The Network and Distributed System Security Symposium (NDSS) 2023

A Robust Counting Sketch for Data Plane Intrusion Detection

Sian Kim[1†], Changhun Jung[1†], Rhongho Jang[3✤]
David Mohaisen[2], DaeHun Nyang[1✤]

1 Ewha Womans University
2 University of Central Florida
3 Wayne State University

† These two authors contributed equally.
✤ Corresponding authors.

CONTENTS

CHAPTER I

Background and Challenges

Intrusion Detection in Networks

- Network traffic measurement: **per-flow statistics** are essential
- Gateway approach relies on **NFV** for scalability

  ※ **[Issue]** High operational cost
- In-network computing (INC) approach with **programmable switches**

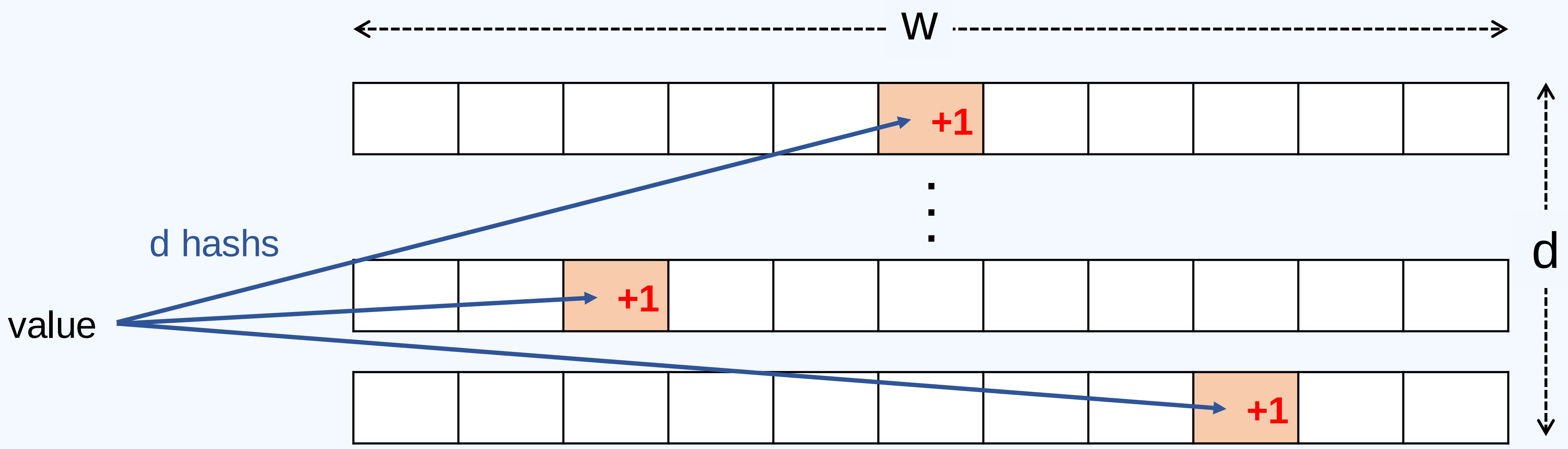
  ※ **[Emerging]** Advantages: High-speed, high flexibility, low cost

  ※ Three ways for **per-flow measurement**:

(1) hardware-based, **(2)** sampling-based, and **(3)** sketch-based approaches

Sketch-based Approaches

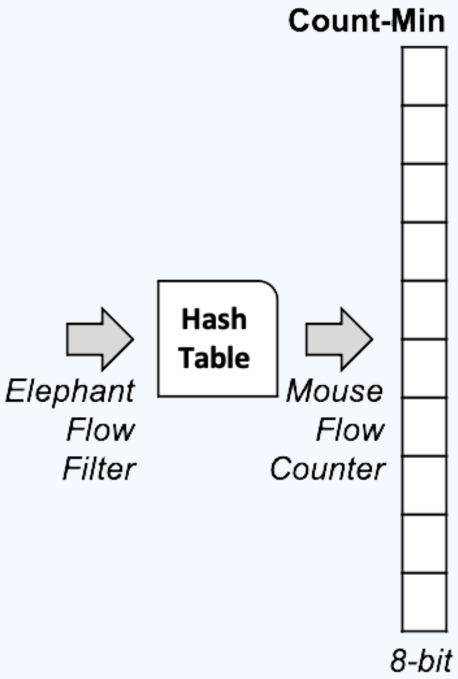
- A **compact** data structure to count a large amount of data
- **Good** estimation accuracy <u>under computation and memory constraints</u>
- Ex. Count-Min Sketch, Elastic Sketch, FCM Sketch etc.



