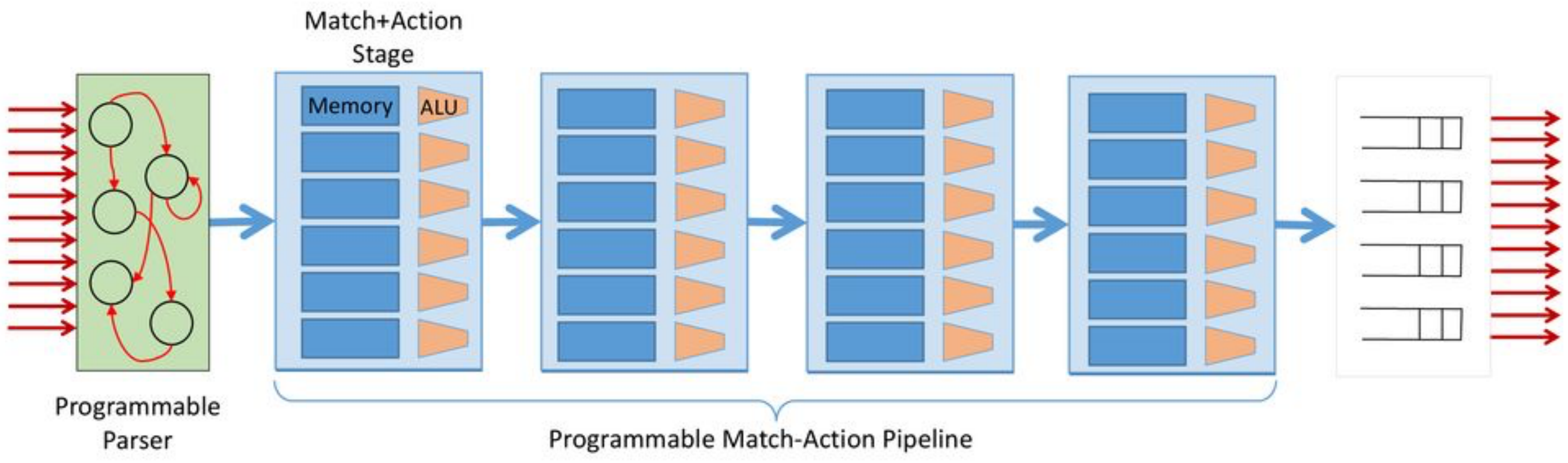
DISCOvering the Heavy Hitters with Disaggregated Sketches

Valerio Bruschi [1], Ran Ben Basat [2], Zaoxing Liu [3], Gianni Antichi [4], Giuseppe Bianchi [1], Michael Mitzenmacher [2]
University of Rome Tor Vergata/CNIT [1], Harvard University [2], Carnegie Mellon University [3], Queen Mary University of London [4]

