New Measurement Abstractions for Data Centers

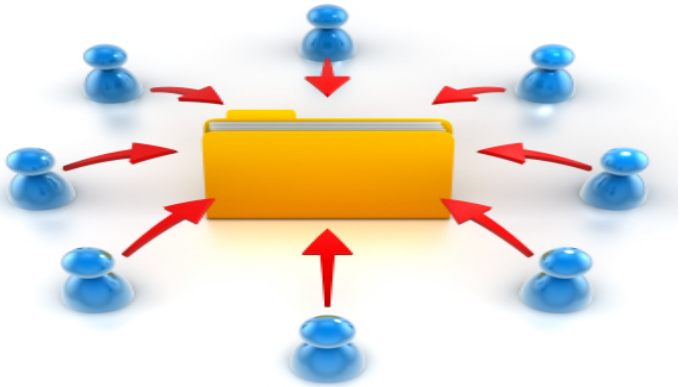
Minlan Yu

University of Southern California

Key Challenges in Data Centers

- Service level agreements
  - Good, predictable performance

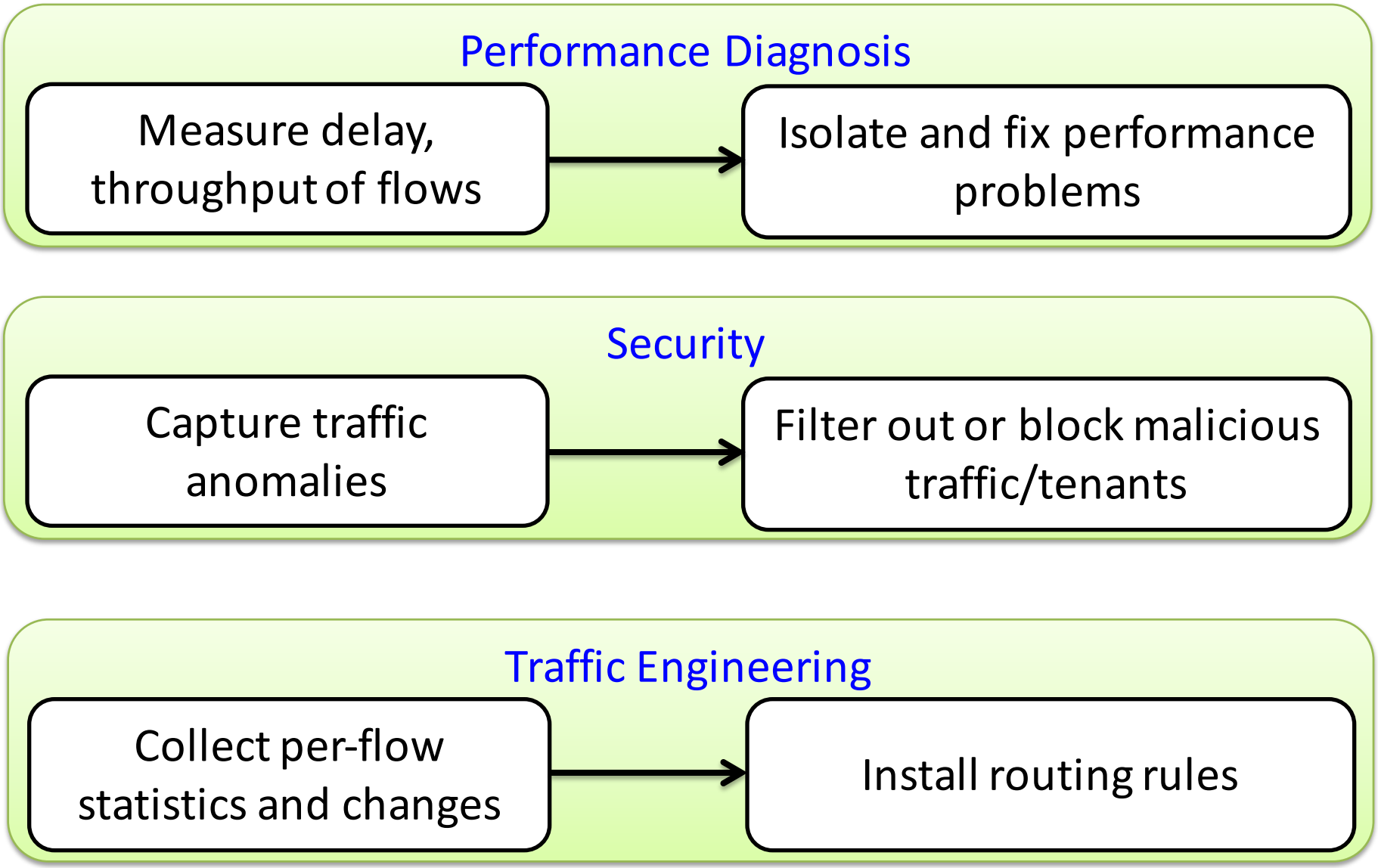
- Multi-tenancy
  - Resource isolation across tenants