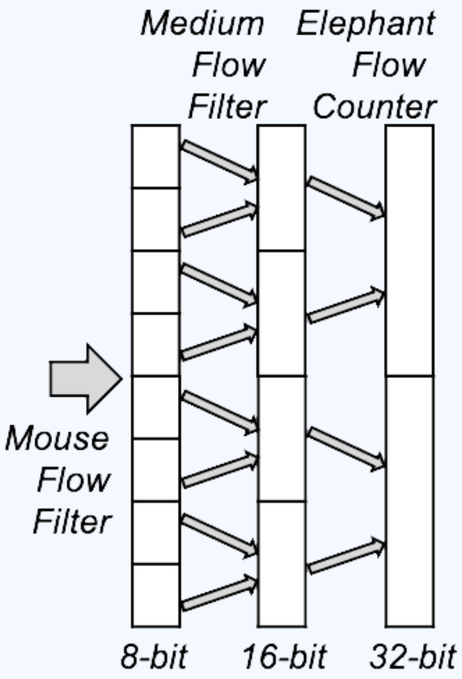
Count-Min Sketch

Advanced: Cascaded Multi-stage Filtering

- Data structure design to adapt to **Zipfian distribution**
- **[Core Idea]** Cascade multiple sketches for a sequential flow filtering according to their sizes



Elastic sketch



FCM sketch (Pyramid Shape)

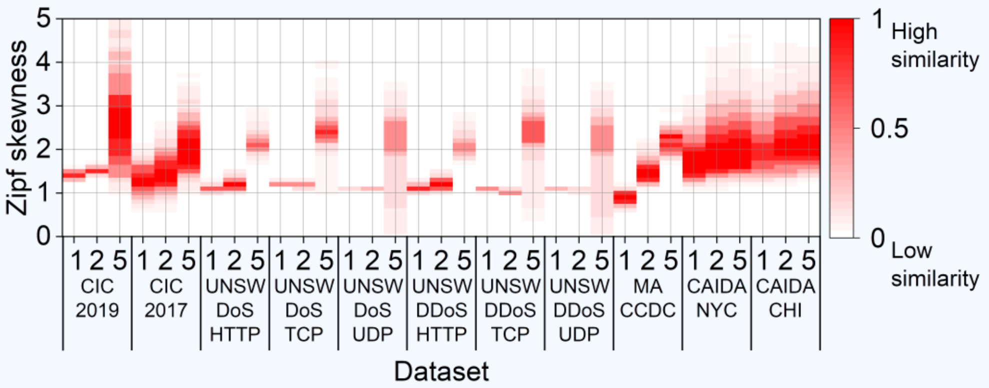
Problem Definition: Traffic Pattern Changes

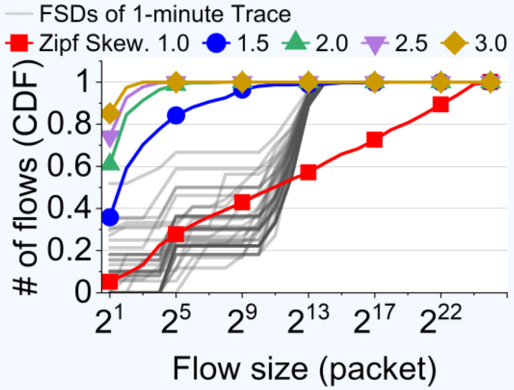
Flow Size Distribution (FSD)