1. Network programmability

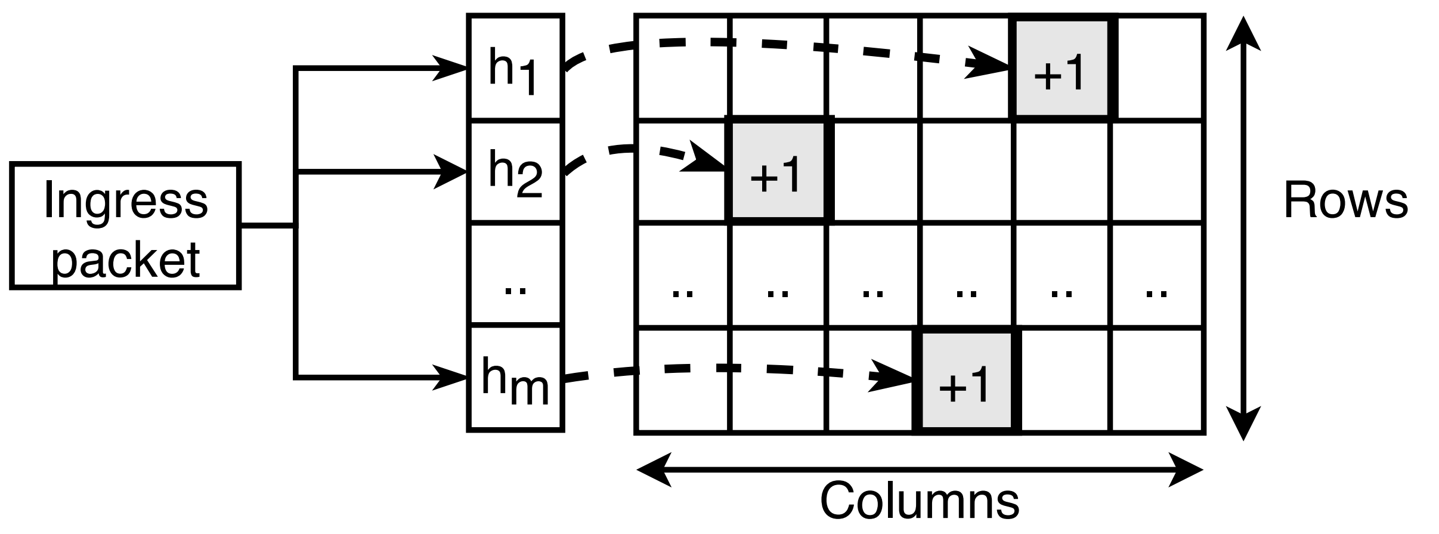
- It allows to customize how packets are processed by individual switches.
- The programmability supported by new network equipment is increasing.
- As well as primitives and resources available in each switch.
- *E.g. PISA and P4 lang in which network functions can be decomposed as programmable sets of match action stage.*



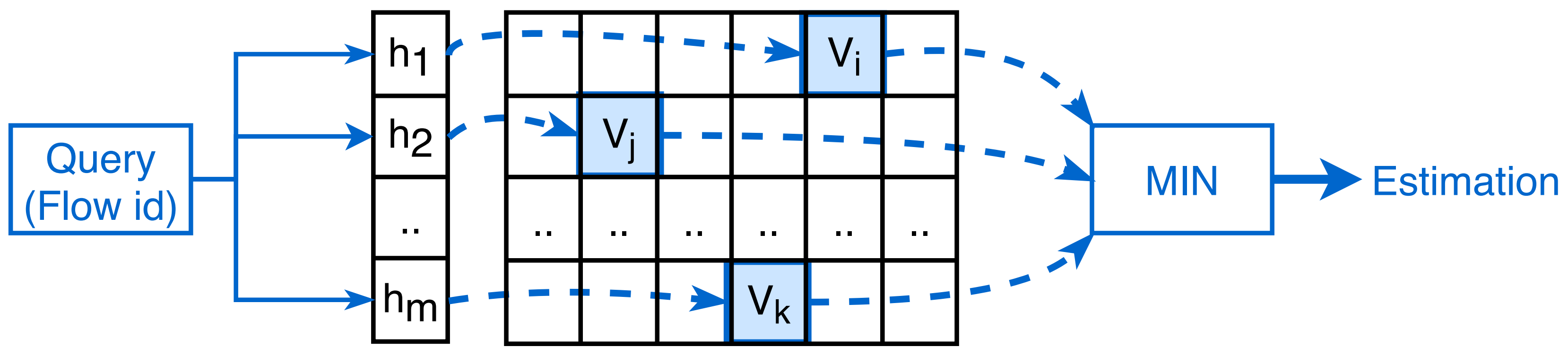
2. Performing network monitoring without starving all switch resources

PROBLEM: Leverage programmability to improve **data plane visibility**
(as the ability to collect packet telemetry in detecting heavy flows, microbursts or changes in traffic patterns.)

