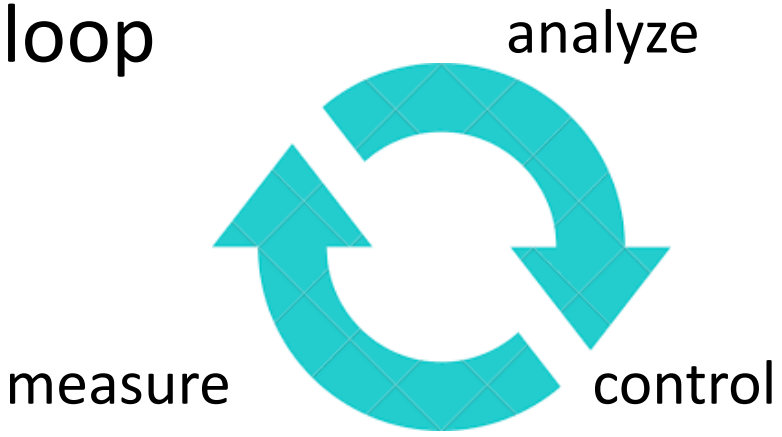
Toward Self-Driving Networks

Jennifer Rexford

PRINCETON UNIVERSITY

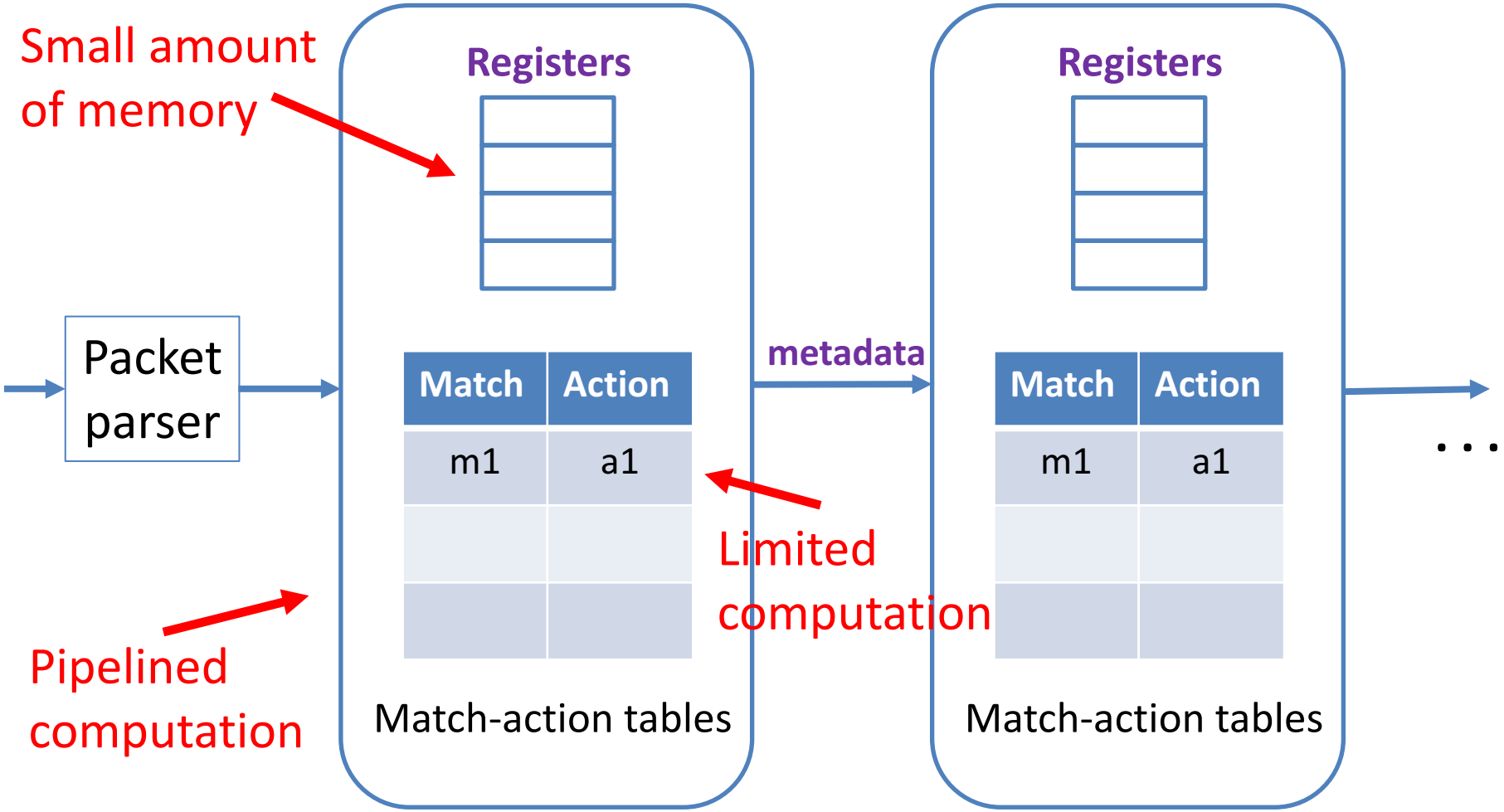
Self-Driving Network

- Complete control loop
  - Measure
  - Analyze
  - Control



- Examples
  - Direct traffic over the best performing path
  - Block or slow the heavy-hitter flows
- Possible now in the data plane!

