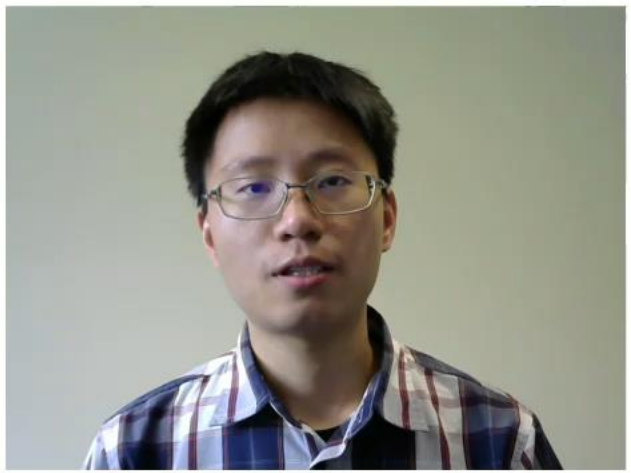
Adaptive Weighted Traffic Splitting in Programmable Data planes

Kuo-Feng Hsu*, Praveen Tammana[+], Ryan Beckett[#],
Ang Chen*, Jennifer Rexford[+], David Walker[+]

Rice University*, Princeton University[+], Microsoft Research[#]

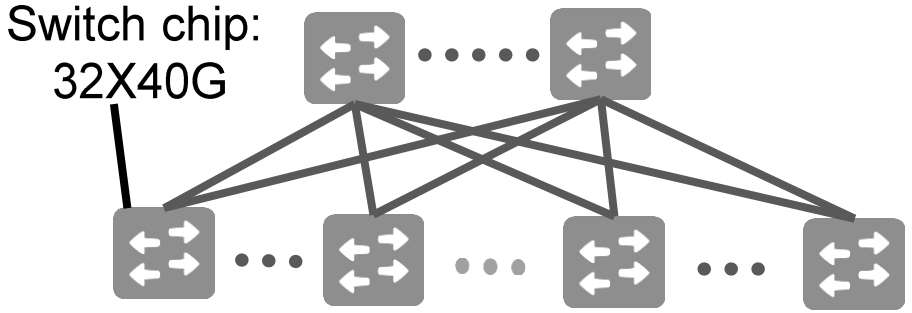


Microsoft®
Research

Multiple paths between source-destination pairs

Datacenter networks

Capacity: 20.4Tbps



16 rack switches*

Private WANs
E.g: B4, SWAN

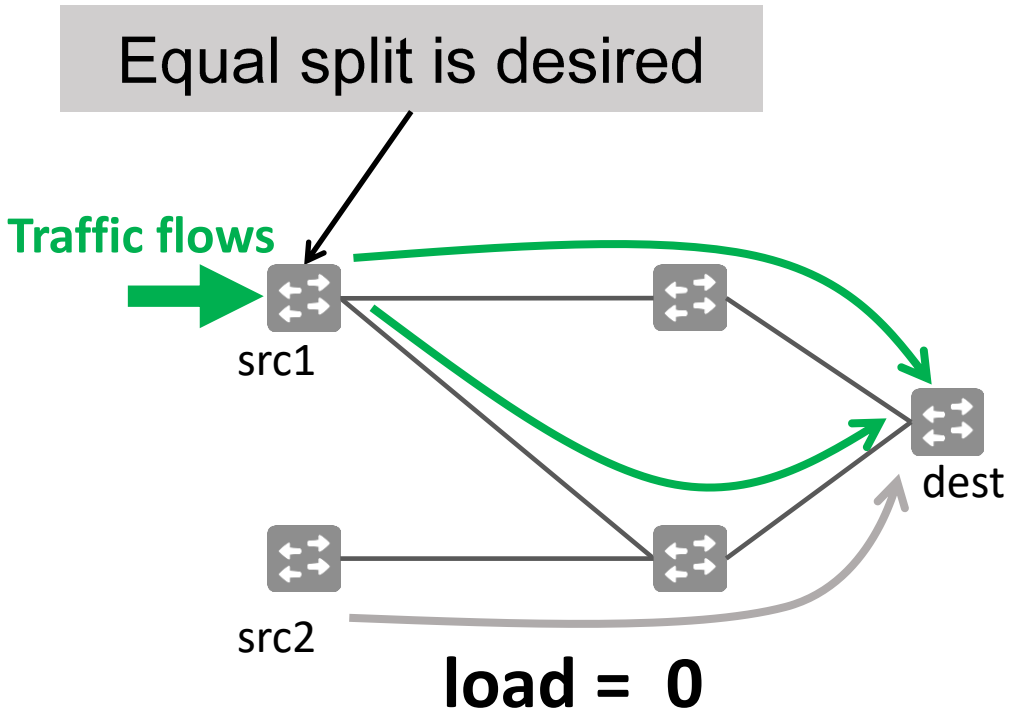
33 datacenter sites* and growing..

