RevealNet: Distributed Traffic Correlation for Attack Attribution on Programmable Networks

Gurjot Singh, Alim Dhanani, Diogo Barradas

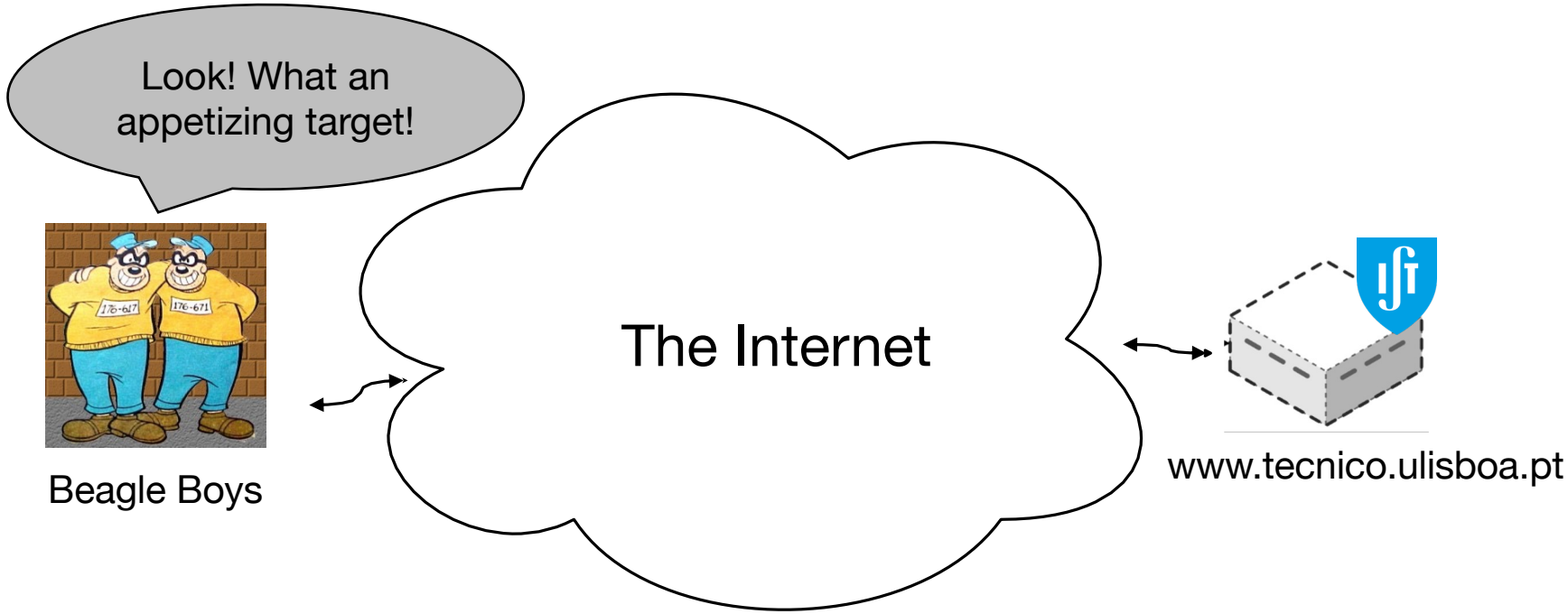


23rd IEEE International Symposium on Network Computing and Applications

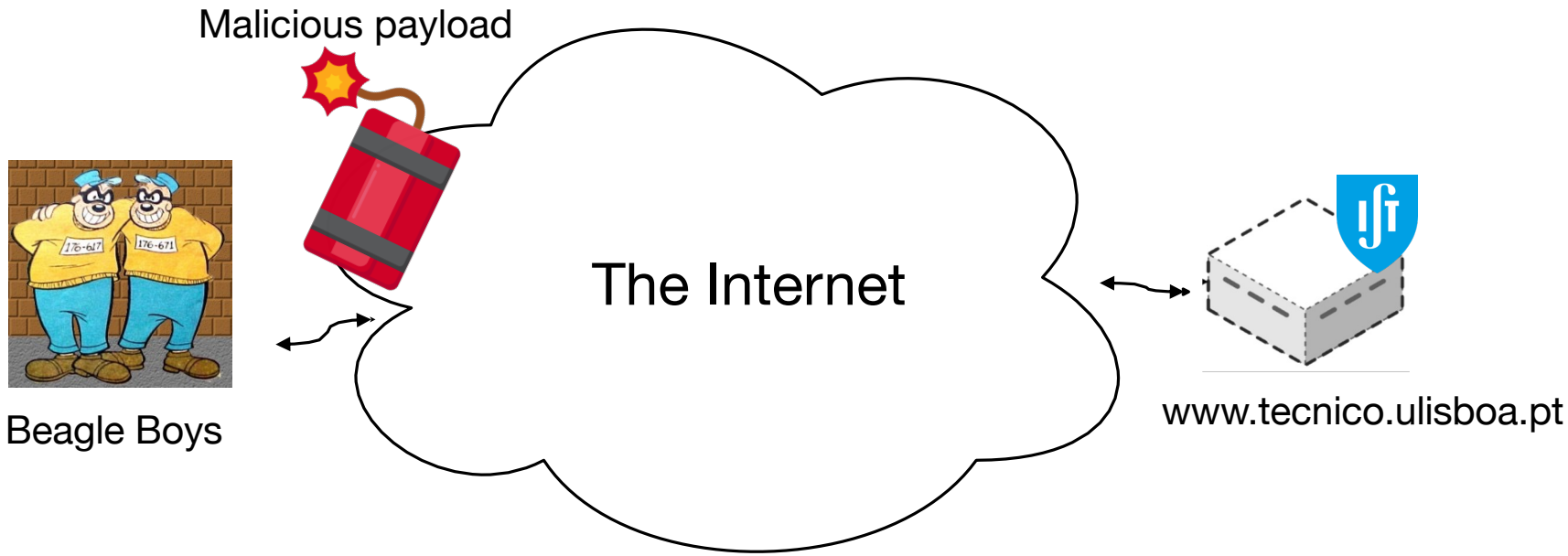
Lisbon, Portugal

November 6th, 2025

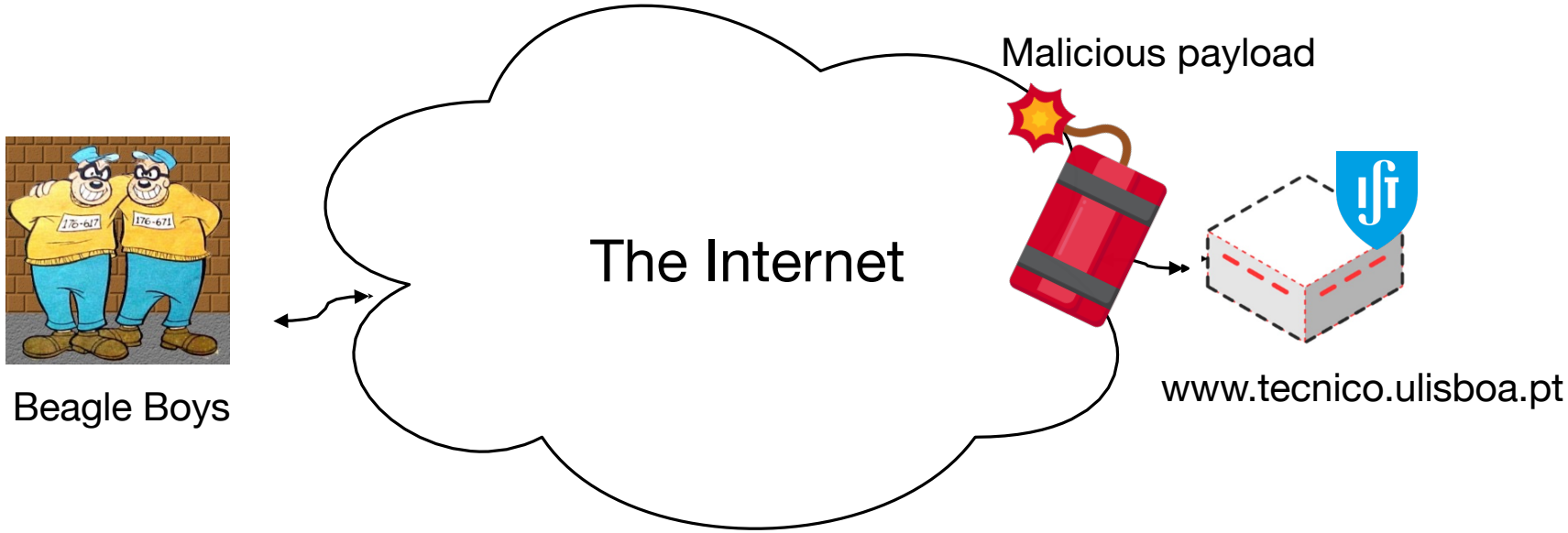
Cyberattacks and Attack Attribution 101



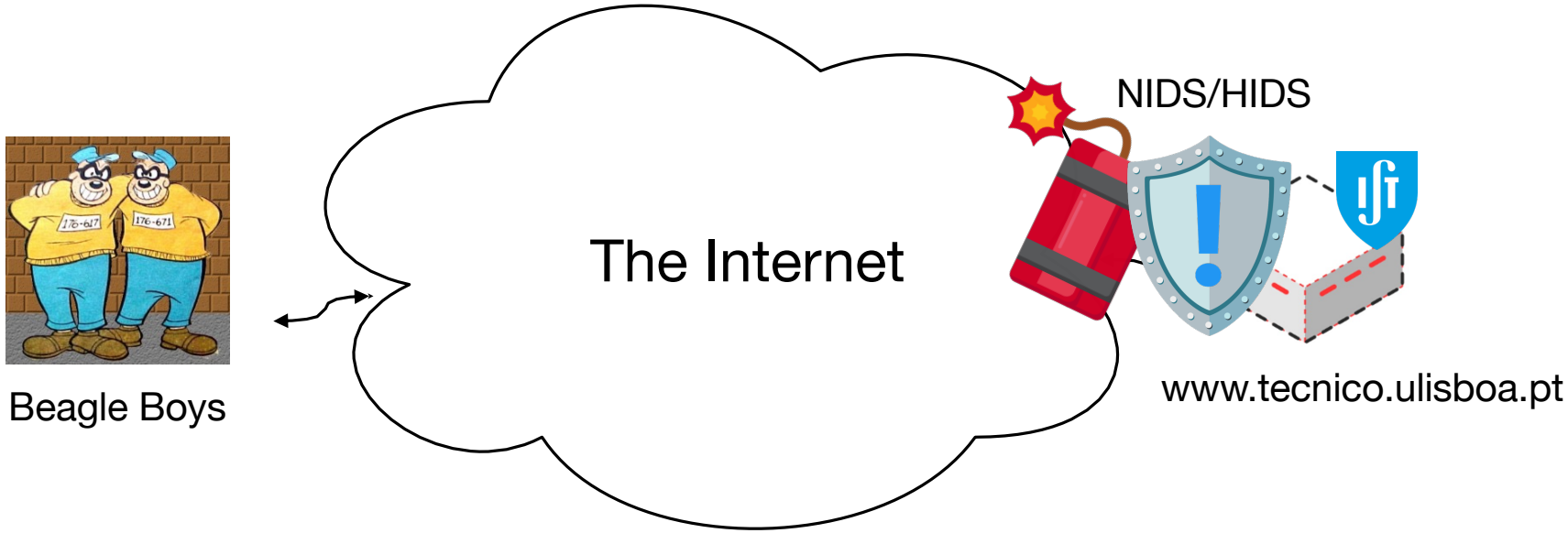
