33rd Network and Distributed System Security Symposium (NDSS'26)

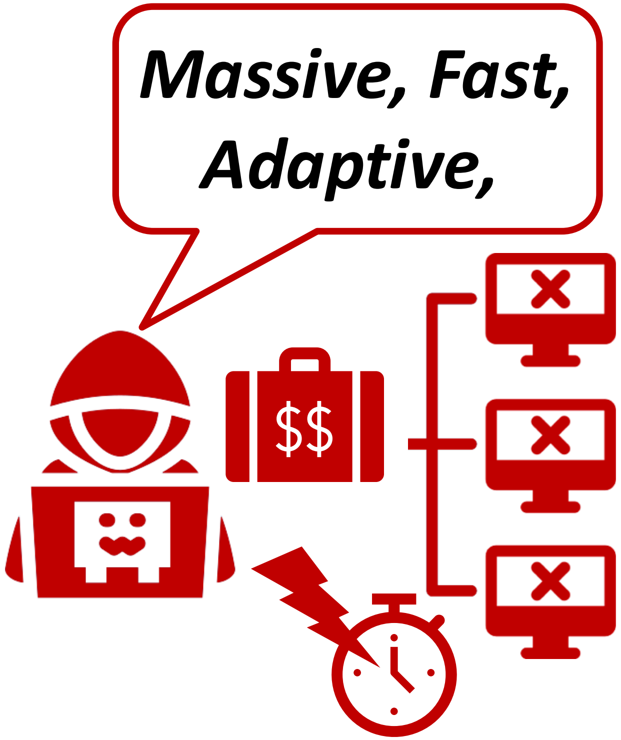
On the Security Risks of Memory Adaptation and Augmentation in Data-plane DoS Mitigation

Hocheol Nam[*], Daehyun Lim[*], Huancheng Zhou[†],
Guofei Gu[†], and Min Suk Kang[*]

[*]*NetS&P Lab,* ***KAIST***
[†]*SUCCESS Lab,* ***Texas A&M University***

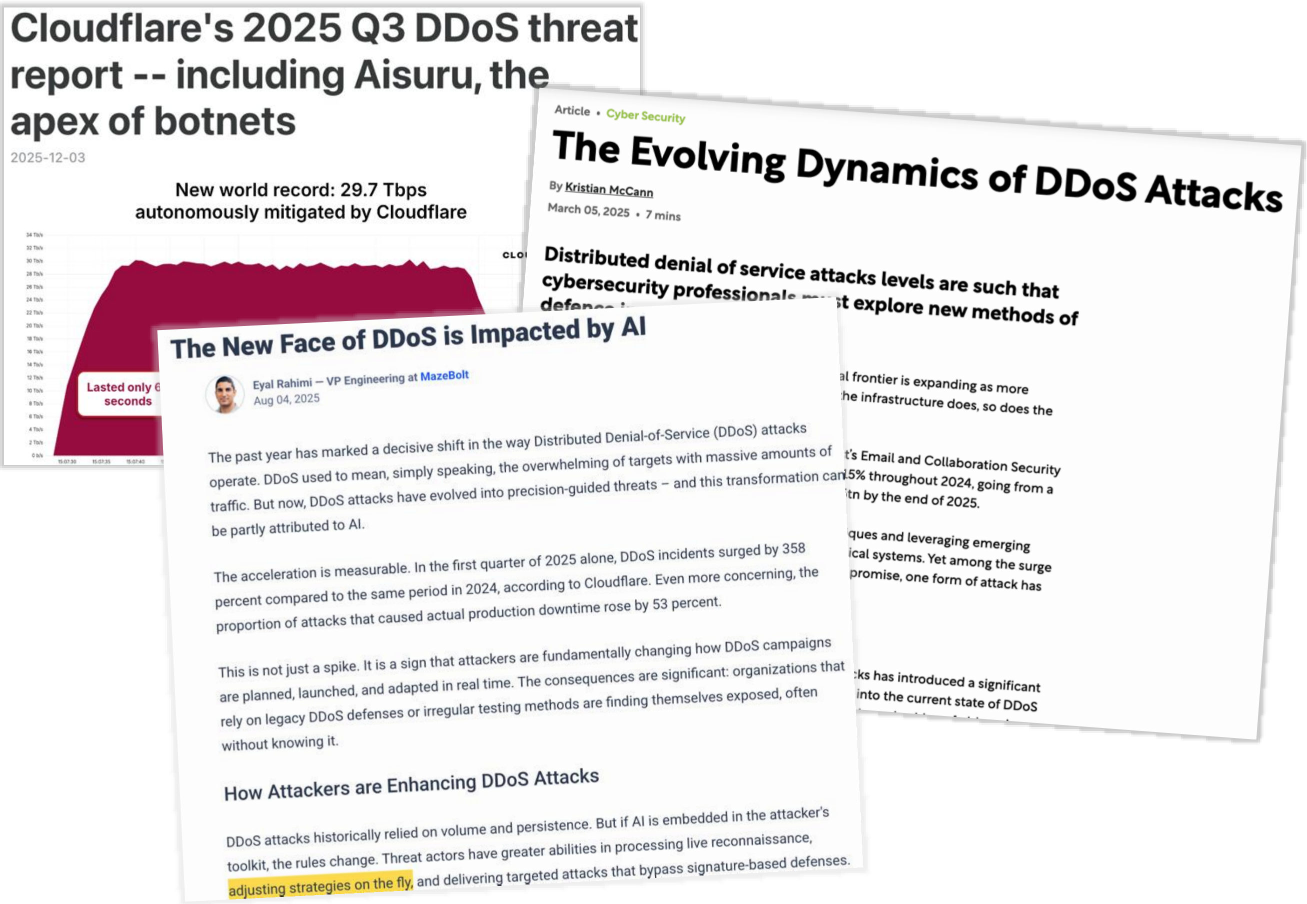
Feb. 24, 2026

DoS ***adversaries*** *have* ***asymmetric advantages*** with their ***adaptive attack strategies***



Massive, Fast, Adaptive,

DDoS attacker



Cloudflare's 2025 Q3 DDoS threat report -- including Aisuru, the apex of botnets

2025-12-03

New world record: 29.7 Tbps autonomously mitigated by Cloudflare

Article • Cyber Security

The Evolving Dynamics of DDoS Attacks

By Kristian McCann

March 05, 2025 • 7 mins

Distributed denial of service attacks levels are such that cybersecurity professionals m t explore new methods of defence i

al frontier is expanding as more he infrastructure does, so does the

t's Email and Collaboration Security 5% throughout 2024, going from a tn by the end of 2025.

ques and leveraging emerging ical systems. Yet among the surge promise, one form of attack has

ks has introduced a significant into the current state of DDoS

The New Face of DDoS is Impacted by AI

Eyal Rahimi — VP Engineering at MazeBolt
Aug 04, 2025

The past year has marked a decisive shift in the way Distributed Denial-of-Service (DDoS) attacks operate. DDoS used to mean, simply speaking, the overwhelming of targets with massive amounts of traffic. But now, DDoS attacks have evolved into precision-guided threats – and this transformation can be partly attributed to AI.

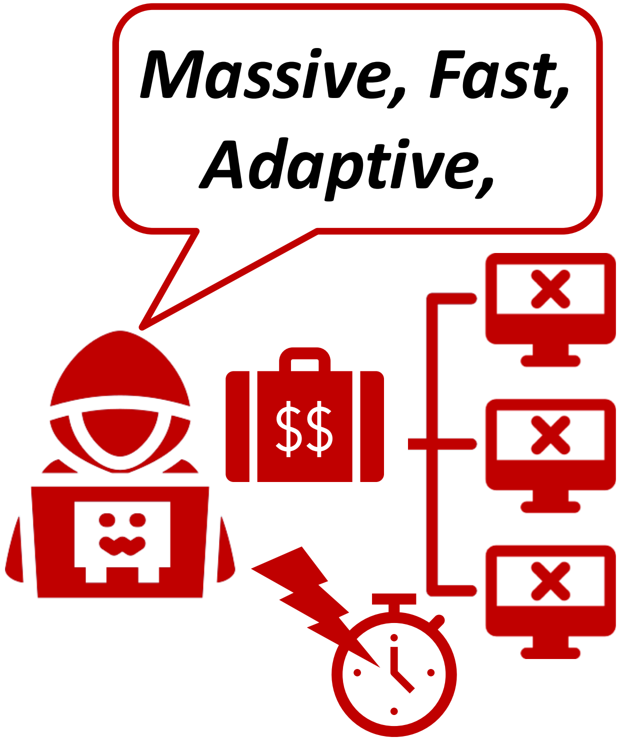
The acceleration is measurable. In the first quarter of 2025 alone, DDoS incidents surged by 358 percent compared to the same period in 2024, according to Cloudflare. Even more concerning, the proportion of attacks that caused actual production downtime rose by 53 percent.

This is not just a spike. It is a sign that attackers are fundamentally changing how DDoS campaigns are planned, launched, and adapted in real time. The consequences are significant: organizations that rely on legacy DDoS defenses or irregular testing methods are finding themselves exposed, often without knowing it.

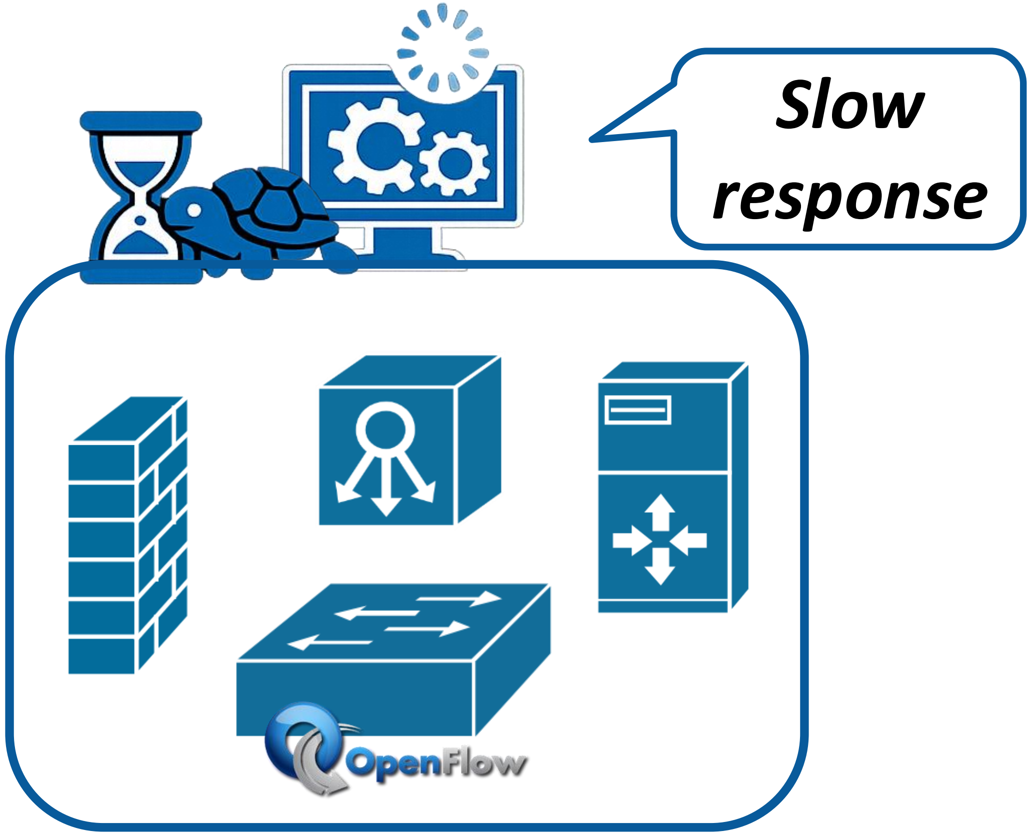
How Attackers are Enhancing DDoS Attacks

DDoS attacks historically relied on volume and persistence. But if AI is embedded in the attacker's toolkit, the rules change. Threat actors have greater abilities in processing live reconnaissance, adjusting strategies on the fly, and delivering targeted attacks that bypass signature-based defenses.