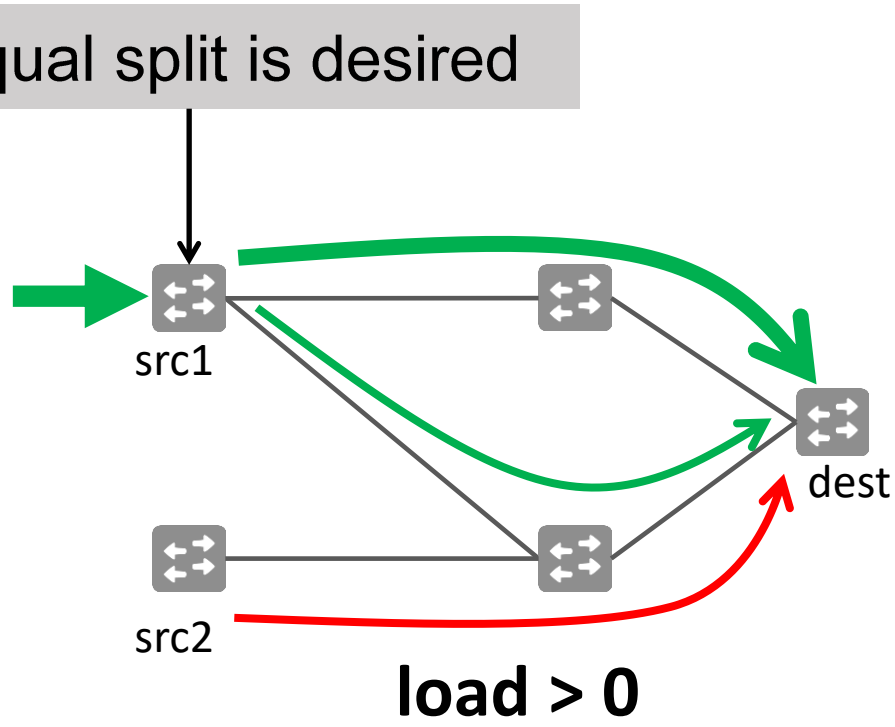
Efficient load balancing is crucial to achieve good performance

*src: B4andAfter [SIGCOMM'18].

Motivation: Load-aware traffic splitting

Fast adaptation to real-time traffic conditions at RTT timescales

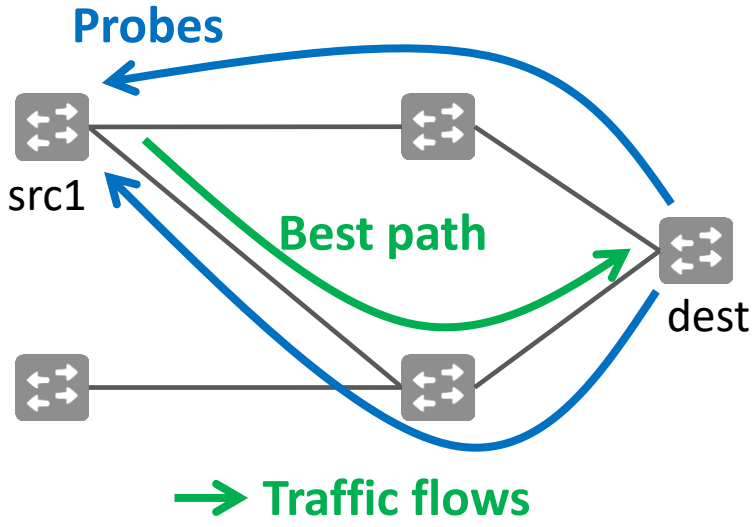




Existing approaches

Operates *entirely* in data plane

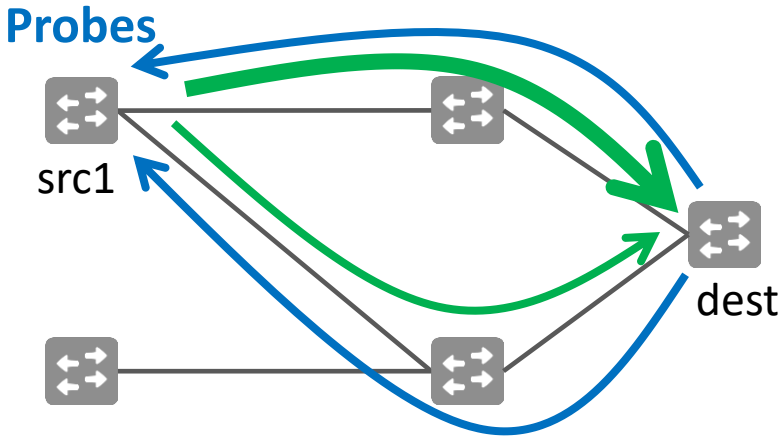
E.g.: HULA [SOSR'16], CONGA [SIGCOMM'14]