Observation 1. FSD of attack and benign traffic are different

Observation 2. FSD varies depending on the flow definition

Observation 3. FSD also changes over time





Challenge1: Data structure must be robust to various traffic patterns

Naïve Approach to Adapting to Changing FSD

Reconfiguration of the sketch's data structure

- Online reconfiguration based on a real-time measured FSD

  → Dynamic data structure (e.g., real-time merge of counters)

  ※ **[Issue]** Infeasible for programmable switch

- Offline reconfiguration based on the FSD periodically

  → Updating the shape of data structure

  ※ **[Issue]** requiring recompile and reload of the entire program

Challenge2: How to adapt to various FSD without reconfiguring data plane switch?

CHAPTER Ⅱ

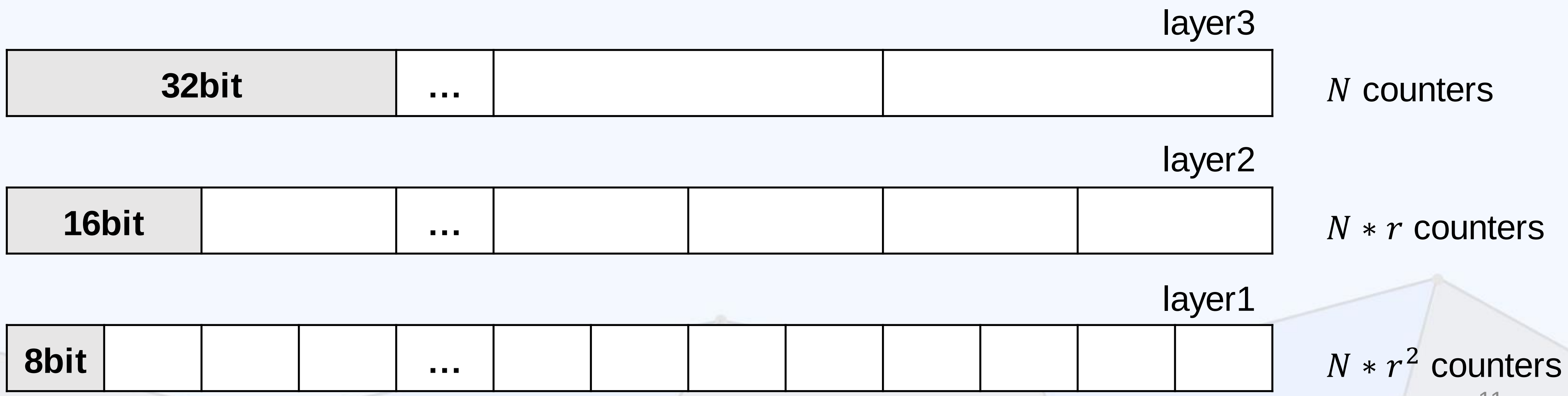
Contributions of Count-Less

Our Contribution: Count-Less (CL)

- A **robust and accurate** network flow measurement tool
  - **(1)** under both attack and benign traffic scenarios
  - **(2)** without dynamic adjustment of data structure.
- A novel **encoding algorithm** called *Minimum Update* is designed
  - **flexible** encoding strategy to maximize memory efficiency
- **Theoretical proof of the error bound**
- **Verified robustness** with security applications
- Data plane implementation supports **in-network flow measurement**

Data Structure of Count-Less

- CL consists of **d layers** of counter arrays
- Top layer uses 32-bit counters for large flows
- Reducing counters' size while going down to the bottom layer
- Number of counter per layer with factor $\boldsymbol{r}$
  - A lower layer array possess $\boldsymbol{r}$ times more counters than its upper layer



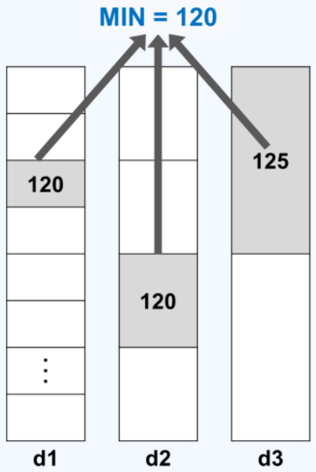
Encoding Algorithm of Count-Less

Conservative Update (optimal version)