*DoS **adversaries** have **asymmetric advantages***
with their ***adaptive attack strategies***

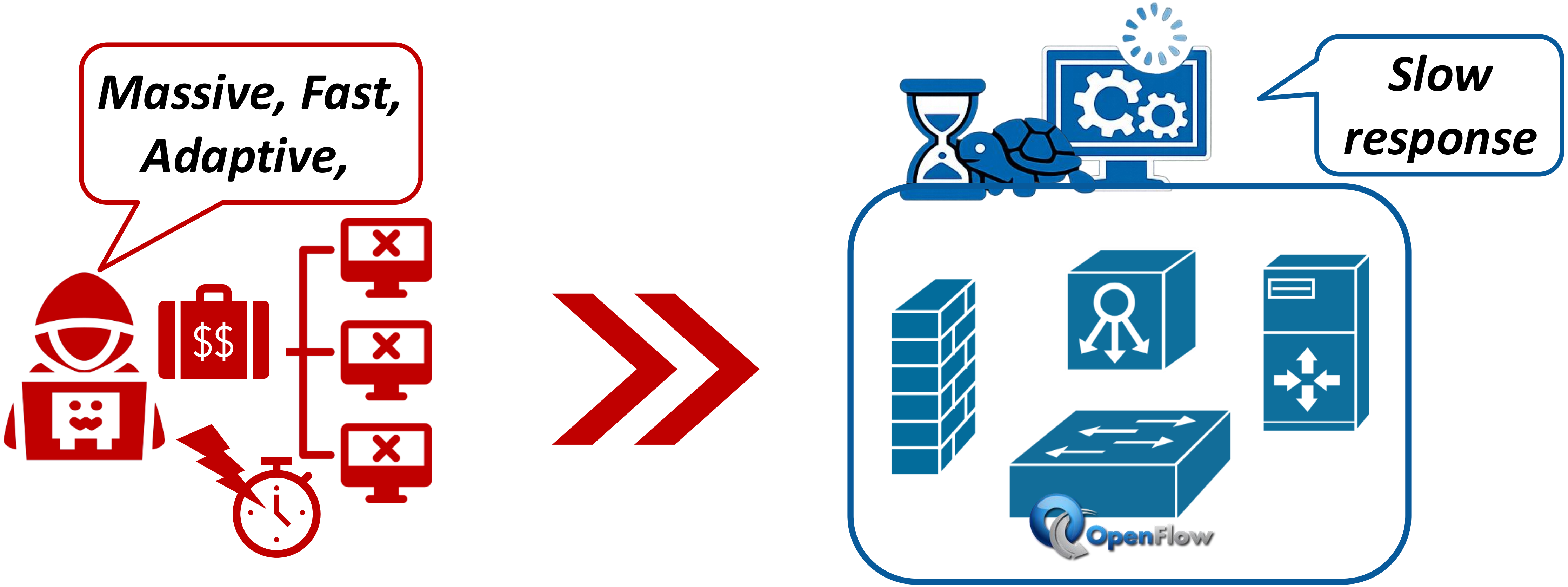


DDoS attacker



DDoS defender

DoS ***adversaries*** *have* ***asymmetric advantages*** with their ***adaptive attack strategies***



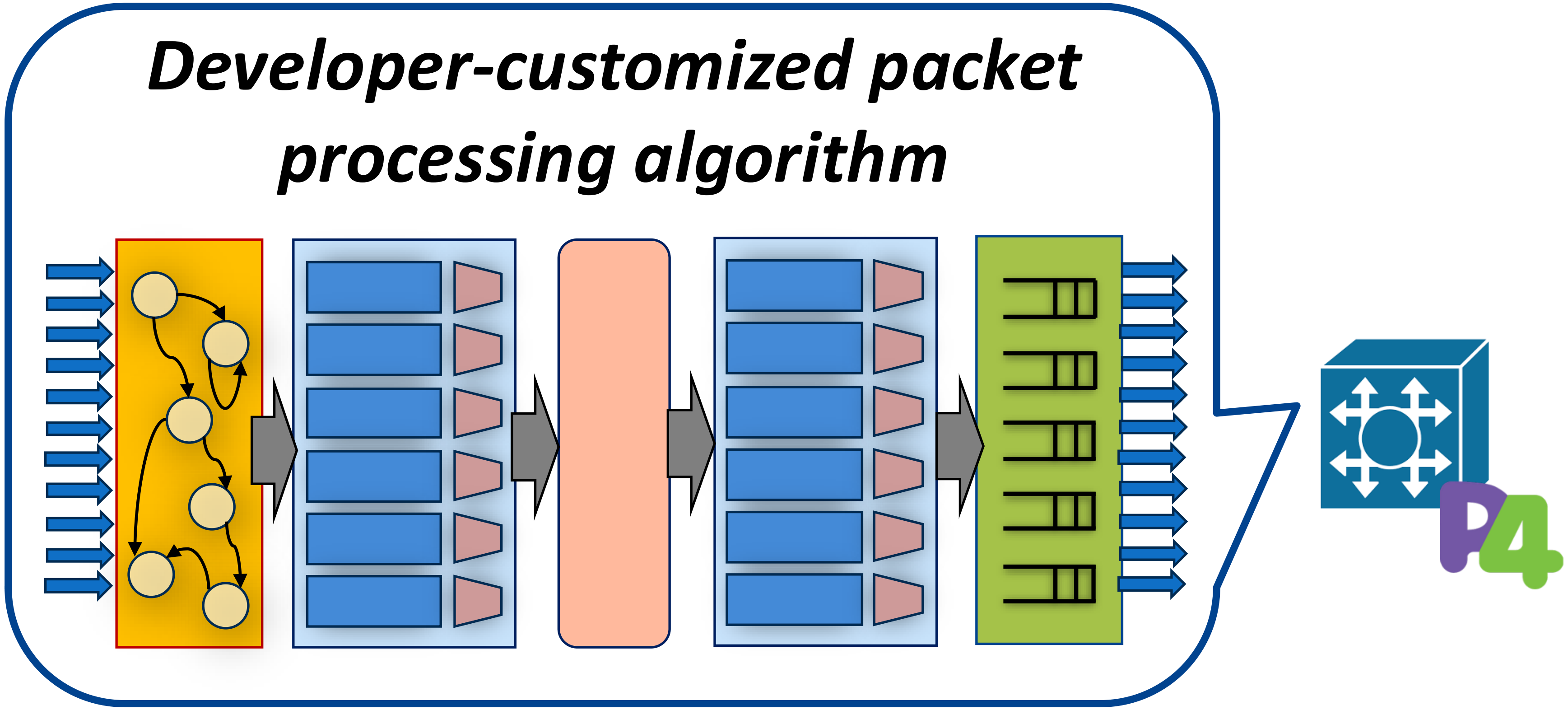
Massive, Fast, Adaptive,

Slow response

DDoS attacker

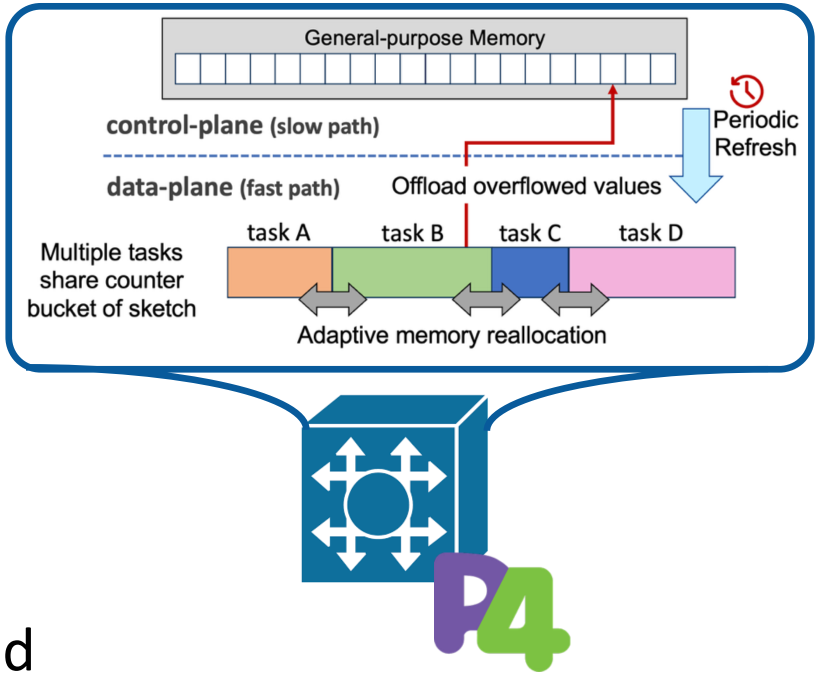
DDoS defender

Programmable switches rebalance the ***asymmetry in DoS defense***



Programmable switches rebalance the ***asymmetry in DoS defense***

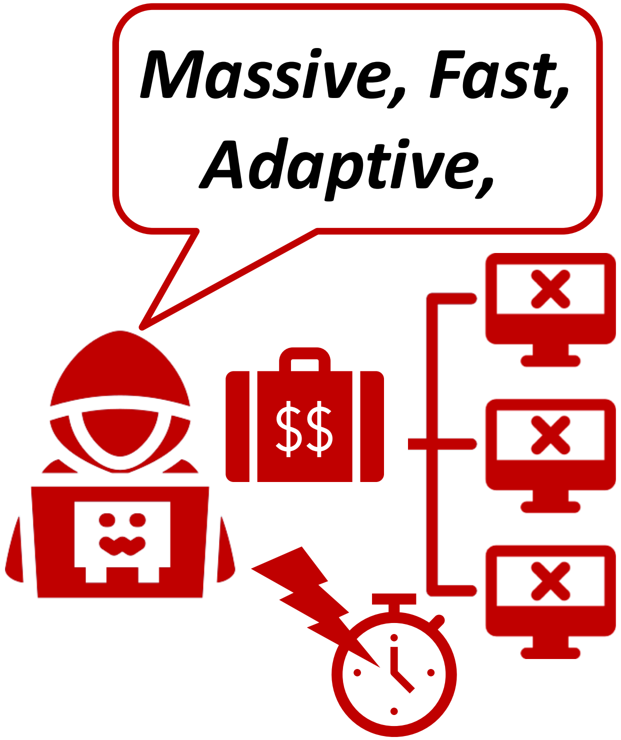
Line-rate adaptive DoS Mitigation [SP’24]


General-purpose Memory
control-plane (slow path)
Periodic Refresh
data-plane (fast path)
Offload overflowed values
task A
task B
task C
task D
Multiple tasks share counter bucket of sketch
Adaptive memory reallocation

DDoS defender

[SP’24] Zhou et al. "Cerberus: Enabling Efficient and Effective In-Network Monitoring on Programmable Switches." *in Proc. IEEE S&P*. 2024.

Programmable switches rebalance the ***asymmetry in DoS defense***

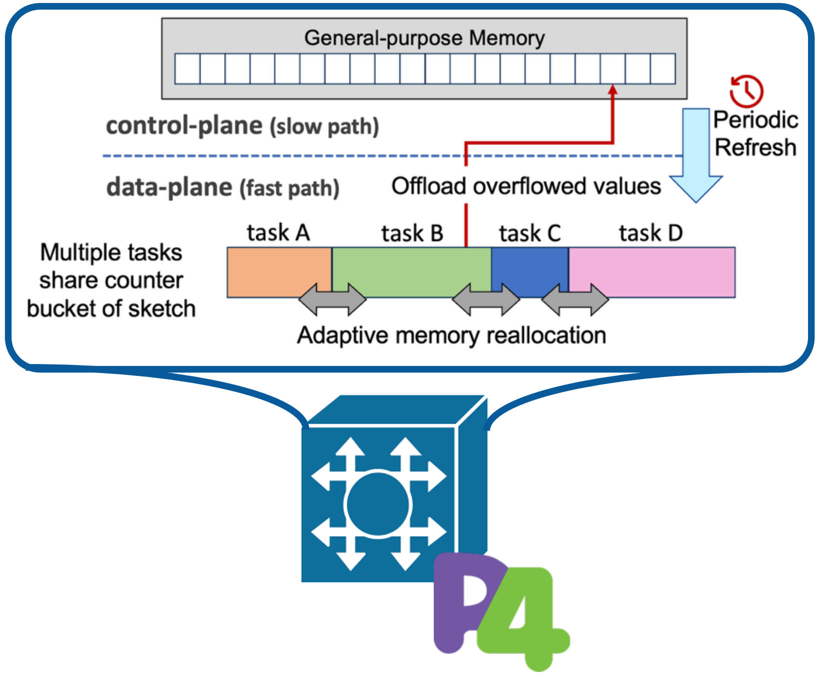


Massive, Fast, Adaptive,

DDoS attacker

≈

Line-rate adaptability



General-purpose Memory

Periodic Refresh

control-plane (slow path)

data-plane (fast path)

Offload overflowed values

task A

task B

task C

task D

Multiple tasks share counter bucket of sketch

Adaptive memory reallocation

DDoS defender

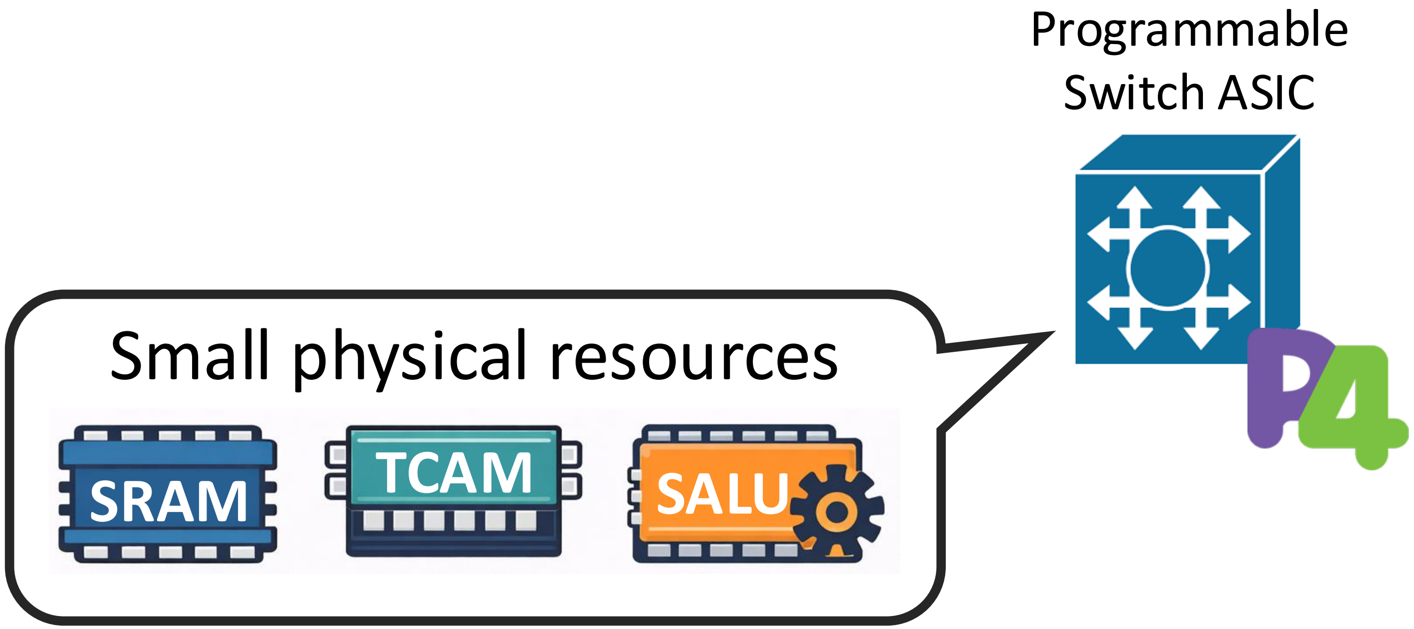
Highly optimized, efficiency-driven design is essential in data plane programming

