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

Zaoxing Liu, Antonis Manousis, Greg Vorsanger,

Vyas Sekar, and Vladimir Braverman

JOHNS HOPKINS UNIVERSITY

Carnegie Mellon

Many Monitoring Requirements

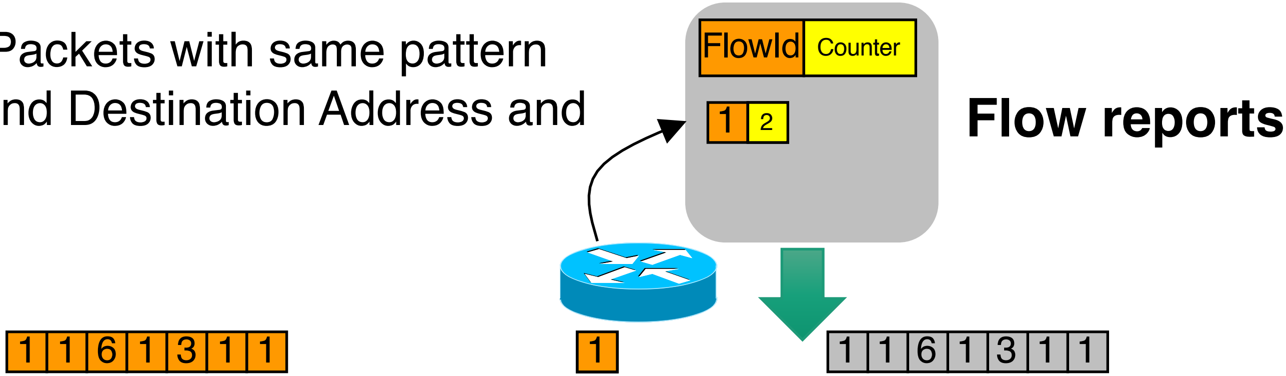
- **Traffic Engineering**
  "Flow Size Distribution"
- **Anomaly Detection**
  "Entropy", "Traffic Changes"
- **Worm Detection**
  "SuperSpreaders"
- **Accounting**
  "Heavy Hitters"

- Who's sending a lot more traffic than 10min ago? (Change)
- Who's sending a lot from 10.0.1.0/16? (Heavy Hitter)
- Are you being DDoS-ed?

Traditional: Packet Sampling

Sample packets at random, group into flows

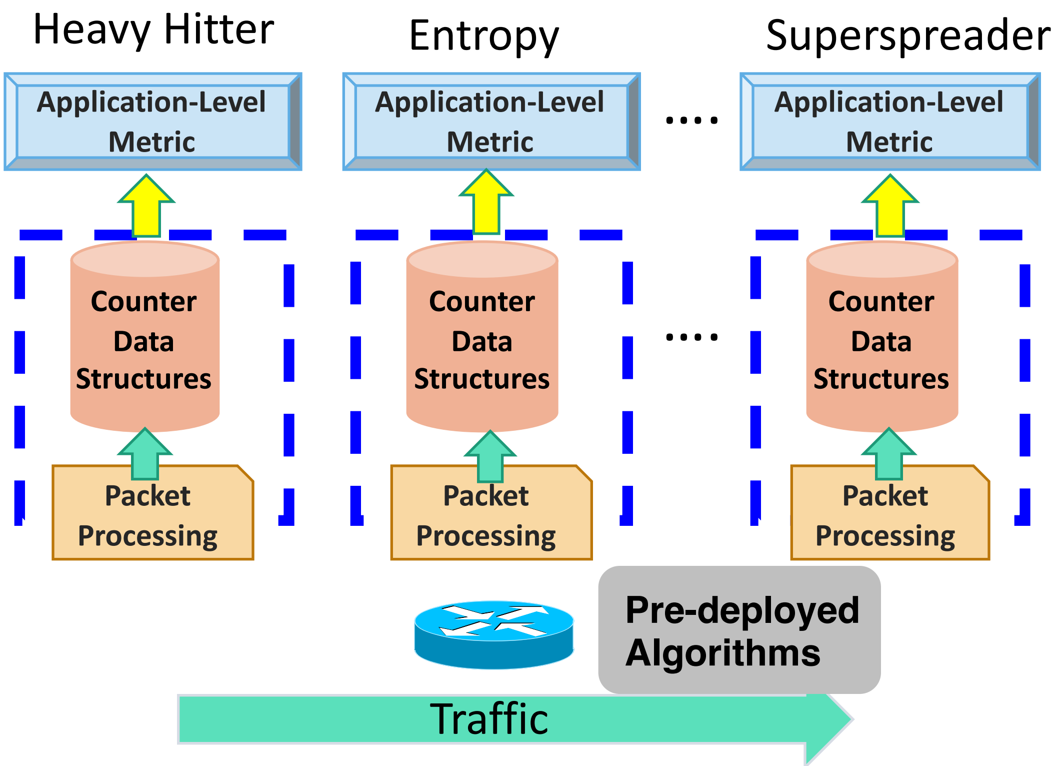
Flow = Packets with same pattern
Source and Destination Address and Ports