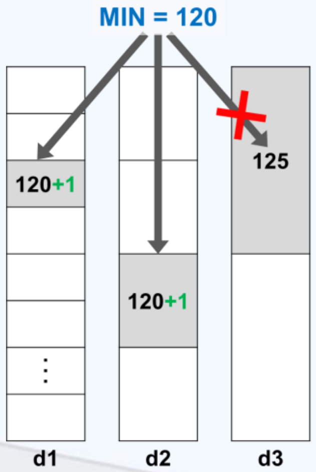
- Find a global minimum value among all layers (left figure)
- Update the counters that contain the minimum value (right figure)

Data Plane Issue

- Conservative Update triggers double-access to the same register, which is **restricted by programmable switch**.



① Find **MIN**

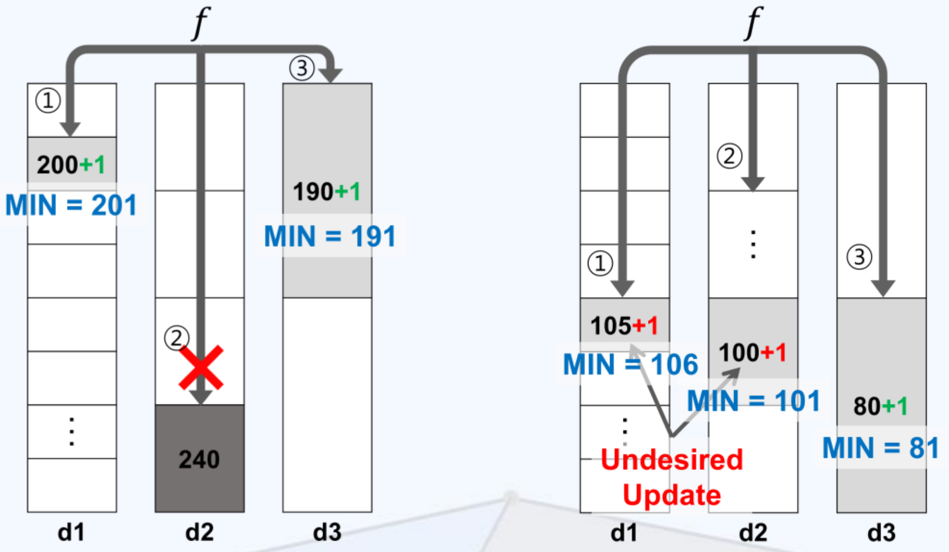


② Update

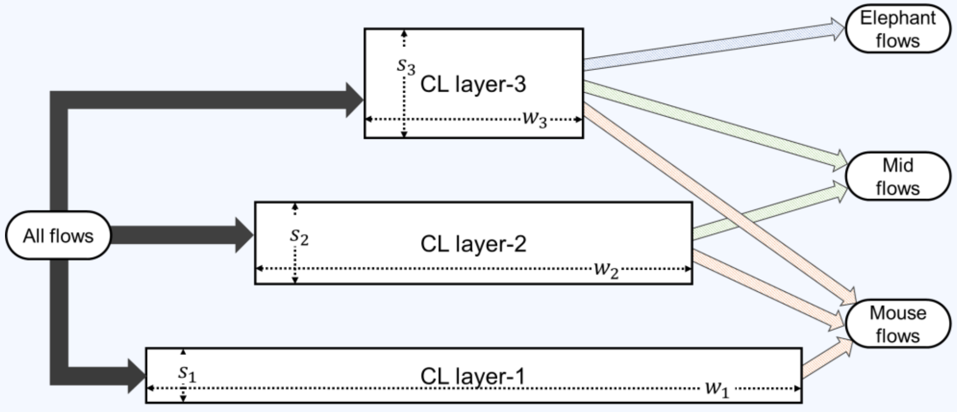
Our Solution: Encoding with Approximation