A Constrained Computational Model



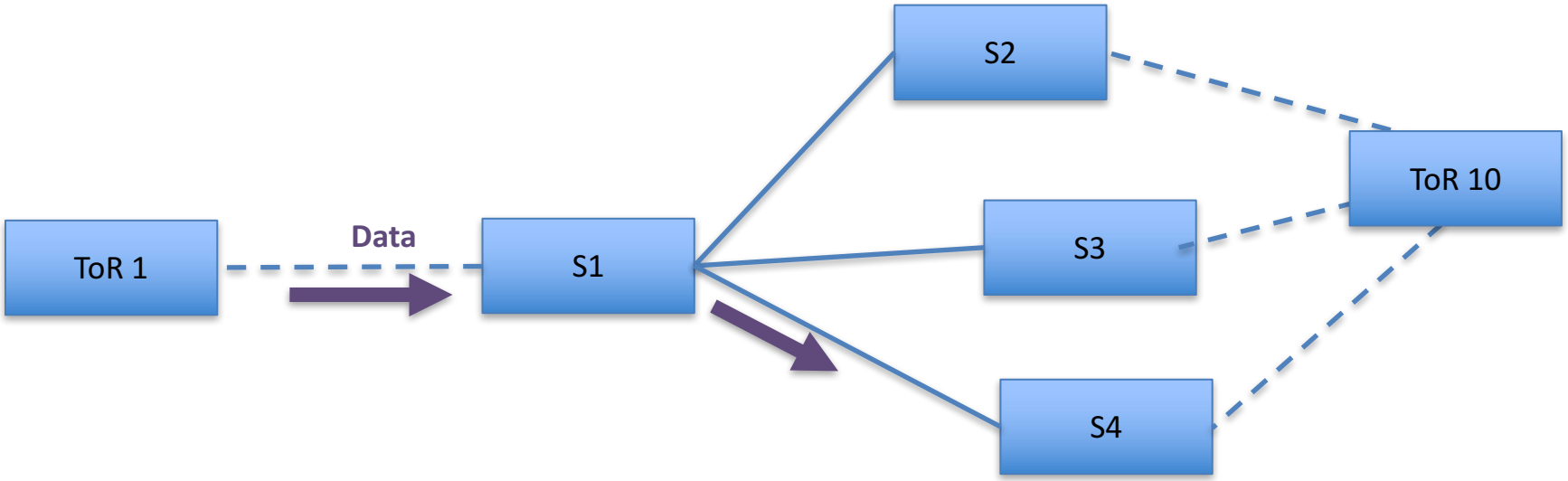
HULA

Hop-by-Hop Utilization-aware Load-balancing Architecture

Naga Katta, Mukesh Hira, Changhoon Kim,
Anirudh Sivaraman, and Jennifer Rexford

http://conferences.sigcomm.org/sosr/2016/papers/sosr_paper67.pdf

HULA Multipath Load Balancing

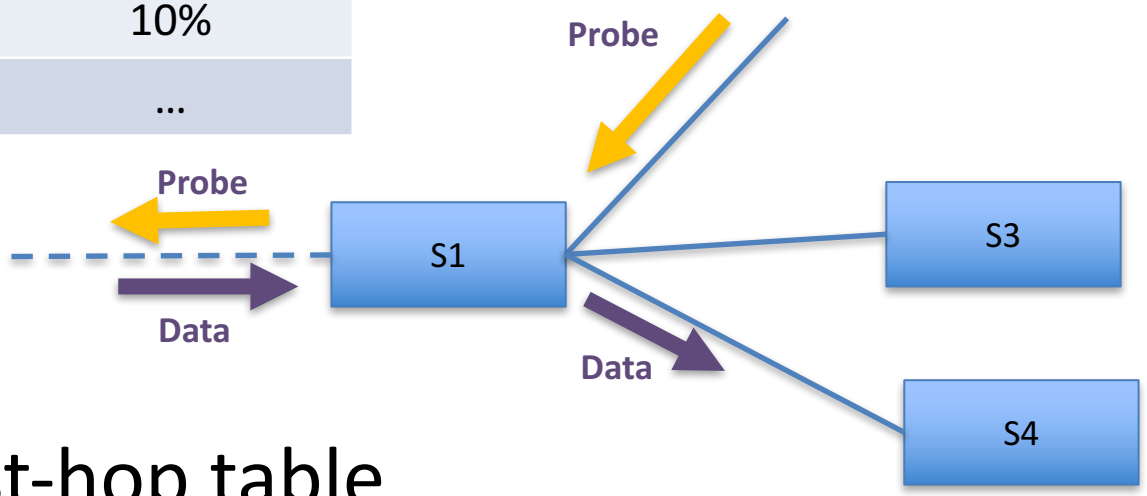


- Load balancing *entirely* in the data plane