- Hardware-constraints

Programmable Switch ASIC

Highly optimized, efficiency-driven design is essential in data plane programming

- Hardware-constraints

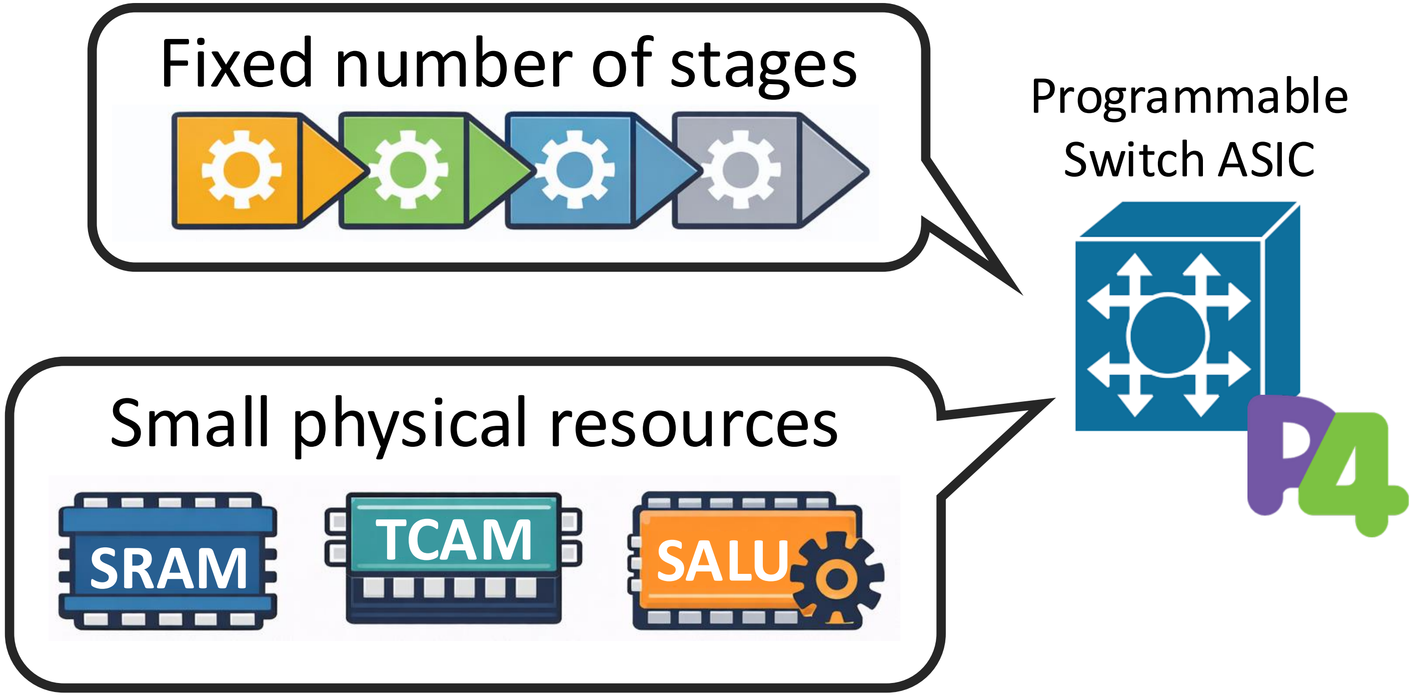


Programmable Switch ASIC

Small physical resources

Highly optimized, efficiency-driven design is essential in data plane programming

- Hardware-constraints



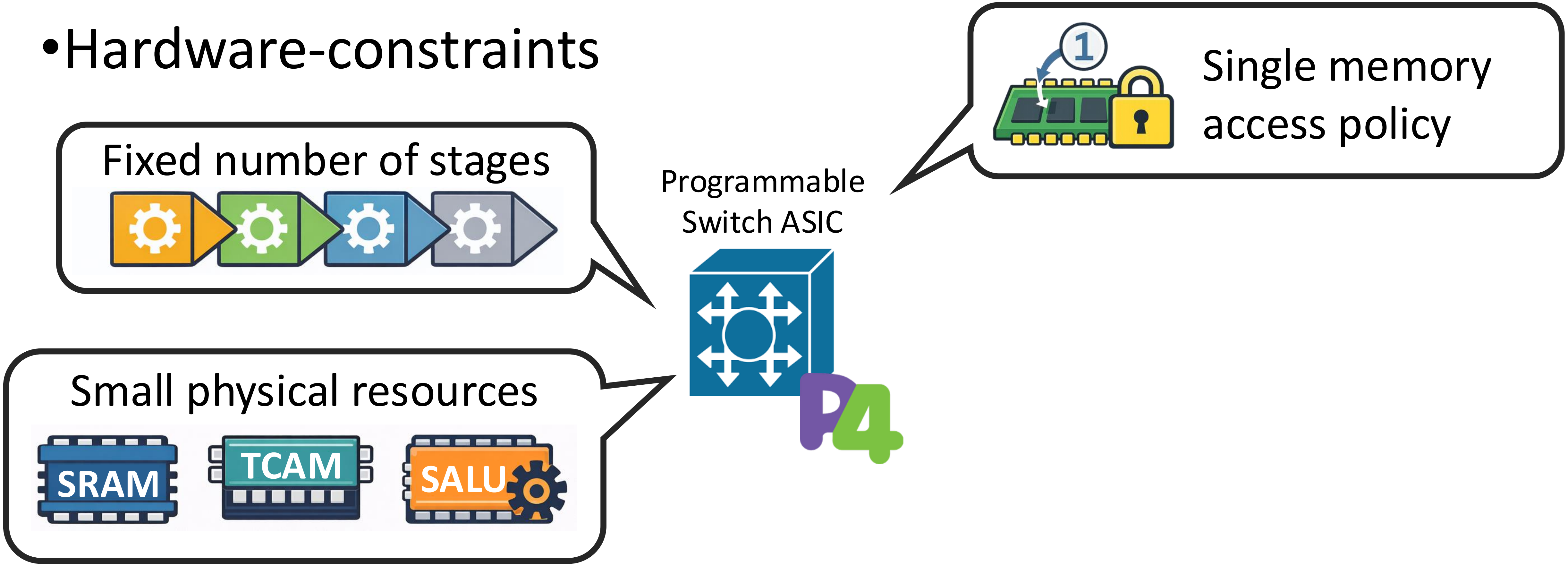
Fixed number of stages

Small physical resources

Programmable Switch ASIC

Highly optimized, efficiency-driven design is essential in data plane programming

- Hardware-constraints



Fixed number of stages

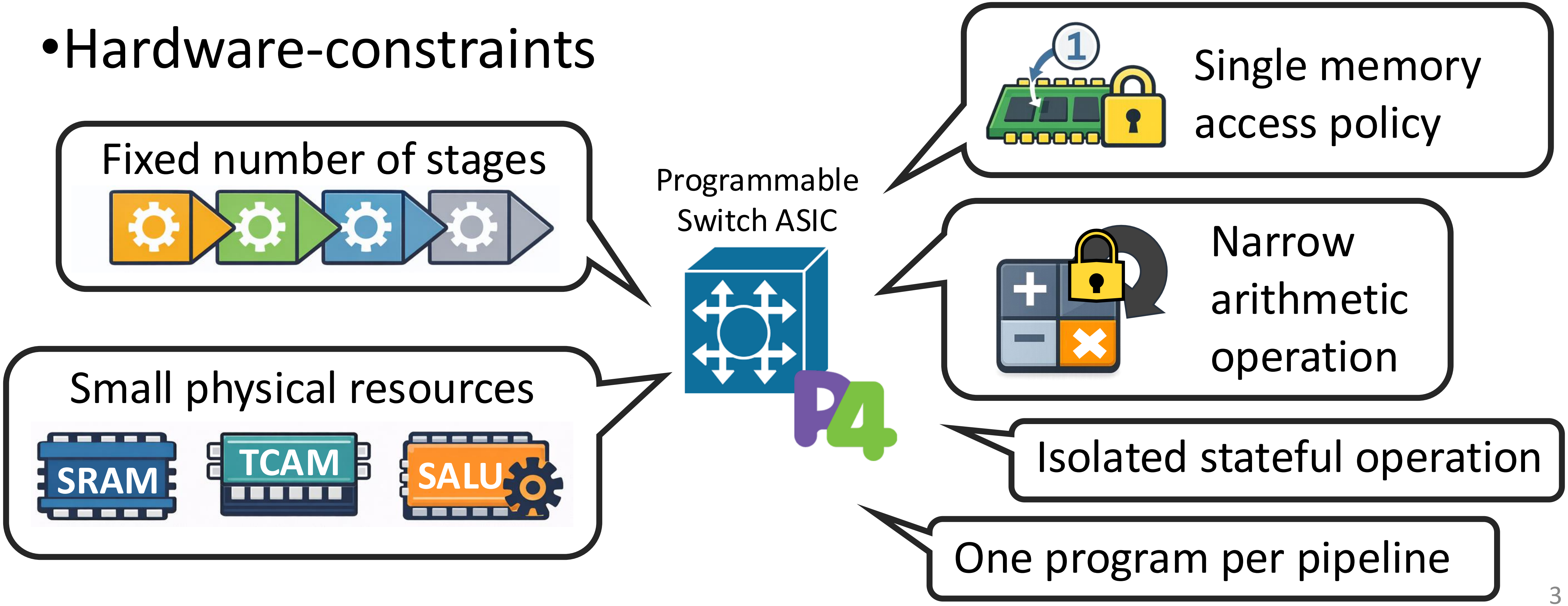
Small physical resources

Programmable Switch ASIC

Single memory access policy

Highly optimized, efficiency-driven design is essential in data plane programming

- Hardware-constraints



Fixed number of stages

Small physical resources

Programmable Switch ASIC

Single memory access policy

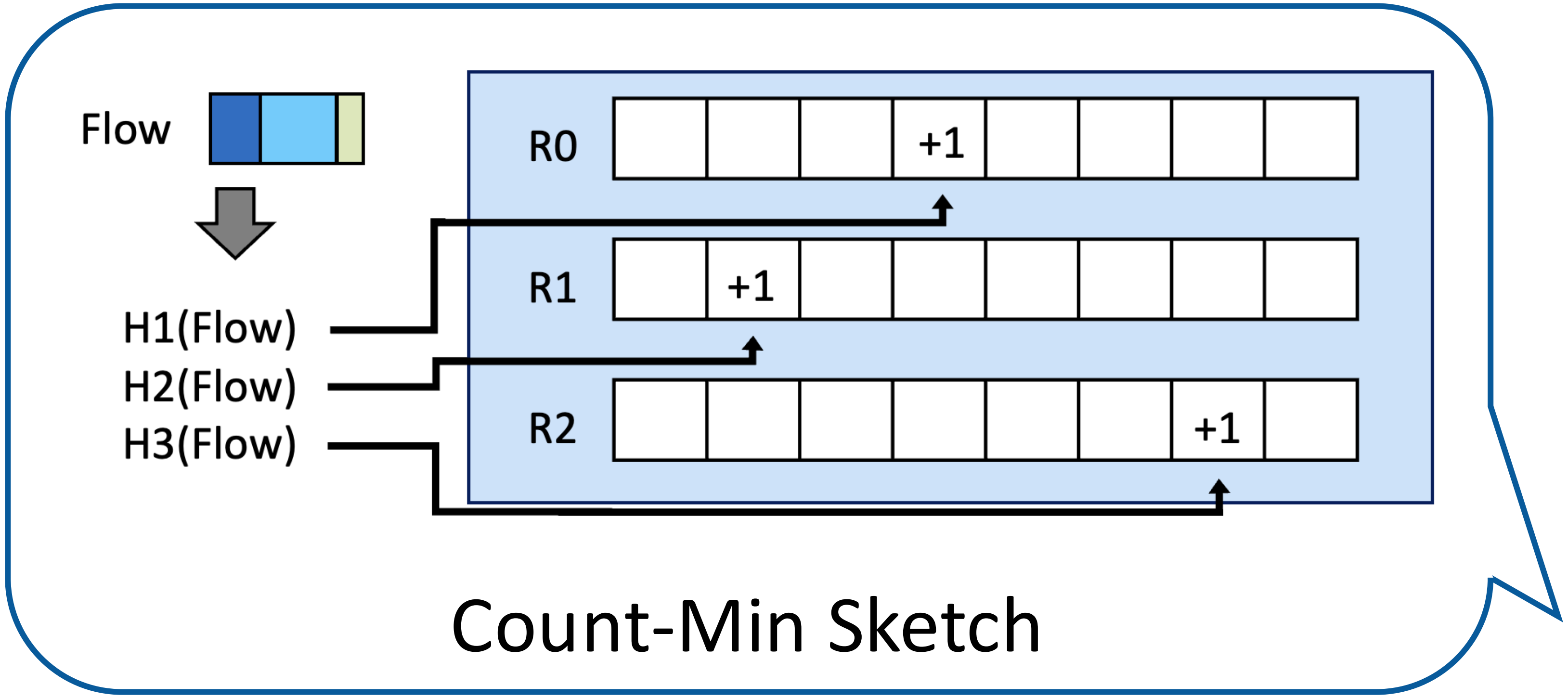
Narrow arithmetic operation

Isolated stateful operation

One program per pipeline

Modern in-network DoS defense relies on ***heavy data-plane optimization***

- Cerberus* [SP'24]

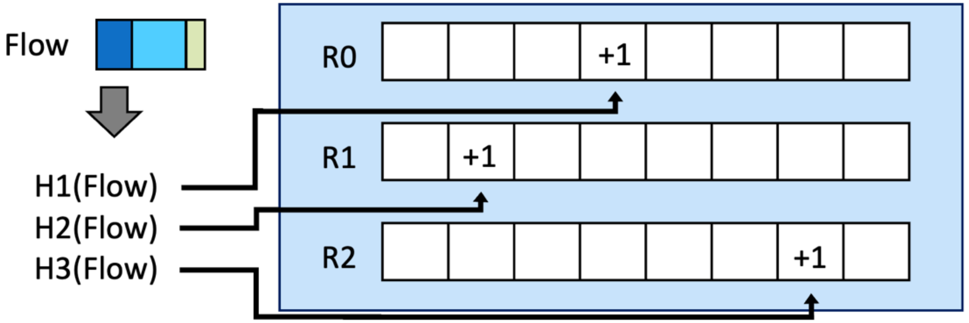


Count-Min Sketch

(*) Zhou et al. "Cerberus: Enabling Efficient and Effective In-Network Monitoring on Programmable Switches." *in Proc. IEEE S&P*. 2024.