Cyberattacks and Attack Attribution 101



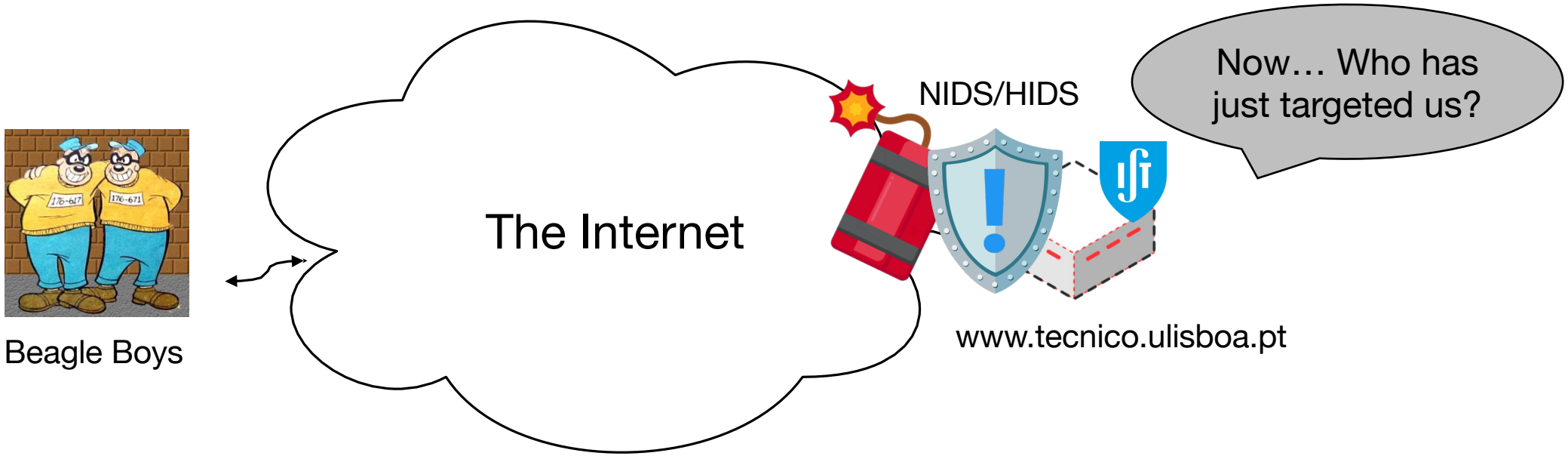
Cyberattacks and Attack Attribution 101



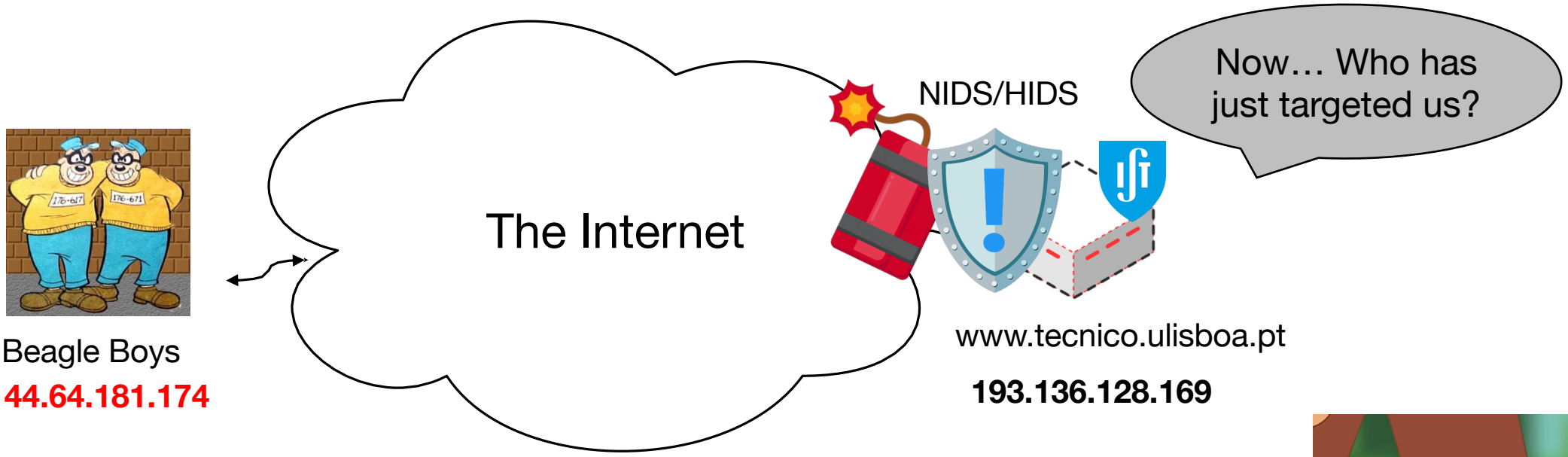
Cyberattacks and Attack Attribution 101



Cyberattacks and Attack Attribution 101



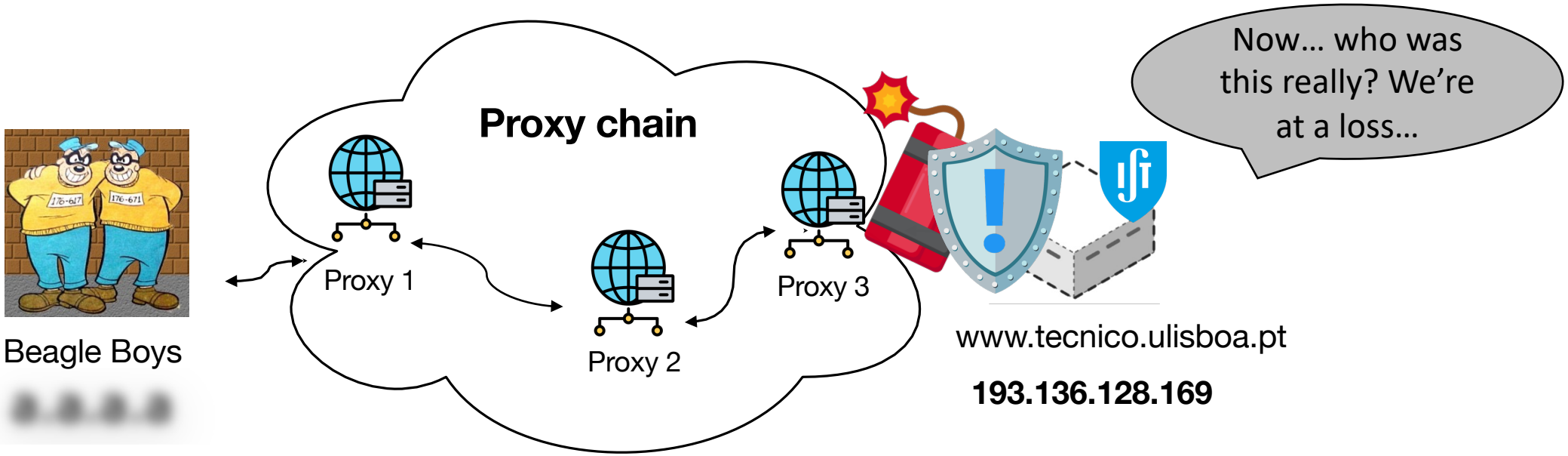
Cyberattacks and Attack Attribution 101



- Attack attribution is simple! Just check the offending flow's 5-tuple!
- But these were the early days…