TYPICAL SOLUTION: approximate data structures (e.g., **sketches**)



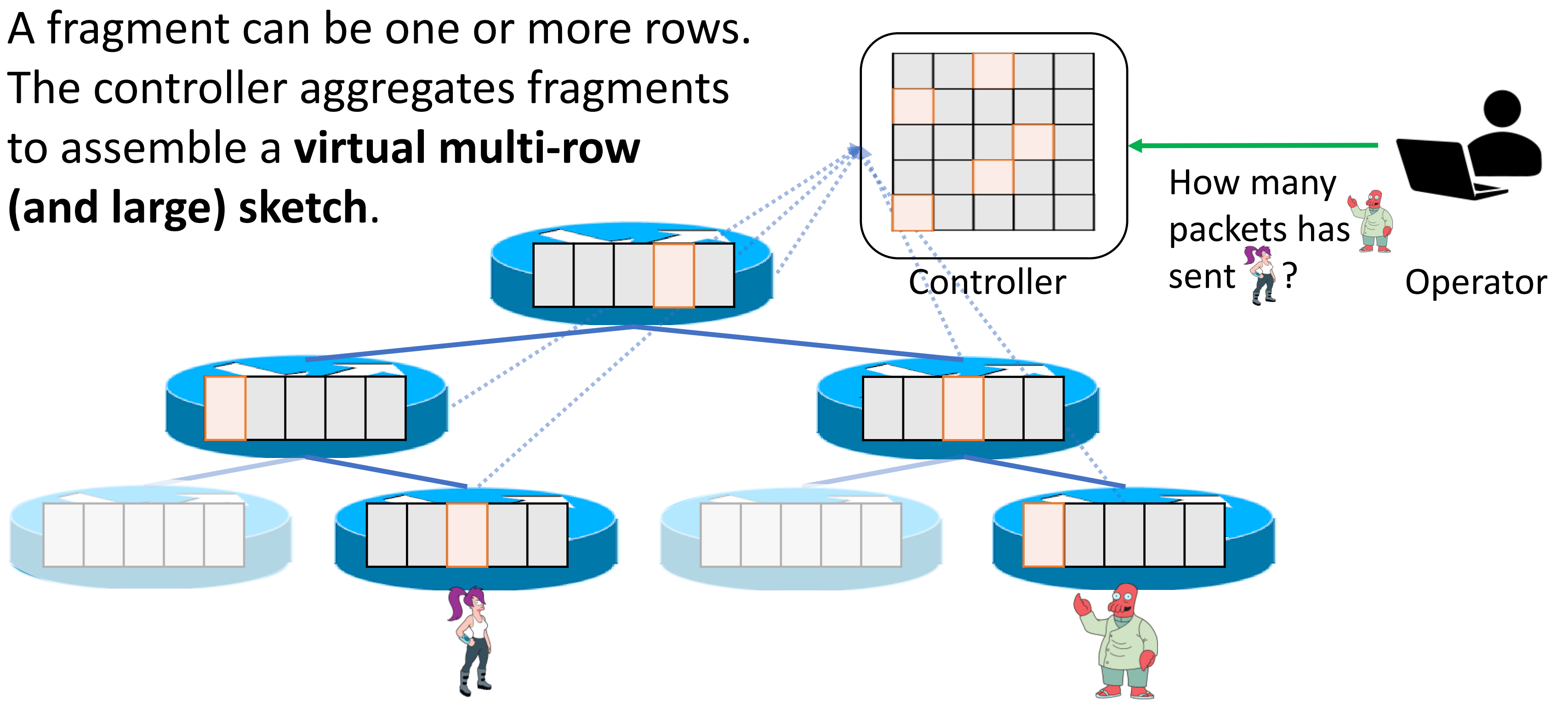
- Interesting trade-off between efficiency and accuracy.
- In practice, it is **hard** to fit sketches **for high accuracy** in every switch.



OUR AIM: a system that **uses a small amount of resources at switches** and does **not compromise accuracy.**

3. The DISaggreggate and COordninate (DISCO) sketch framework

- DISCO stores a **sketch fragment** on each switch.
- A fragment can be one or more rows.
- The controller aggregates fragments to assemble a **virtual multi-row (and large) sketch**.