Modern in-network DoS defense relies on ***heavy data-plane optimization***

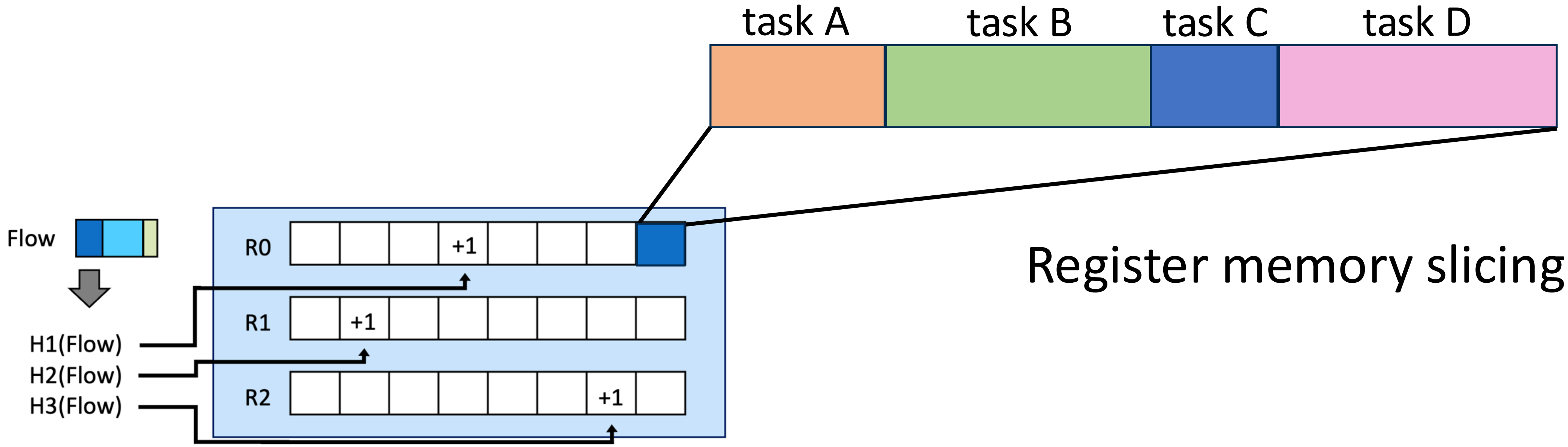
- Cerberus [SP’24]



[SP’24] Zhou et al. "Cerberus: Enabling Efficient and Effective In-Network Monitoring on Programmable Switches." *in Proc. IEEE S&P*. 2024.

Modern in-network DoS defense relies on ***heavy data-plane optimization***

- Cerberus [SP’24]



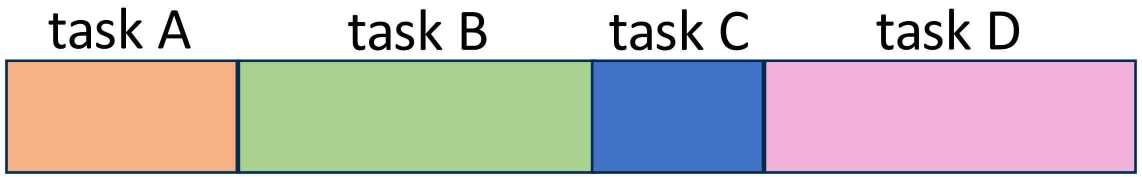
Register memory slicing

[SP’24] Zhou et al. "Cerberus: Enabling Efficient and Effective In-Network Monitoring on Programmable Switches." *in Proc. IEEE S&P*. 2024.

Modern in-network DoS defense relies on ***heavy data-plane optimization***

- Cerberus [SP’24]

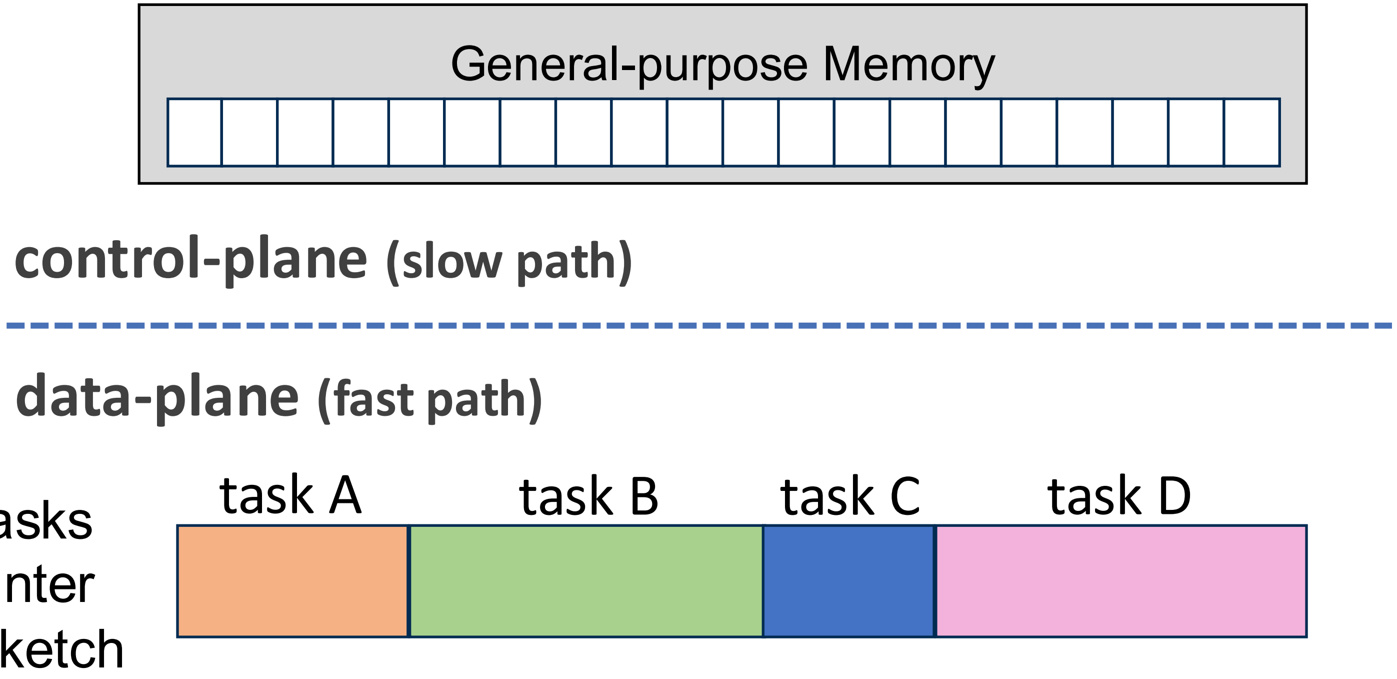
Multiple tasks share counter bucket of sketch



[SP’24] Zhou et al. "Cerberus: Enabling Efficient and Effective In-Network Monitoring on Programmable Switches." *in Proc. IEEE S&P*. 2024.

Modern in-network DoS defense relies on ***heavy data-plane optimization***

- Cerberus [SP'24]

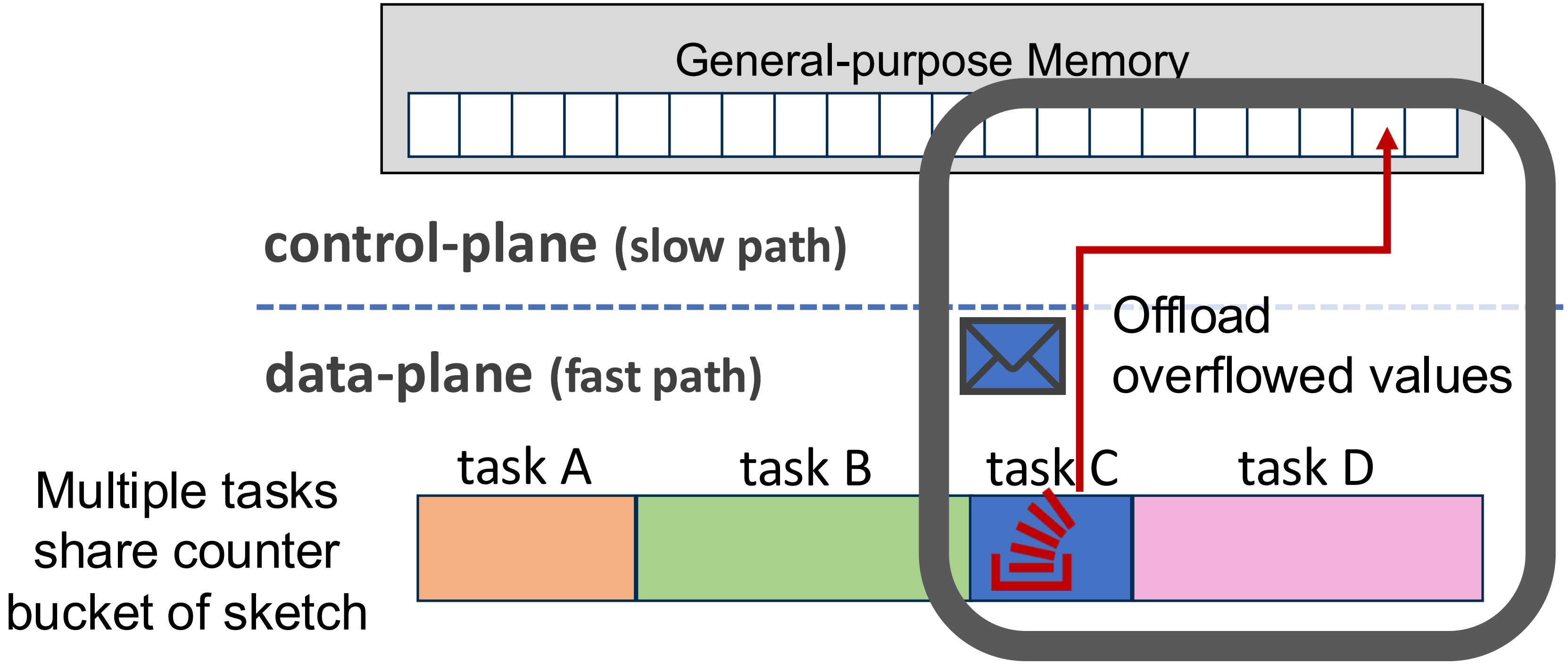


Multiple tasks share counter bucket of sketch

[SP'24] Zhou et al. "Cerberus: Enabling Efficient and Effective In-Network Monitoring on Programmable Switches." *in Proc. IEEE S&P*. 2024.

Modern in-network DoS defense relies on ***heavy data-plane optimization***

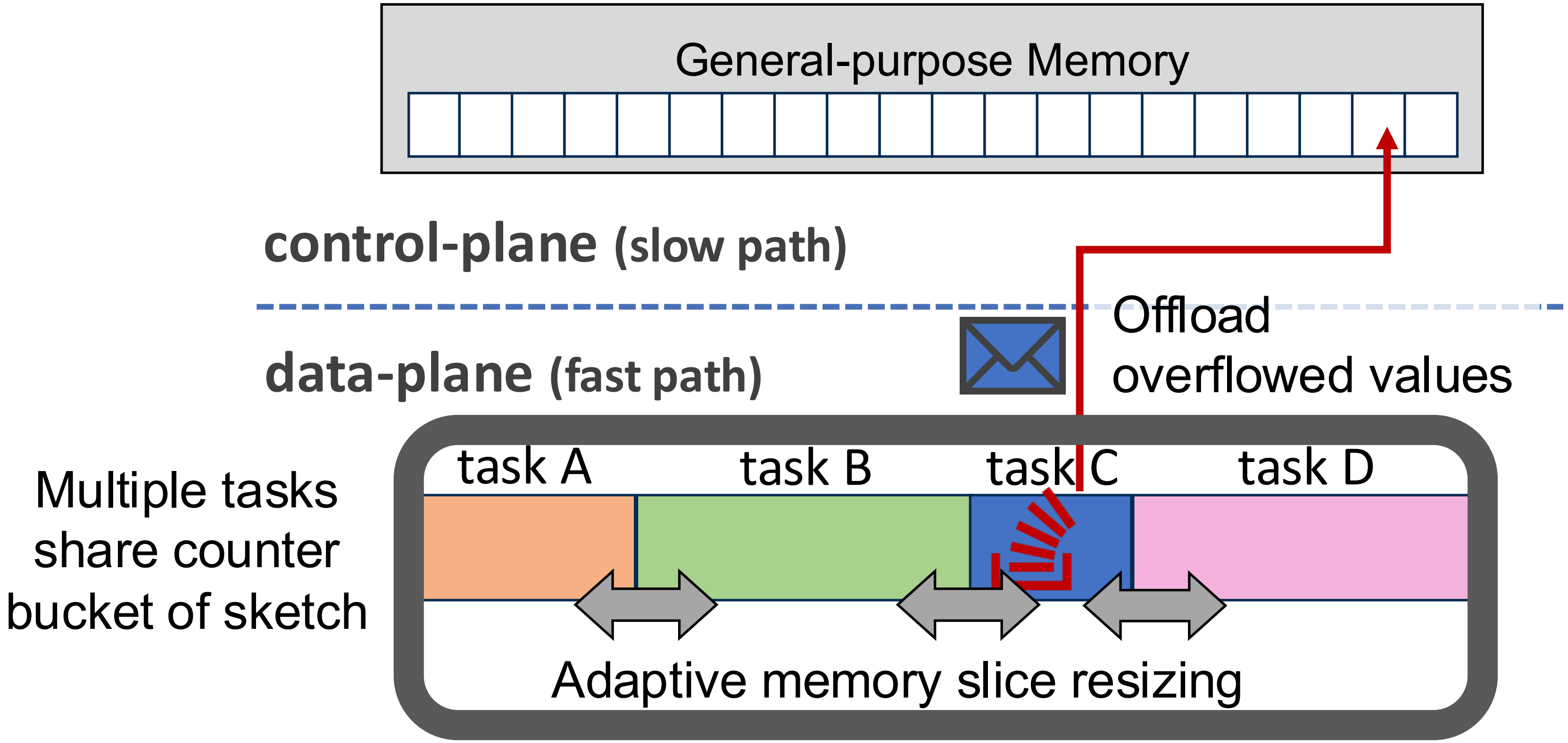
- Cerberus [SP’24]



[SP’24] Zhou et al. "Cerberus: Enabling Efficient and Effective In-Network Monitoring on Programmable Switches." *in Proc. IEEE S&P*. 2024.