Overloads the best path when RTTs are large

Control plane + Data plane

E.g.: HALO [ToN'15], TeXCP [SIGCOMM'05]

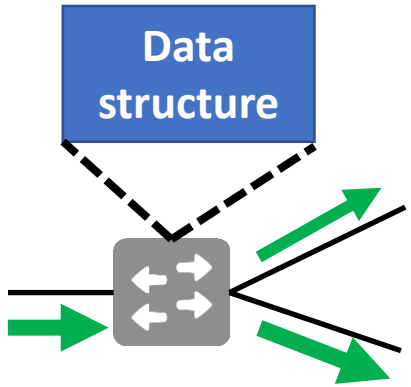


Unclear how these solutions can be realized using commodity data planes

In this work..

Question

How to balance load dynamically ***across multiple paths*** *in the data plane?*

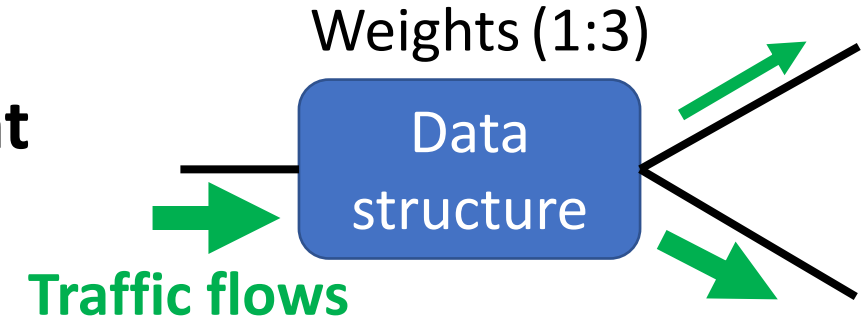


Contributions

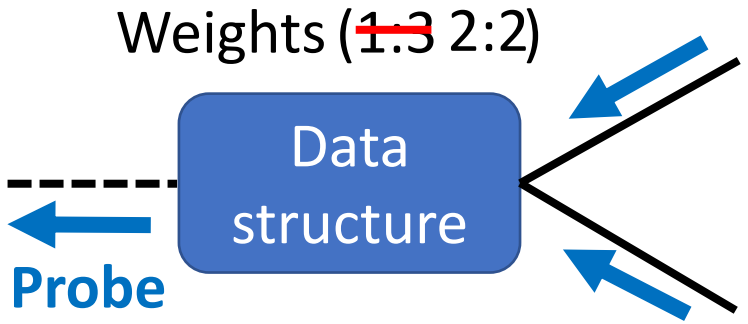
1. We design **new data structures for load-aware traffic splitting**
2. We characterize and study tradeoffs of these data structures
3. We propose a data structure called **DASH**