Estimate: FSD, Entropy, Heavy Hitters …

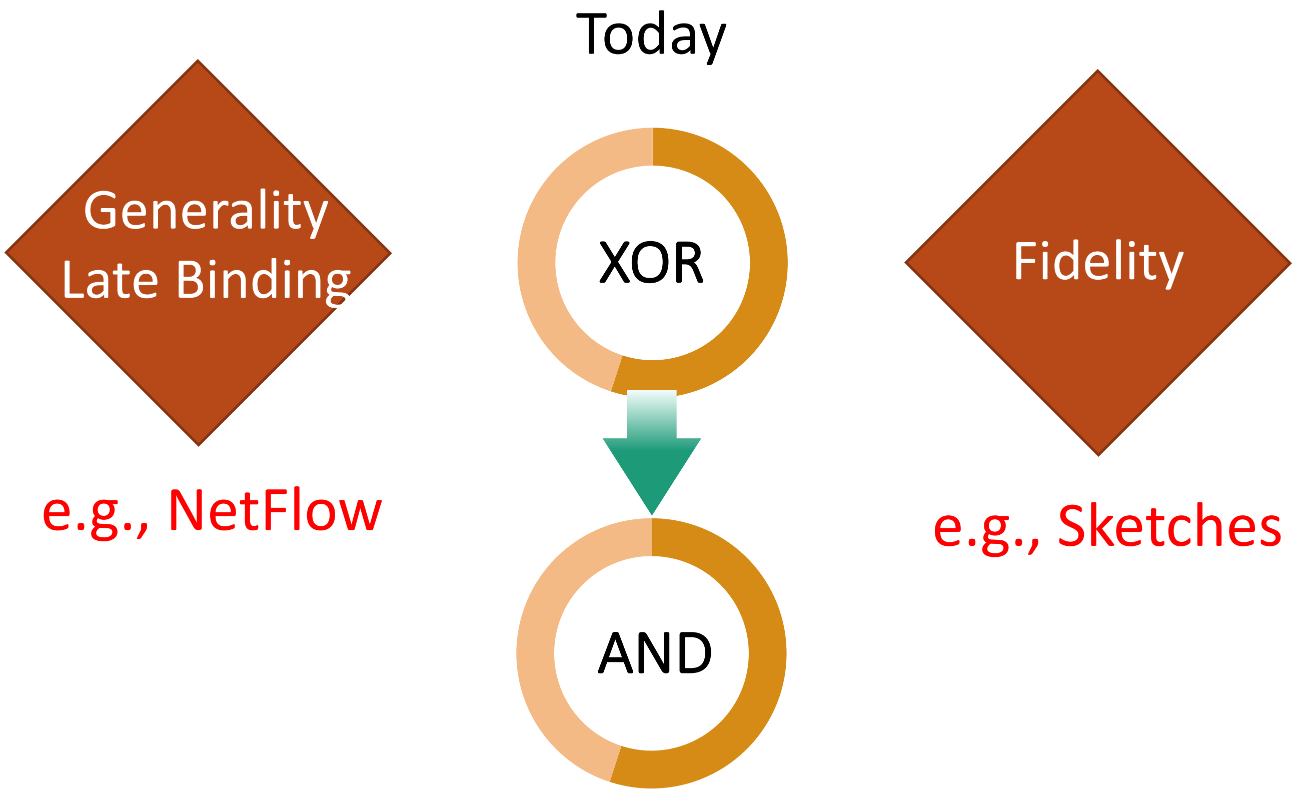
Prior work: Not good for fine-grained analysis!

Alternative: App-Specific Sketches



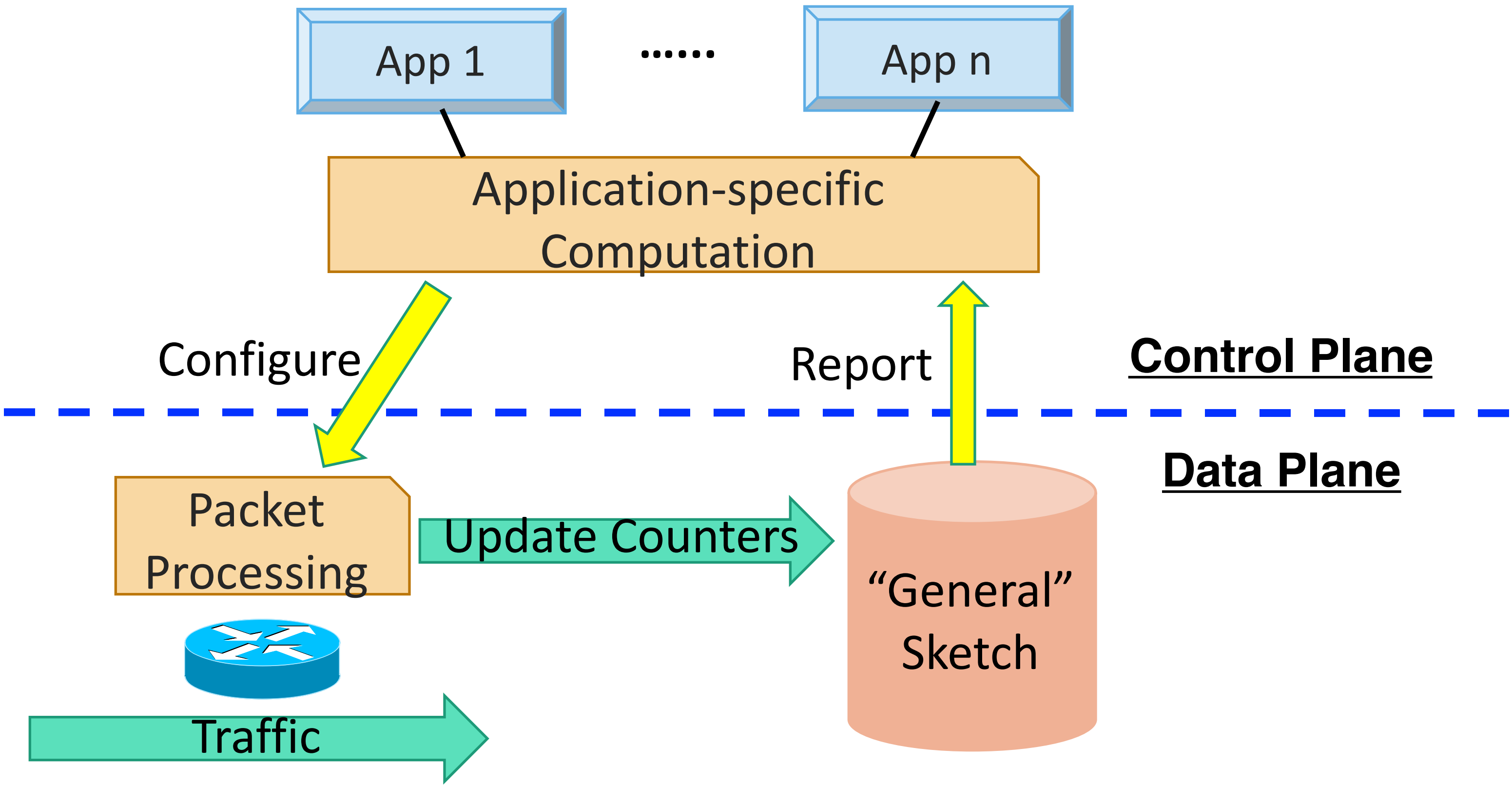
Higher Complexity with more applications
Higher development time as new applications appear
Tight Binding between monitoring data and control plane

Motivating Question



Can we achieve this?