Modern in-network DoS defense relies on ***heavy data-plane optimization***

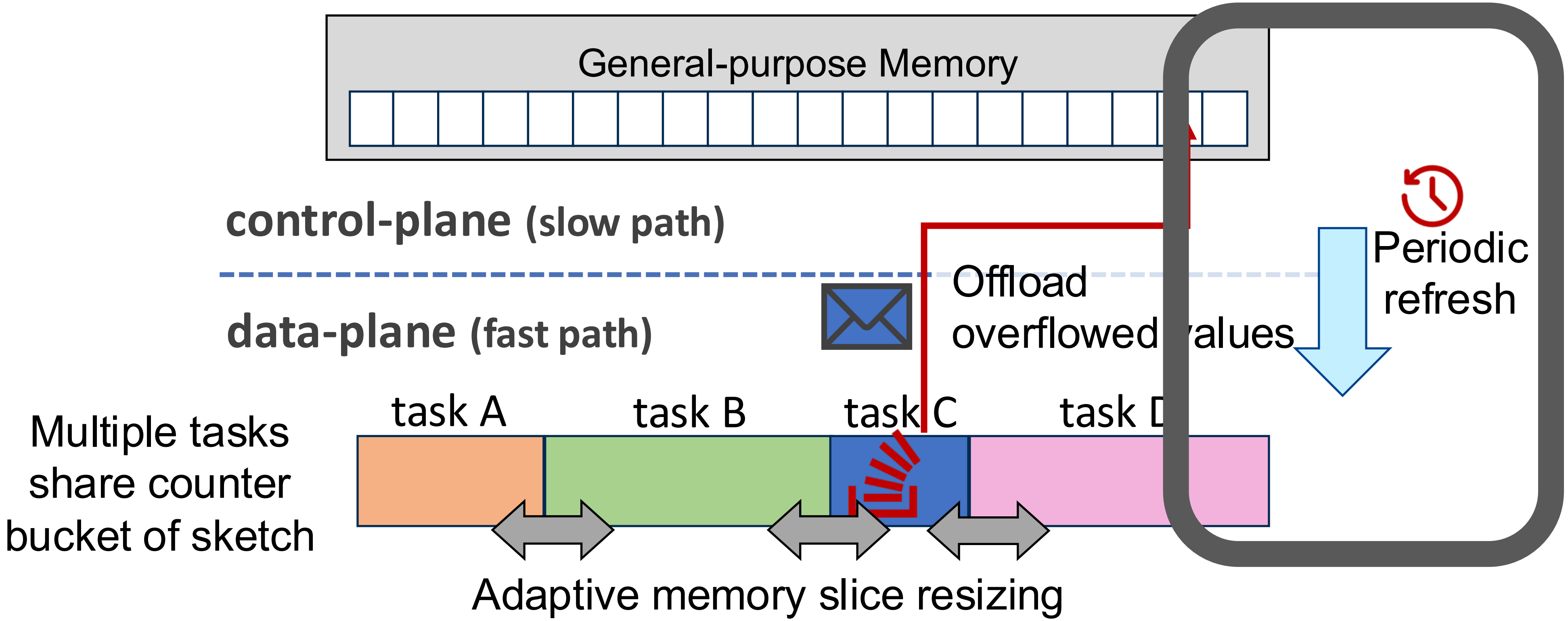
- Cerberus [SP’24]



[SP’24] Zhou et al. "Cerberus: Enabling Efficient and Effective In-Network Monitoring on Programmable Switches." *in Proc. IEEE S&P*. 2024.

Modern in-network DoS defense relies on ***heavy data-plane optimization***

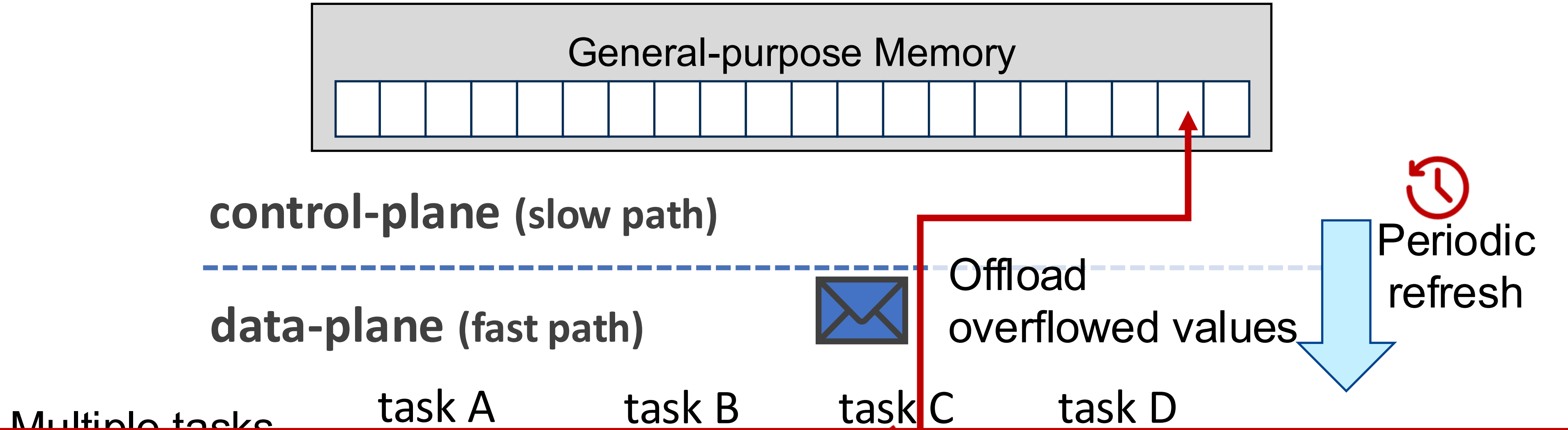
- Cerberus [SP’24]



[SP’24] Zhou et al. "Cerberus: Enabling Efficient and Effective In-Network Monitoring on Programmable Switches." *in Proc. IEEE S&P*. 2024.

Modern in-network DoS defense relies on ***heavy data-plane optimization***

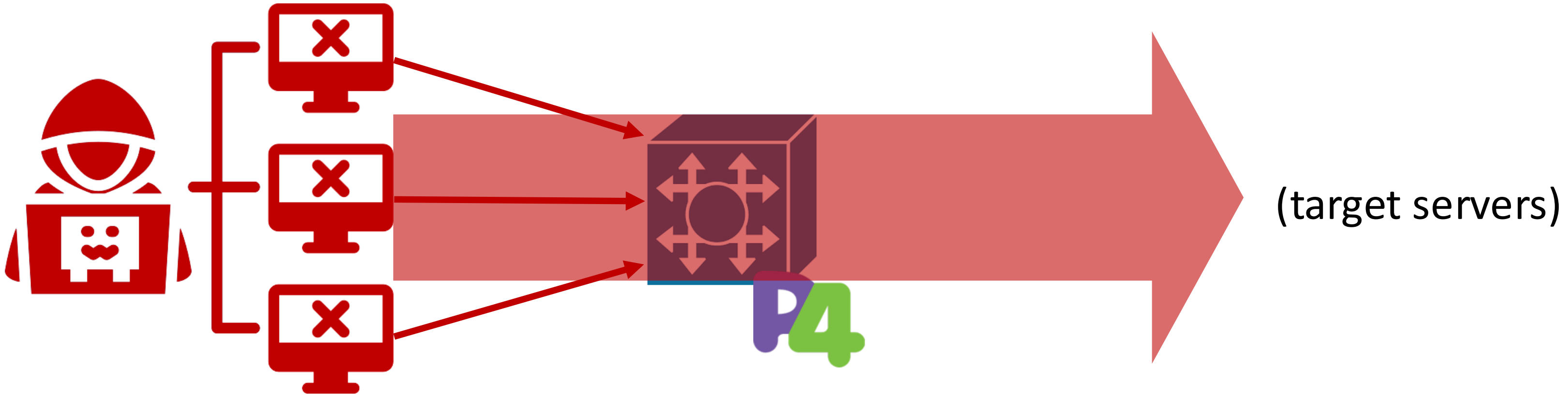
- Cerberus [SP’24]



However, these optimizations combined ***creates new attack surfaces***

Adversary Model

- ***Goal***: To send most volumetric DoS traffic undetected

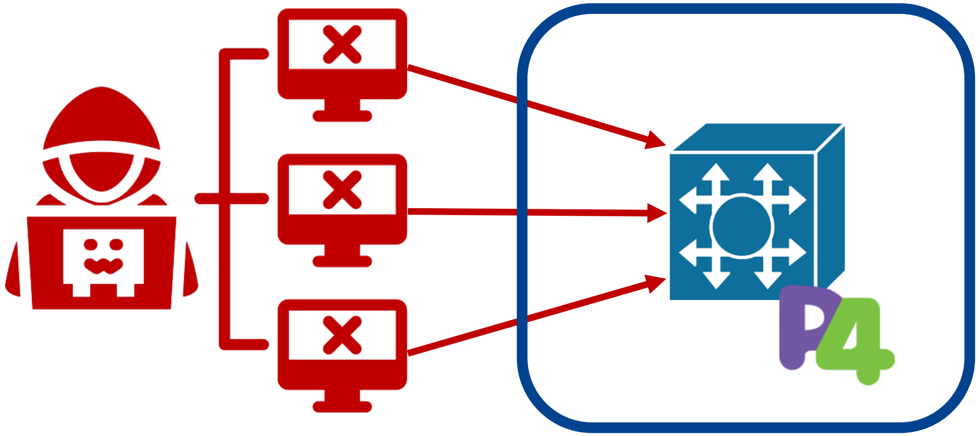


Adversary Model

- ***Goal***: To send most volumetric DoS traffic undetected

System model

- Switches at network chokepoints (e.g., scrubbing centers)
- DoS defense (e.g., Cerberus) deployed