Key problems

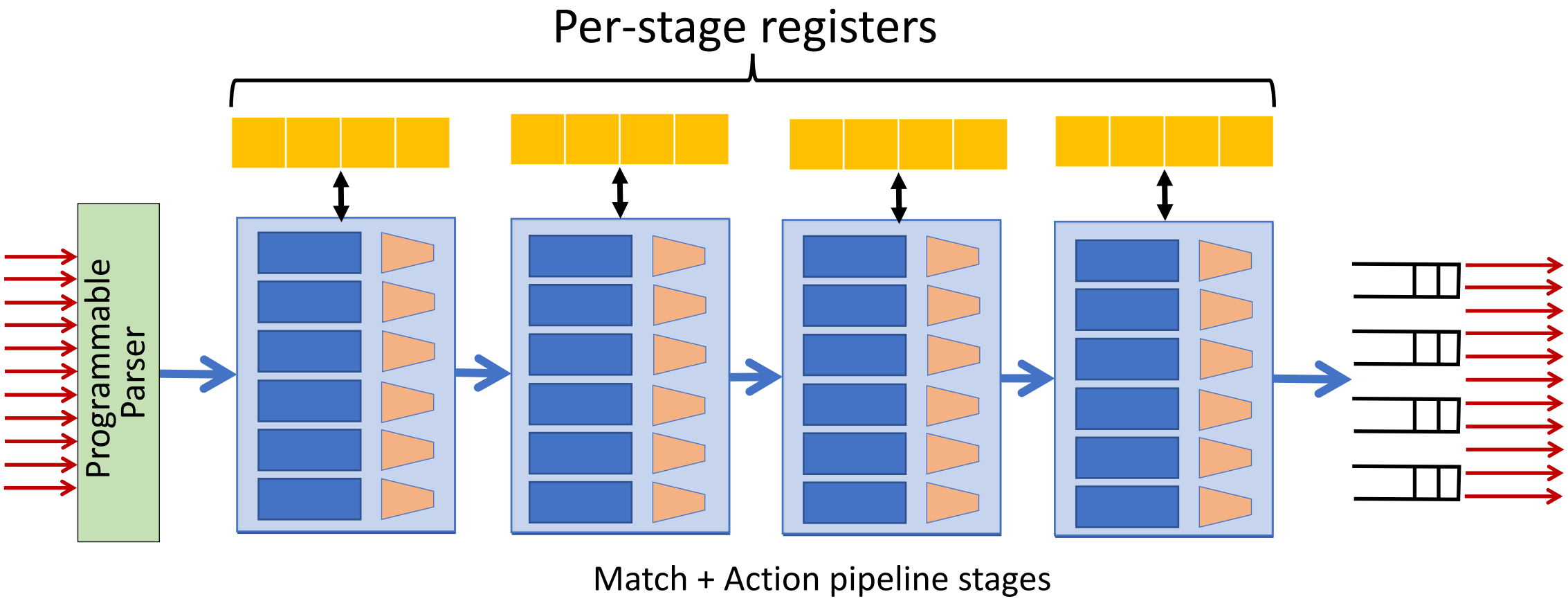
(1) Spreading flows using a **Path-to-Weight** data structure



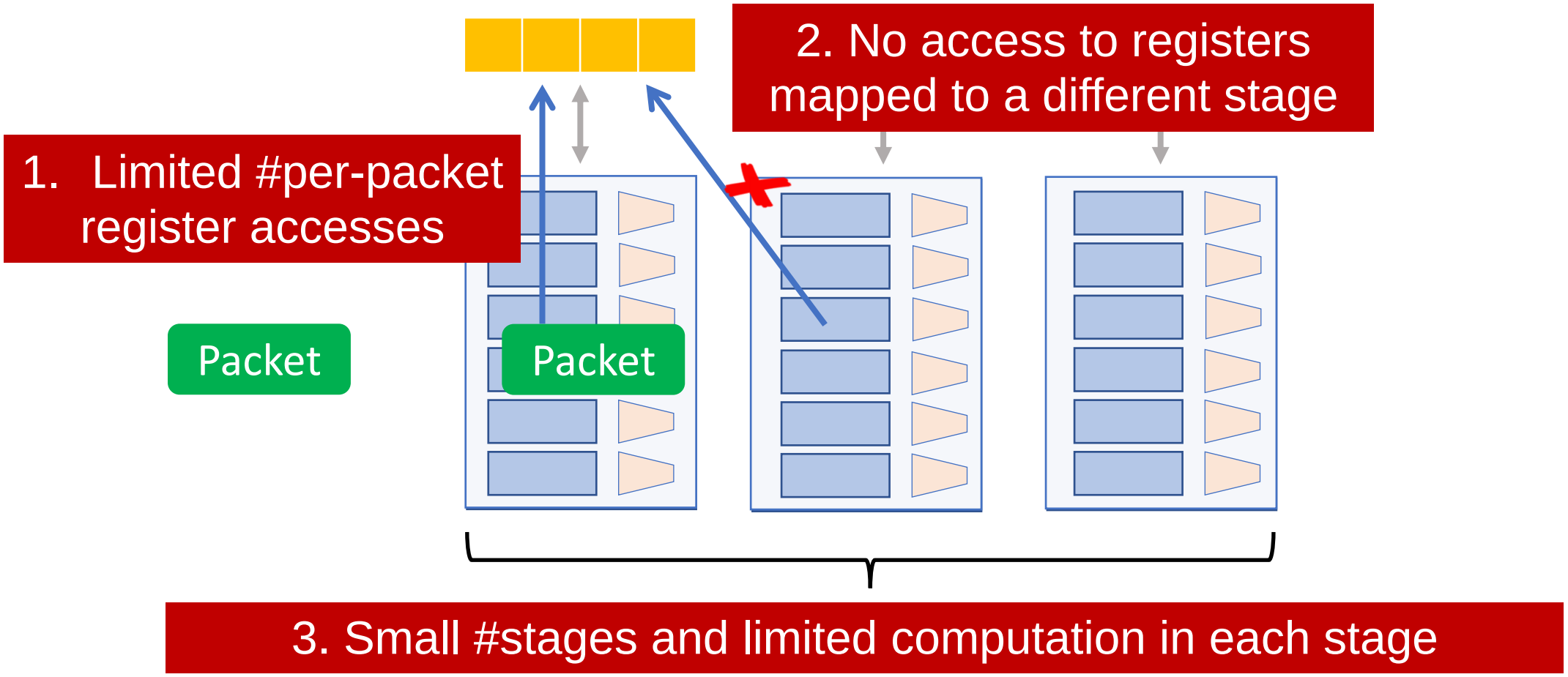
(2) Updating that data structure as probes arrive