UnivMon Vision



- **One Sketch** for multiple tasks
- Naturally **Late-binding**

Many Natural Challenges!

Does such a construction exist?

If it exists, is it feasible to implement?

Does it extend to a network-wide setting?
e.g., Multiple paths, Multiple dimensions

Is it competitive w.r.t. custom algorithms?

This Talk

Does such a construction exist?

If it exists, is it feasible to implement?

Does it extend to a network-wide setting?
e.g., Multiple paths, Multiple dimensions

Is it competitive w.r.t. custom algorithms?

This Talk

Does such a construction exist?

If it exists, is it feasible to implement?

Does it extend to a network-wide setting?
e.g., Multiple paths, Multiple dimensions

Is it competitive w.r.t. custom algorithms?

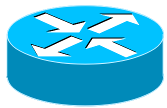
Concept of Universal Streaming

- Basic Streaming Algorithms:

(A stream of length m with n unique items)

1	1	5	1	3	3	1	2	4	6	5	……

frequency vector $< f_1, f_2 \dots f_n >$



Frequency Moments $F_k = \sum_{i=1}^{n} f_i^k$

F_2 : AMS Sketch, Count Sketch

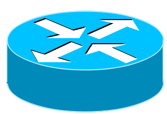
……

One algorithm solves one problem

- Universal Streaming?

1	1	5	1	3	3	1	2	4	6	5	……

frequency vector $< f_1, f_2 \dots f_n >$



Universality:
arbitrary g() function?

G-sum = $\sum_{i=1}^{n} g(f_i)$

Theory of Universal Streaming [BO'10, BO'13]

Thm 1:

There exists a universal approach to estimate G-sum when g() function is non-decreasing such that g(0)=0, and $g(f_i)$ doesn't grow monotonically faster than f_i^2 .

Thm 2:

A universal sketch construction can be used to estimate G-sum with high probability using polylogarithmic memory.

Intuition of Universal Sketch

Informal Definition: Item i is a g-heavy hitter if changing its frequency f_i significantly affects its G-sum.

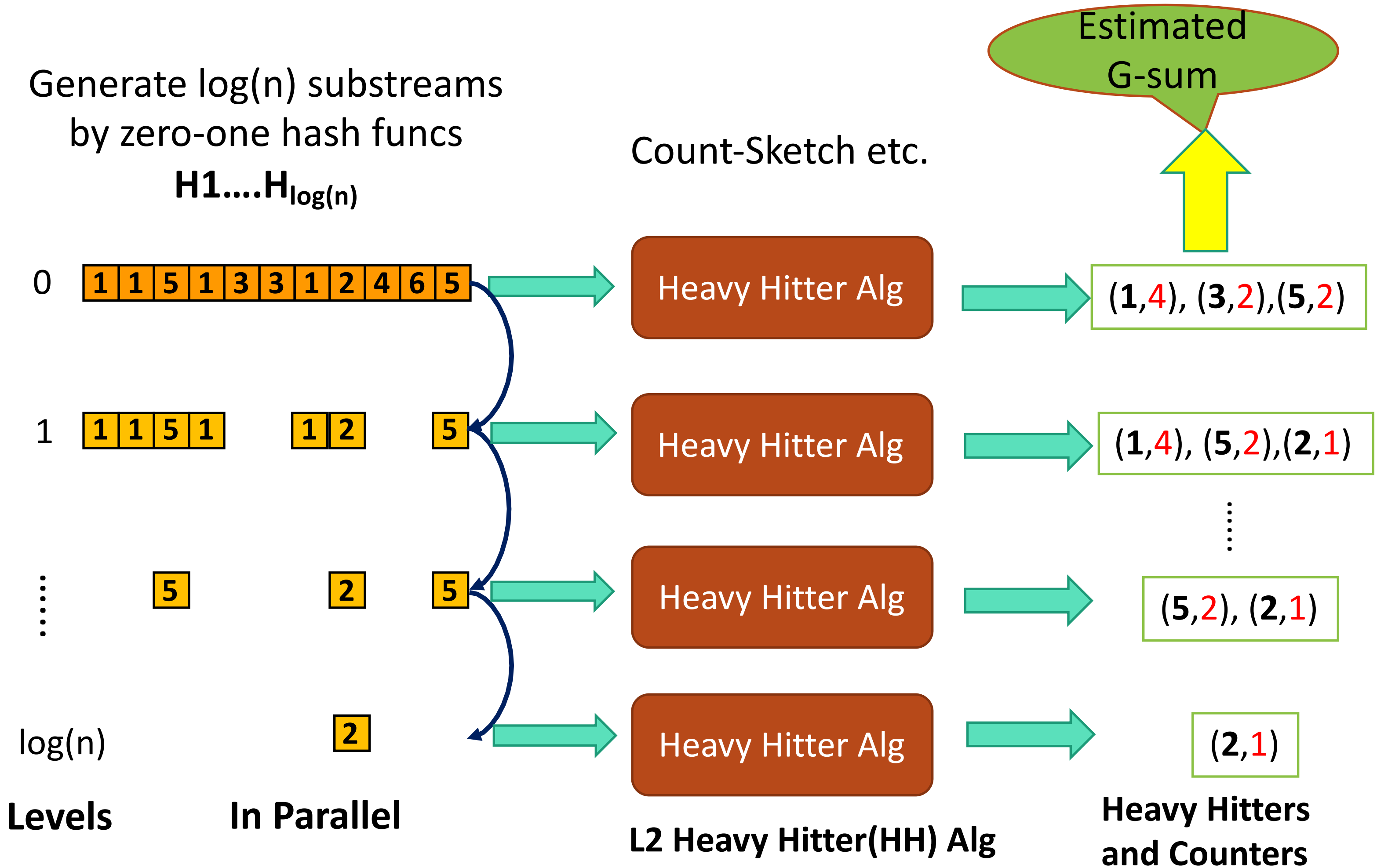
Case 1: there is one sufficiently large a g-heavy hitter

Most of mass is concentrated in this heavy hitter.
Use L2 Heavy-Hitter algorithm to find such a heavy hitter.

