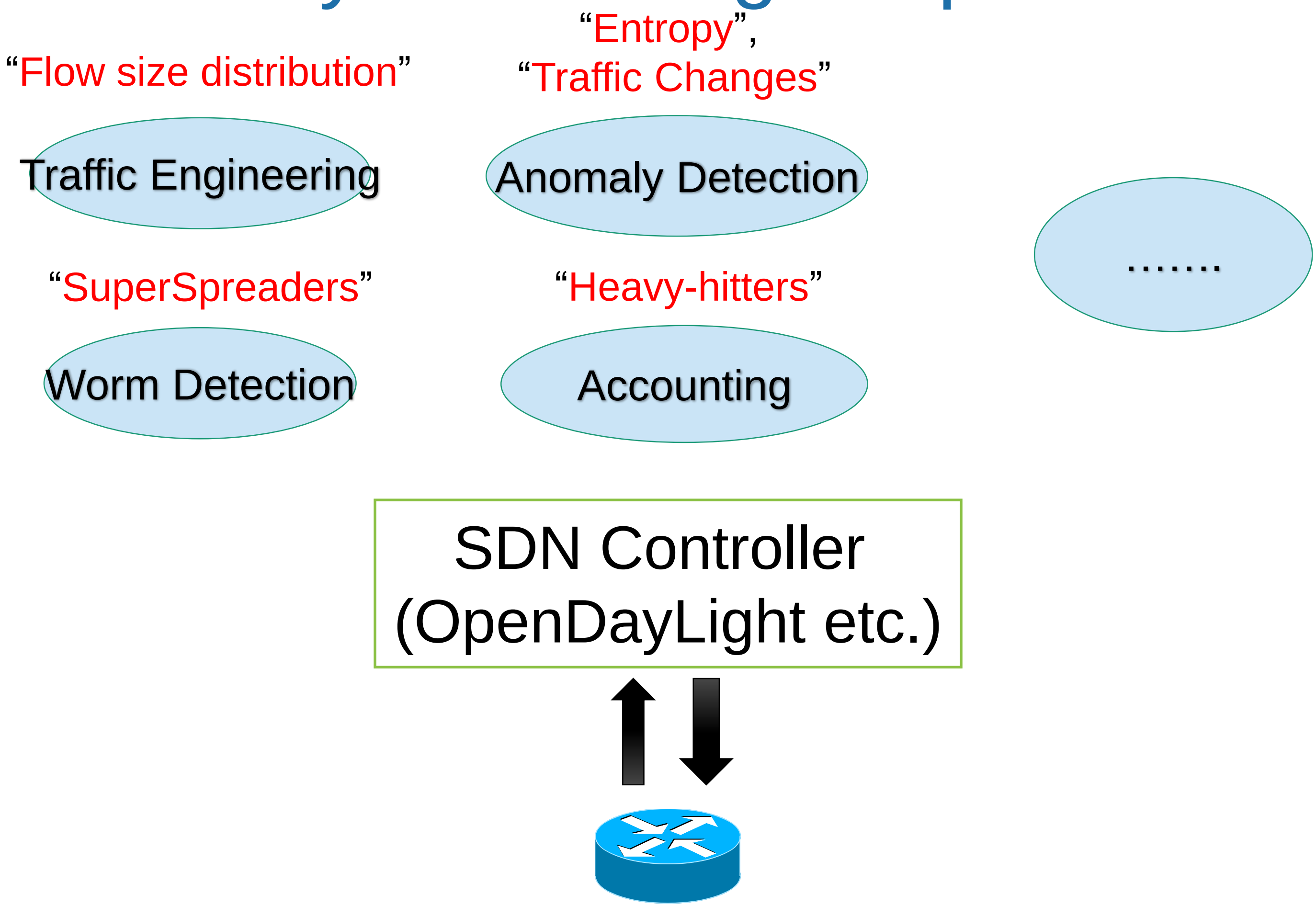
UnivMon: Software-defined Monitoring with Universal Sketch

Zaoxing (Alan) Liu

JOHNS HOPKINS UNIVERSITY

Joint work with Antonis Manousis (CMU), Greg Vorsanger(JHU), , Vyas Sekar (CMU), and Vladimir Braverman(JHU)

Network Management: Many Monitoring Requirements



Network Management: Many Monitoring Requirements

"Flow size distribution"

Traffic Engineering

"Entropy", "Traffic Changes"

Anomaly Detection

"SuperSpreaders"

Worm Detection

"Heavy-hitters"