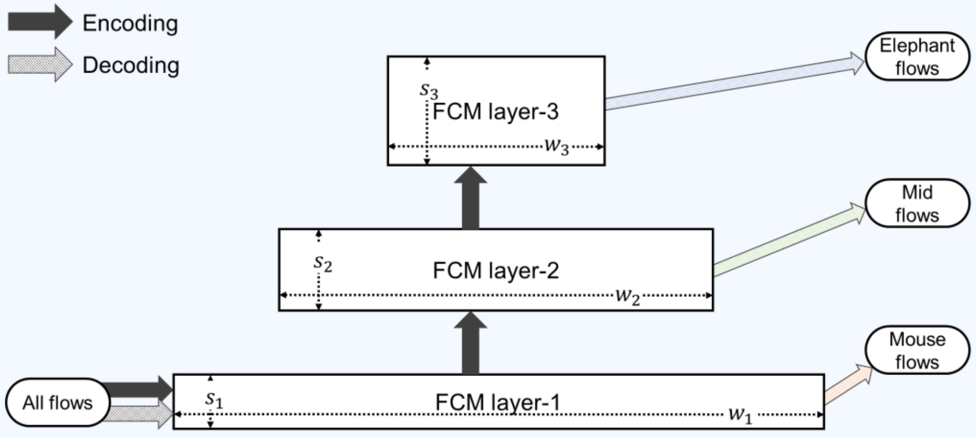
Minimum Update (approximate version)

- Update occurs with a sequential order, from the lowest to highest layer
- It stores the temporal minimum value (MIN) during the process
- Update happens only when its value is smaller than the current MIN



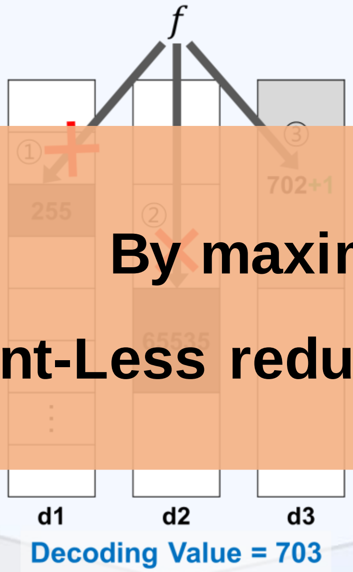
Comparing Encoding Algorithms

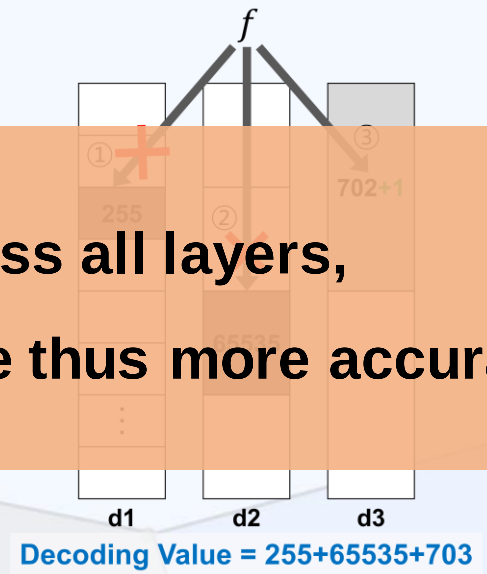




Minimum Update (Count-Less)

Cascading approach (FCM Sketch)