Case 2: there is NO single sufficiently large g-heavy hitter

Find heavy hitters on a series of sampled substreams of increasingly smaller size.

Universal Sketch Data Structure



This Talk

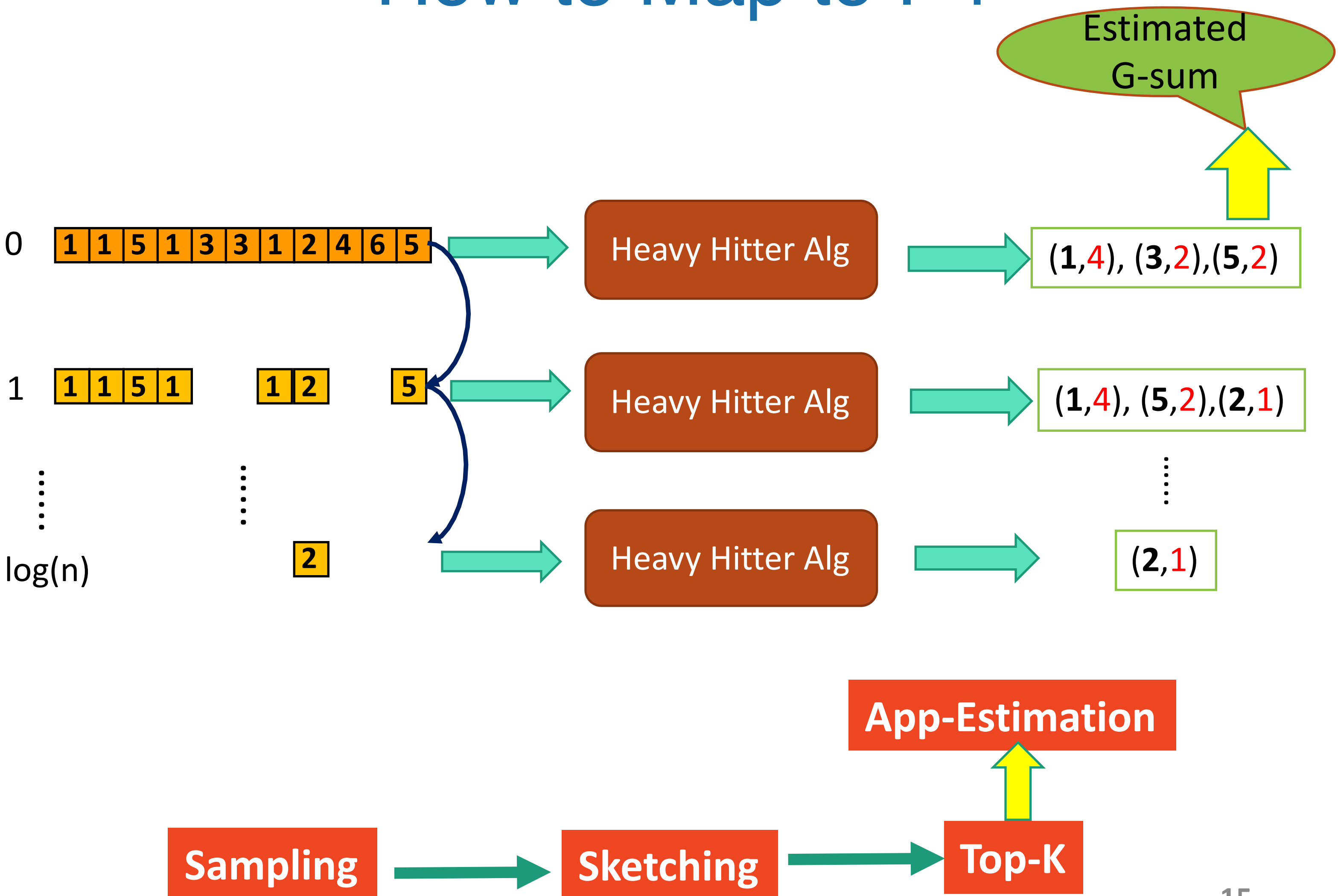
Does such a construction exist?

If it exists, is it feasible to implement?