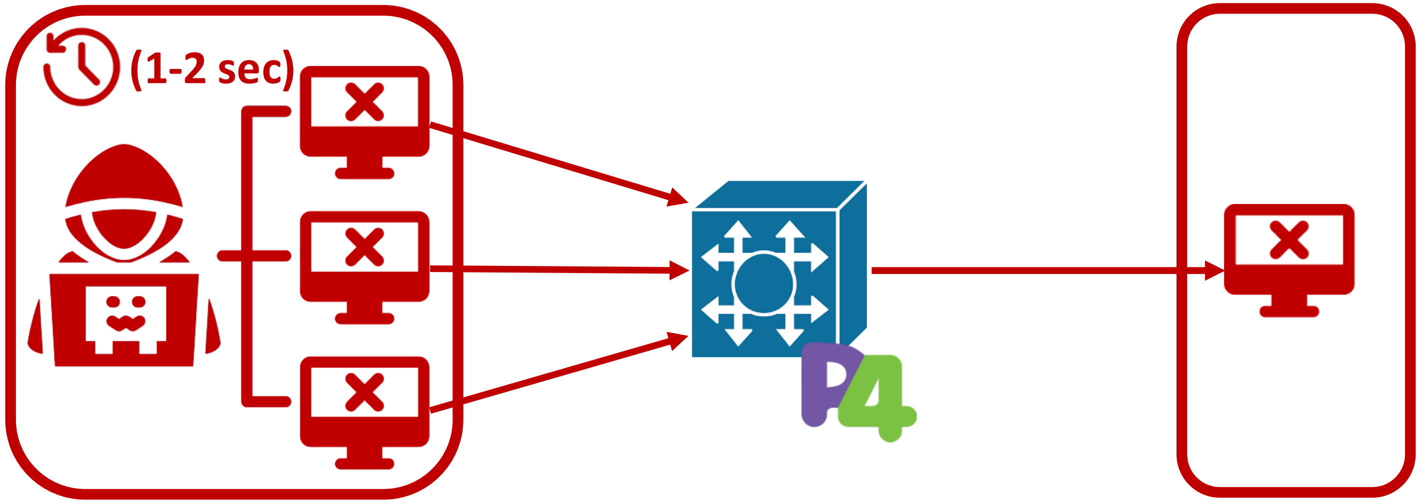


Adversary Model

- ***Goal***: To send most volumetric DoS traffic undetected

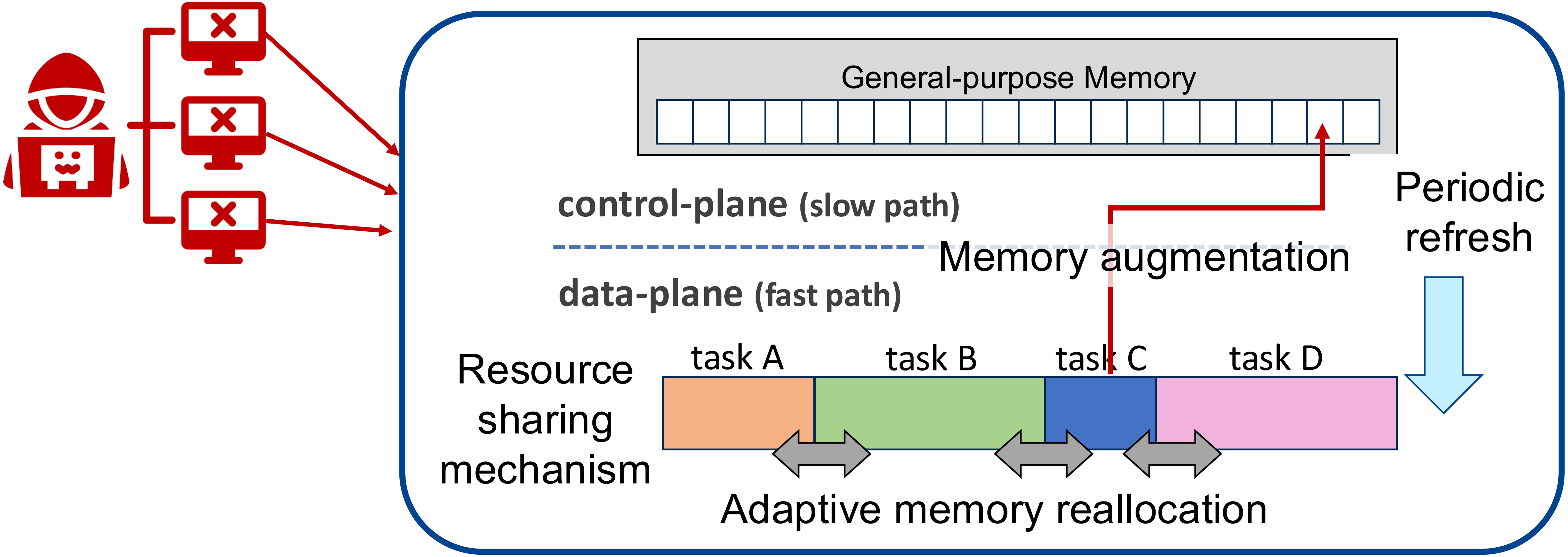
Attacker capabilities

- Adversary controls a large-scale, globally distributed botnet
- Adversary coordinates ***loosely synchronized*** attack traffic (≈1–2s)



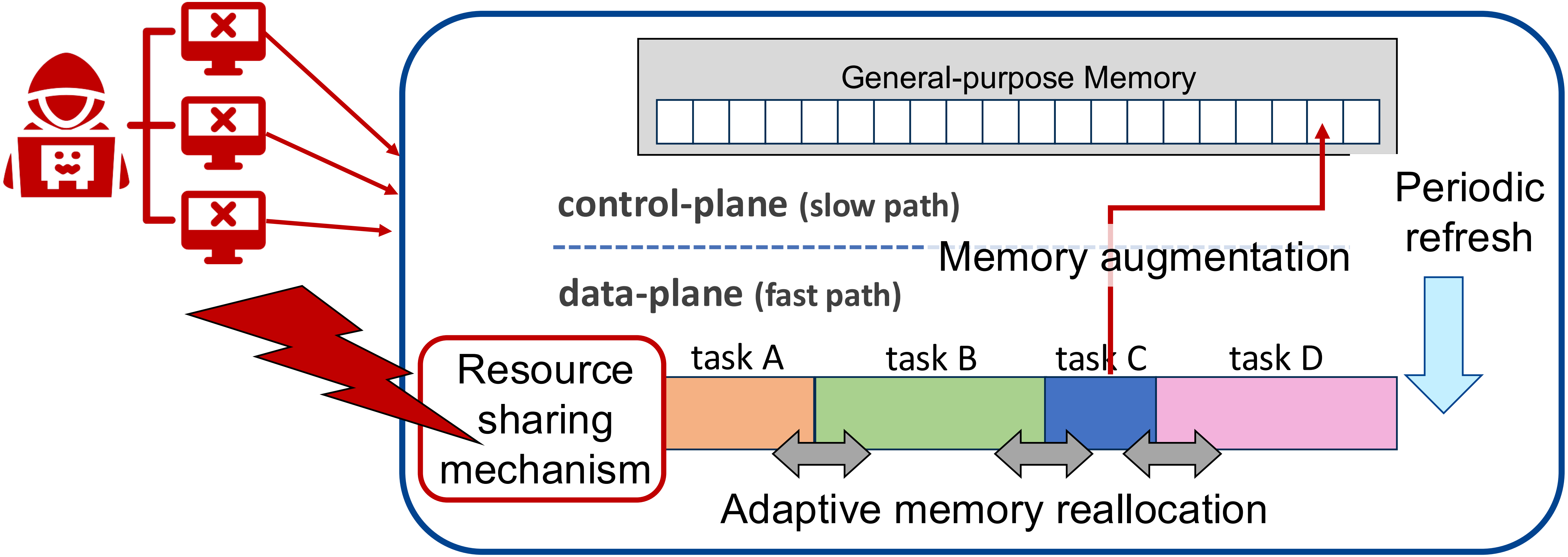
The *Heracles* attack ***exploits data-plane optimizations*** to evade DoS detection

- The Heracles attack



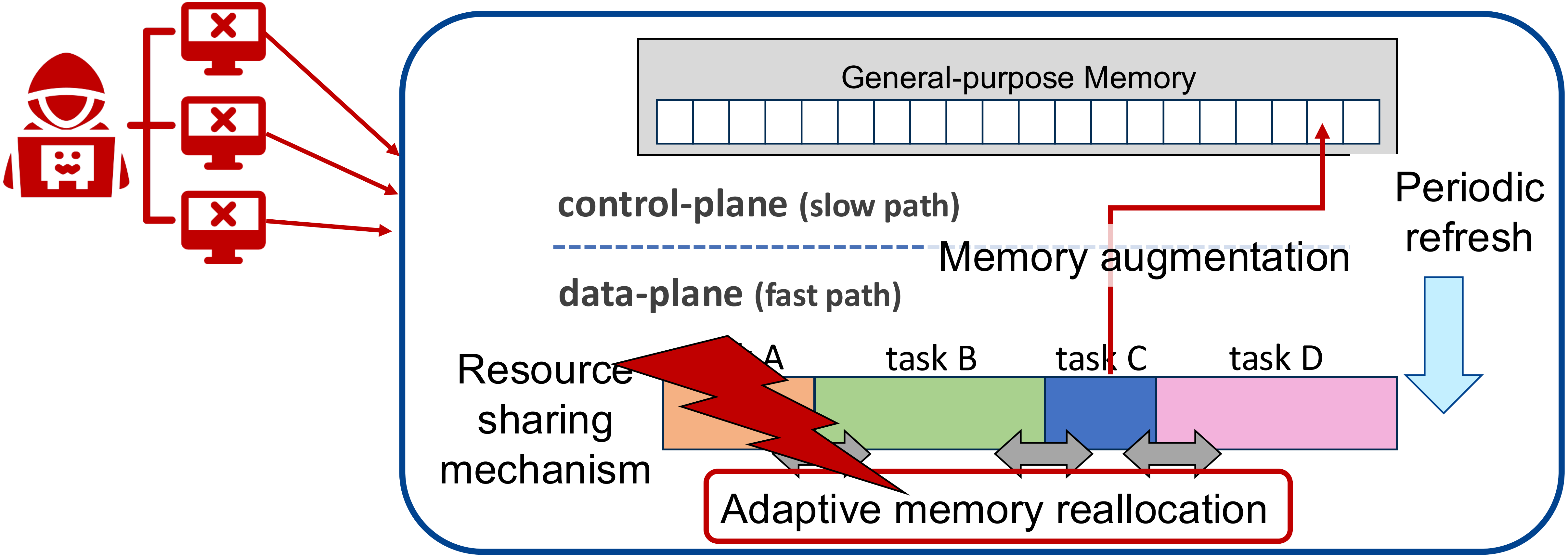
The ***Heracles*** attack ***exploits data-plane optimizations*** to evade DoS detection

- The Heracles attack



The ***Heracles*** attack ***exploits data-plane optimizations*** to evade DoS detection

- The Heracles attack



General-purpose Memory

control-plane (slow path)

data-plane (fast path)

Memory augmentation

Periodic refresh

Resource sharing mechanism

task A

task B

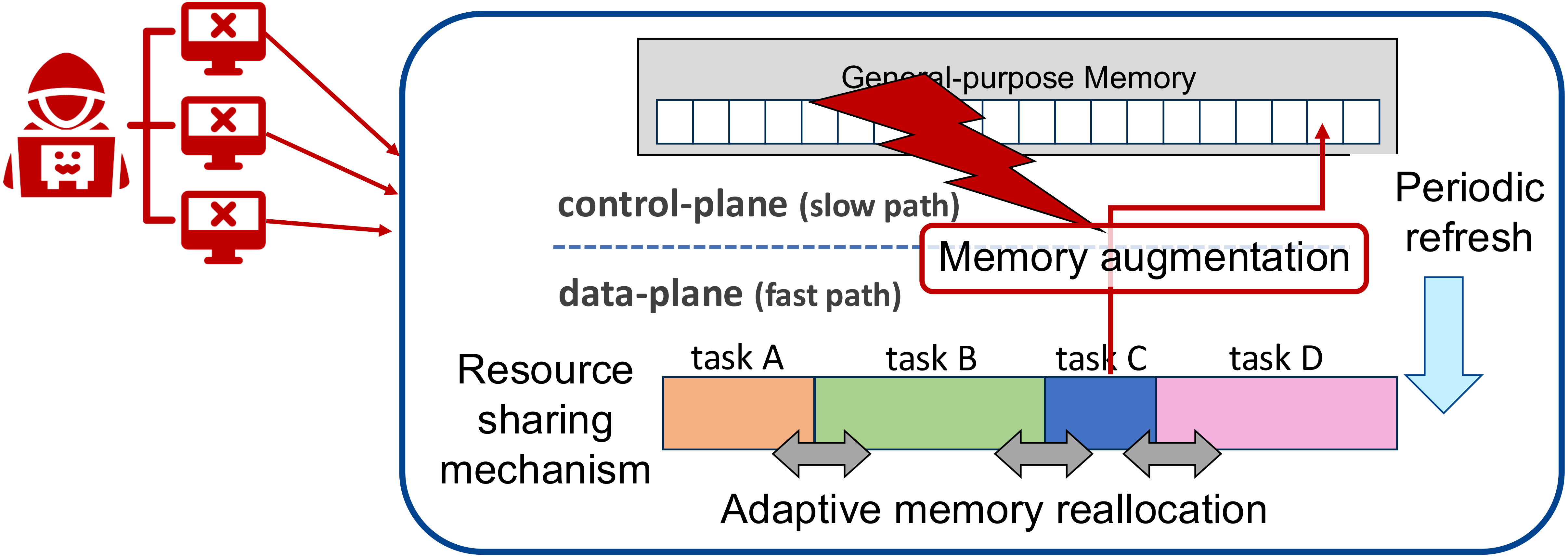
task C

task D

Adaptive memory reallocation

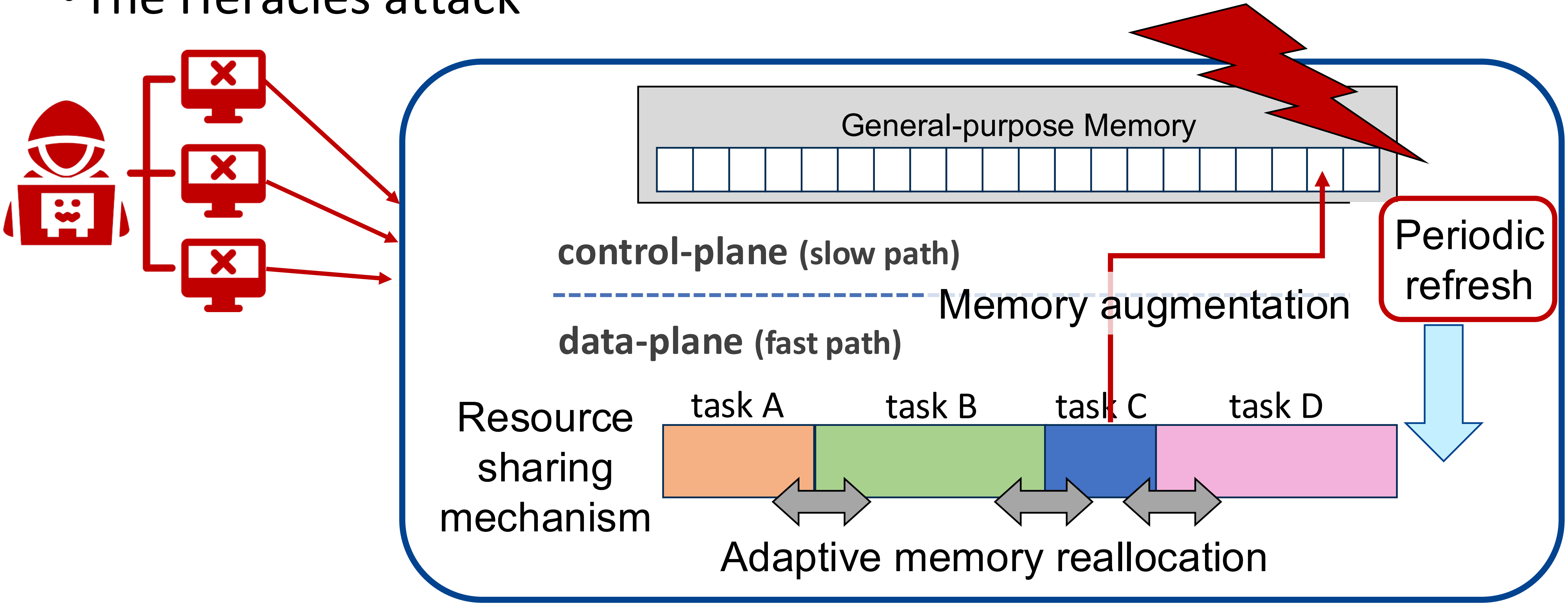
The ***Heracles*** attack ***exploits data-plane optimizations*** to evade DoS detection