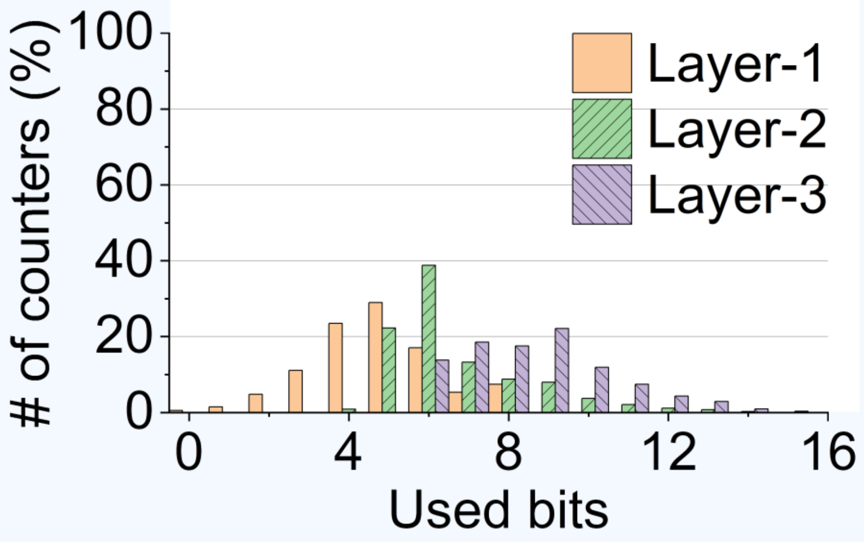


By maximizing counter usage across all layers,

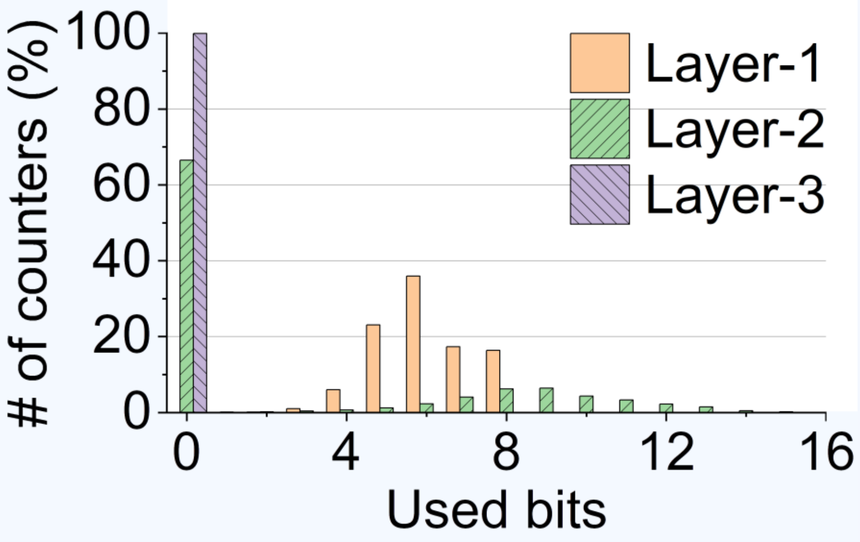
Count-Less reduces the hash collision rate thus more accurate