- Cost reduction
  - Drive the network to high utilization

All require fine-grained network management

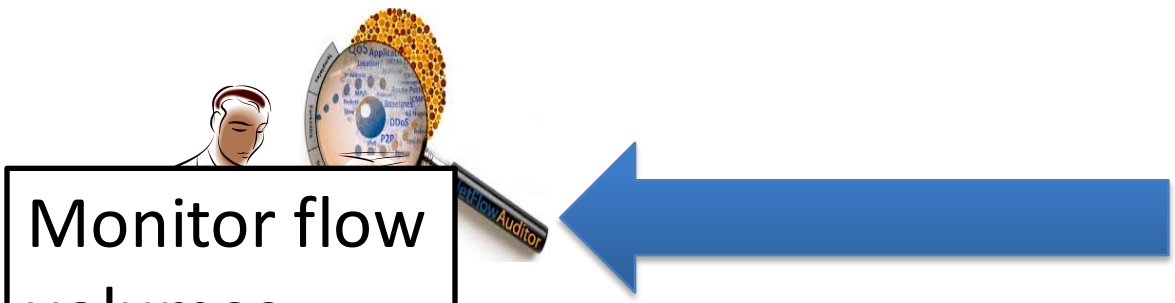
Management = *Measurement* + Control

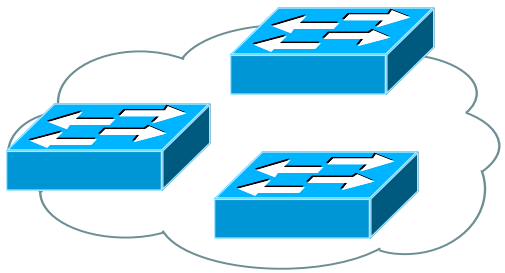


The state of today's Datacenter measurement

The need for new measurement abstractions

Many Measurement Programs





Monitor flow volumes

Performance diagnosis

Anomaly detection

Query independence:
Specify many measurement queries
with simple, programmable abstractions

