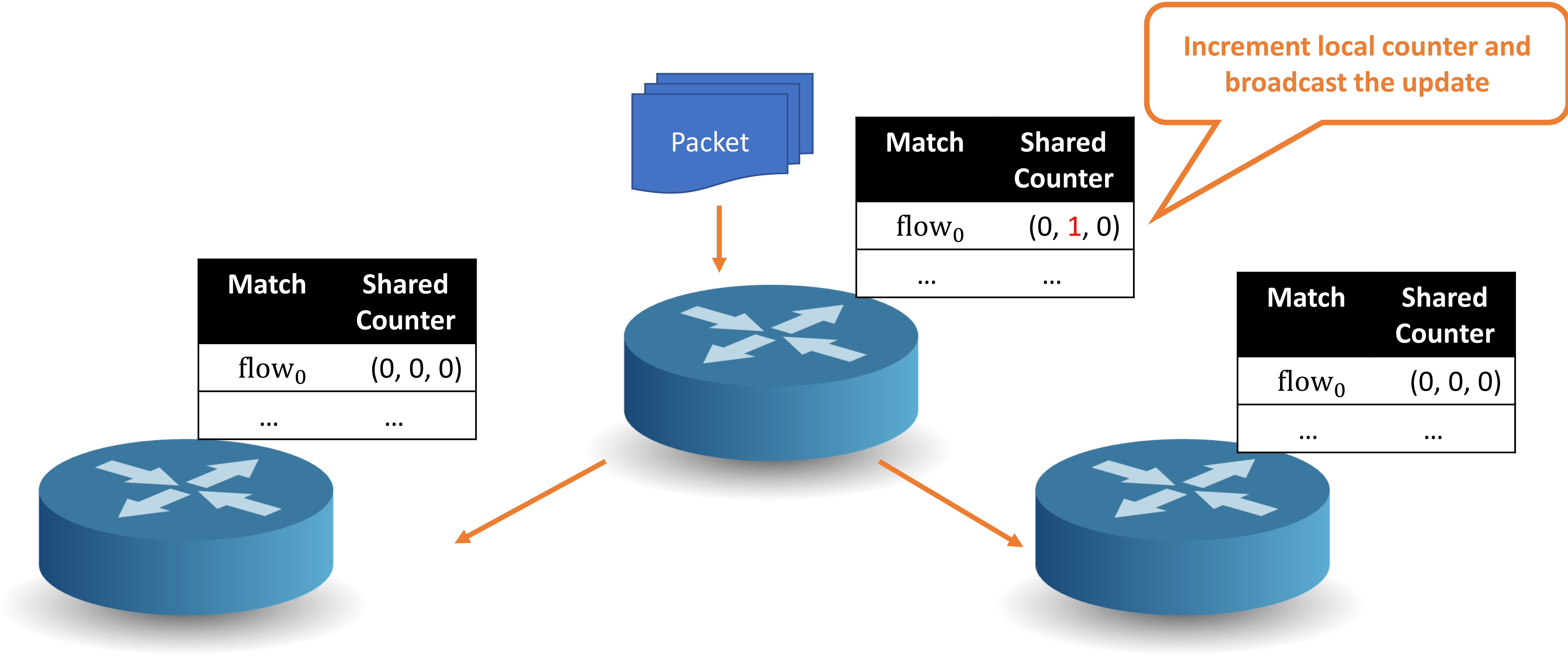


Match	Shared Counter
flow$_0$	(0, 0, 0)
…	…

Match	Shared Counter
flow$_0$	(0, 0, 0)
…	…

Match	Shared Counter
flow$_0$	(0, 0, 0)
…	…

Eventual Write-Optimized: HH Detection



Increment local counter and broadcast the update

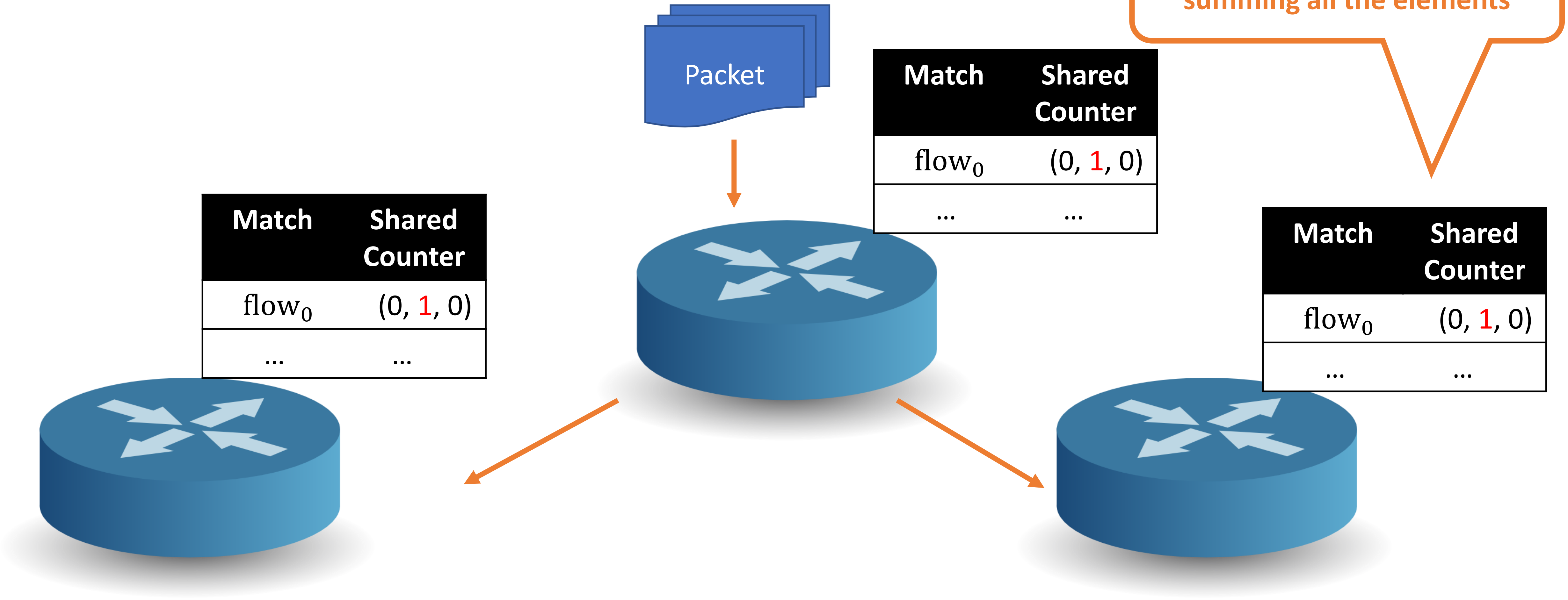