Advant. 1. Count-Less improves memory efficiency

Dataset Description: benign one-minute CAIDA dataset



CL-MU Sketch

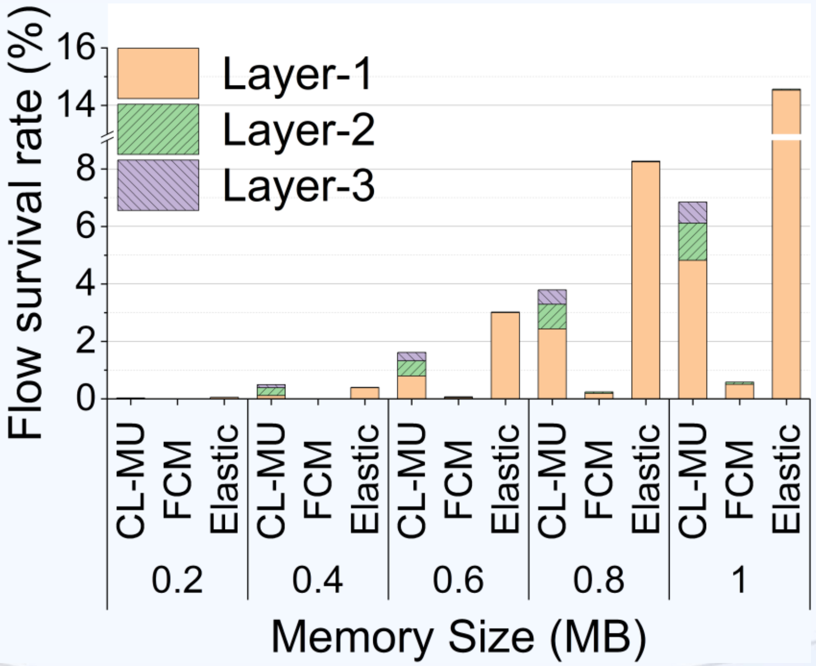


FCM Sketch

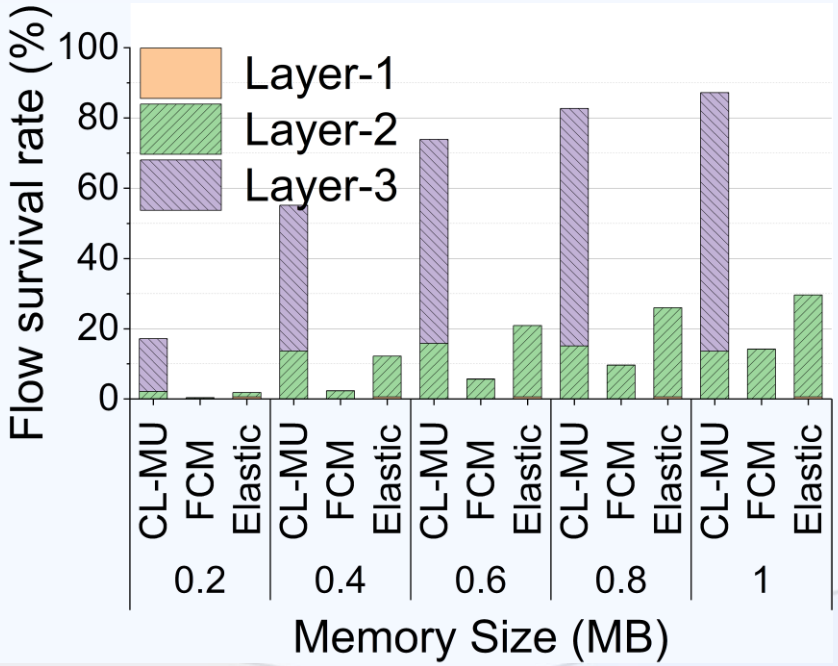
Advant. 2. More flows survives

Flow Survival Rate: fraction of flows that are below a certain relative error after decoding

- FSR for Mouse Flow (≤ 255): say survive if the estimated relative error is below 0.1
- FSR for Elephant Flow (>255): say survive if the estimated relative error is below 0.01