Many Data Collection Primitives

- Passively collect different sources of data

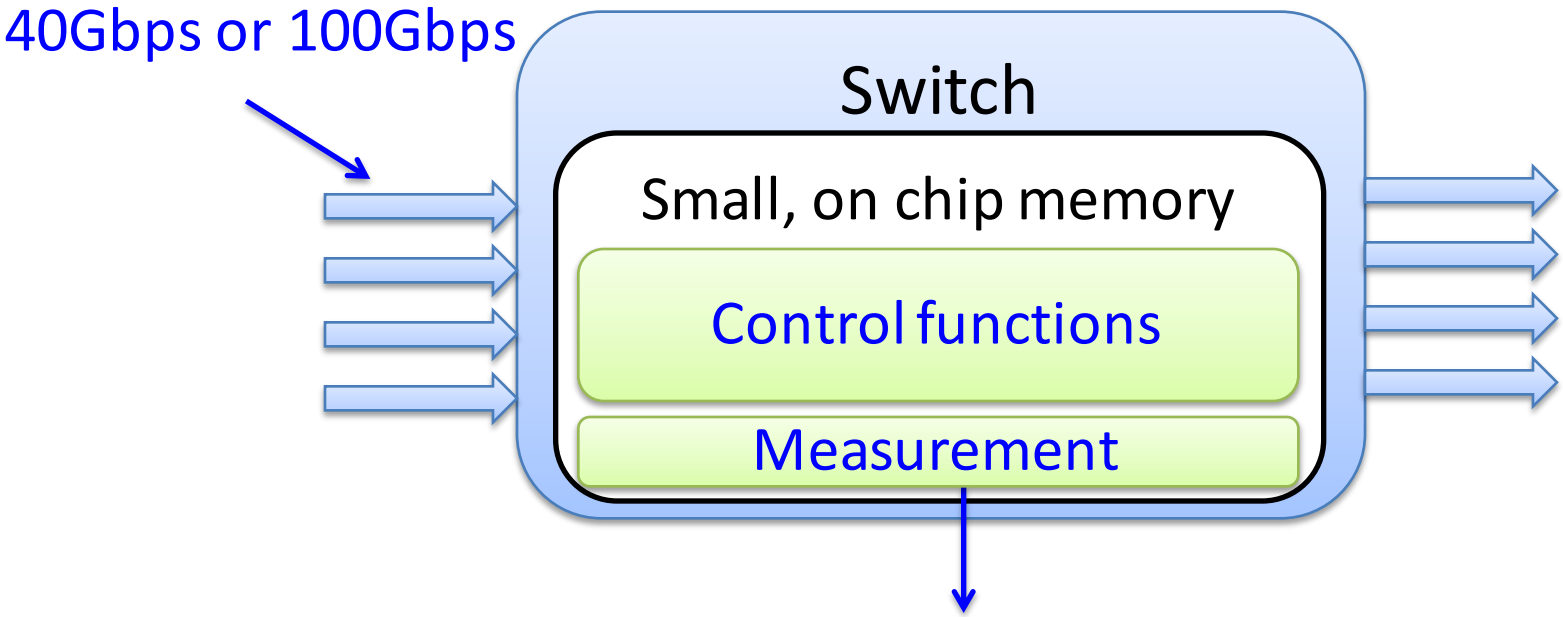
→Little control on what (not) to measure

Target independence:

Specify which data to collect

independent of where and how it is collected

Limited Measurement Resources



- Low sampling rates
- Miss many important information

Reconfigurable:
Dynamically change the way resources are used for different measurement queries and traffic

A New Abstraction for Measurement

Specify measurement queries

Measurement Framework

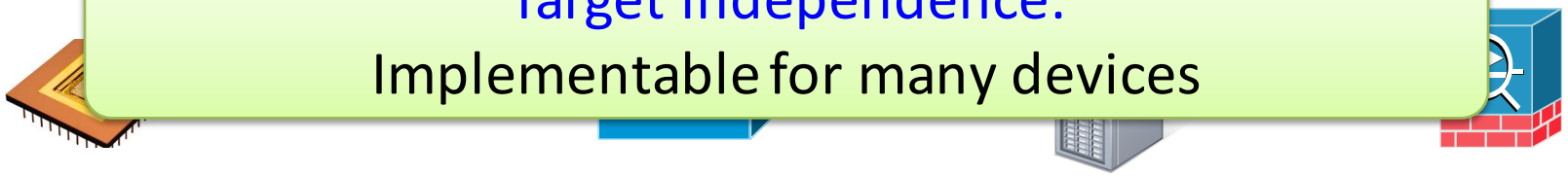
Query independence:
Expressive Abstractions for many queries