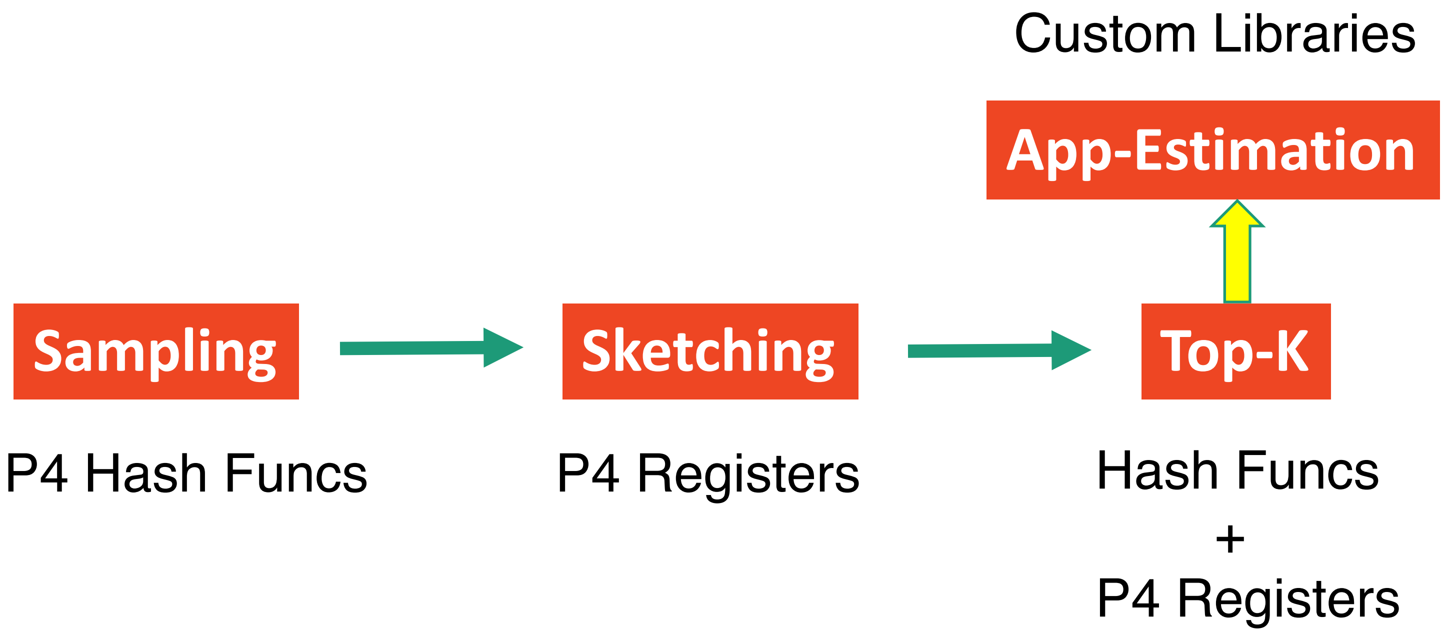
Does it extend to a network-wide setting?
e.g., Multiple paths, Multiple dimensions

Is it competitive w.r.t. custom algorithms?