Attackers commonly obscure their identity

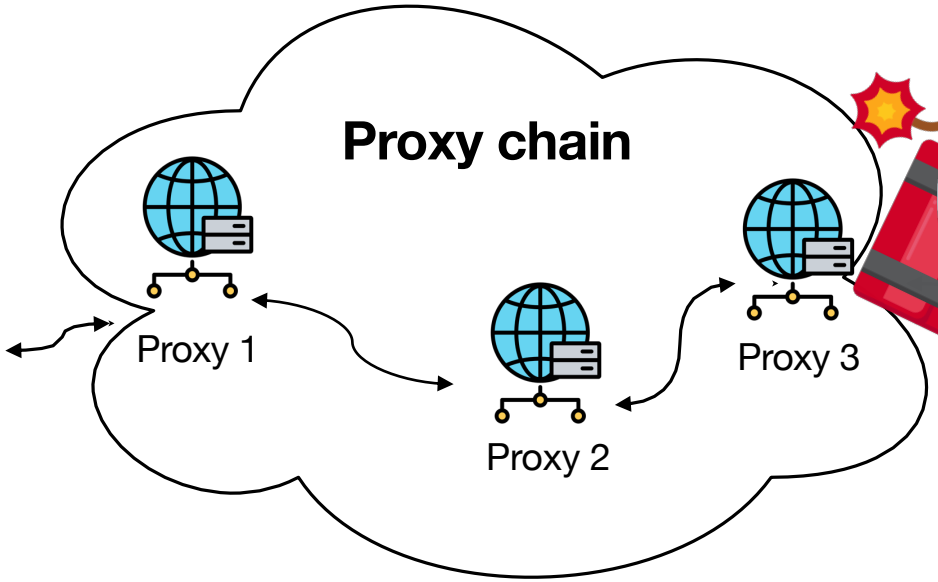
Stepping-stone attacks

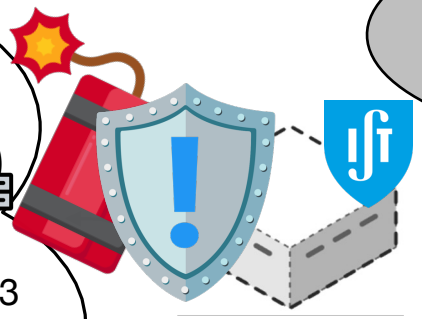


- Proxy chains, anonymity networks, VPN's provide online anonymity by sending users' network traffic through multiple relays

Traffic correlation for attack attribution

Beagle Boys





www.tecnico.ulisboa.pt

193.136.128.169

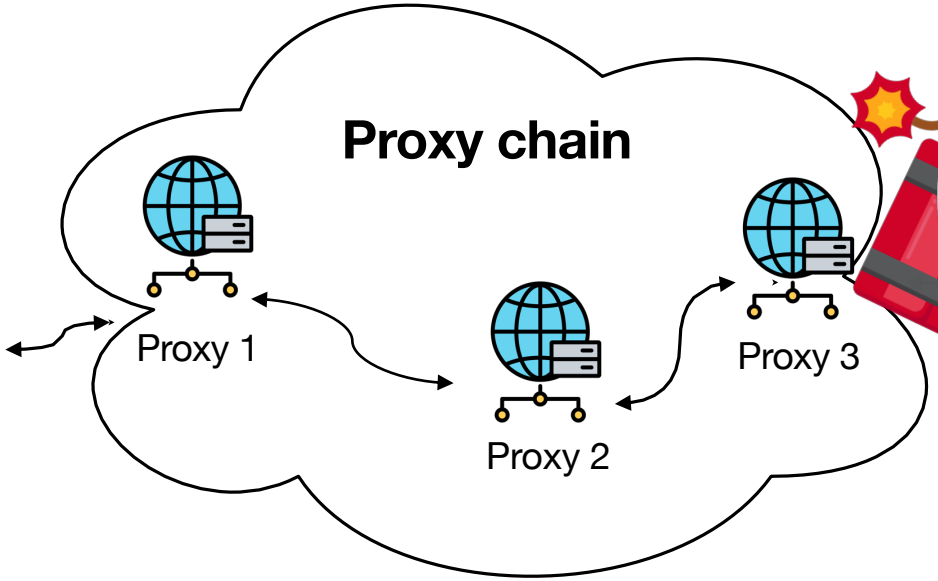
Hey ISPs, can you give me a hand?

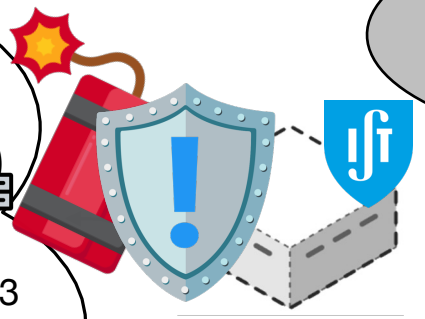
MEO

Sure thing!

Traffic correlation for attack attribution

Beagle Boys

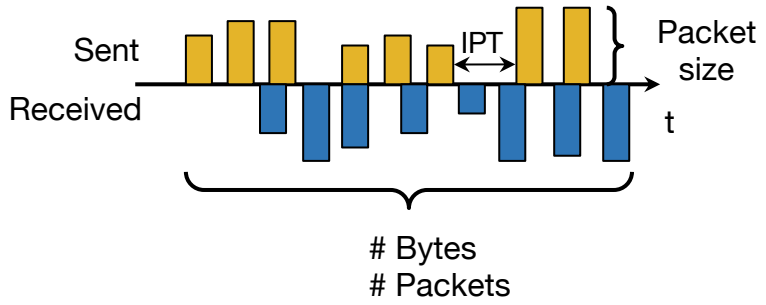




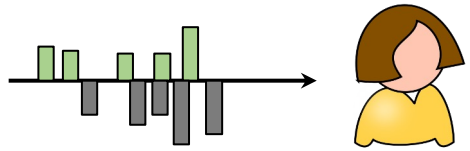
Here's what I saw…

www.tecnico.ulisboa.pt

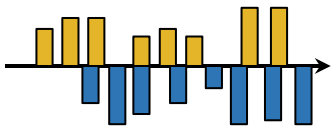
193.136.128.169



Traffic correlation for attack attribution



Alice



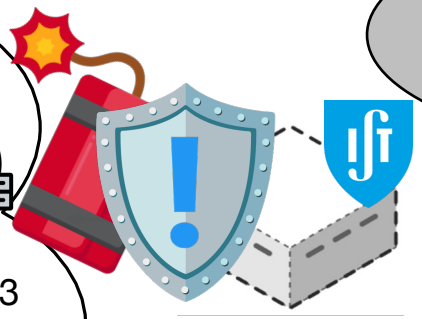
Beagle Boys

Proxy chain

Proxy 1

Proxy 2

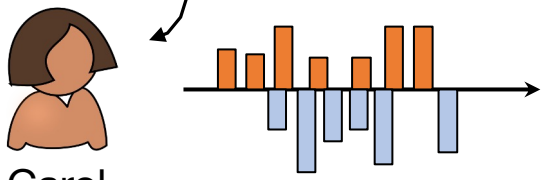
Proxy 3



Here's what I saw…

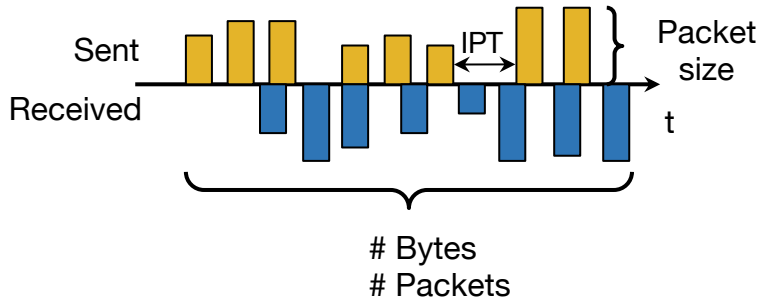
www.tecnico.ulisboa.pt

193.136.128.169

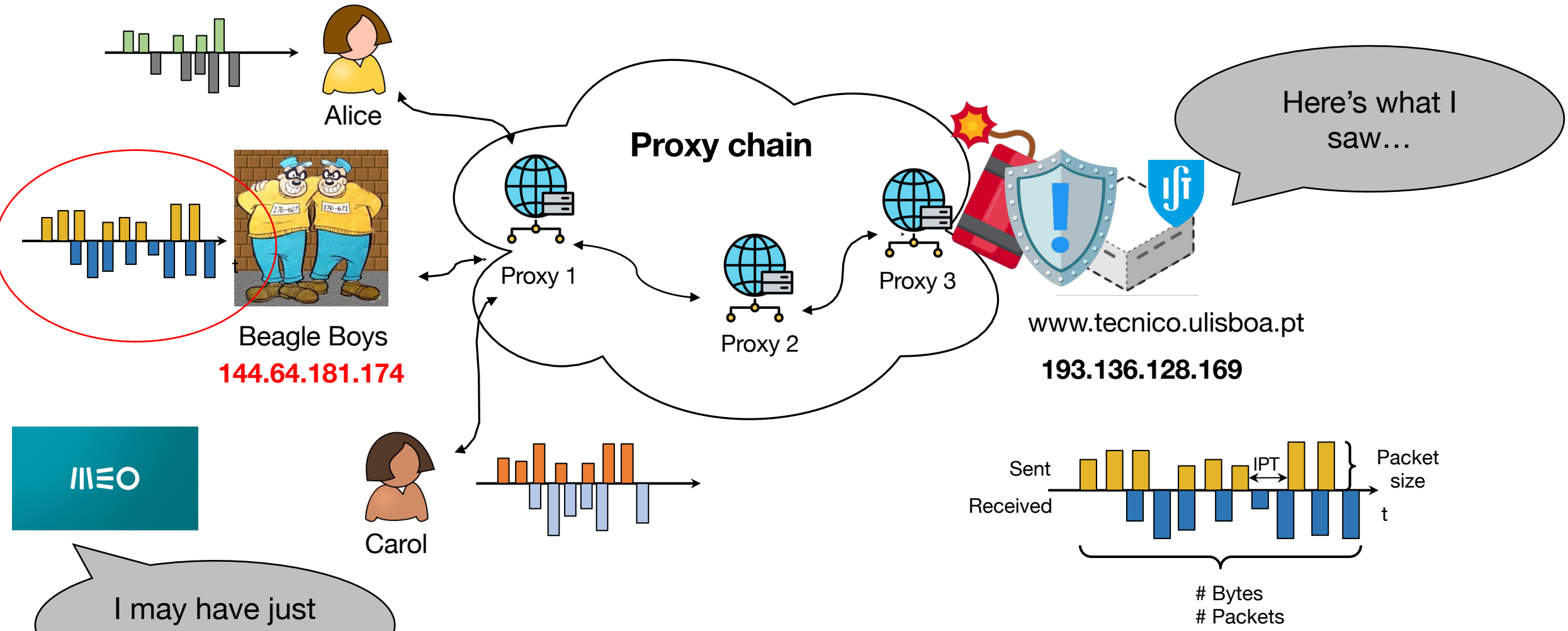


Carol

Let's see what we got here…



Traffic correlation for attack attribution



Formalizing the attack attribution problem

Network Model: The Internet is modeled as multiple interconnected networks (ISPs, CSPs, enterprises) that cooperate by sharing network flows' features.