Reconfigurable: Efficient Runtime

Dynamically configure devices & manage resources

Automatically collect the right data

Target independence:
Implementable for many devices



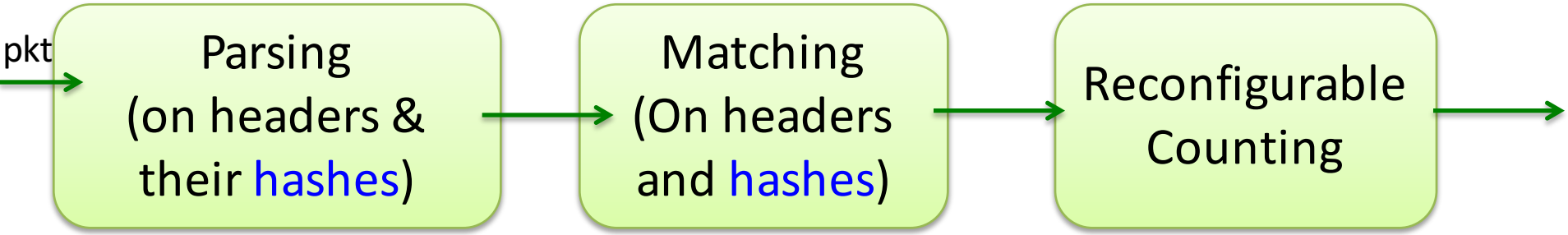
Expressive Abstractions (OpenSketch)

Picking the packets to measure

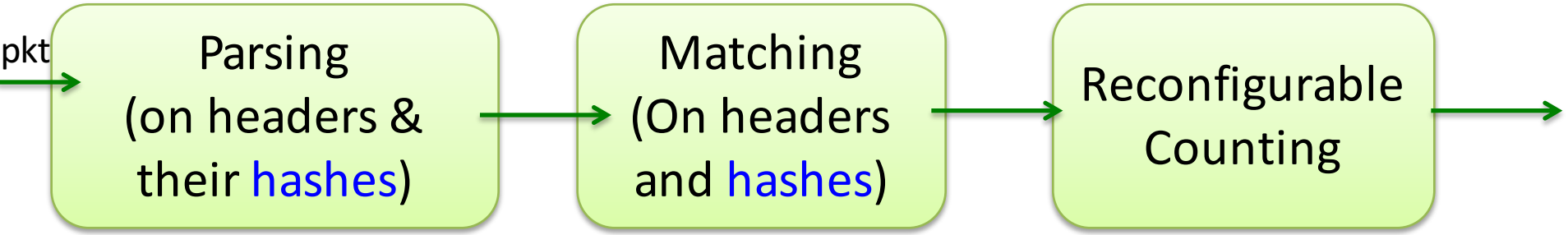
Storing & exporting data

Filter traffic aggregates
(e.g., to host A)
Classify a set of flows
(e.g., a set of malicious source IPs)

Dynamically flows to counters
(e.g., more counters for elephant flows)