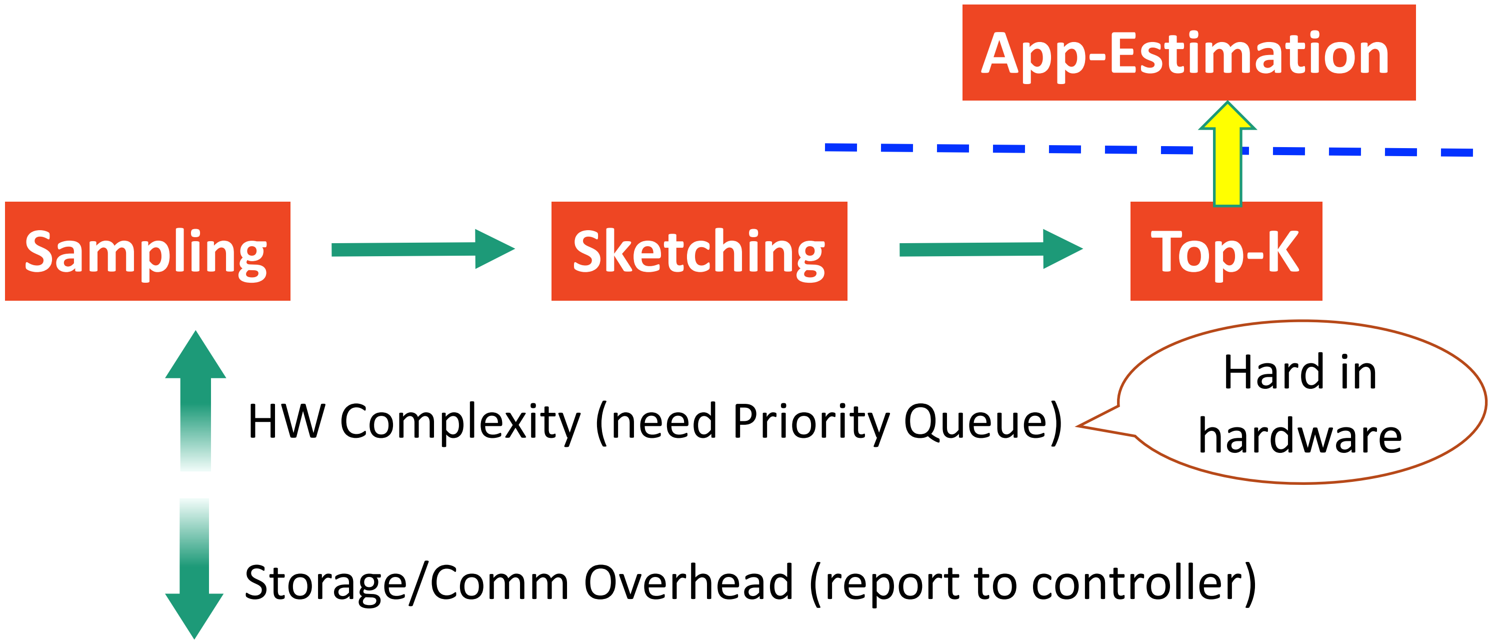
How to Map to P4



Mapping to P4



Top-K Stage on Switch



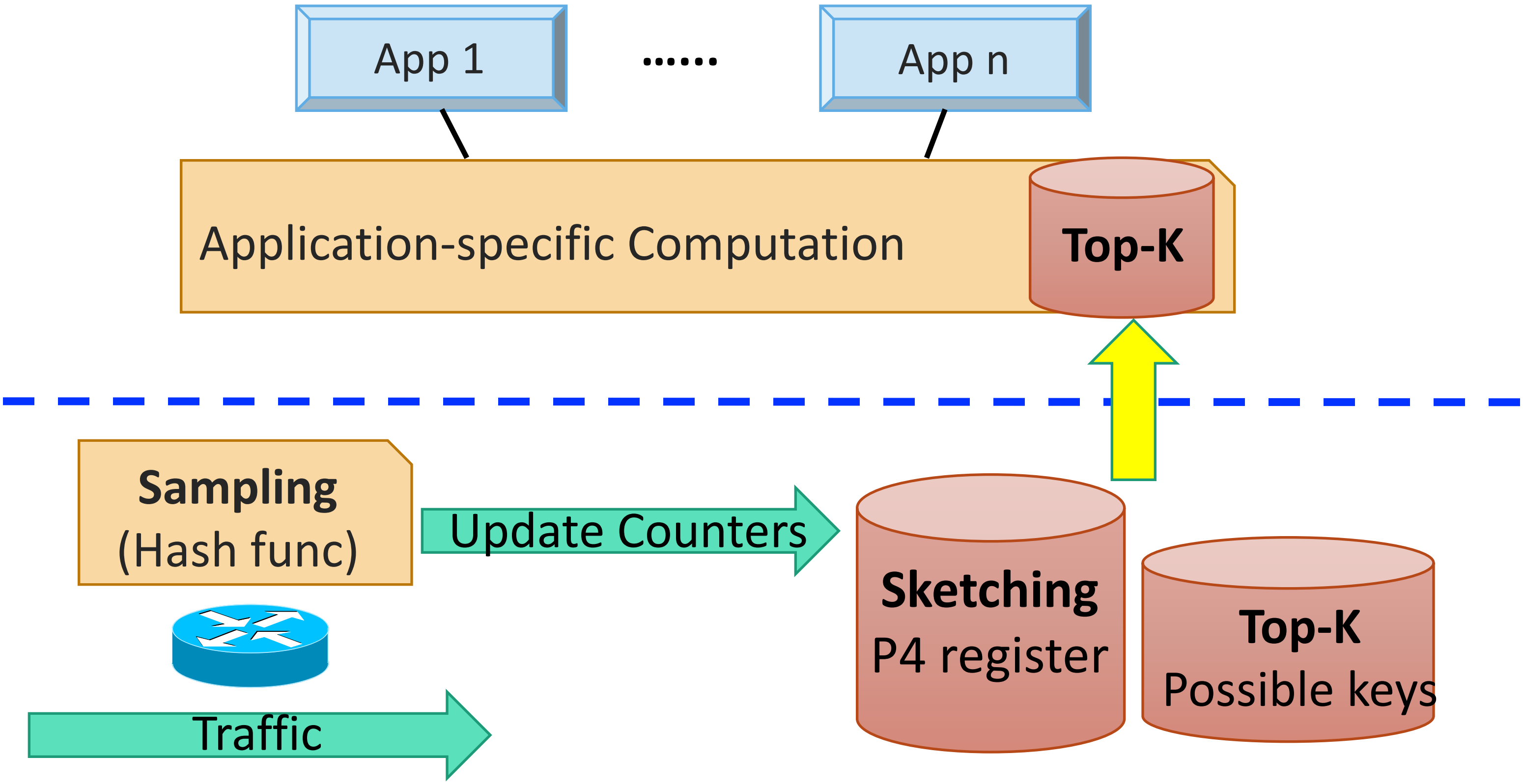
Split Top-K Stage

HW Complexity (w/o Priority Queue)

Storage/Comm. Overhead (report to controller)

Several MBs more

Implementation Summary



Sampling → **Sketching** → **Top-K** → **App-Estimation**

This Talk

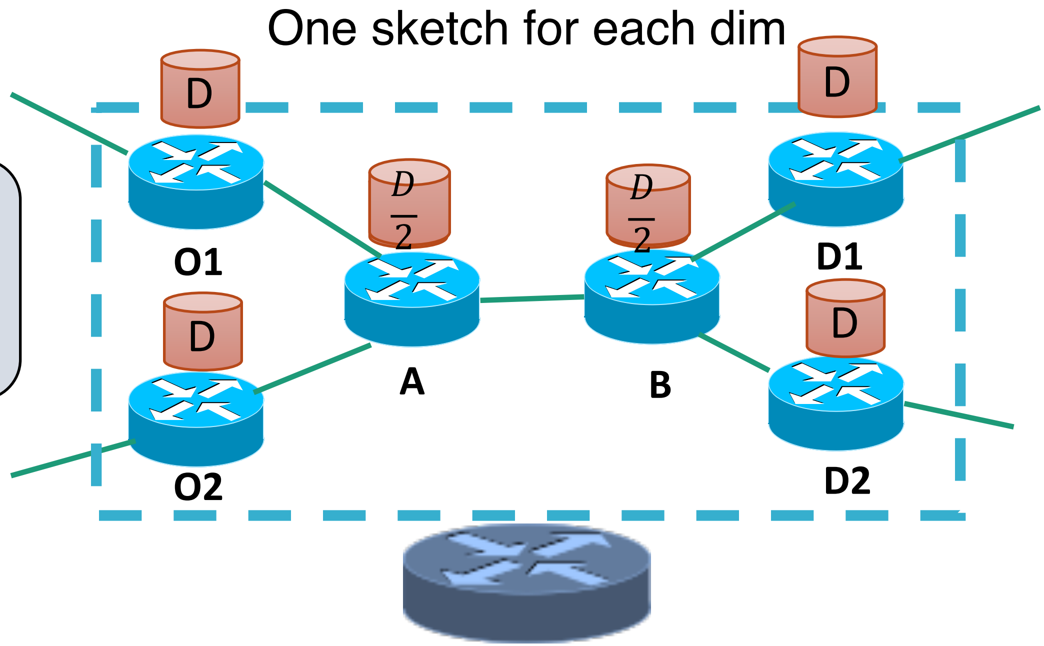
Does such a construction exist?

If it exists, is it feasible to implement?

Does it extend to a network-wide setting?
e.g., Multiple paths, Multiple dimensions

Is it competitive w.r.t. custom algorithms?