Match	Shared Counter
flow_0	(0, 1, 0)
…	…

Match	Shared Counter
flow_0	(0, 0, 0)
…	…

Match	Shared Counter
flow_0	(0, 0, 0)
…	…

Eventual Write-Optimized: HH Detection

Reads are performed locally by summing all the elements



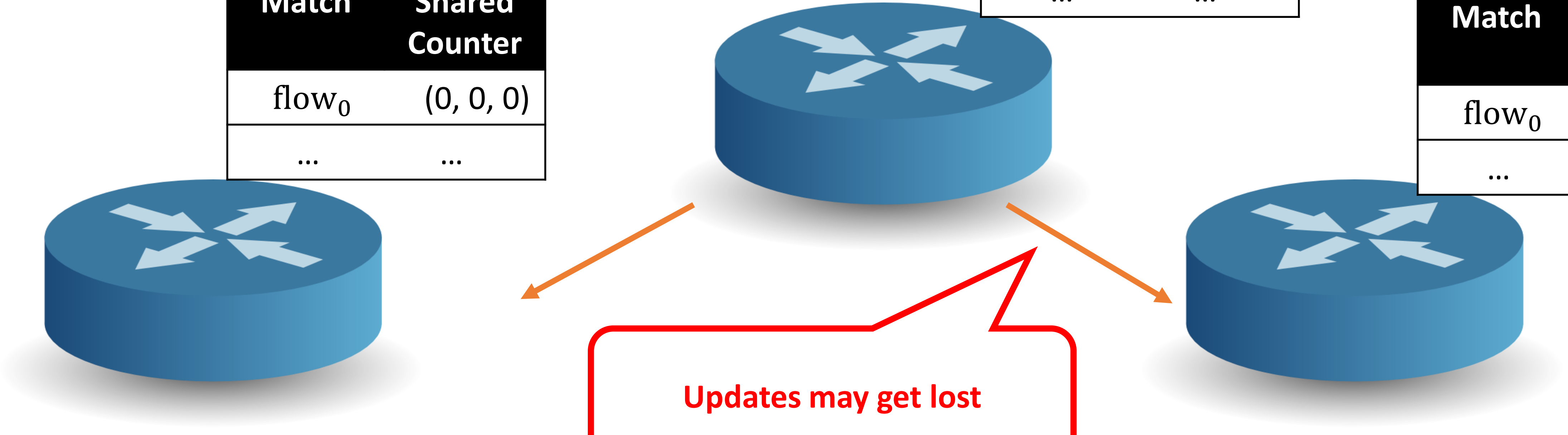
Match	Shared Counter
flow_0	(0, 1, 0)
…	…