Setup: The attacked network N_i compares its malicious flow f_m^i with candidate flows f_k^j from other networks. Comparison can be performed via similarity metrics, for instance.

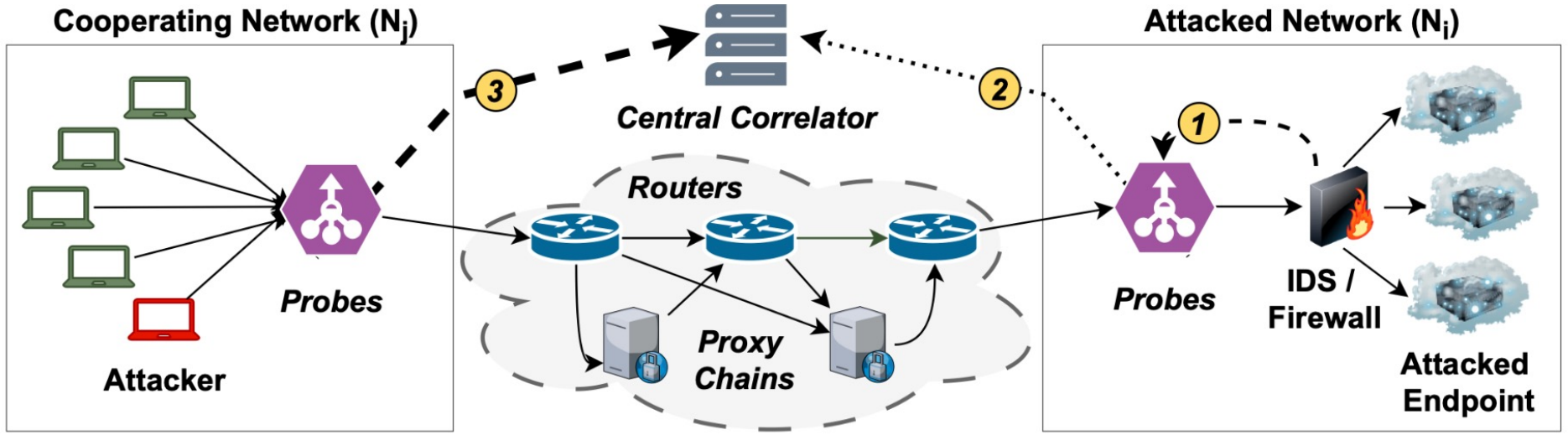
Goal: Guess the attacker IP $\hat{\mathcal{A}}$ by choosing the IP a that maximizes correlation with the malicious flow:

$$\hat{\mathcal{A}} = \arg\max_{a \in \mathcal{N}_{\text{IP}}} \sum_{N_j \neq N_i} \sum_{f_k^j \in \mathcal{F}(N_j, a)} \rho(f_m^i, f_k^j).$$

Constraint: Only account for strong matches: include $\rho\left(f_m^i, f_k^j\right)$ terms where $\rho \geq \eta$ (similarity threshold)

Challenge: attack attribution is centralized

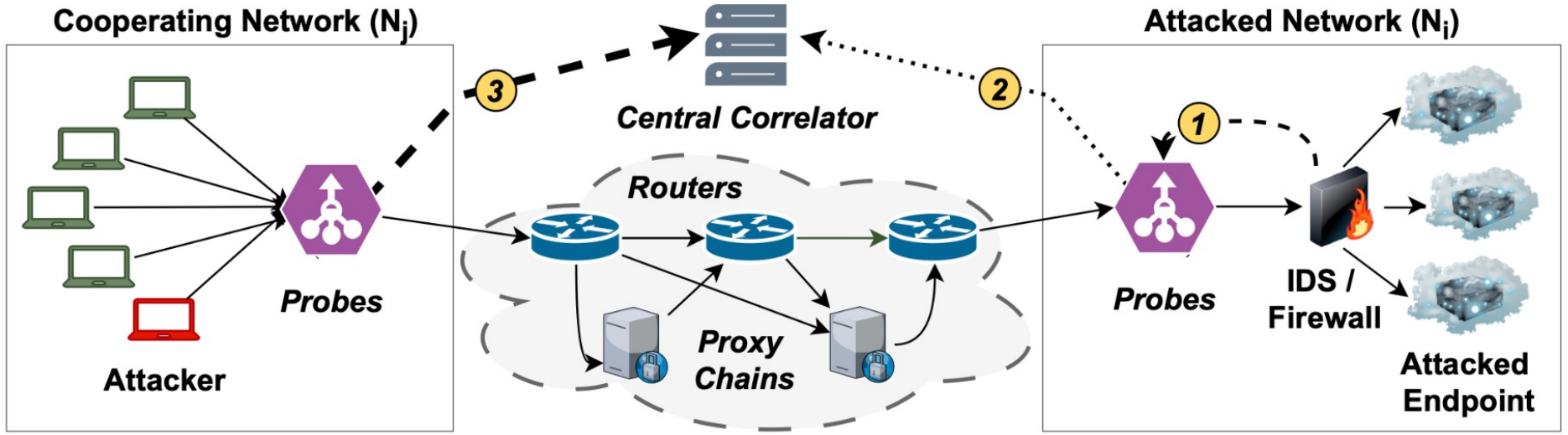
- Upon detection of an attack, cooperating networks send all of their **flows' features** to a **central correlation node** for matching. This incurs **severe overheads** on:
  - Computation – one server does all the processing
  - Bandwidth – probes must send out all potential matching flows
  - Storage – probes must record all observed flows



Challenge: attack attribution is centralized

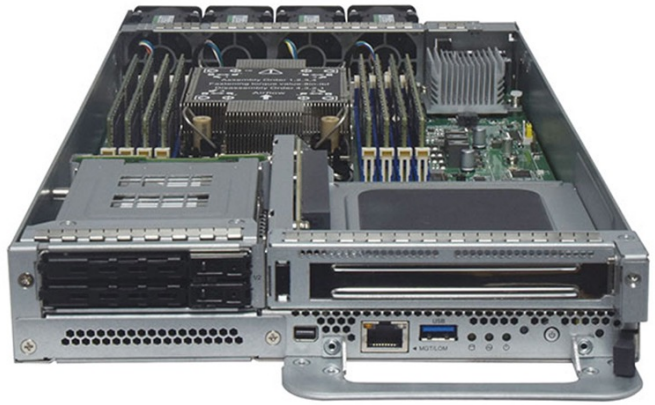
- **Optimizations?**
  - Computation **–> split candidate flows across correlation nodes**
    -> still requires special-purpose servers
  - Bandwidth **-> still need to exchange flow features in bulk**
  - Storage **–> compact data structures (e.g., flow sketches)**

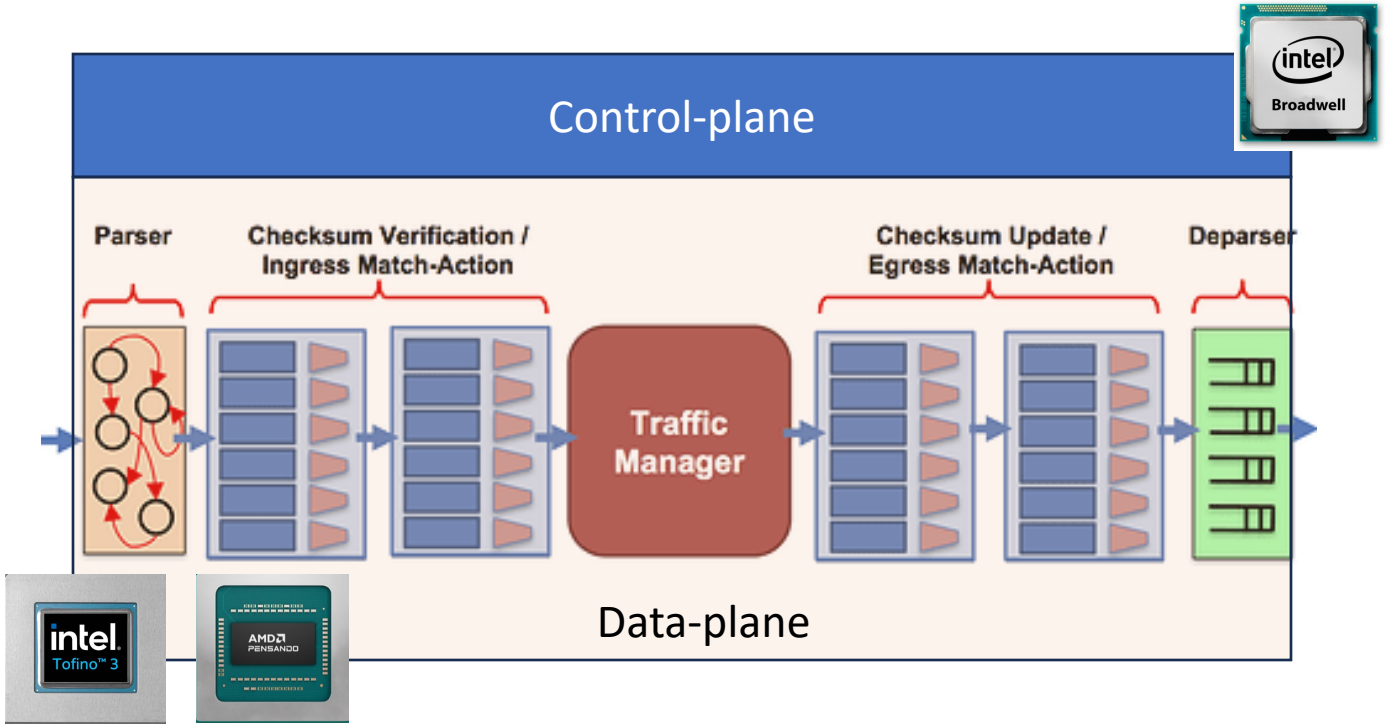
Oops?