Network-wide Problem



N nodes
D dimensions
(e.g., src, srcdst)

Trivial sol: place D*N sketches
Our goal: Place s sketches, where s<<D*N
One-big-switch abstraction

This talk

Does such a construction exist?

If it exists, is it feasible to implement?

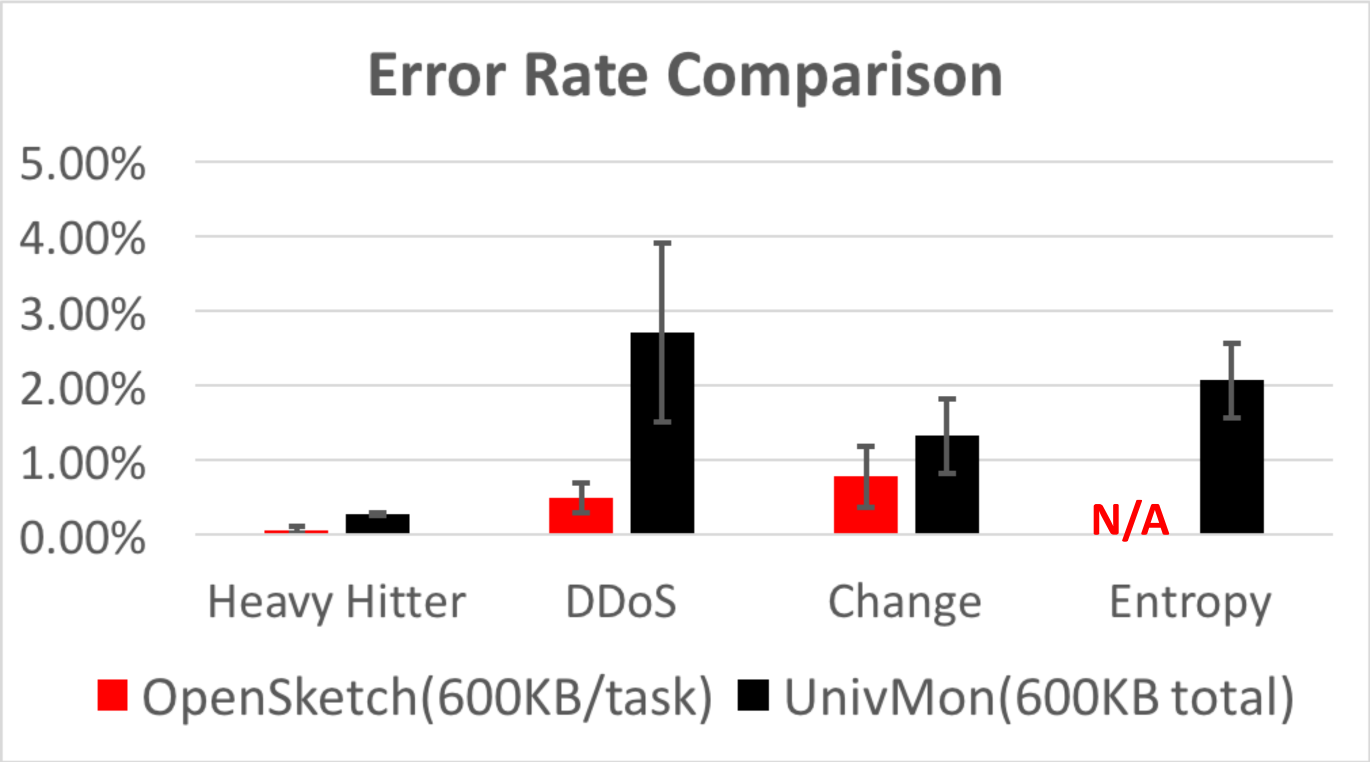
Does it extend to a network-wide setting?
e.g., Multiple paths, Multiple dimensions

→ Is it competitive w.r.t. custom algorithms?

Evaluation Setup

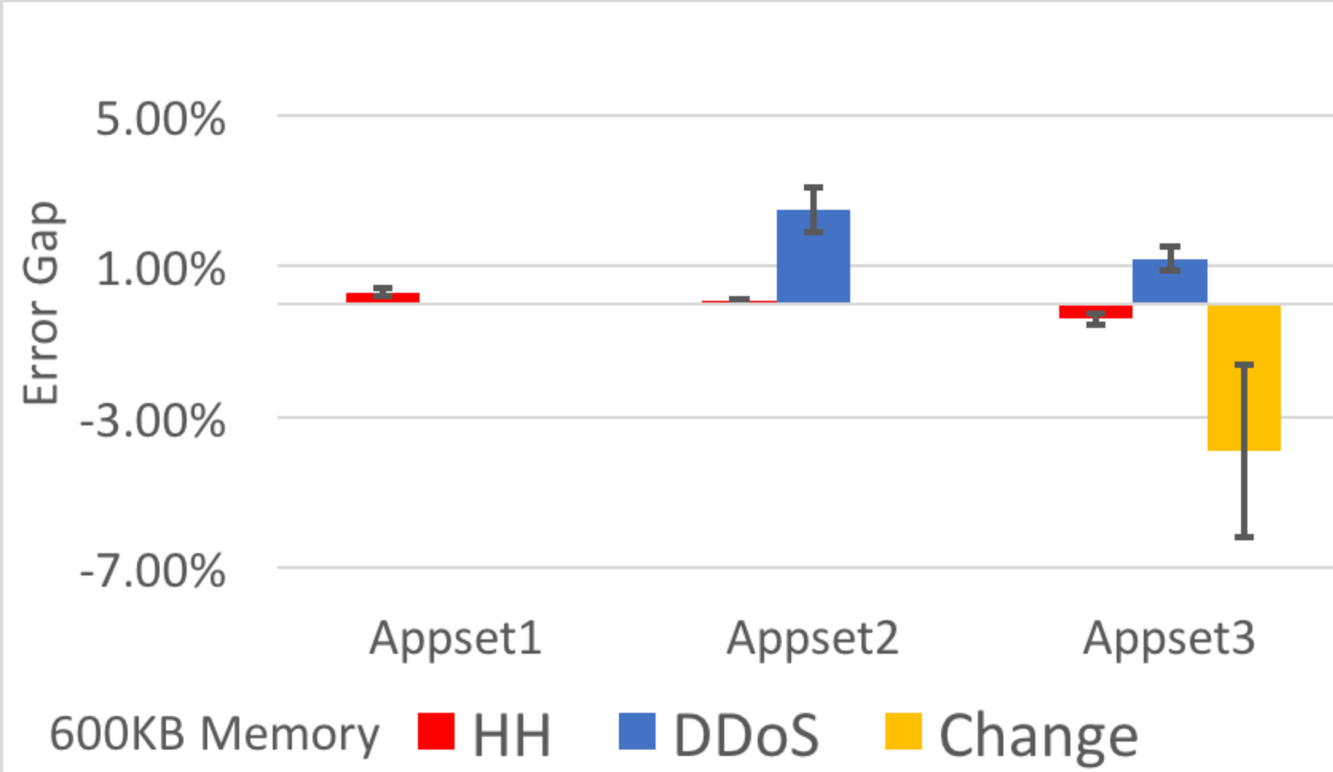
- Traces: CAIDA backbone traces
  - Split into different “epoch” durations
- Memory setup: 600KB—5MB
- Application metrics: HH, Change, DDoS, etc.
- Custom algorithms from OpenSketch