PISA: Protocol Independent Switch Architecture

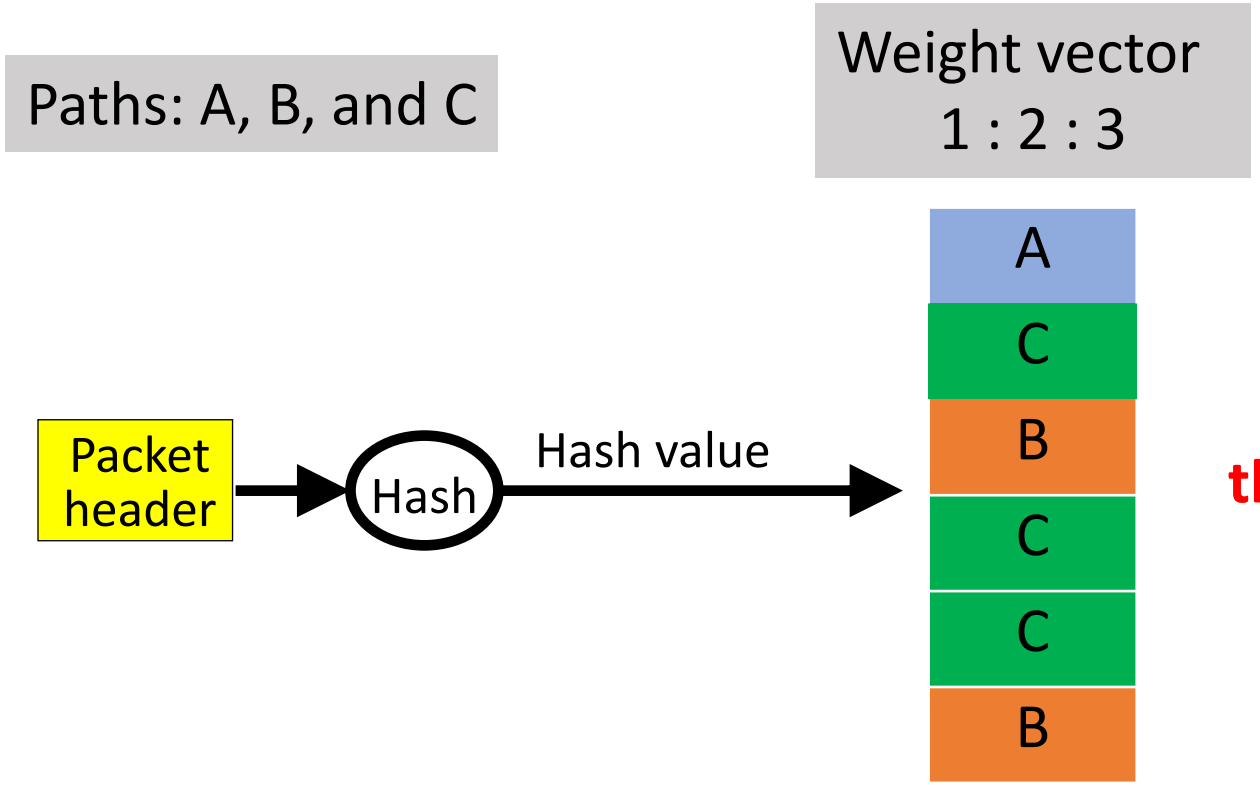


But, a constrained computational model



Existing technique: Weighted Cost Multipathing (WCMP)