  - Collect real-time, path-level performance statistics
  - Group packets into “flowlets” based on time & headers
  - Direct each new flowlet over the current best path

Path Performance Statistics

Best-hop table

Dest ToR	Best Next-Hop	Path Utilization
0	S3	50%
1	S4	10%
…	…	…



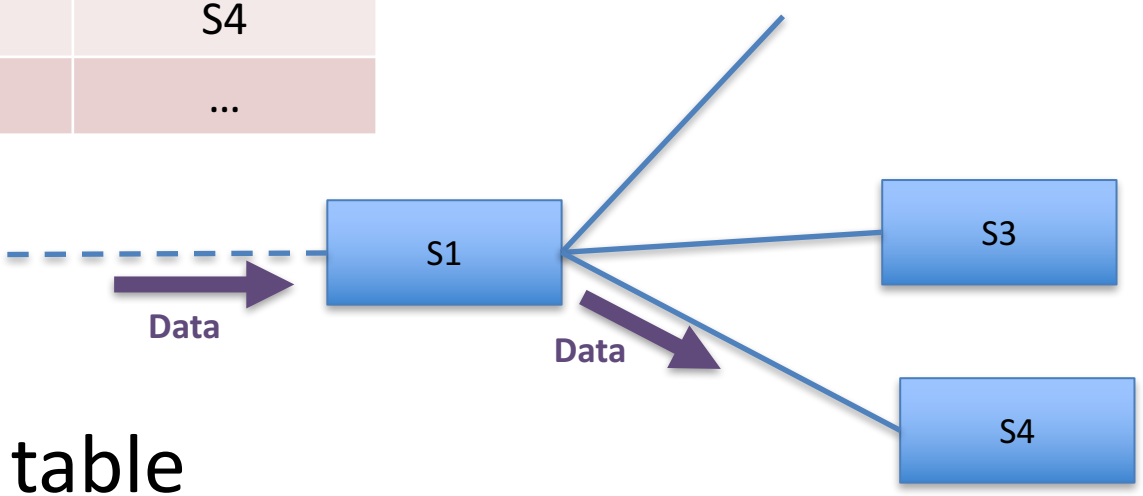
- Using the best-hop table
  - *Update* the best next-hop upon new probes
  - *Assign* a new flowlet to the best next-hop