Idea: P4 switches as distributed correlation engines

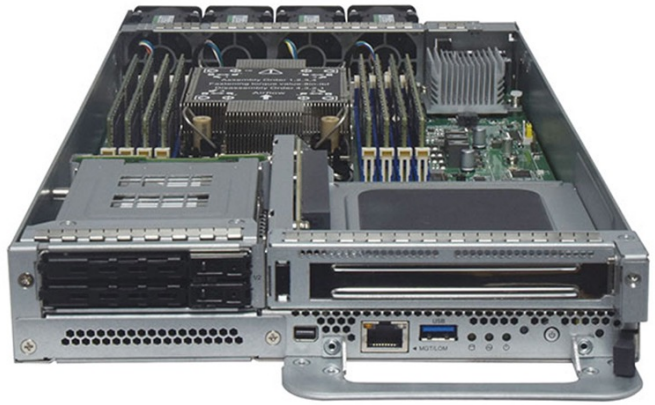
- Programmable switches support **custom packet-processing** behavior (e.g., via P4)
- Currently available in **edge networks**' border routers

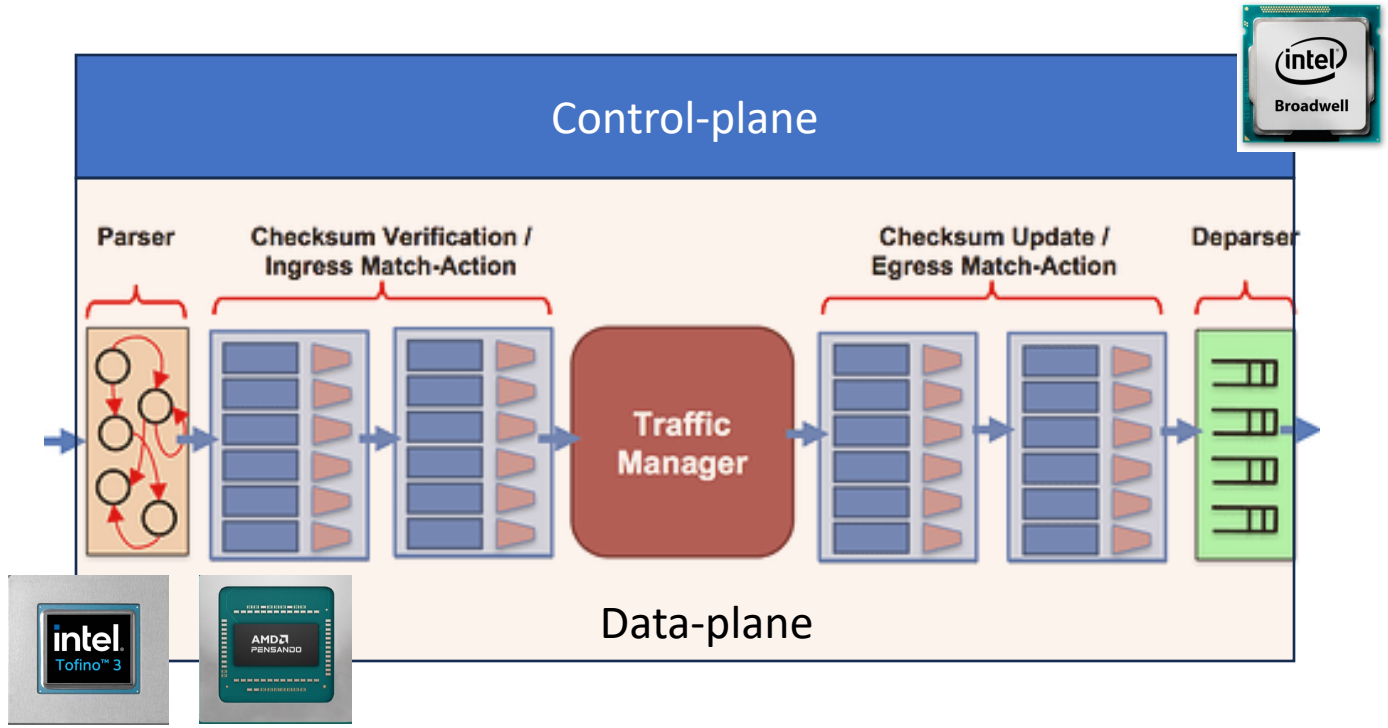




Idea: P4 switches as distributed correlation engines

- Programmable switches support **custom packet-processing** behavior (e.g., via P4)
- Currently available in **edge networks**' border routers





1. Compute flows' features **2. Compress flows' features** **3. Correlate in control plane**

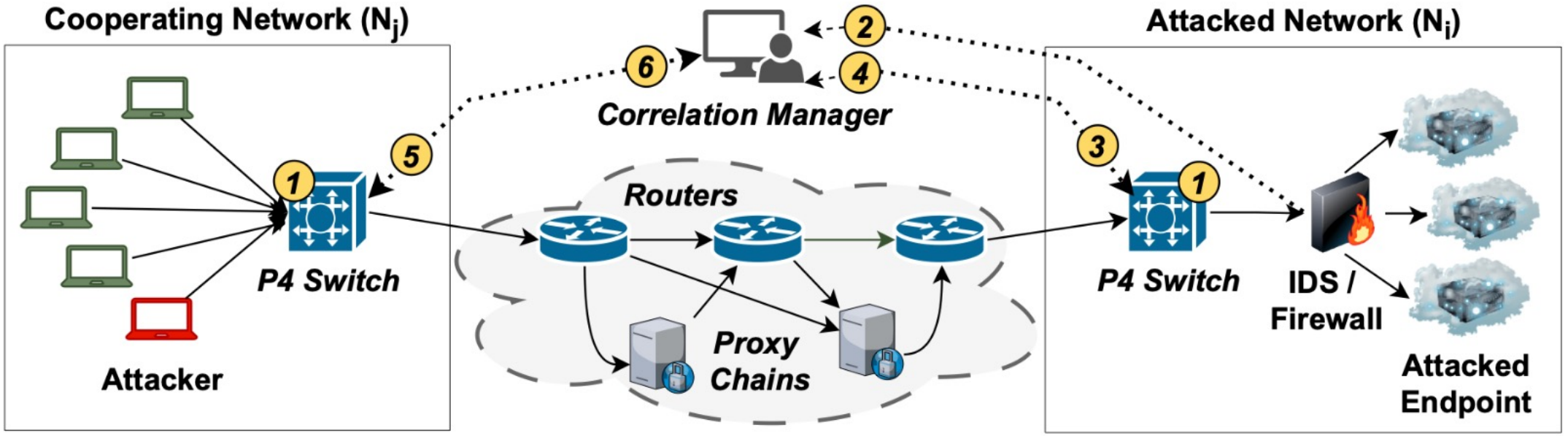
Contributions

- Introduce **RevealNet**, a *decentralized attack attribution framework* based on the orchestration of P4 programmable switches for enabling flow correlation

- Implemented **RevealNet** in `bmv2`, the reference P4 switch, together with existing *flow sketching schemes* for flow features' compression

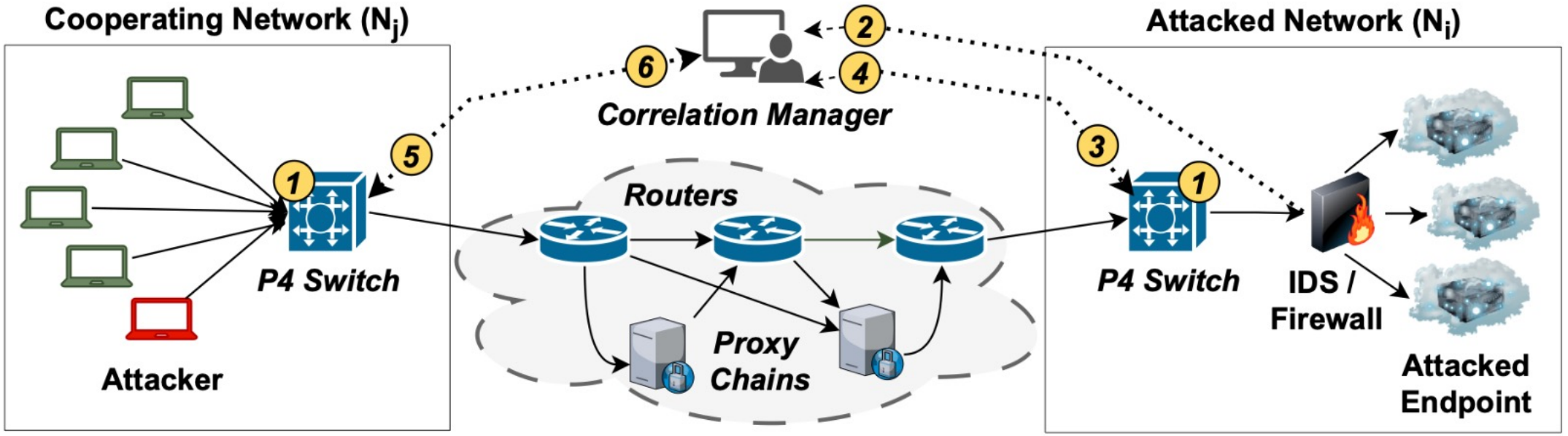
- Evaluated **RevealNet**'s *correlation accuracy* as well as its *computational and bandwidth overheads* when identifying the source of malicious flows