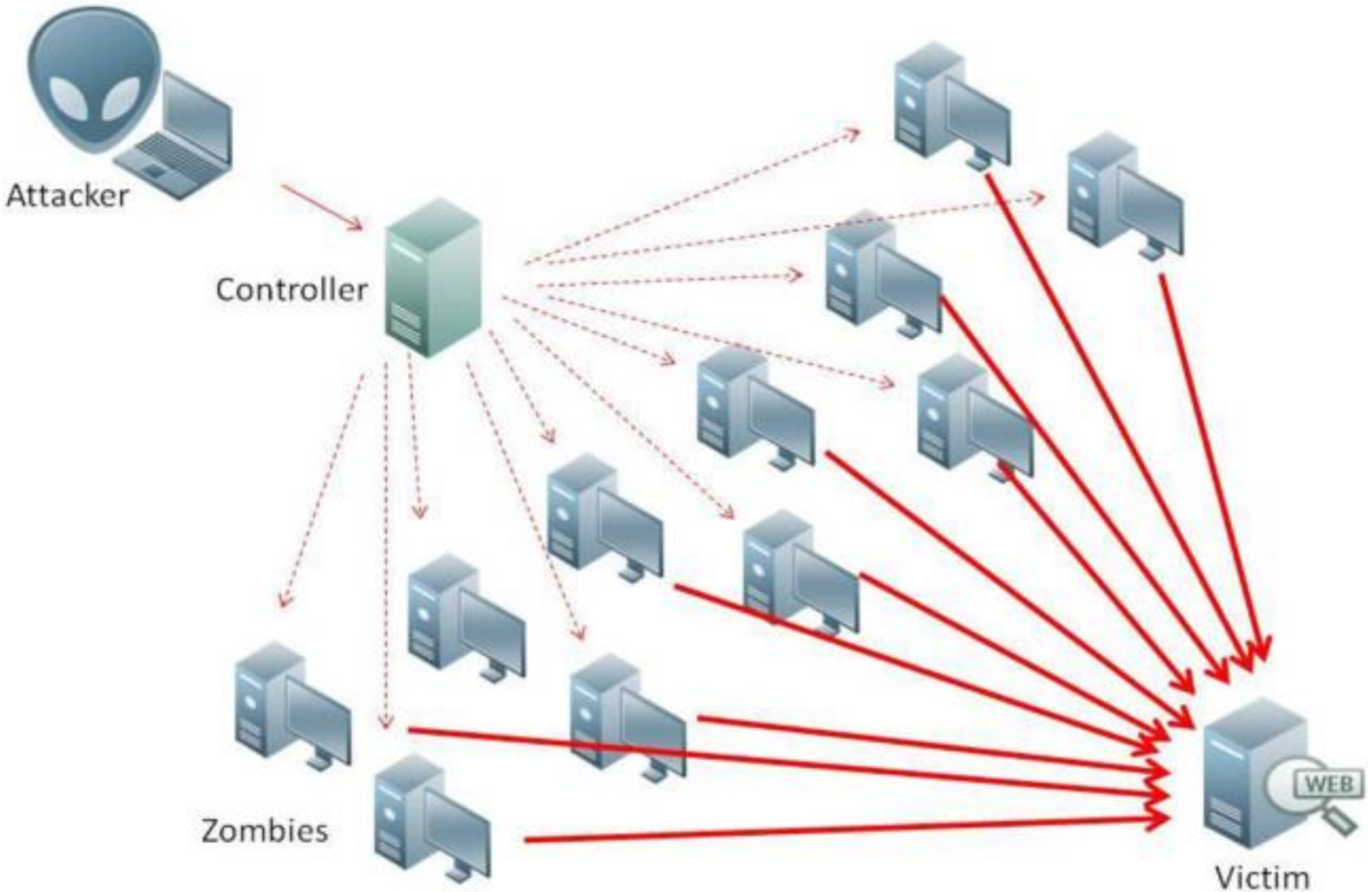
Accounting

....

Typical Measurement Questions:

- 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?