The resource requirements at each switch **are significantly reduced.**

Different switches process **different traffic (!)**

DISCO can track various types of heavy hitters:
Assuming a controller with full network visibility (e.g., routing, topology)

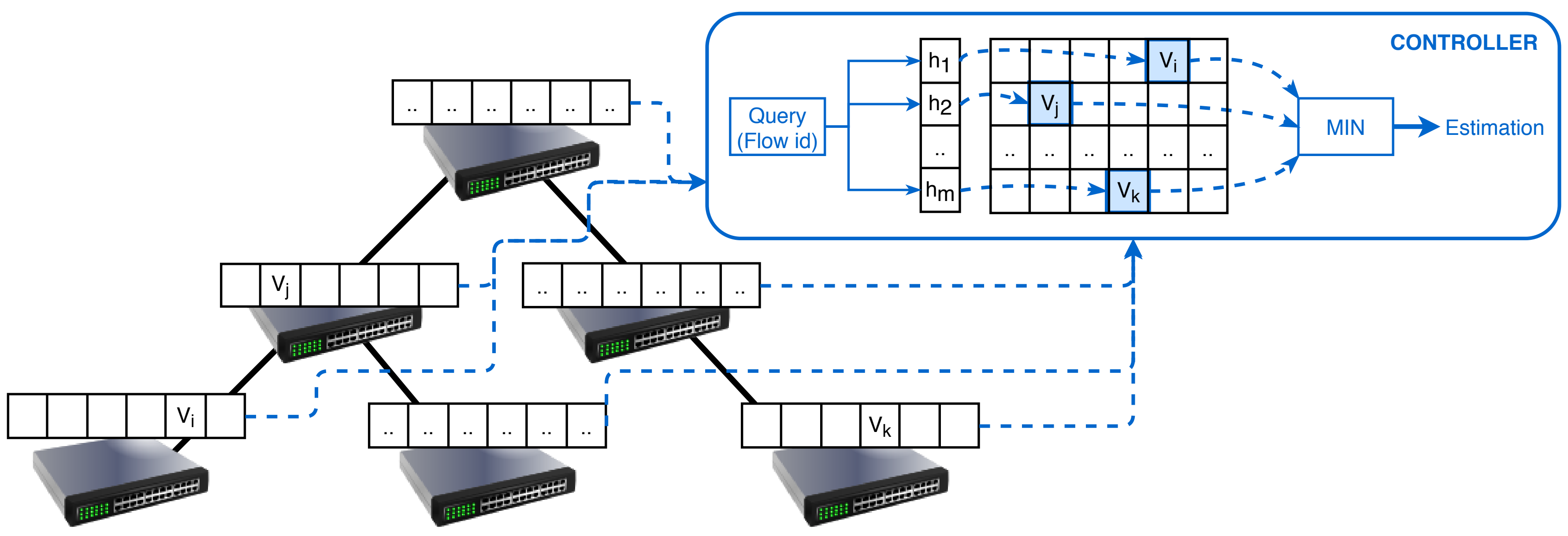
1. **Path-based Heavy Hitters.**

DISCO can detect heavy hitters on a given path efficiently.

2. **Network-wide Heavy Hitters.**

DISCO offers a "one big switch" abstraction with multiple disjoint Origin-Destination (OD)-paths.