RevealNet's decentralized correlation pipeline



1. Flow feature extraction
2. Attack detection
3. Request for attacking flows
4. Propagation of attacking flows' features
5. Correlation directive
6. Flow correlation and reporting

RevealNet's decentralized correlation pipeline

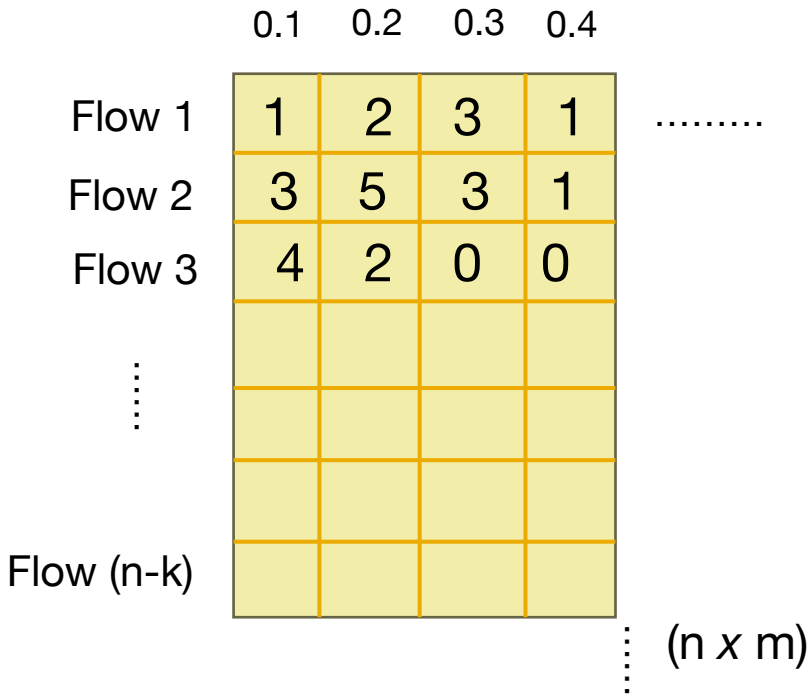


1. Flow feature extraction
2. Attack detection
3. Request for attacking flows
4. Propagation of attacking flows' features
5. Correlation directive
6. Flow correlation and reporting

1. Compute flows' features

Essentially, start by building a **Traffic Aggregation Matrix (TAM):**

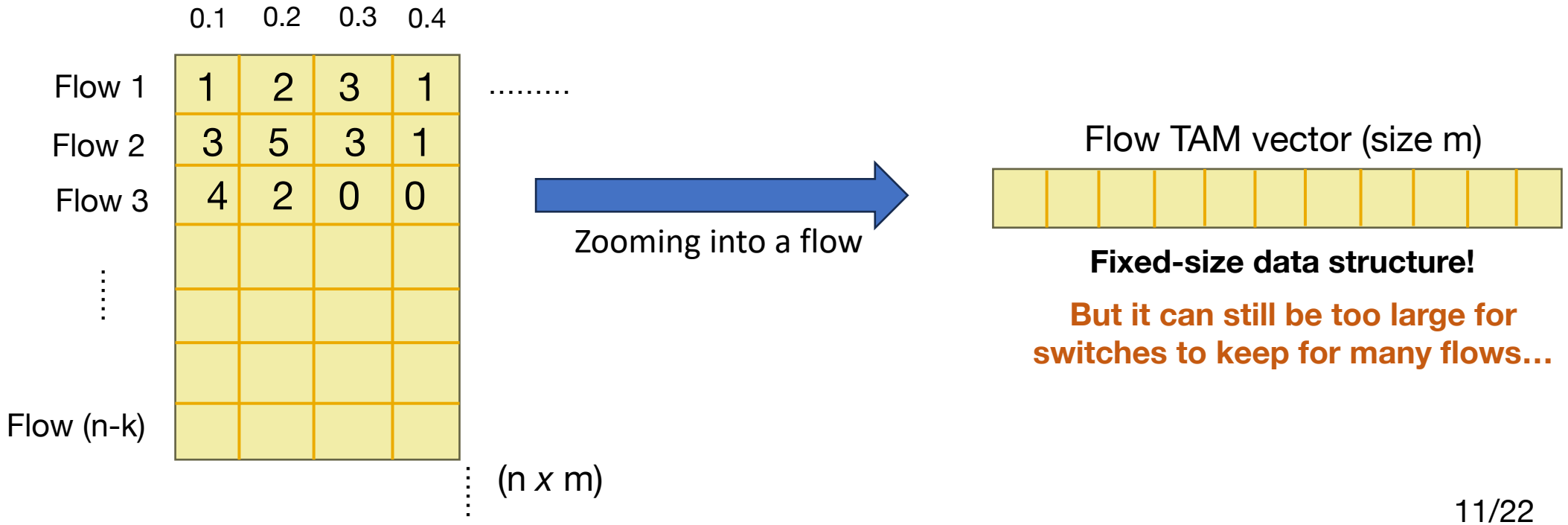
- Keeps track of number of packets transmitted by a flow at some configurable time interval
  - E.g., #packets sent every 0.1s



1. Compute flows' features

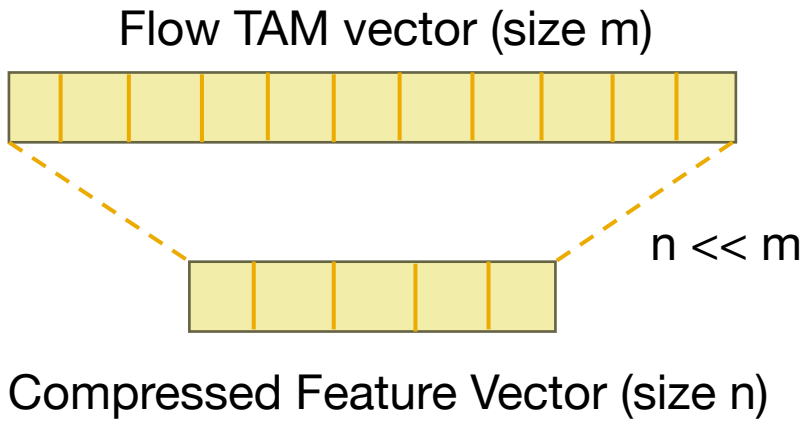
Essentially, start by building a **Traffic Aggregation Matrix (TAM):**

- Keeps track of number of packets transmitted by a flow at some configurable time interval
  - E.g., #packets sent every 0.1s for a total of 1s



2. Compress flows' features

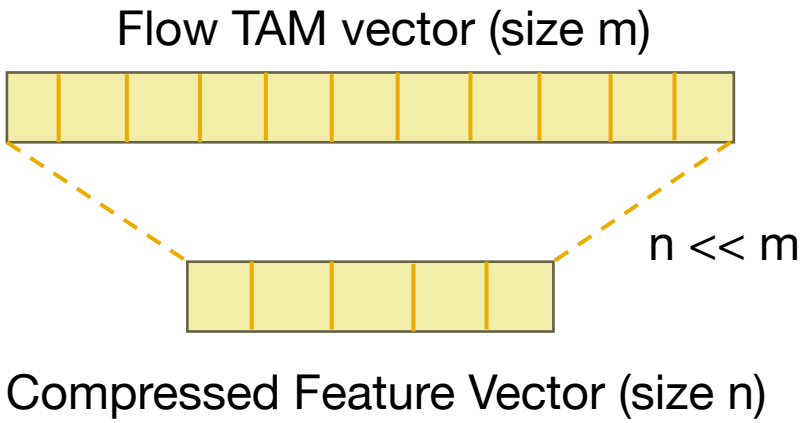
Ideally…



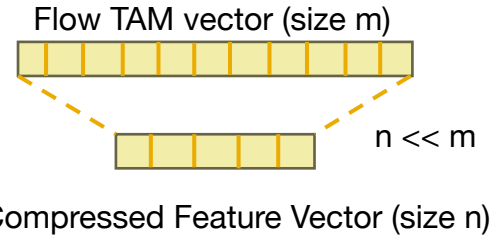
2. Compress flows' features

Ideally…

Sketches to the rescue!