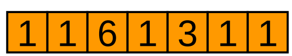
Example: A Victim being DDoSed

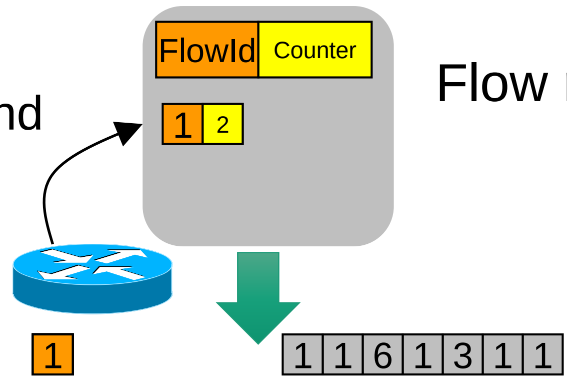


Traditional: Packet Sampling

Sample packets at random,
aggregate into flows

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



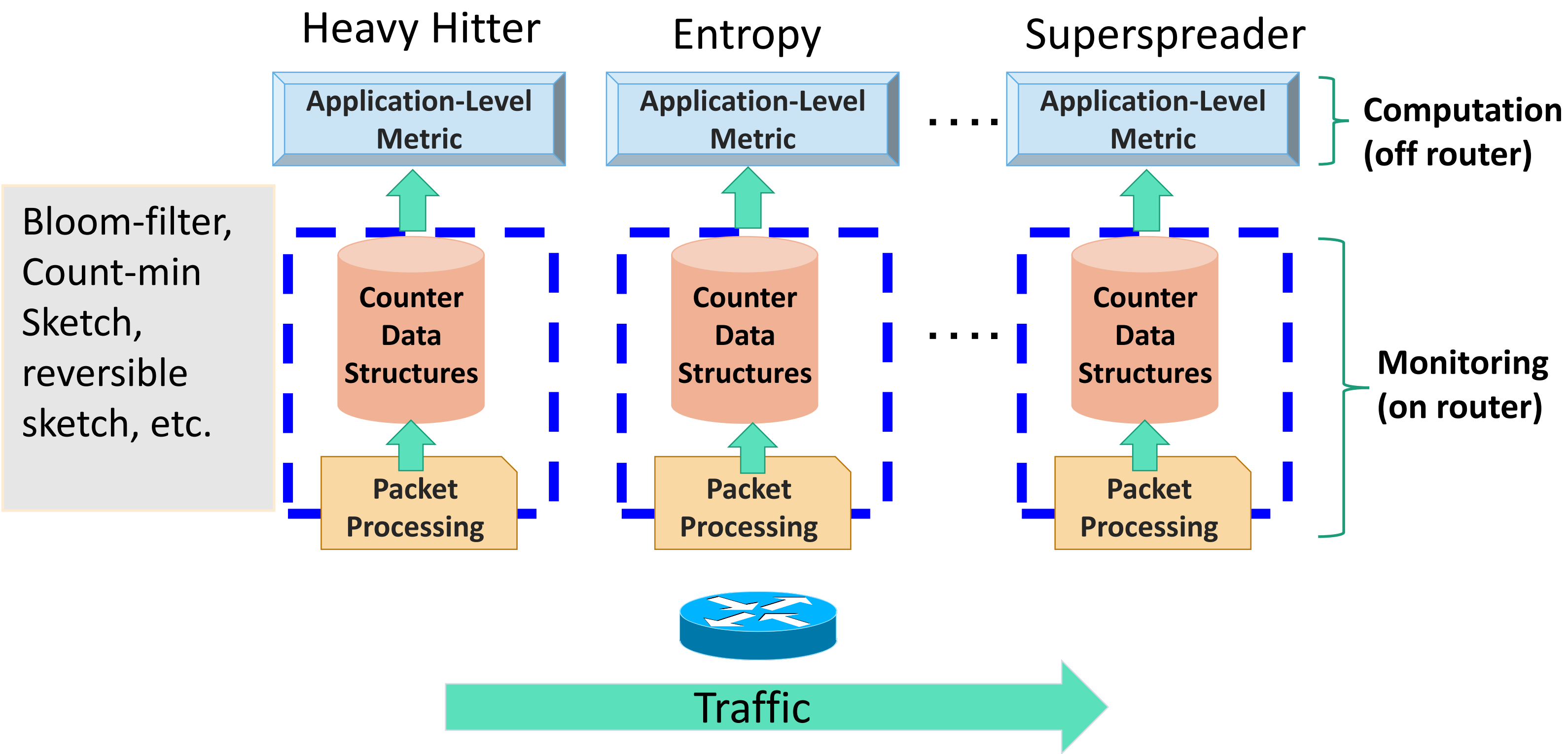


Flow reports

Estimate: FSD, Entropy, Heavyhitters …

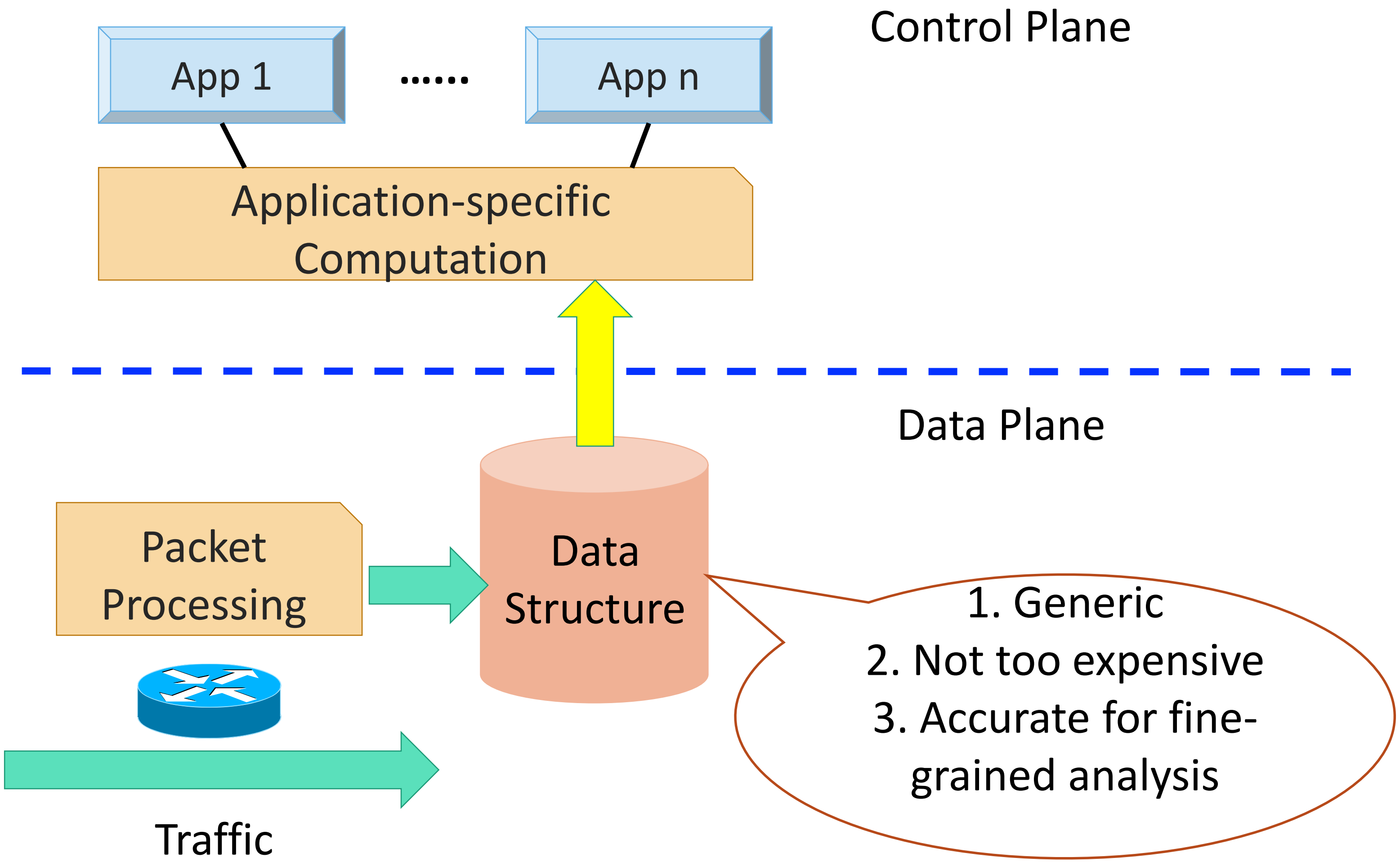
Not good for fine-grained analysis
Extensive literature on limitations for many tasks!

Application-Specific Sketches



Complexity: Need per-metric implementation
Recent Example: OpenSketch [NSDI'13]
Trend: Many more applications appear!