Replicates table entries with same pathID in proportion to its weight

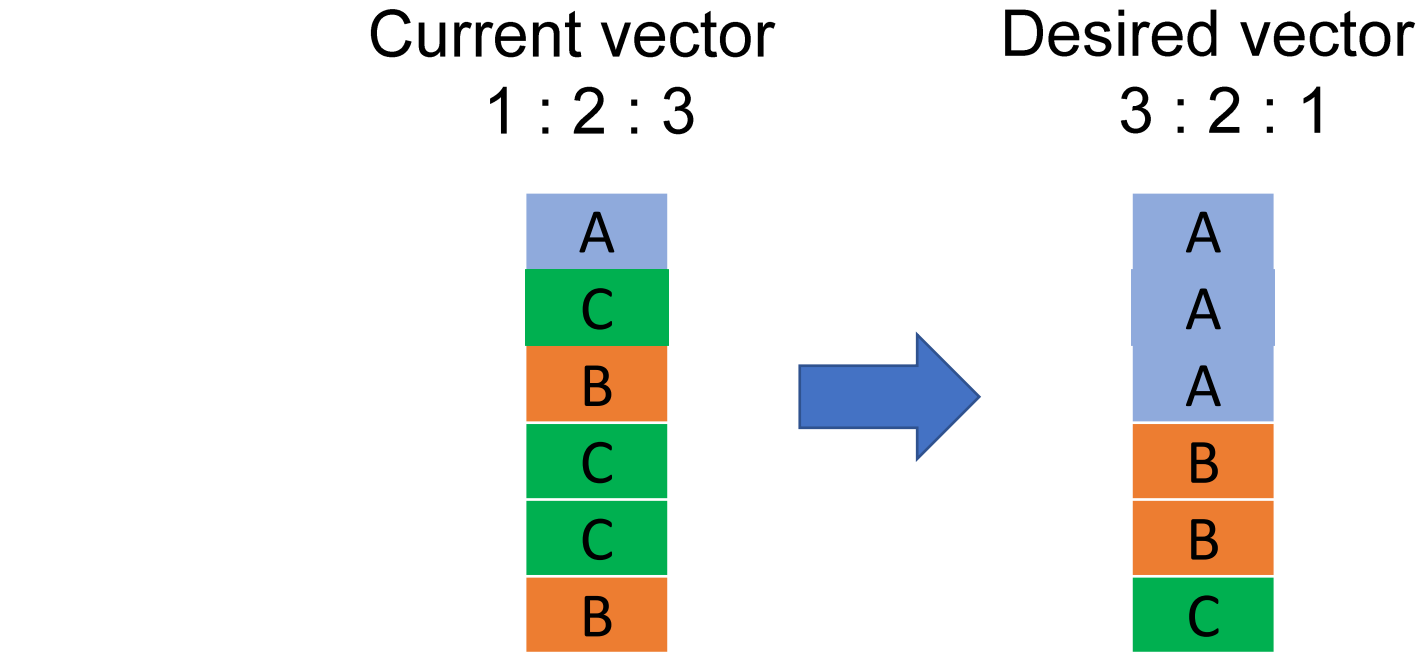


How can we update the table as probes arrive?