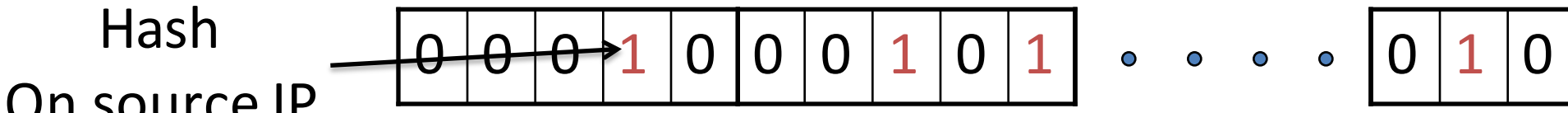


Support Many Queries

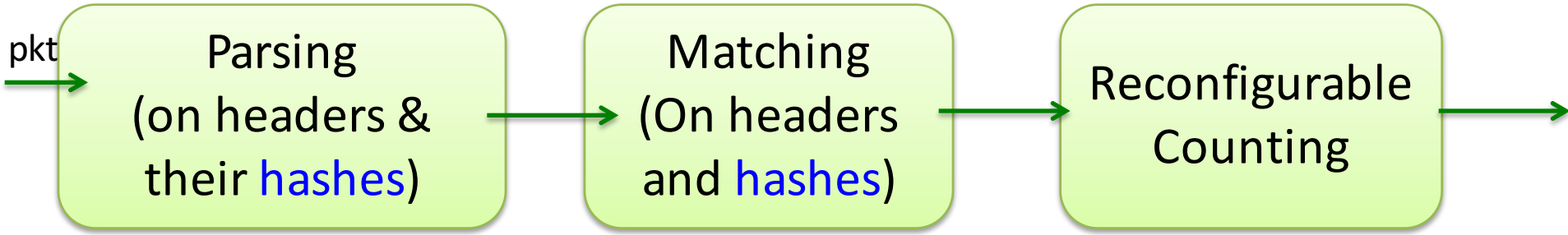


How many unique IPs send traffic to host A?

Bitmap



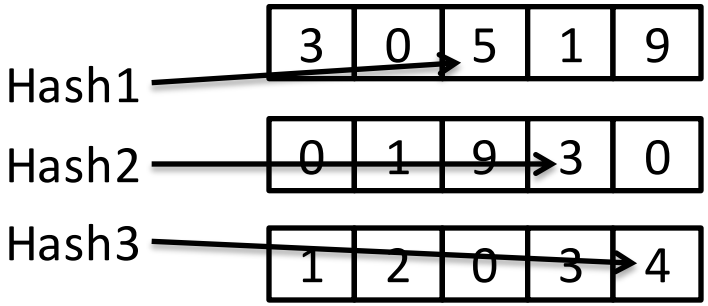
Support Many Queries



Who's sending a lot to host A?

Count-Min Sketch

Dst = host A
3 hashes on source IP



Multiple Tables

- Multiple tables for one query
  - Who's sending a lot to host A?
  - *count-min sketch* to count volume of flows
  - *reversible sketch* to identify flows with heavy counts in the count-min sketch

- Multiple tables for many queries
  - Bitmap for counting unique source IPs
  - CountMin sketch for counting traffic for each source IP

Support NetFlow

- NetFlow
  - Store flows and their counters in a hash table
  - Accounting network utilization, capacity planning, troubleshooting, and attack detection
  - It is challenging to implement NetFlow in hardware

SrcIPadd	DstIPadd	Protocol	SrcPort	DstPort	Pkts	Bytes/Pkt
173.100.21.2	10.0.227.12	11	00A2	00A2	11000	1528
173.100.3.2	10.0.227.12	6	15	15	2491	740
173.100.20.2	10.0.227.12	11	00A1	00A1	10000	1428
173.100.6.2	10.0.227.12	6	19	19	2210	1040
173.100.7.2	10.0.227.12	1	41	41	3456	1140
173.100.4.2	10.0.227.12	11	16	16	1929	840
173.100.2.2	10.0.227.12	6	0	0	2228	628
173.100.5.2	10.0.227.12	11	17	17	1325	940

A Better NetFlow

- How to handle hash collisions
  - A larger hash table to reduce collisions

  →Too much memory usage
  - A linked list of buckets, or send to the control plane

  →Too much delay per packet
- Our idea: Embrace hash collisions
  - Encode all the collided (flow, counter) pairs in one bin
  - Network-wide decode in the controller
  - Simple fixed per packet processing
  - Memory usage similar to perfect hashing and actually smaller