2. Compress flows' features



Coskun et al. [1] sketches

TAM Vector

$$C_F(i) = \sum_{t=1}^{T} \sum_{i=1}^{k} B_i(t) \cdot V_F(t),$$

Integer Sketch: linear transformation w/ Random Bayesian Projections

$$S_F = \{s_i \mid s_i = \mathbb{I}[C_F(i) > 0],\ i = 1, 2, \ldots, k\}$$

Binary Sketch: compresses integer sketch for further space savings

Coskun, B. and Memon, N., 2009, December. Online sketching of network flows for real-time stepping-stone detection. In *2009 Annual Computer Security Applications Conference* (ACSAC)

Nasr et al. [2] sketches

TAM Vector | Sensing Matrix | Final Sketch

$$F = \Phi V \quad \text{with } V \in \mathbb{R}^n,\ \Phi \in \mathbb{R}^{m \times n},\ F \in \mathbb{R}^m$$

Nasr et al. Sketch: Relies on Compressed Sensing linear projection algorithms

Nasr, M., Houmansadr, A. and Mazumdar, A., 2017, October. Compressive traffic analysis: A new paradigm for scalable traffic analysis. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (ACM CCS)

2. Compress flows' features

Coskun et al.'s sketch example

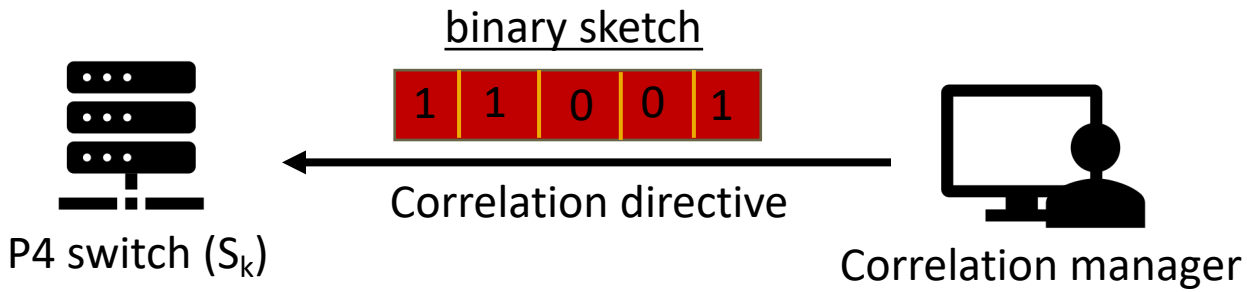
Table Type	Rows (Flows)	Columns	Entry Size	Total Size
Flow Packet Count	100	100 (time slots)	2 bytes (16 bits)	$100 \times 100 \times 2 = 20,000$ bytes (20 KB)
Integer Sketch	100	5 (sketch length)	2 bytes (16 bits)	$100 \times 5 \times 2 = 1,000$ bytes (1 KB)
Binary Sketch	100	5 (sketch length)	1 bit	$100 \times 5 \times 1 = 500$ bits (62.5 bytes)

	Table Type	Flows Storable (200 MB)
1	Flow Packet Count	10485
2	Integer Sketch	209715
3	Binary Sketch	3355443

As we will see, sketches retain correlation accuracy vs. the full TAM!