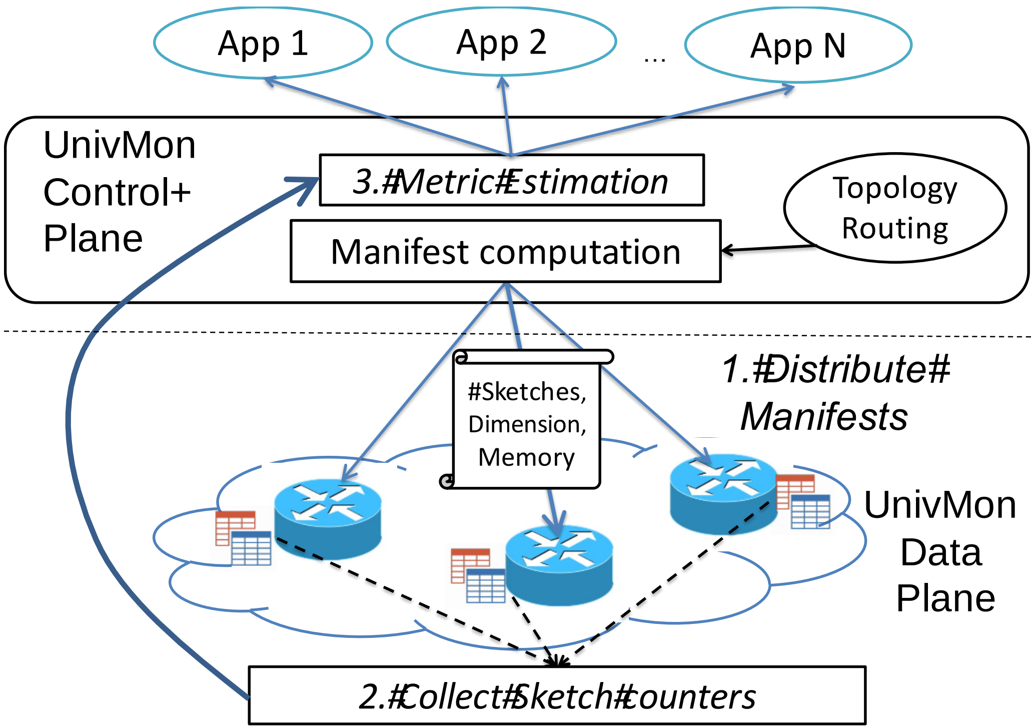
A Generic Method



Outline

- Motivation
- UnivMon System Design
- UnivMon Algorithm
- Evaluation

Our Design: UnivMon



Control plane

- Metric estimation
- Monitoring manifest distribution

Data Plane

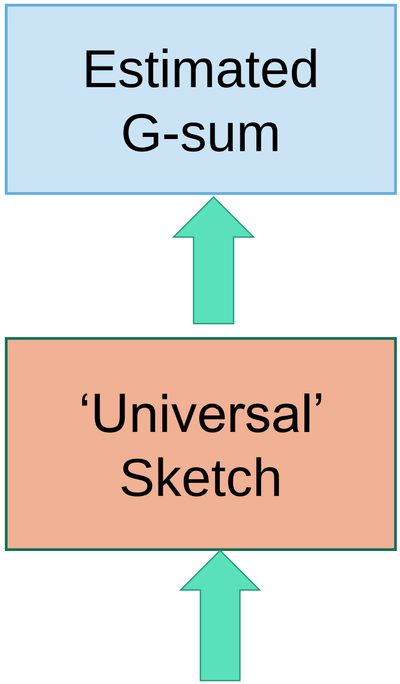
- Traffic data collection
- Sketch counter updates

- **Late-binding** for applications: data plane is general-purpose.
- **One Sketch**: no need of memory allocation for multiple tasks

Outline

- Motivation
- UnivMon Design
- UnivMon Algorithm
- Evaluation

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



As long as $g(fi)$ does not grow monotonically faster than f_i^2

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

frequency vector is < $f_1, f_2 \ldots f_n$ >

1	1	5	1	3	3	1	2	4	6	5

……(A stream of length m with n unique items)

- When $g(x) = x^0$, G-sum is # of distinct items. [SuperSpreader]
- When g(x) = xlog(x), G-sum is the entropy norm. [Entropy]

Basics: Streaming Algorithms

1	1	5	1	3	3	1	2	4	6	5

……(A stream S of length m with n unique items)