Flowlet Routing

Flowlet table

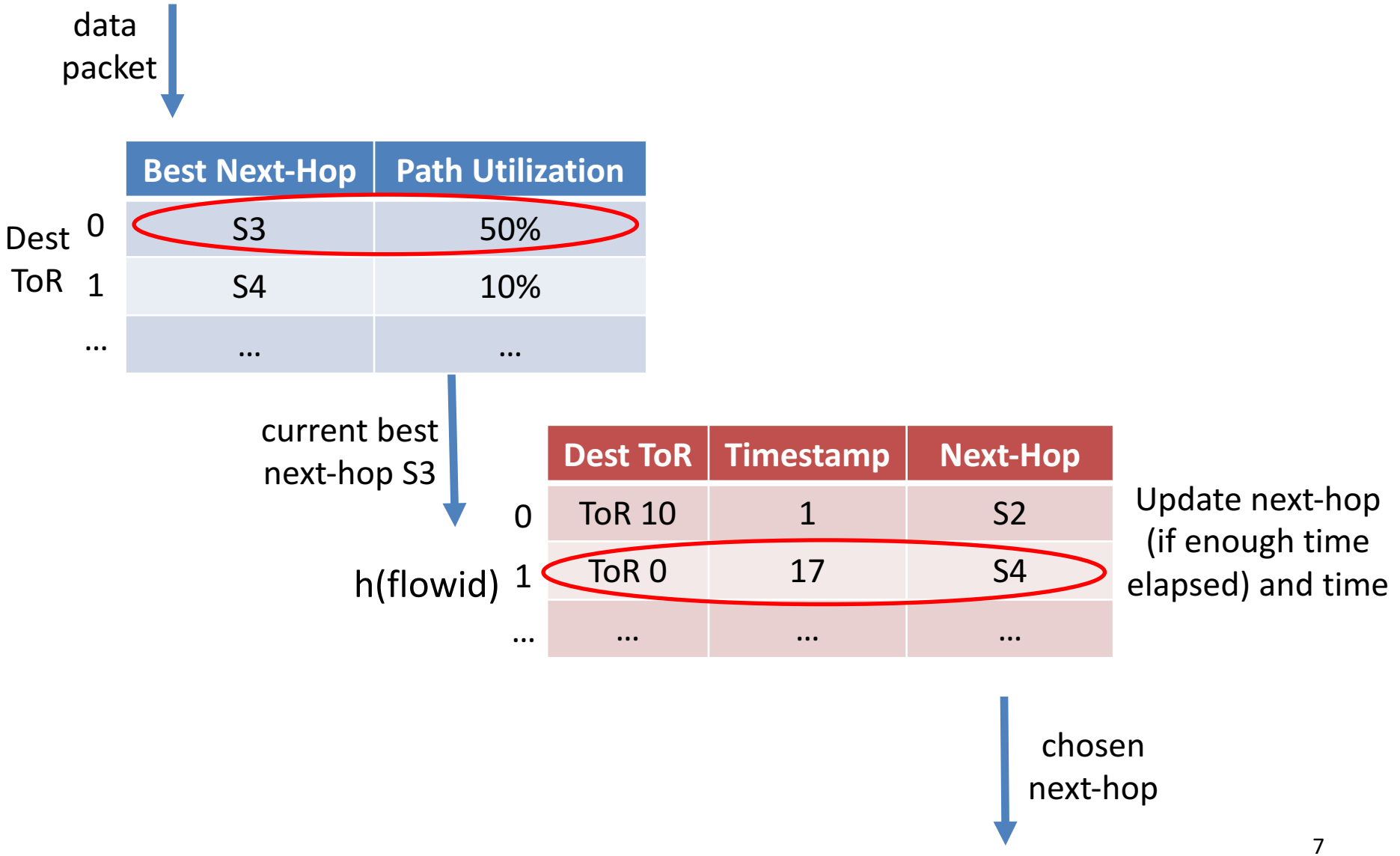
h(flowid)