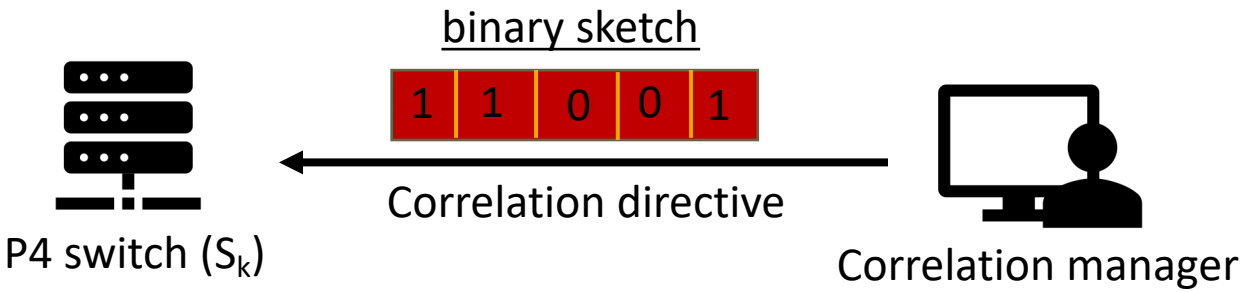
3. Correlate in the control plane

1. Receive target flow sketch

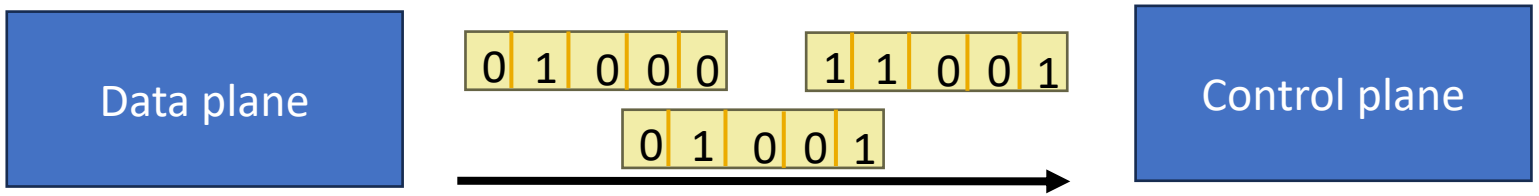


3. Correlate in the control plane

1. Receive target flow sketch

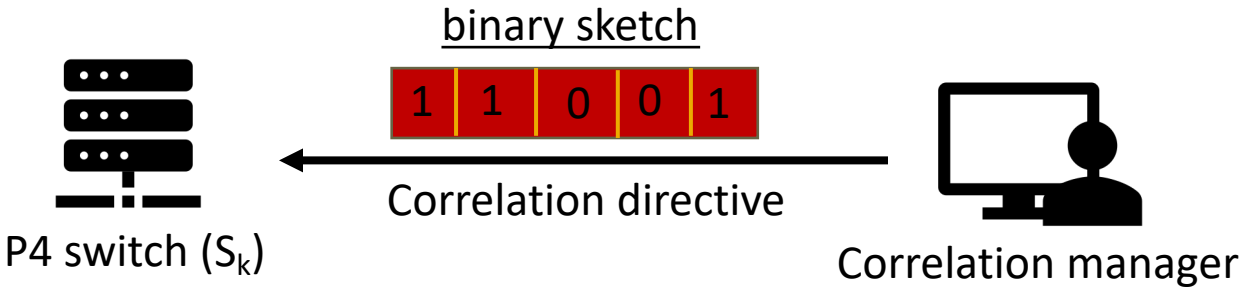


2. Load existing flow sketches from the switch's data plane into the control plane

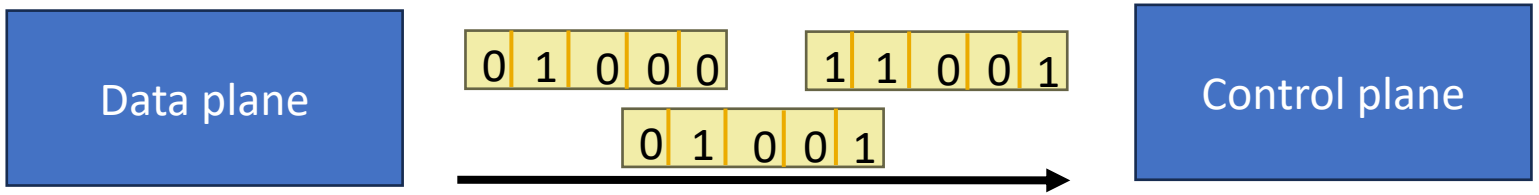


3. Correlate in the control plane

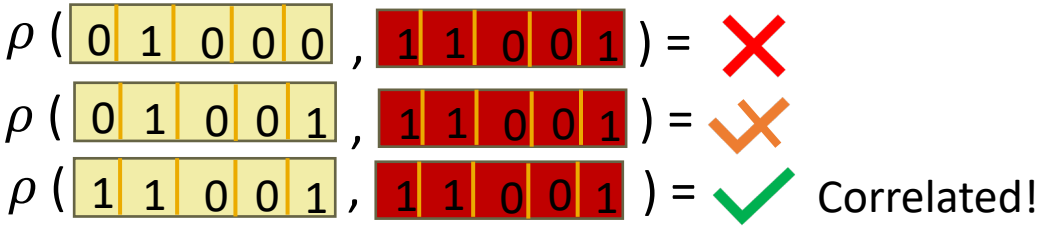
1. Receive target flow sketch



2. Load existing flow sketches from the switch's data plane into the control plane

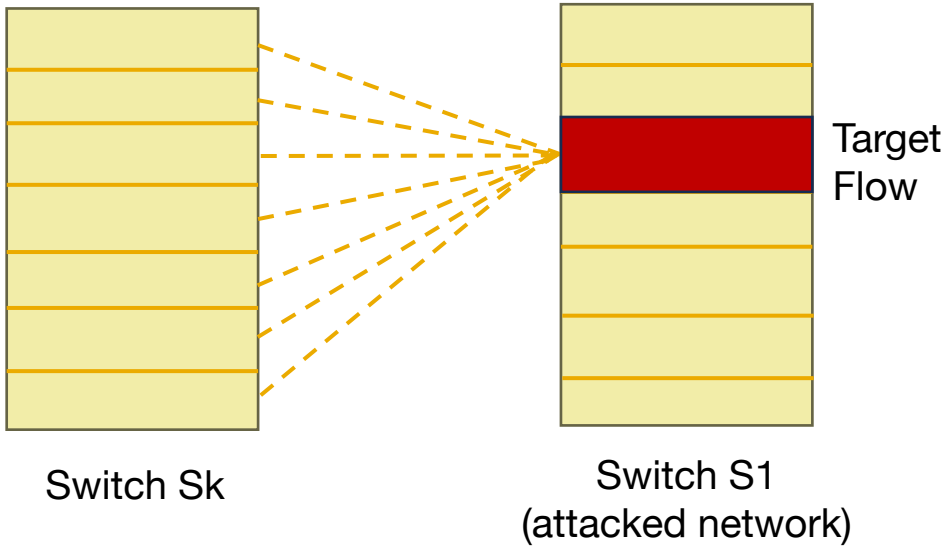


3. Compute pair-wise statistical similarity between the received target sketch and local sketches

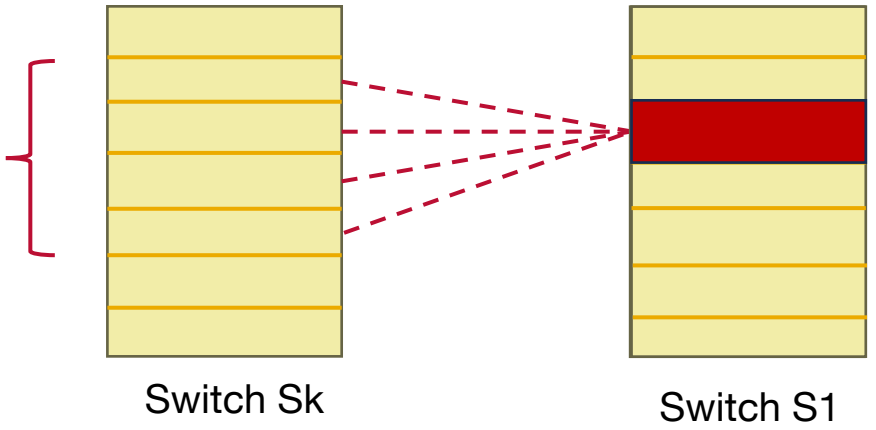


ρ – Hamming distance

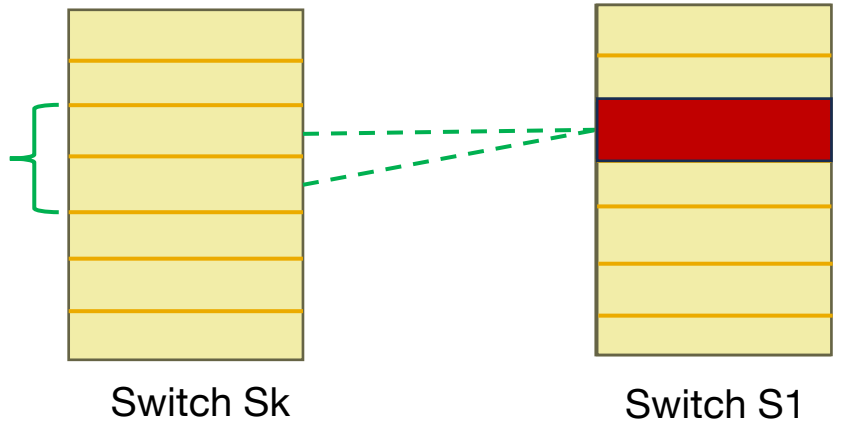
3. Correlate in control plane (+ heuristics)



Initial Timestamp Heuristic



+ Total Packet Counts



Evaluation

Evaluation Goals and Metrics

- **Correlation effectiveness**
  - True Positive Rate
  - False Positive Rate
- **Efficiency**
  - Computational cost (# of pair-wise flow comparisons across switches)
- **Scalability**
  - Number of flows that can be stored and processed
  - Communication overhead

Experimental testbed

- **Datasets**:

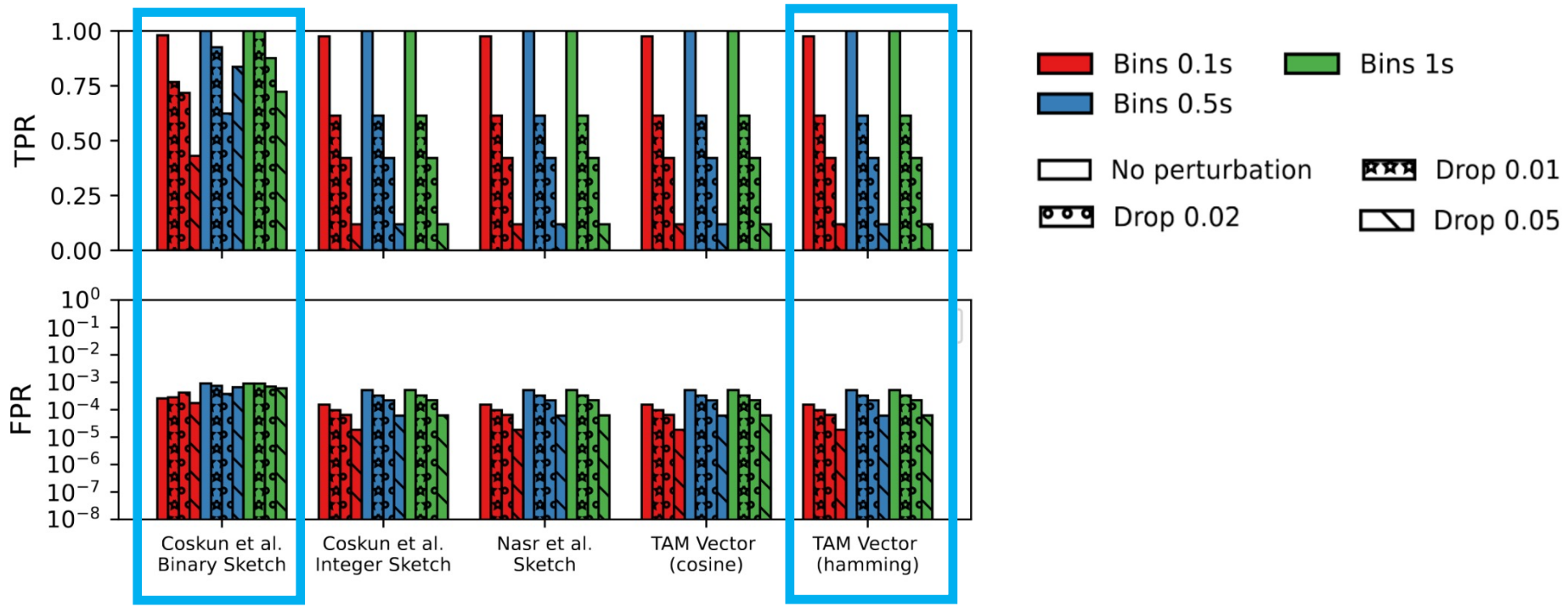
Dataset	Category	Description	Flows		Span (s)
			Benign	Malicious	
Snojan	Botware	PPI malware downloading.	206 723	1 607	45.64
Dridex	Ransomware	Victim locations uploading.	125 424	3 202	54.75
Adload	Adware	Resources for PPI adware.	125 417	602	54.80
Oracle	Web	TLS padding Oracle.	294 110	224	64.14
Penetho	Spyware	Wifi cracking APK spyware.	293 808	1 006	55.64
Bitcoinminer	Miner	Abnormal encrypted channels.	125 418	202	61.01

We assume a perfect IDS! Ask about it ☺

- **Trace augmentation** for distributed monitoring across two vantage points:
  - Simulated transmission of attacking flows over WAN via proxy chain (+ **latency**, + **loss**)

- **TAM/Sketch parameters**:
  - TAM granularity = {0.1s, 0.5s, 1s}
  - Compressed Sketch Length = 10 cells

RevealNet can correlate flows effectively



Sketches achieve similar TPR/FPR rates w.r.t. TAM vectors, even under perturbed network conditions (Effectiveness)

RevealNet's correlation pipeline is efficient

Dataset	None	Creation Time	Packet Count	Both
Dridex	128 626	19 409	2 403	824
Adload	126 019	17 386	1 168	258
Snojan	208 330	20 975	2 735	693
Oracle	294 334	41 580	95	84
Bitcoinminer	125 620	17 288	791	207
Penetho	294 814	28 040	1 735	454

Heuristics help reduce overall number of comparisons by 3 orders of magnitude (Computation)

Method/ Storage	Per flow (in Bytes)	Stored Flows (in 256 MB)
Coskun et al. (bin.)	1.25	$\approx 204.8 \times 10^6$
Coskun et al. (int.)	40	$\approx 6.4 \times 10^6$
Nasr et al. (int.)	40	$\approx 6.4 \times 10^6$
TAM (0.1s bins)	2864	$\approx 1.05 \times 10^5$
TAM (0.5s bins)	576	$\approx 5.20 \times 10^5$
TAM (1s bins)	291	$\approx 1.03 \times 10^6$

Sketches allow switches to accommodate additional flows (Storage)

RevealNet's pipeline is scalable

Communication overhead (in bits) for the `bitcoinminer` dataset. Topology with Sw_c=20 switches

Method	Centralized			Distributed (RevealNet)				OH Red. (%)
	$Sw_c \rightarrow CS$	$Sw_a \rightarrow CS$	Total	$Sw_a \rightarrow CM$	$CM \rightarrow Sw_c$	$Sw_c \rightarrow CM$	Total	
Coskun et al. (bin.)	1 193 390	2 020	1 195 410	2 020	38 380	736 896	777 296	35.0%
Coskun et al. (int.)	38 188 480	64 640	38 253 120	64 640	1 227 680	736 896	2 029 216	94.7%
Nasr et al. (int.)	38 188 480	64 640	38 253 120	64 640	1 227 680	736 896	2 029 216	94.7%
TAM	229 085 280	387 840	229 473 120	387 840	7 368 960	736 896	8 493 696	96.3%

Sw_a – attacked network
Sw_c – cooperating network
CS – central correlation server
CM – correlation manager

Distributed correlation reduces communication overheads by 35 to 96% vs. centralized attack attribution (Bandwidth)

Takeaways

- Network attackers increasingly make use of proxy chains to hide their tracks
  - Traffic correlation is an attractive method to perform attack attribution in these conditions
- Current attack attribution frameworks are centralized, bearing severe storage, communication, and computation overheads
- **RevealNet**:
  - Distributed attack attribution framework based on programmable switches' orchestration
  - Implements flow feature compression schemes and heuristics to alleviate overheads

Diogo Barradas
diogo.barradas@uwaterloo.ca

Thank you!