UnivMon is Competitive Per-App



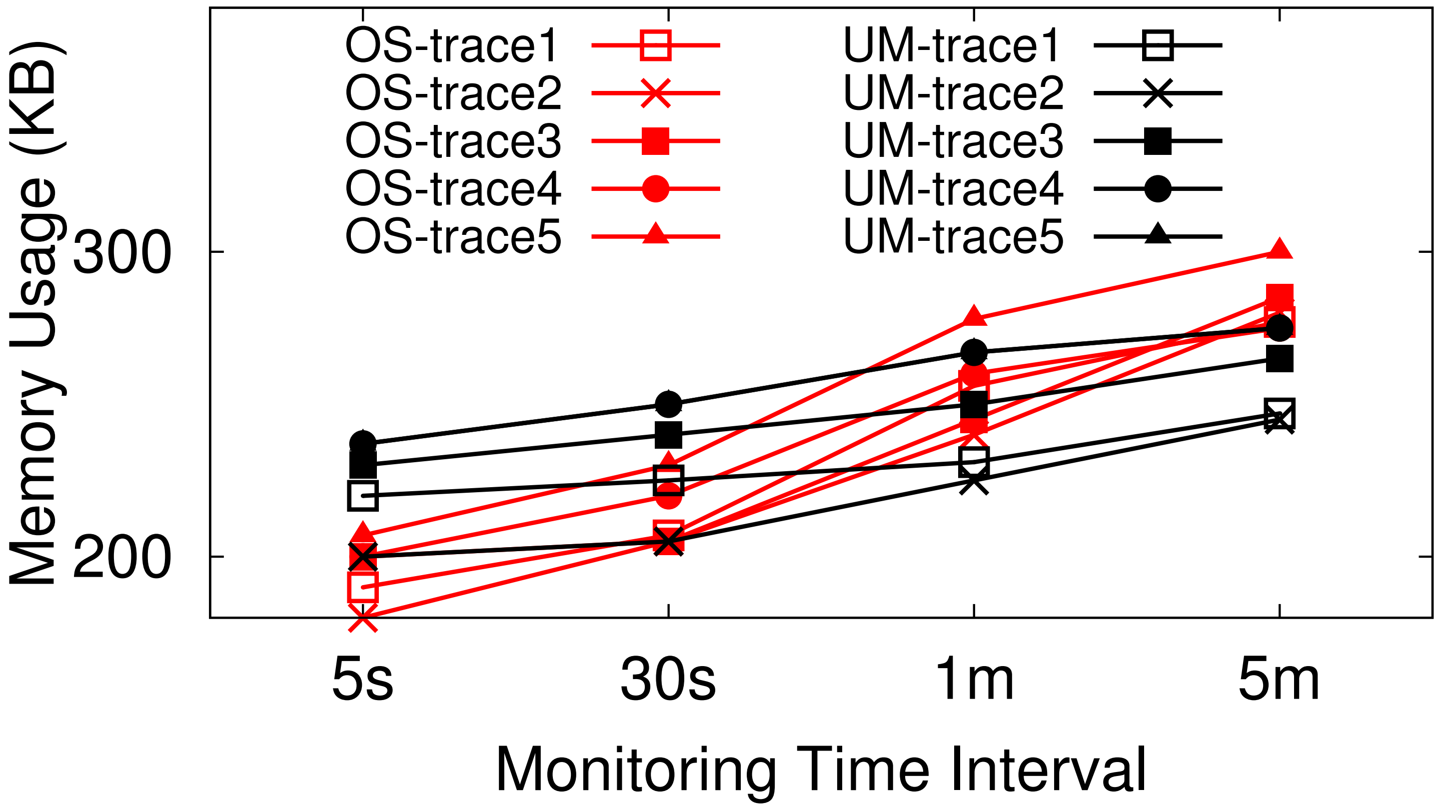
Max error gap < 3.6%; Results hold across multiple traces

UnivMon Better for Larger Portfolio



Clear advantages when handling more applications

Memory needs are reasonable



Slow increase (logarithmically) and supports larger windows

Conclusions

- Network management needs many metrics
- Traditional: Generality XOR Fidelity
  - E.g., NetFlow vs Custom Sketches
- New opportunity: Universal Sketches!
  - Generality AND Fidelity AND Late Binding
- UnivMon brings this opportunity to fruition
  - Practical, realizable in P4
  - Comparable (and better) than custom
  - Amenable to "network-wide" abstractions
  - Many exciting future directions:
    - Theoretical improvements, Native multidimensional, etc.

Network-wide coordination helps