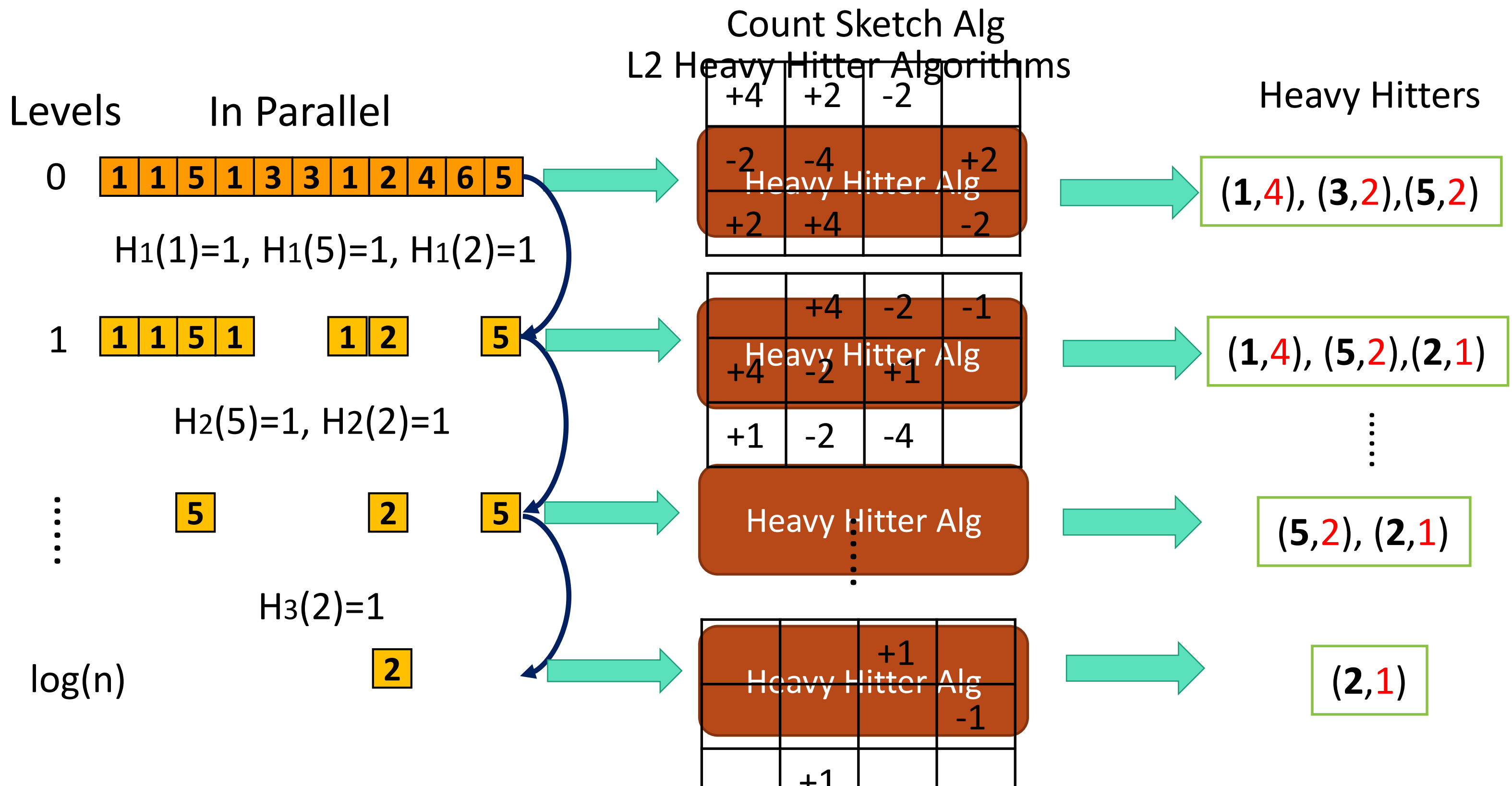
- **K-th Frequency Moments:**

  $F_k(S) = \sum_{i=1}^{n} f_i^k$

  F_0 is n: the number of distinct items in S.
  F_1 is m: the length of S.
  F_2 is Gini's index of homogeneity.
  When k>2, the space lower bound is $\Omega(n^{1-2/k})$ [CKS'03]
  (For current applications, the case of k>2 is not that interesting)

- **Heavy Hitters (Frequent Items):**
  g-heavy item i : $g(f_i) > \alpha$ G-sum for some α
  Count-min sketch [CM'04] is a popular L1-heavy hitter algorithm