Match	Shared Counter
flow_0	(0, 1, 0)
…	…

Match	Shared Counter
flow_0	(0, 1, 0)
…	…

What About Packet Loss?

Periodic synchronization



Match	Shared Counter
flow$_0$	(0, 0, 0)
…	…

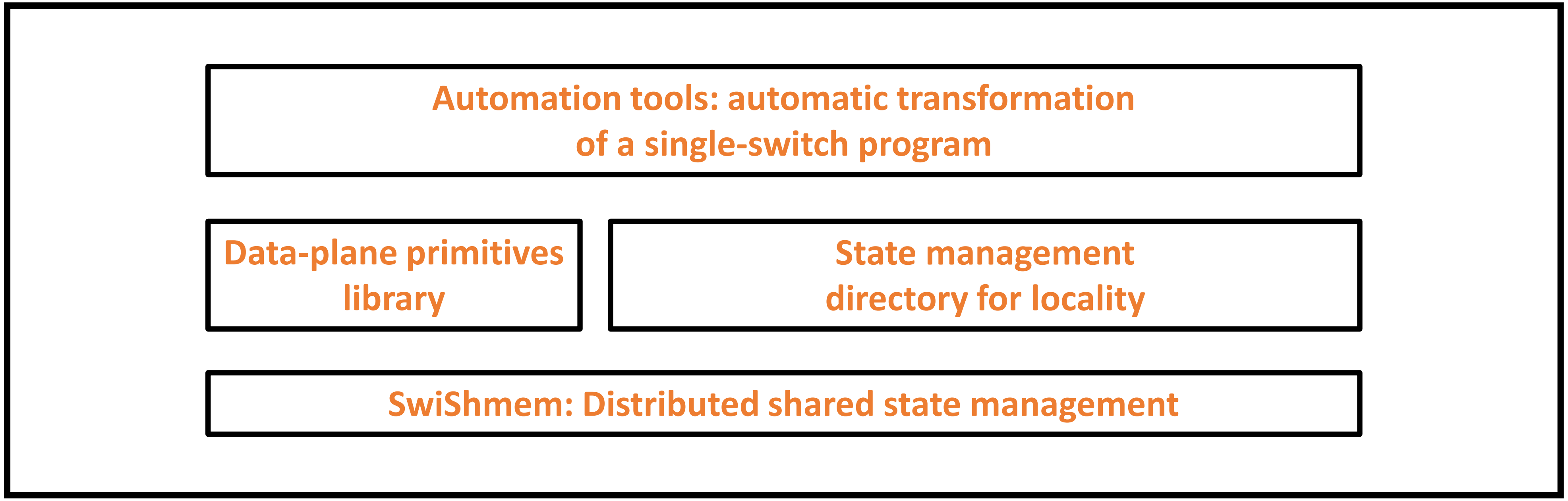
Match	Shared Counter
flow$_0$	(0, 0, 0)
…	…

Match	Shared Counter
flow$_0$	(0, 0, 0)
…	…

Updates may get lost

Vision: “The One Big Switch Abstraction”

The One Big Switch Abstraction



Automation tools: automatic transformation of a single-switch program

Data-plane primitives library

State management directory for locality

SwiShmem: Distributed shared state management

Thank you!
Questions?