	Dest ToR	Timestamp	Next-Hop
0	ToR 10	1	S2
1	ToR 0	17	S4
…	…	…	…



- Using the flowlet table
  - *Update* the next hop if enough time has elapsed
  - *Update* the timestamp to the current time
- *Forward* the packet to the chosen next hop

Putting it all Together



HashPipe

Heavy Hitter Detection Entirely in the Data Plane

Vibhaalakshmi Sivaraman, Srinivas Narayana, Ori Rottenstreich, S. Muthukrishnan, and Jennifer Rexford

https://conferences.sigcomm.org/sosr/2017/papers/sosr17-heavy-hitter.pdf

Heavy-Hitter Detection

- Heavy hitters
  - The *k* largest trafic flows
  - Flows exceeding threshold *T*
- Space-saving algorithm
  - Table of (key, value) pairs
  - Evict the key with the minimum value

New Key K7

Id	Count
K1	4
K2	2
K3	7
K4	10
K5	1
K6	5

Table scan

Approximating the Approximation

- Evict minimum of *d* entries
  - Rather than minimum of all entries
  - E.g., with *d = 2* hash functions

New Key K7

Id	Count
K1	4
K2	2
K3	7
K4	10
K5	1
K6	5

Multiple memory accesses

Approximating the Approximation

- Divide the table over *d* stages
  - One memory access per stage
  - Two different hash functions