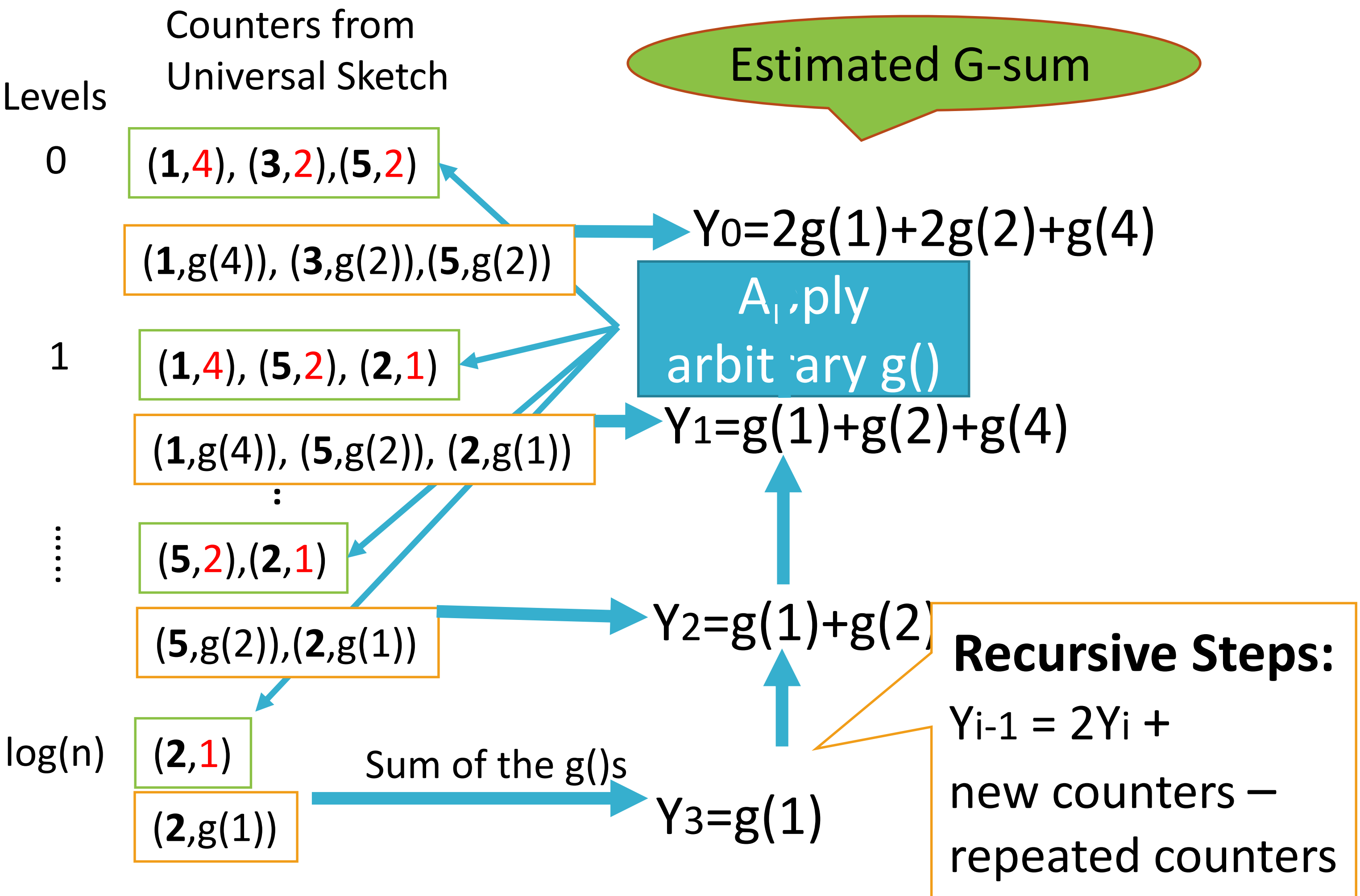
Universal Sketch Data Structure



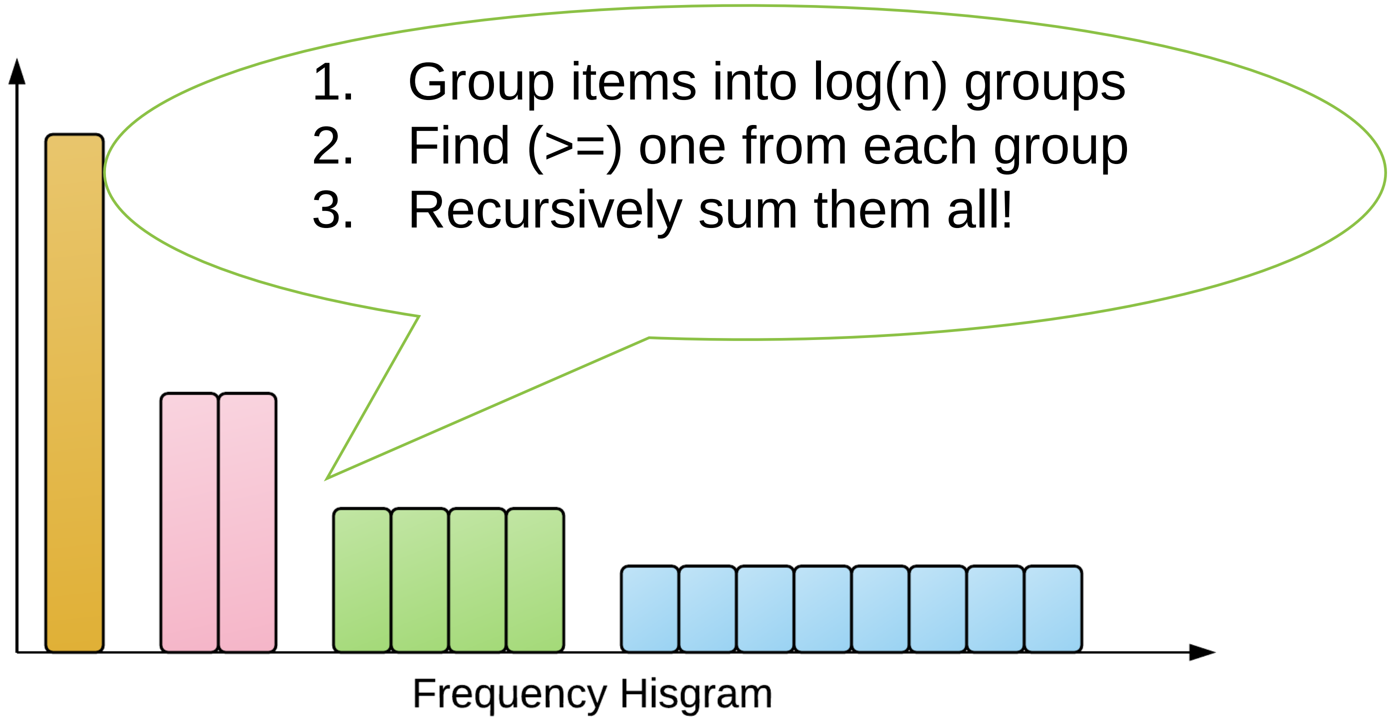
Generate log(n) substreams by zero-one hash funcs $H1....H_{log(n)}$