Mouse Flow



Elephant Flow

CHAPTER Ⅲ

Analysis and Evaluations

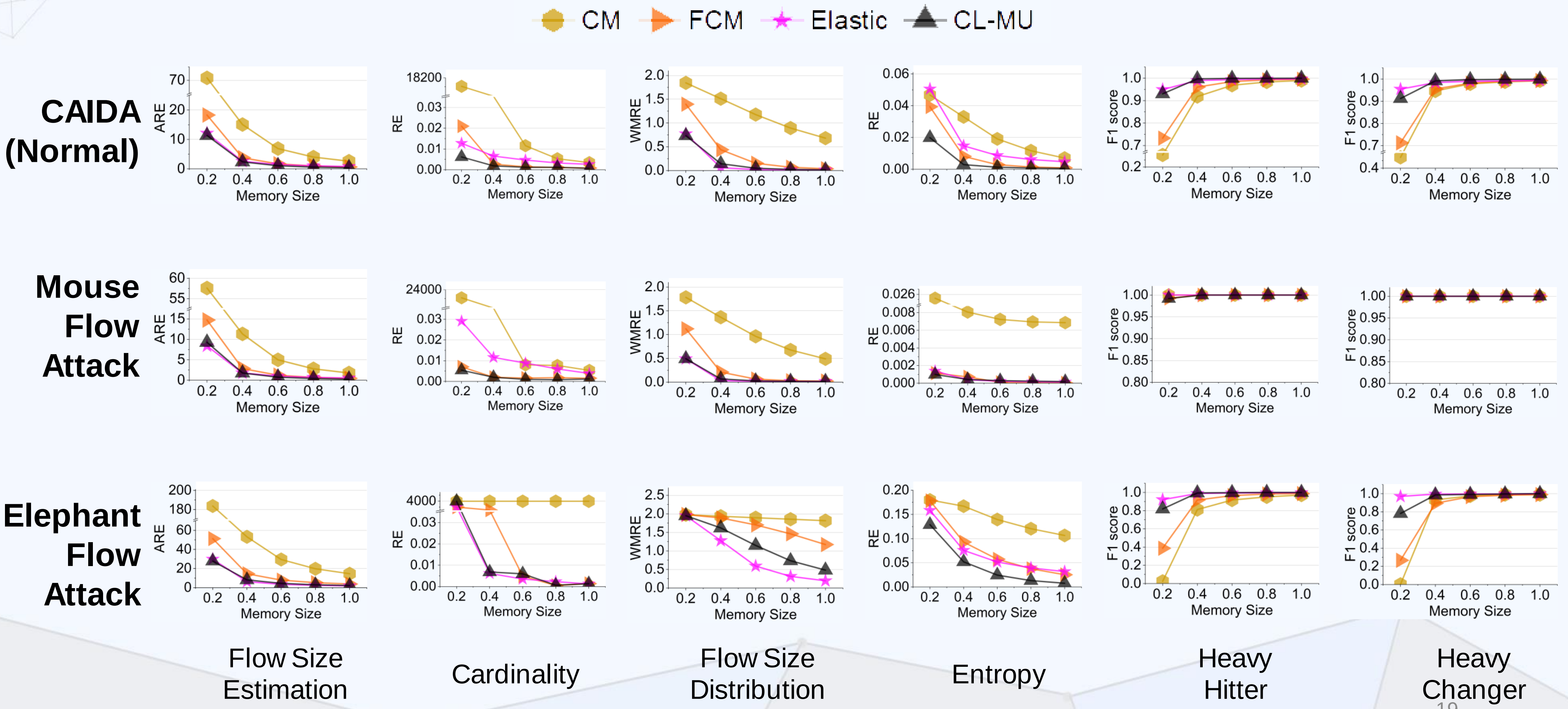
Robustness of Count-Less

Note: Count-Less achieves comparable performance with **Elastic sketch** in large flow-heavy trace (skewness 1.0 and 1.2)**, even though Elastic uses dedicated hardware for large flows.**

		Skewness							
		1.0	**1.2**	**1.4**	**1.6**	**1.8**	**2.0**	**2.2**	**2.4**
Sketch	**CL-MU**	0.01	1.48	3.52	15.14	47.97	102.38	151.41	184.69
	FCM(k=4)	0.04	1.08	7.38	25.45	83.56	232.92	605.06	851.19
	Elastic	0.00	0.13	4.27	19.89	78.17	163.80	198.40	208.86

Average Relative Error (ARE) varying skewness of traffic's flow size distribution

Security Applications Varying Traces



Data Plane Overheads: Comparison

Resource Usage	CM	CL-MU	FCM
Hash Bit (%)	2.88	3.06	4.97
SRAM (%)	5.72	6.14	7.29
ALU (%)	4.16	6.25	16.67
Used stages	2	5	4

Latency (Normalized)	CM	CL-MU	FCM
Layer-1	0.02	0.12	0.09
Layer-2	0.02	0.64	0.75
Layer-3	0.05	0.24	0.91
Total	0.09	1.00	1.75

Data Plane Implementation: Hardware resource usage and added latency in the **programable switch**

CHAPTER IV
Conclusion

Conclusion

- **Flow size distribution changes** by many factors
- Count-Less with a novel **Minimum Update strategy**
  - It adapts to sudden changes in traffic patterns
  - It fits into the pipeline design of the data plane
- **Low latency** and **high throughput** in-network per-flow measurement
- Verified **high accuracy and robustness** through analysis and experiments

Thank you

Questions?

ksy60a@ewha.ac.kr