A Better Netflow

- Encode Hash a flow into multiple bins

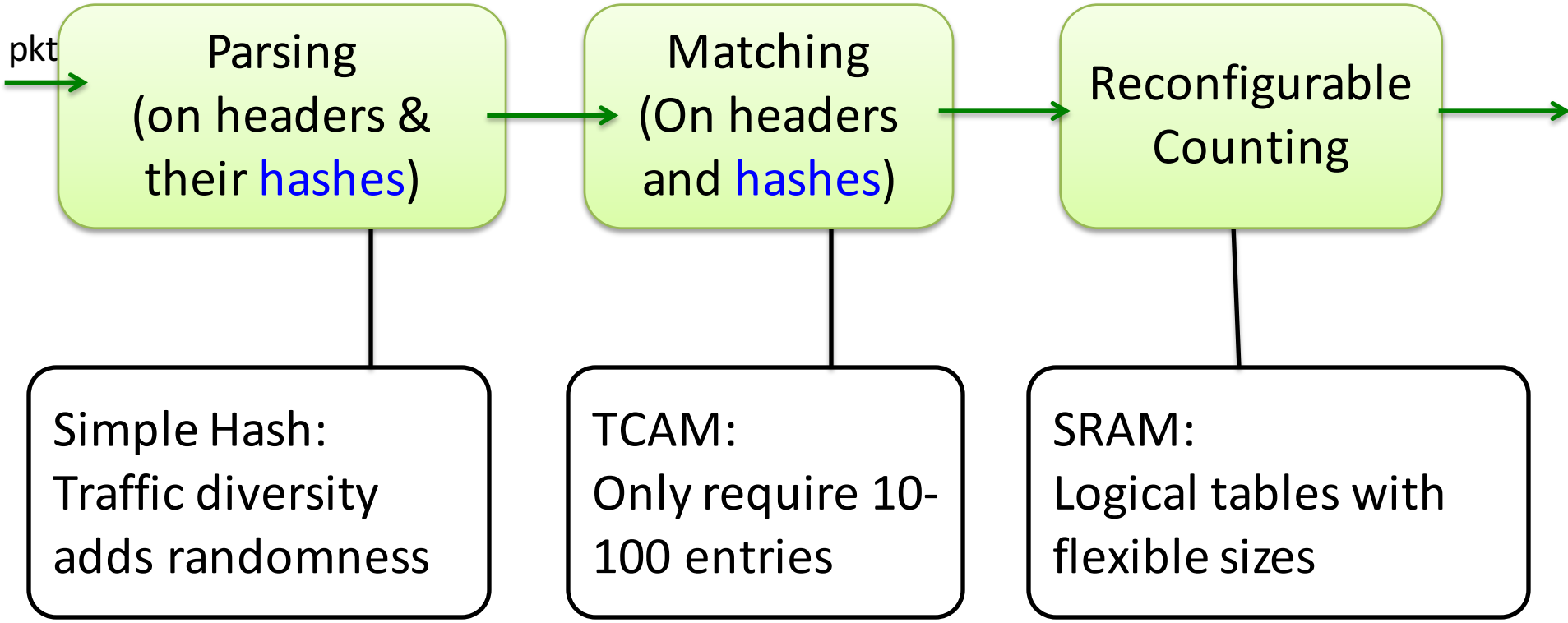
FlowXOR	a	a⊕b	b⊕c	…
FlowCounter	1	2	2	
PacketCounter	C_a	C_a+C_b	C_b+C_c	

- Decode
  - Identify bins with a single flow
  - Remove these flows, and then iterate
- Extensions
  - Identify more bins with a single flow
  - Leverage flow information from other hops

Even More Queries…

Measurement Programs	**Reconfigurable, hash-based counting table**
Loss Detection	Inversible Bloom Filter
Superspreaders	Count-min sketch; Bitmap; Reversible sketch
Traffic change detection	Count-min sketch; Reversible sketch
Traffic entropy on port field	Multi-resolution classifier; Count-min sketch
Flow size distribution	multi-resolution classifier; hash table

Build on P4-compatible Switches



A New Abstraction for Measurement

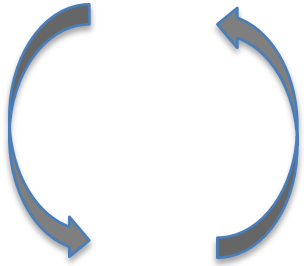
Specify measurement queries

Measurement Framework

Query independence:
Expressive Abstractions for many queries

Reconfigurable: Efficient Runtime

Dynamically configure devices & manage resources



Automatically collect the right data

Target independence:
Implementable for many devices