5. DISCO supports disaggregating common design such as CMS and CS

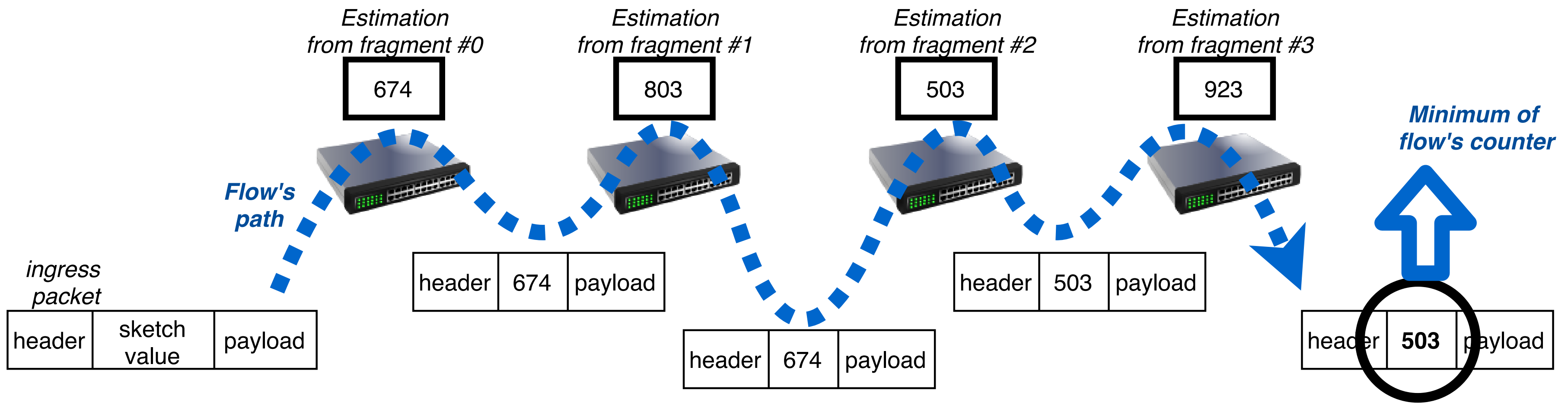


the minimum (or median) is taken among the sketch rows of the virtual sketch.

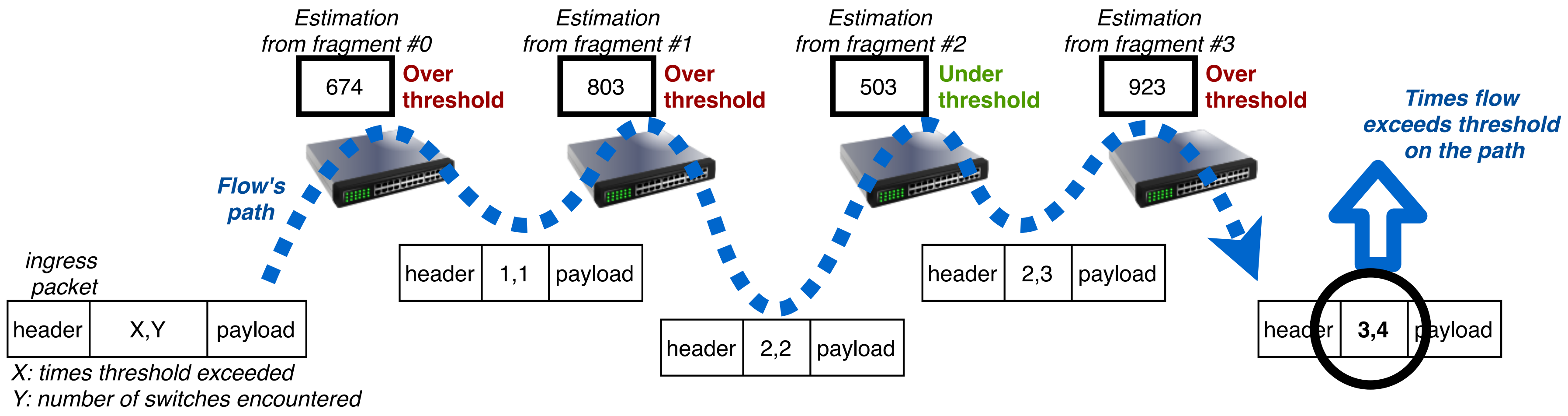
To notice:
60k counters ~ 2 sec latency
< 10k counters ~ < 0.5 sec latency

6. DISCO also supports finding the IDs of the Heavy Hitters on the fly

- **Count Min Sketch (CMS):** DISCO would carry on packets the minimal estimate seen. The last hop would report the ID to the controller if it exceeded a pre-determined threshold.