New Key K7

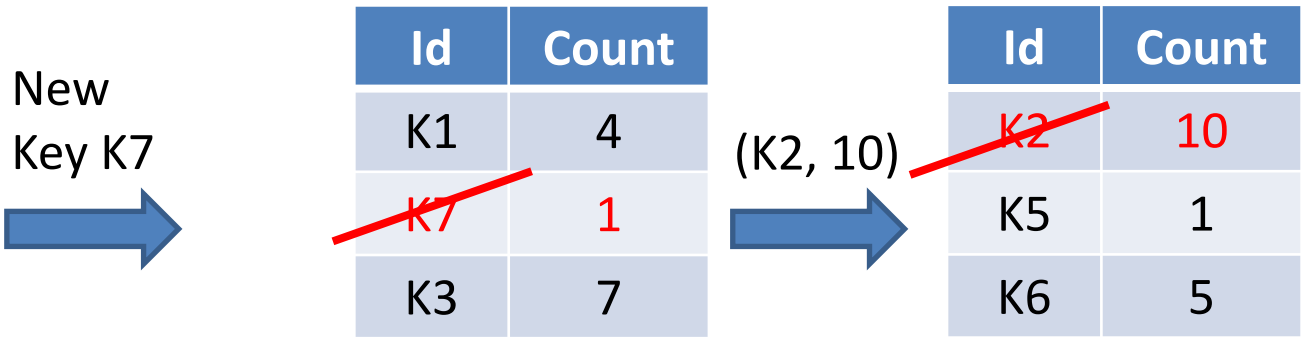
Id	Count
K1	4
K2	2
K3	7

Id	Count
K4	10
K5	1
K6	5

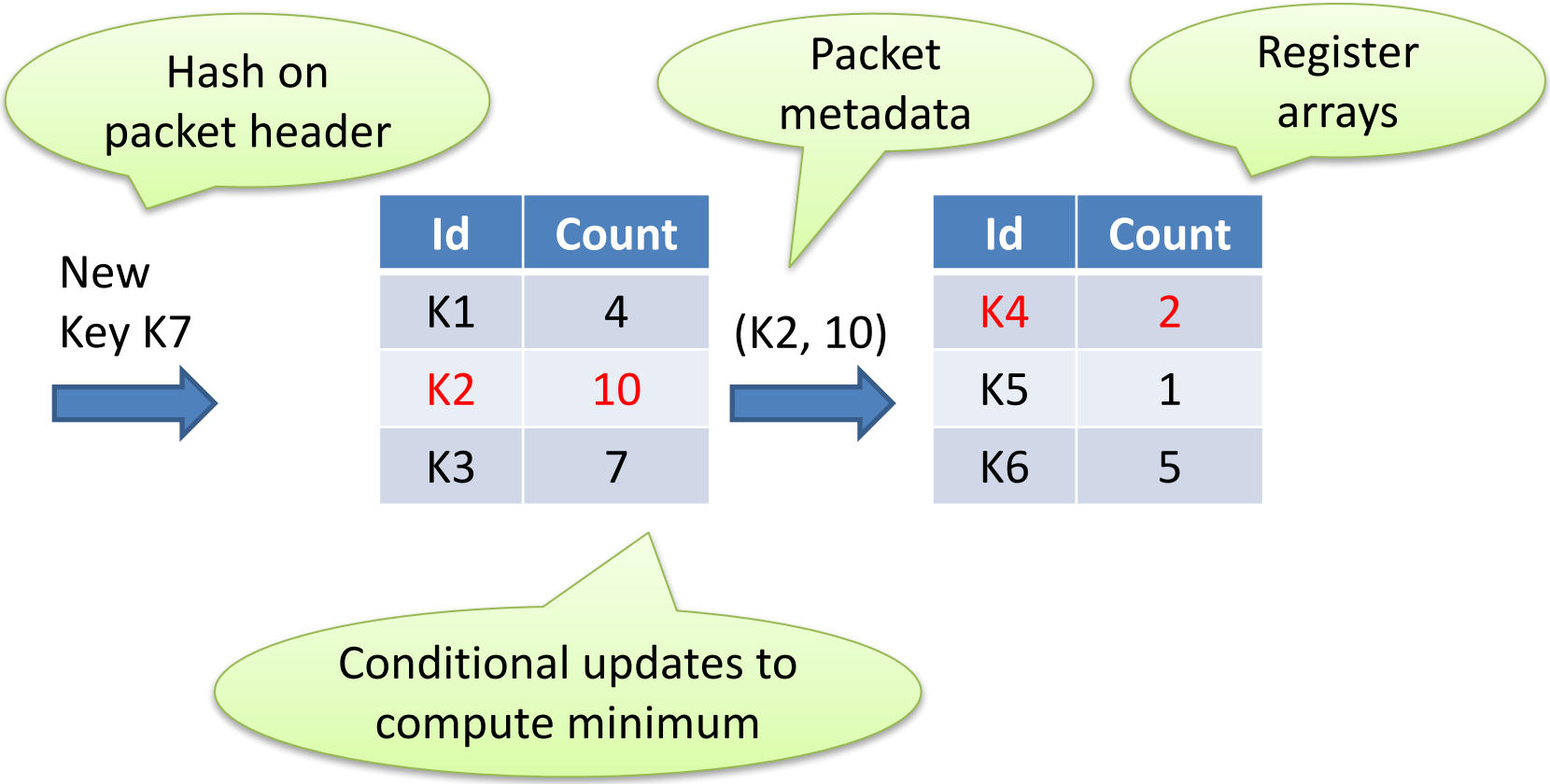
Going back to the first table

Approximating the Approximation

- Rolling min across stages
  - Avoid recirculating the packet
  - … by carrying the minimum along the pipeline



P4 Prototype and Evaluation



High accuracy with overhead proportional to # of heavy hitters

Conclusion

- Self-driving networks
  - Integrate measure, analyze, and control
  - Distribute across the network devices
- Enabled by programmable switches
  - Parsing, processing, and state
- Approximate data structures
  - Limited memory for storing state
  - Limited processing per packet