WCMP table is stored in a stage registers

Updating WCMP table

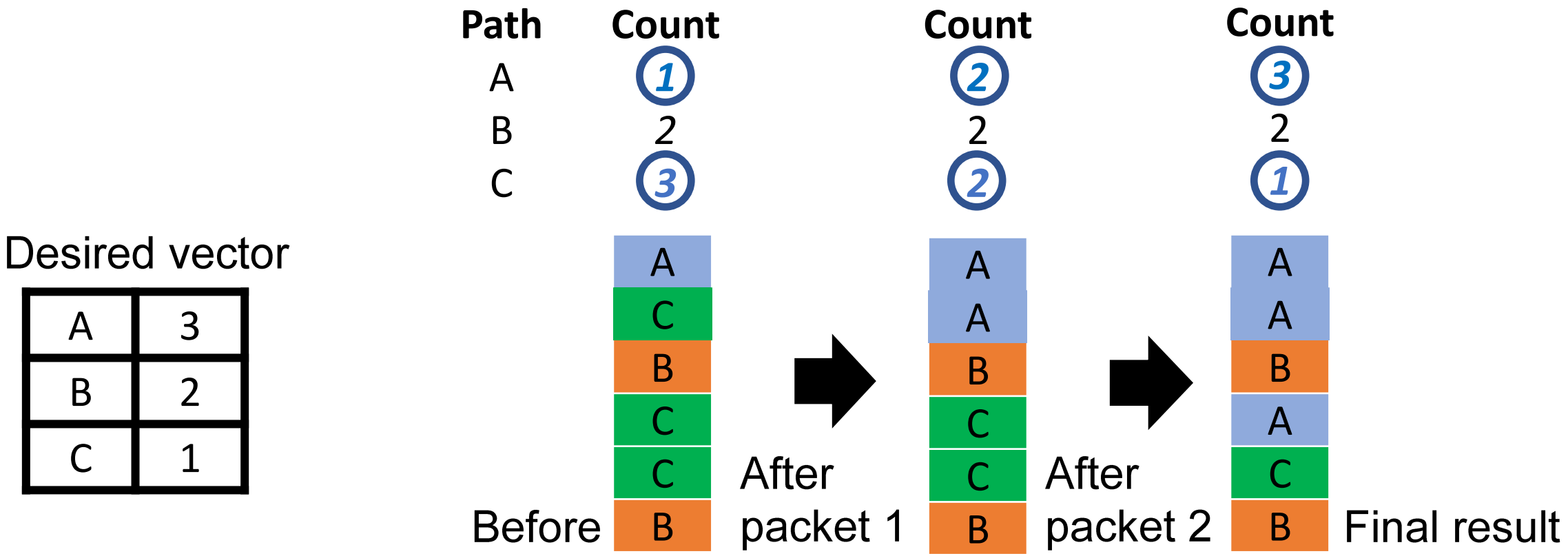
Iterate over the table



✗ Requires many per-packet accesses to a stage registers

Updating WCMP table

Assign entries of non-deficit paths to deficit paths

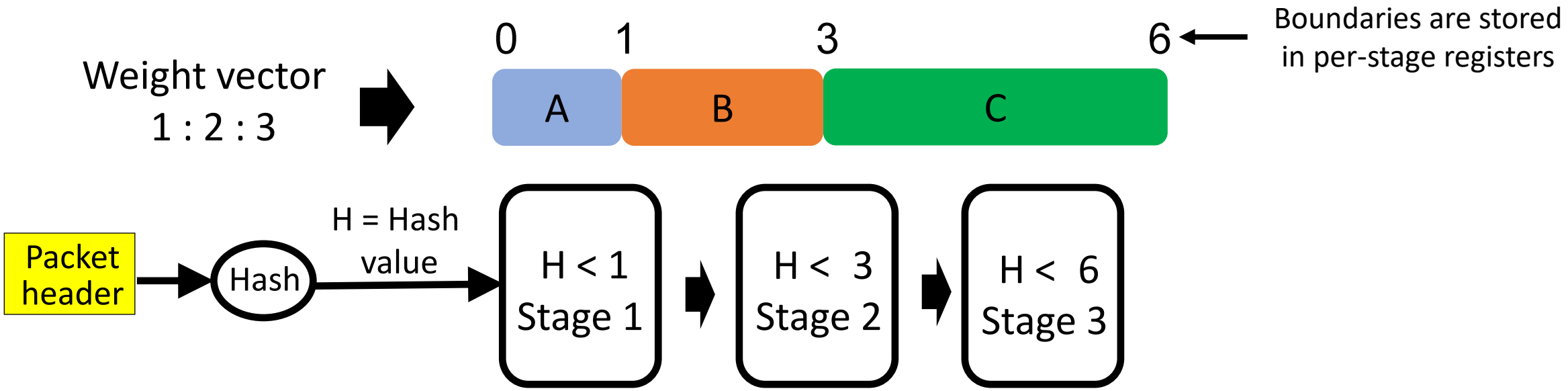


✗ Requires read and write to same register from different stages

DASH: Data-plane Adaptive Splitting with Hash threshold

~~Replicate pathID in WCMP table~~

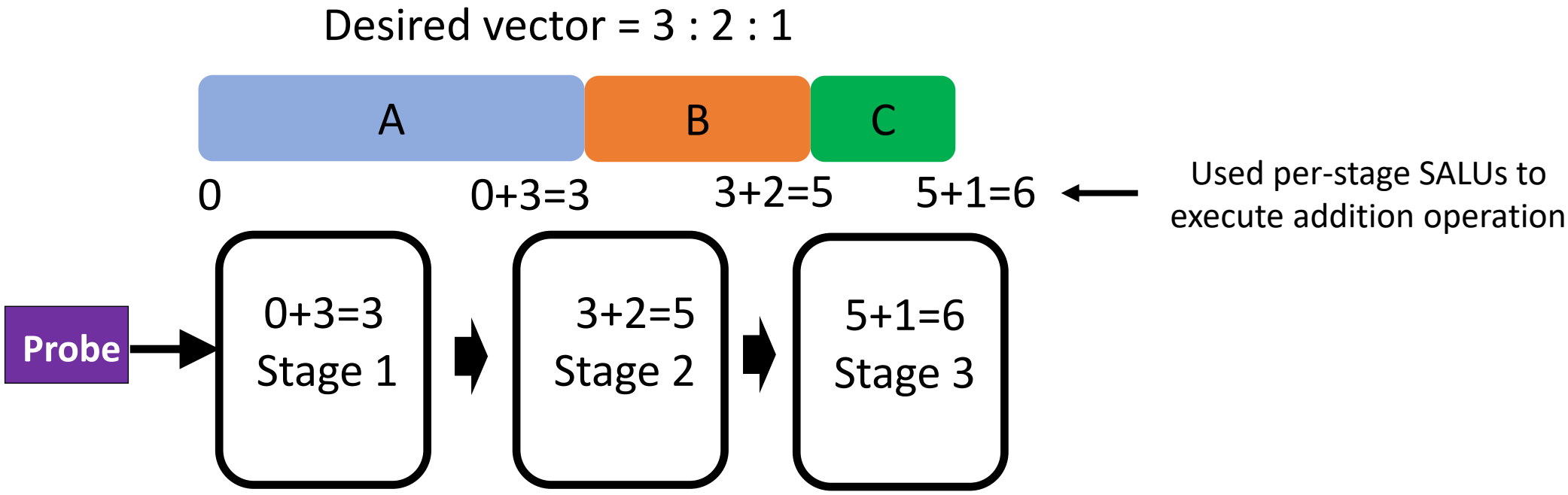
Idea: Partition hash space into unique regions of size proportion to path weights