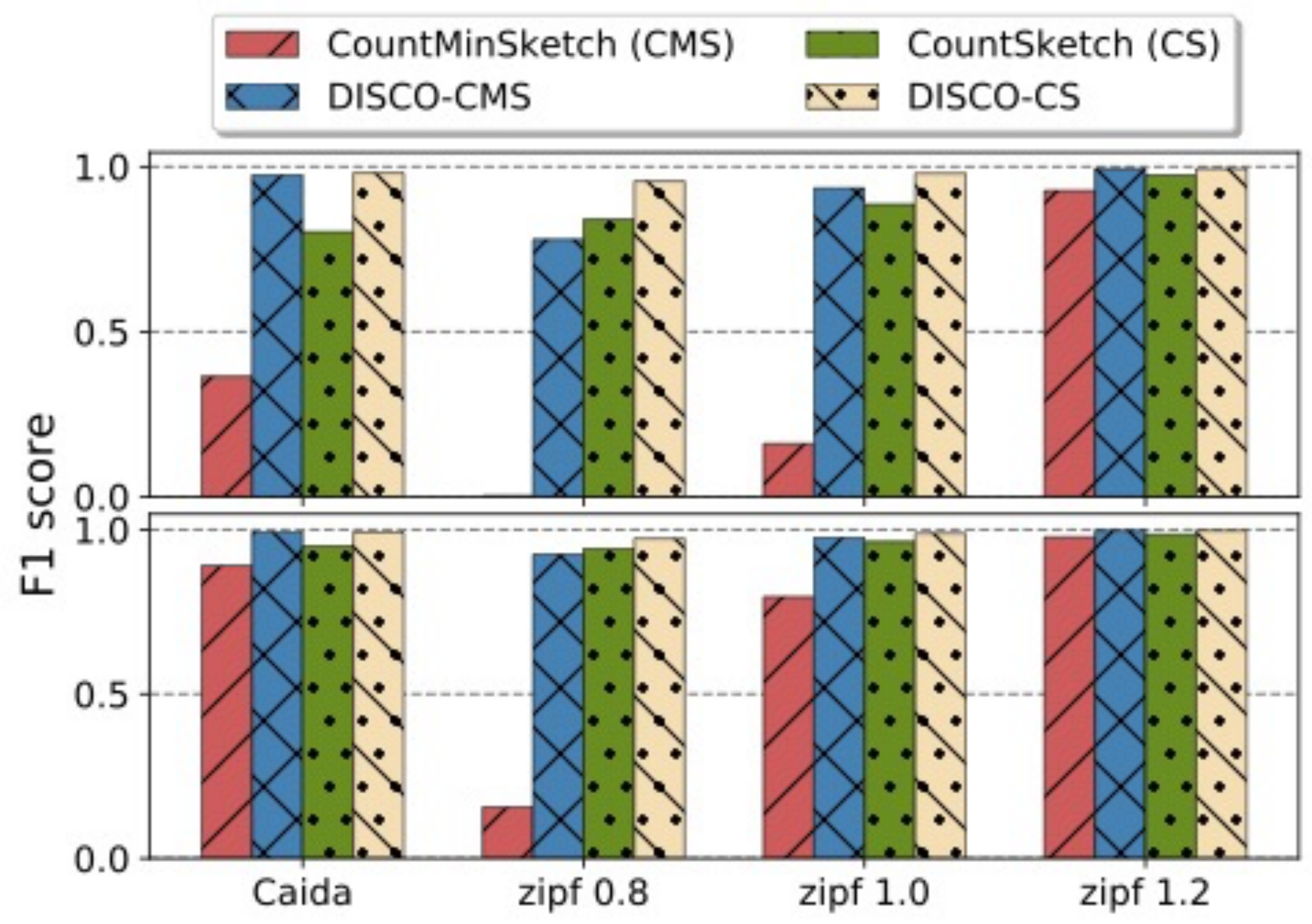
- **CountSketch (CS):** DISCO would carry on packets the number of counters on the path that exceed the threshold. The last hop would report the ID to the controller if at least half the counters exceed the threshold.



X: times threshold exceeded
Y: number of switches encountered

7. Experiments (preliminary insights)

- We find that disaggregation **positively affects accuracy**,
- while keeping a **lightweight footprint on per-switch resources.**
- DISCO achieves **32×** (with CMS) and **1.1×** (with CS) higher F1 score than one large sketch while reducing the per-switch resource footprint by **2.5×.**
- The topology simulated reproduces a k = 8 fat-tree.
- F1 score ($F1 = TP/(TP + 0.5(FP + FN))$)



CMS,CS:
5 rows and 8192 columns

DISCO-CMS,DISCO-CS:
1 row of 32768 columns in each switch.

CMS,CS:
5 rows and 16384 columns

DISCO-CMS,DISCO-CS:
1 row of 65536 columns in each switch.