Count-Sketch etc. Similar to counting bloom filter

Estimating G-sum

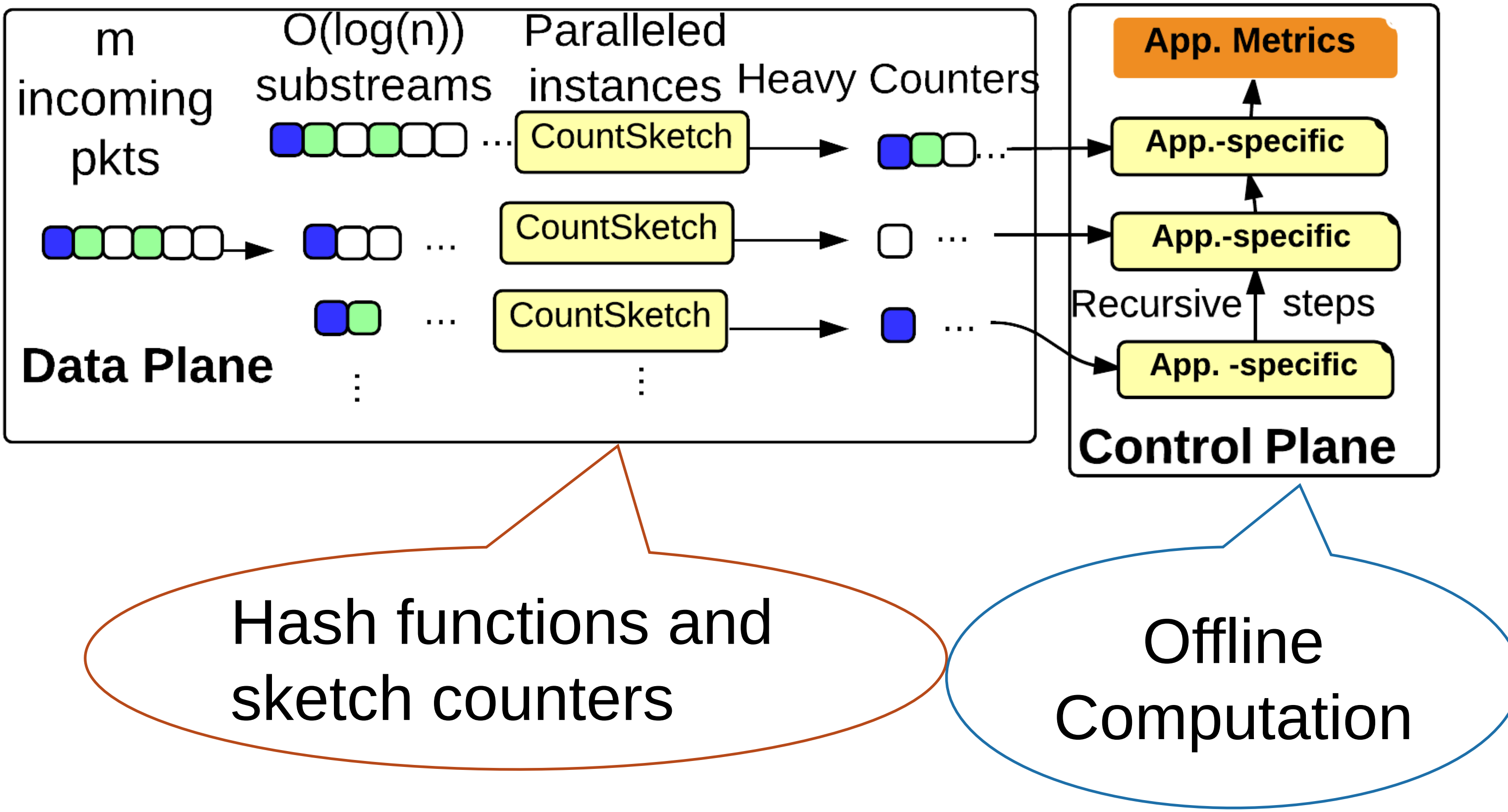


Intuitions of Universal Sketch

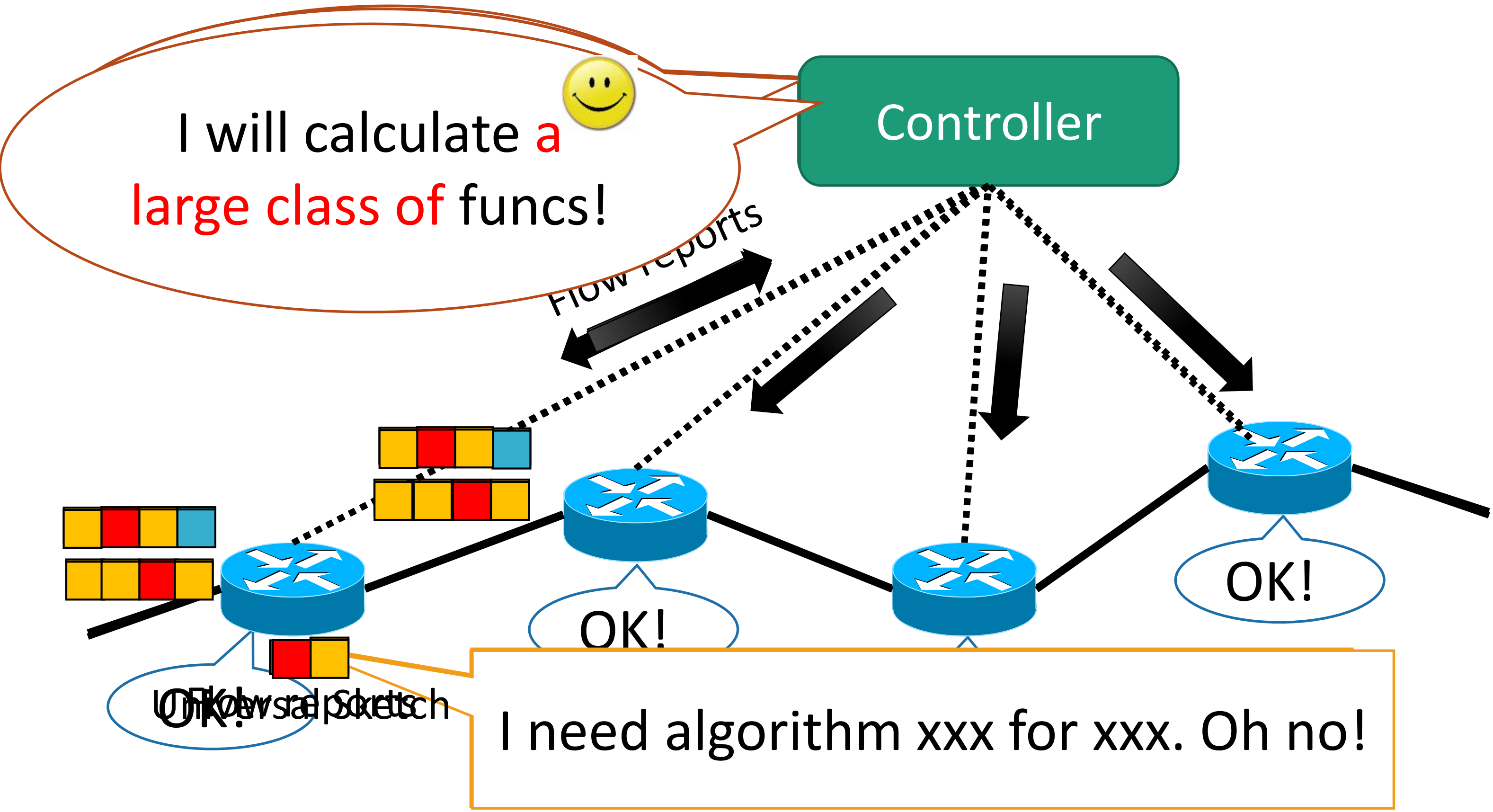


Idea: Detect items that **contribute** most to G-sum

Putting it together: UnivMon



Our Approach: UnivMon
Packet Sampling Solution
Application-specific Approach



Outline

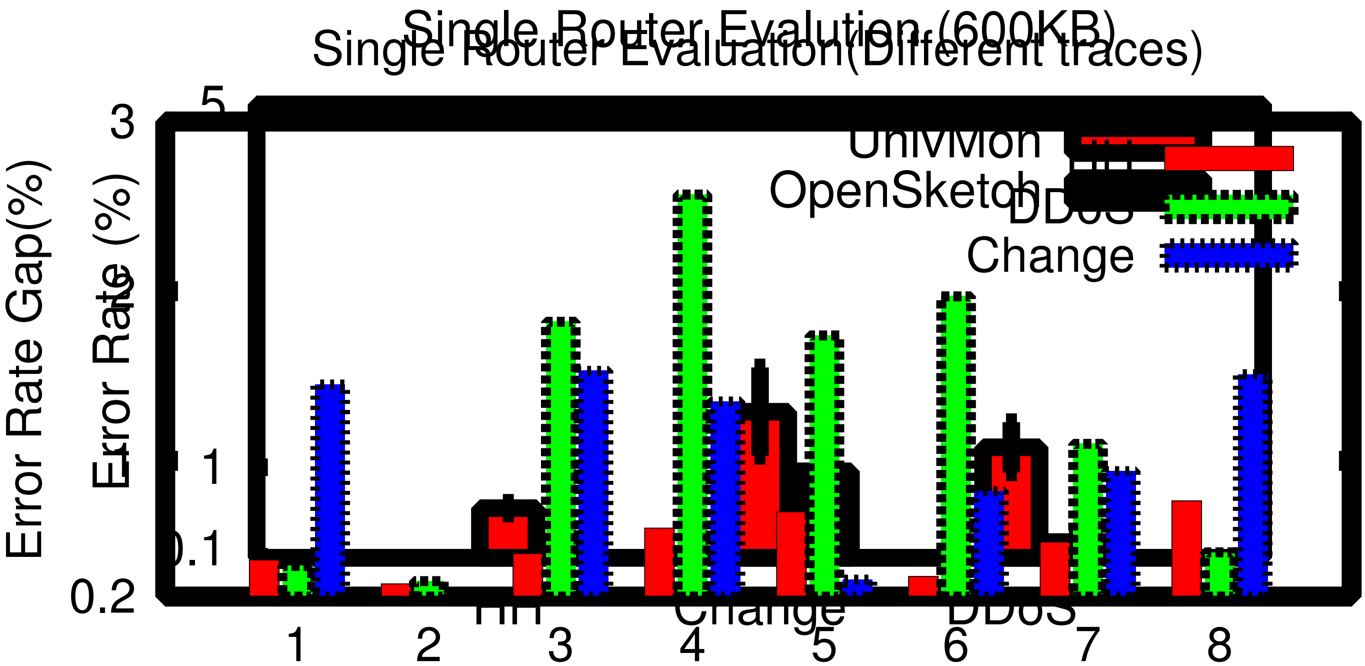
- Motivation
- UnivMon Design
- UnivMon Algorithm
- Evaluation

Key Evaluation Questions

- Feasibility of hardware deployment?
  - ✓ *Removed expensive data structures*
  - ✓ *Implemented in P4*
- UnivMon's accuracy
  - ✓ *Compare to the up-to-date sketch algorithms*
- UnivMon's stability
  - ✓ *Stabilities over different traces*

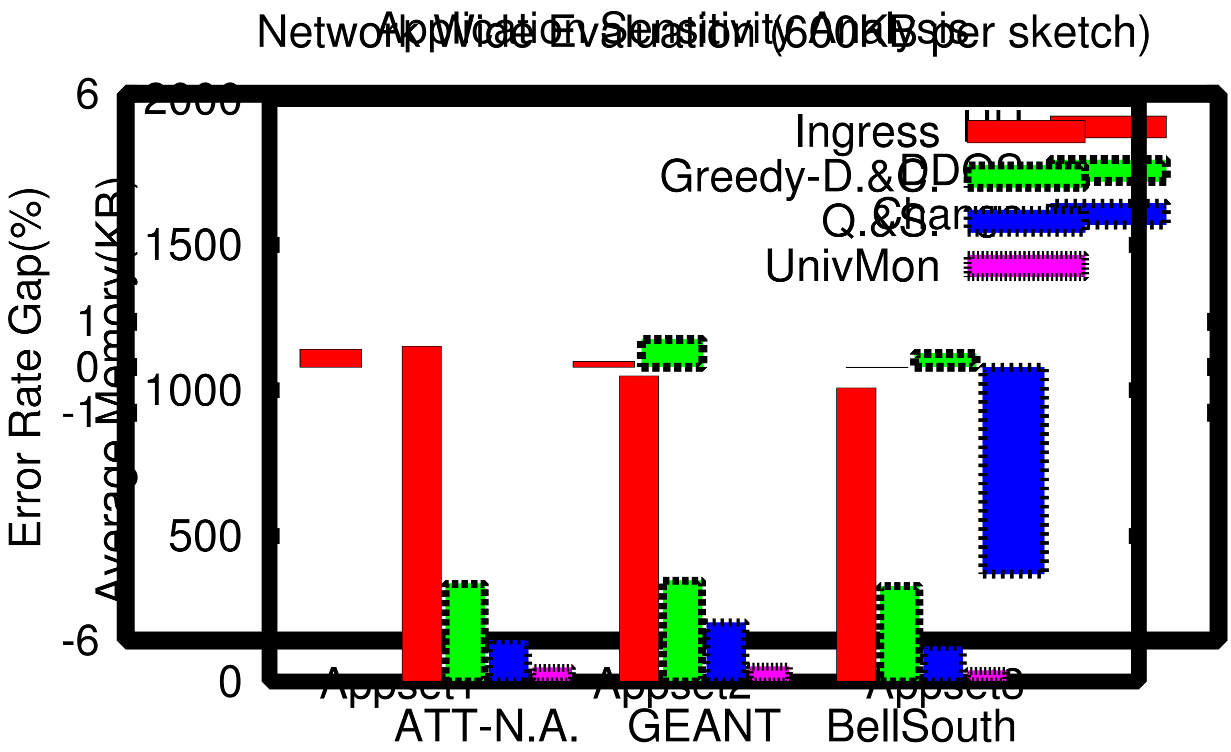
Evaluation

- Comparison with custom sketches via OpenSketch



Evaluation

- Comparison with custom sketches via OpenSketch



Conclusions

- Traditional packet sampling based approaches have limitations for fine-grained analysis.
- Custom Sketches (e.g. OpenSketch)
  - Need to know applications beforehand
  - High Implementation cost
  - More expensive on multi-tasks
- UnivMon: an efficient and general sketching based approach with late-binding on applications.

Future Directions

- Multi-dimensional data
- Performance optimization for hardware
- Dynamically change monitoring scope
- New Theories on Universal Sketch:
  - Sliding Window Model [SODA'16]
  - Functions of One Variable [PODS'16]
  - Symmetric Norms [arXiv 1511.01111]

Thank you!