Check packet's hash value against boundaries

Updating DASH boundaries is simple

New path boundary = Previous path's boundary + Path region size

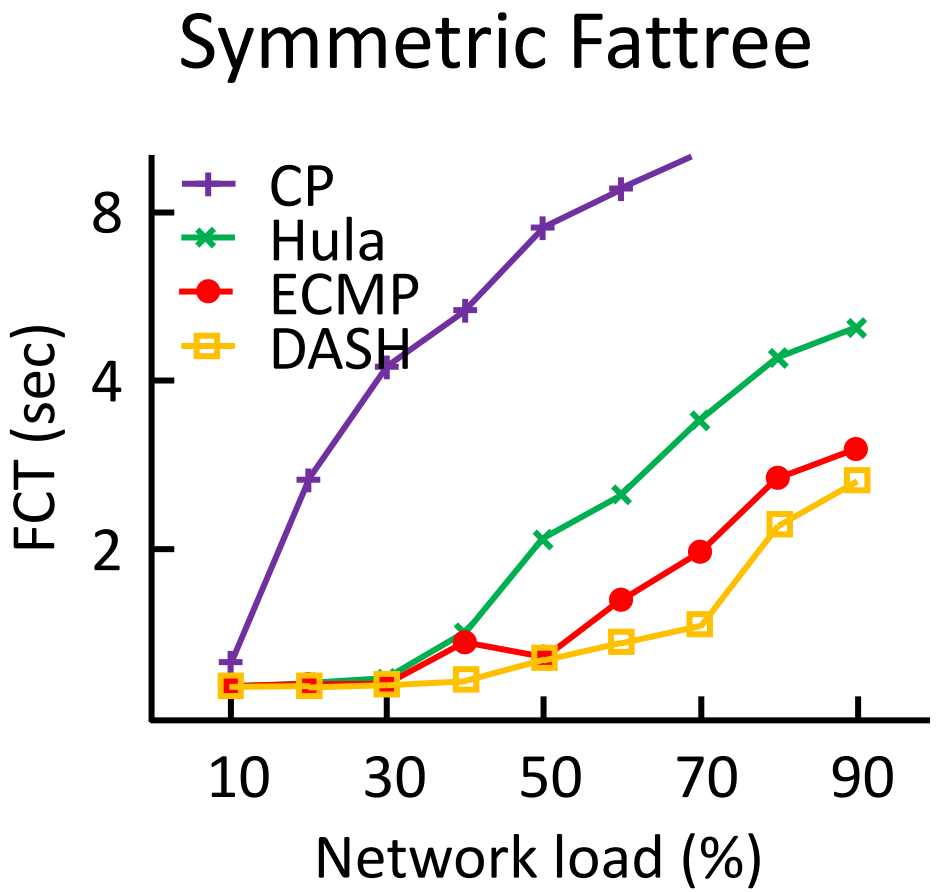


✓ One per-packet access to stage registers
✓ Read-write in the same stage

Fast and efficient

Evaluation

- Prototype: Bmv2
- Environment: Ns3
- Workload:
  - Web search [Alizadeh-SIGCOMM'15]
- Schemes
  - CP (Control plane): slow adaptation
  - Hula: fast but single best path
  - ECMP: traffic splitting but no adaptation
  - DASH: traffic splitting and fast adaptation



Summary and future work

- Adaptive weighted traffic splitting in the data plane
- **DASH** is fast and efficient
- Future work:
  - Implementation in commodity switches (e.g., P4-Tofino)
  - Building a distributed online traffic engineering system

Thank you