- The Heracles attack



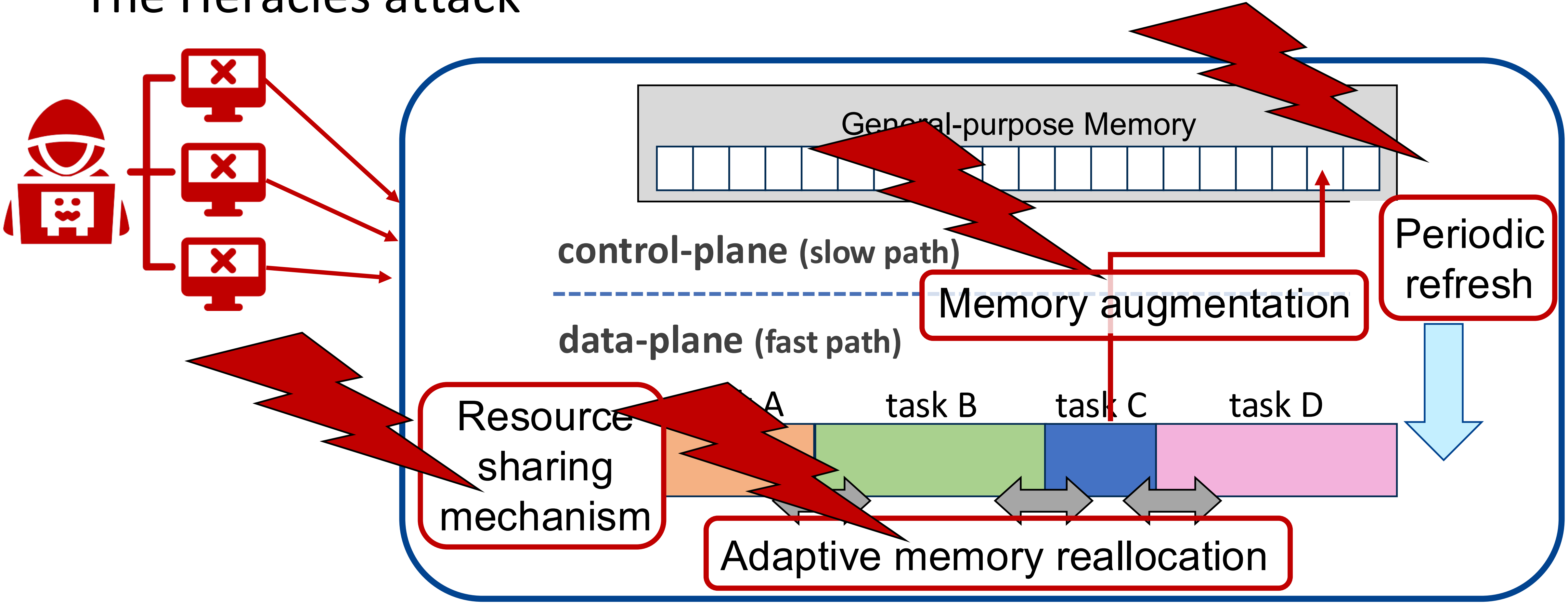
The *Heracles* attack ***exploits data-plane optimizations*** to evade DoS detection

- The Heracles attack



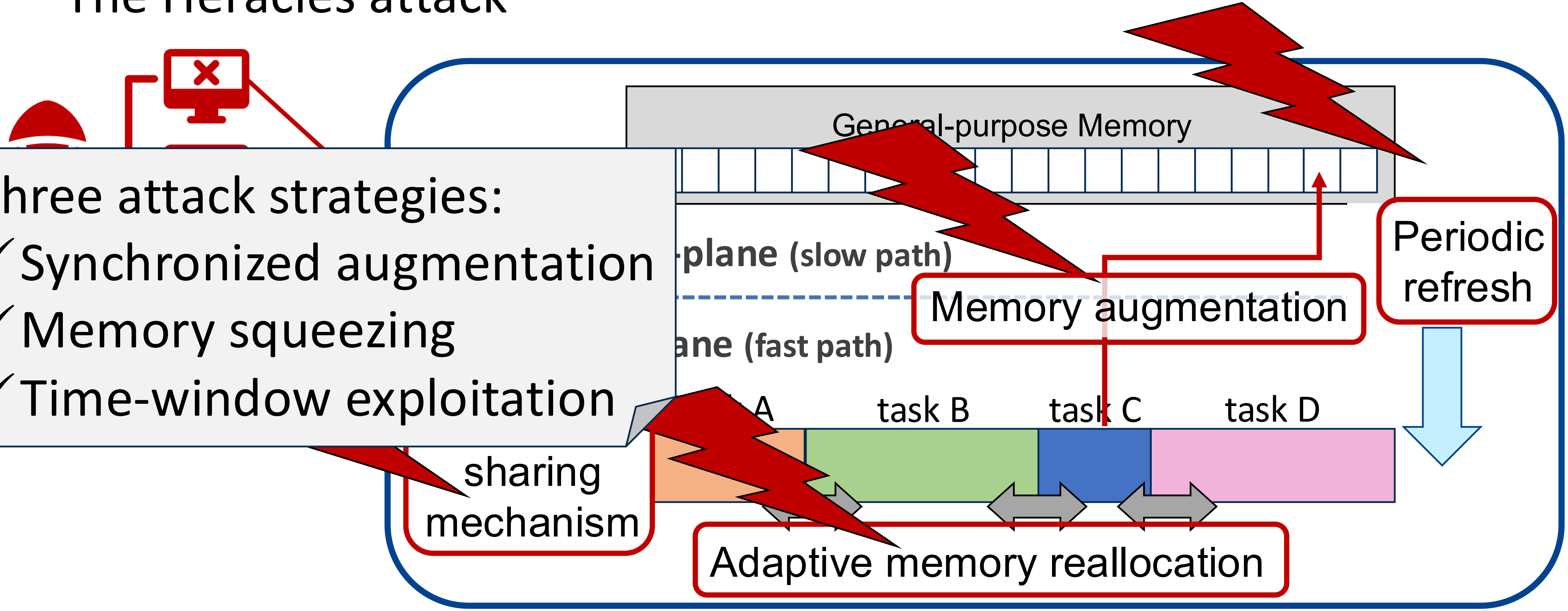
The *Heracles* attack ***exploits data-plane optimizations*** to evade DoS detection

- The Heracles attack



The ***Heracles*** attack ***exploits data-plane optimizations*** to evade DoS detection

- The Heracles attack



Three attack strategies:

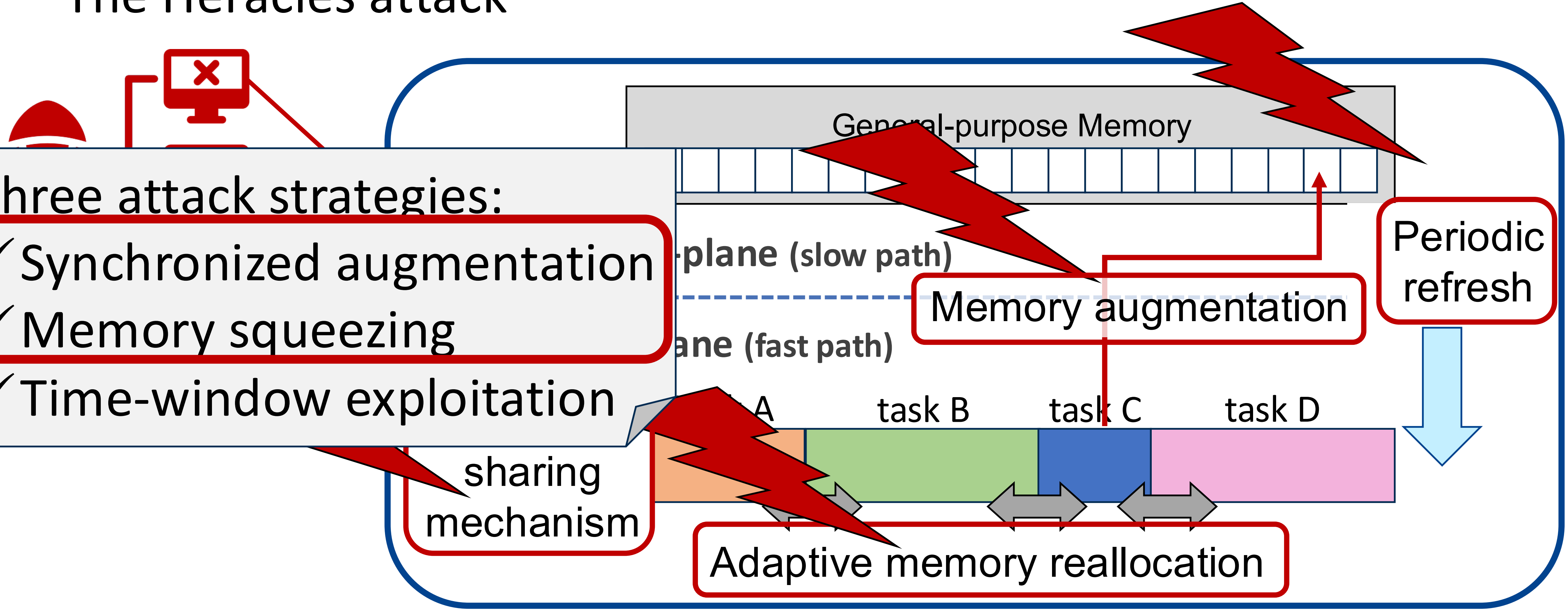
✓Synchronized augmentation

✓Memory squeezing

✓Time-window exploitation

The *Heracles* attack *exploits data-plane optimizations* to evade DoS detection

- The Heracles attack



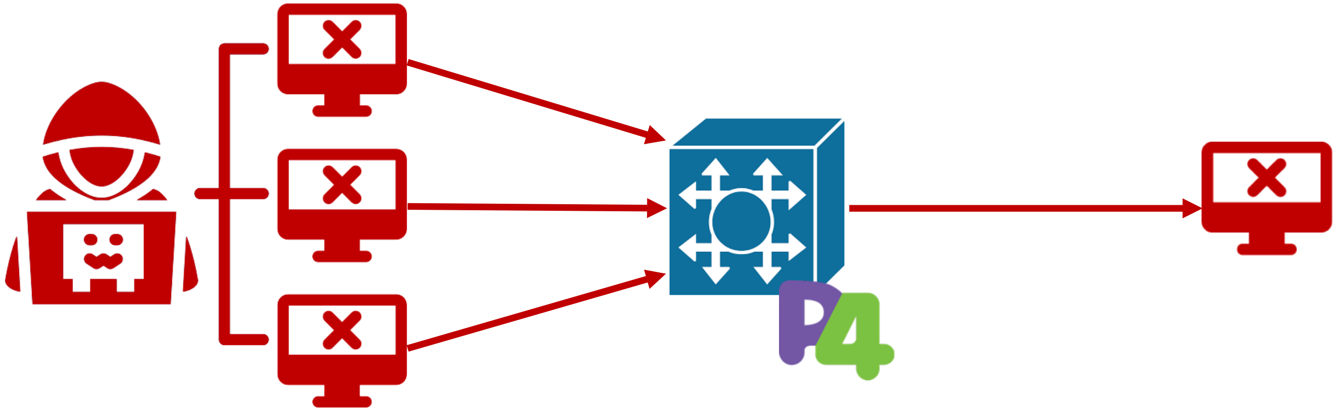
Three attack strategies:

✓ Synchronized augmentation

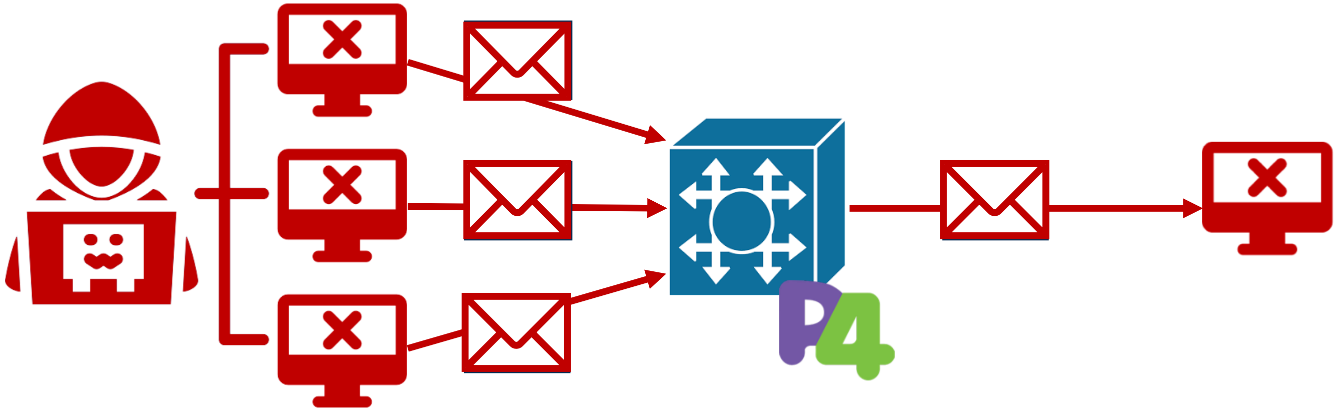
✓ Memory squeezing

✓ Time-window exploitation

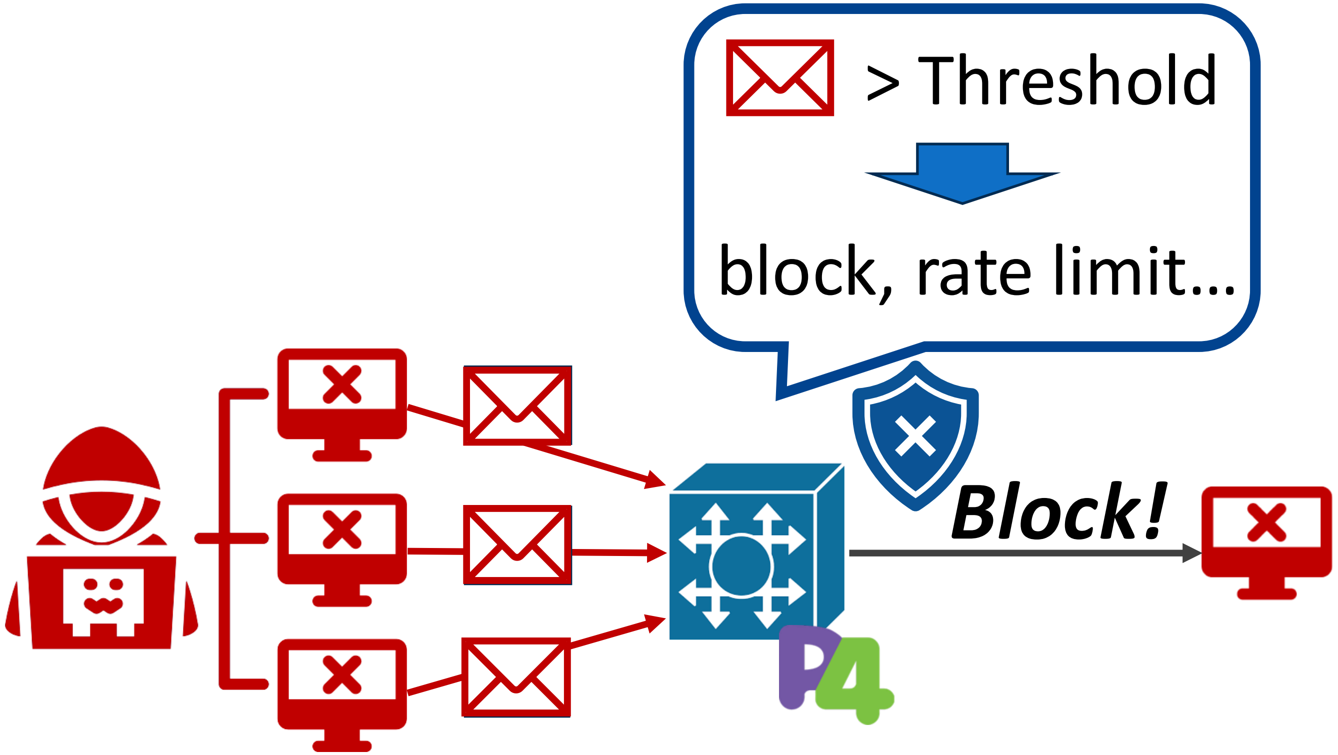
Attack preparation: Exploiting ***sketch-based filtering*** to infer ***internal timings***



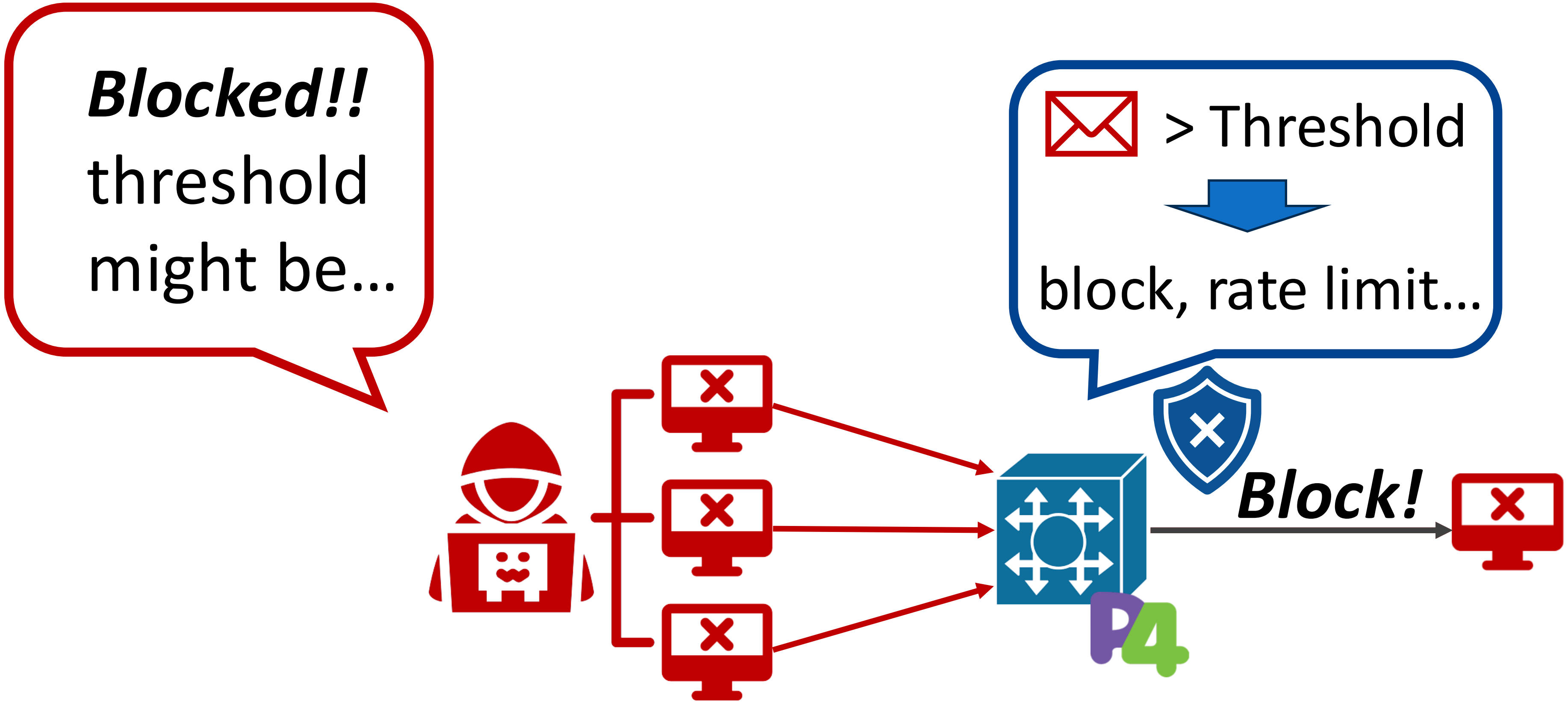
Attack preparation: Exploiting ***sketch-based filtering*** to infer ***internal timings***



Attack preparation: Exploiting ***sketch-based filtering*** to infer ***internal timings***

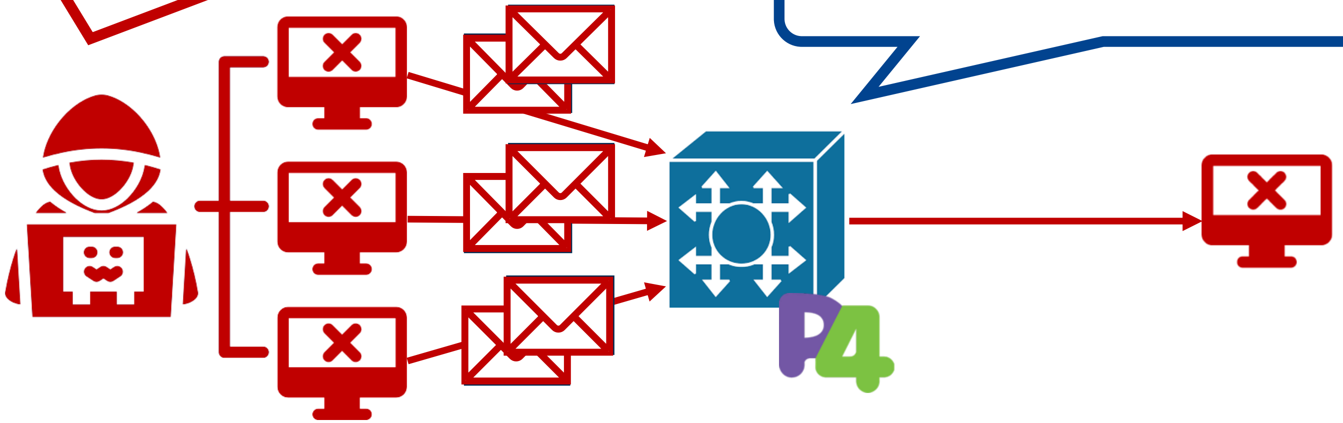


Attack preparation: Exploiting ***sketch-based filtering*** to infer ***internal timings***



Attack preparation: Exploiting ***sketch-based filtering*** to infer ***internal timings***

Adjust sending rate (PPS) :
Time-window period P might be $\frac{T}{PPS_{high}} \leq P \leq \frac{T}{PPS_{low}}$

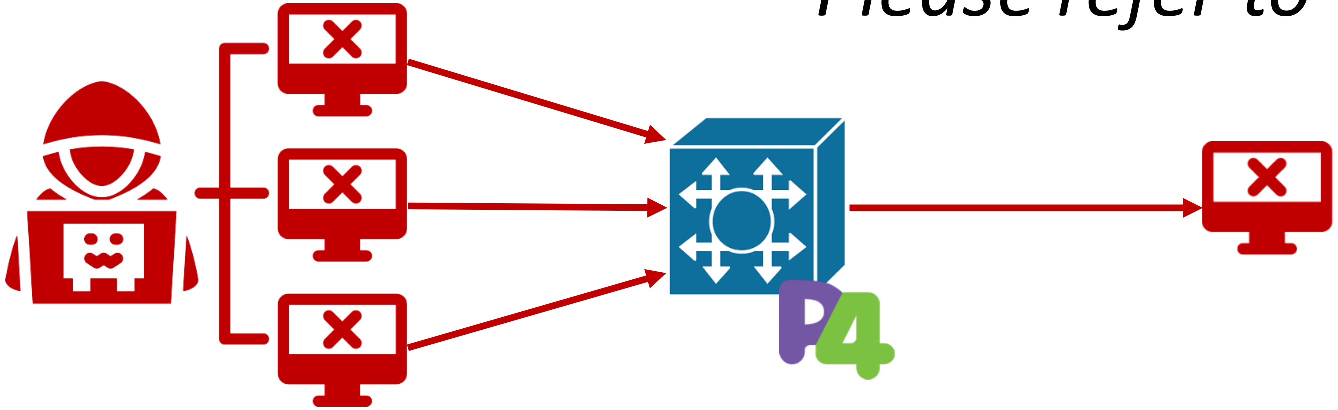


Periodic sketch refresh

Attack preparation: Exploiting ***sketch-based filtering*** to infer ***internal timings***

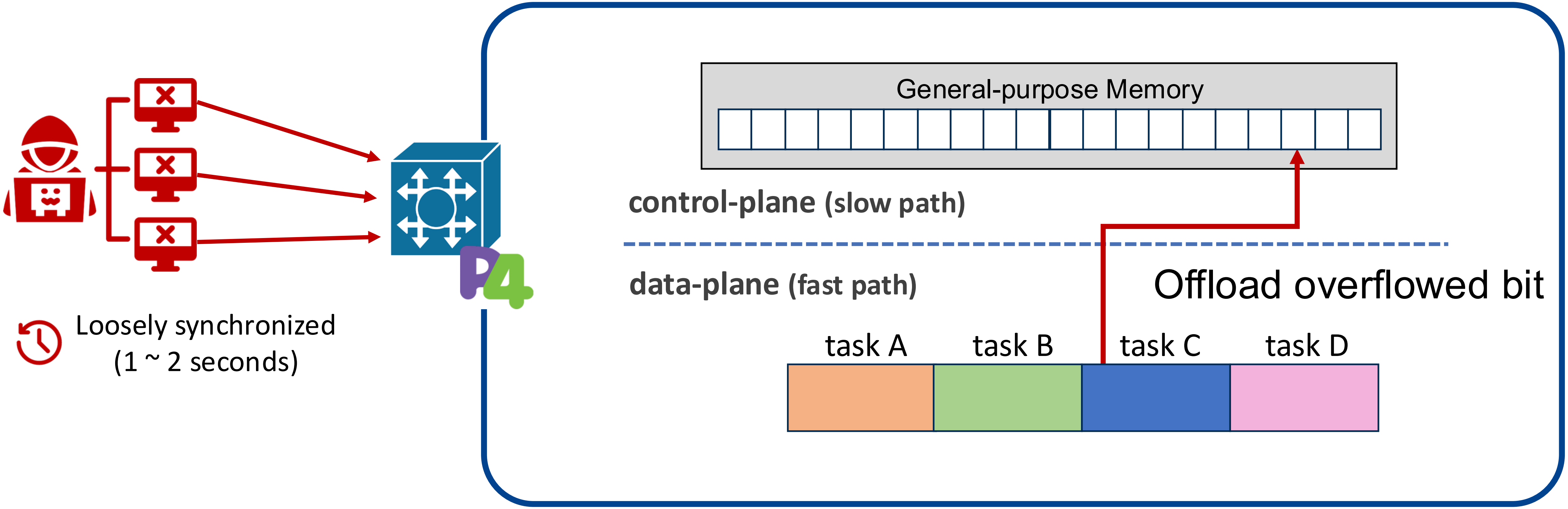
- Number of time-window
- Starting time of time-window
- Register-sharing tasks

Please refer to § III-A of our paper



Attack strategy 1: Synchronized augmentation exploits resource sharing and augmentation

- Overwhelm ***data-to-control plane path***



Loosely synchronized (1 ~ 2 seconds)

General-purpose Memory

control-plane (slow path)

data-plane (fast path)

Offload overflowed bit

task A

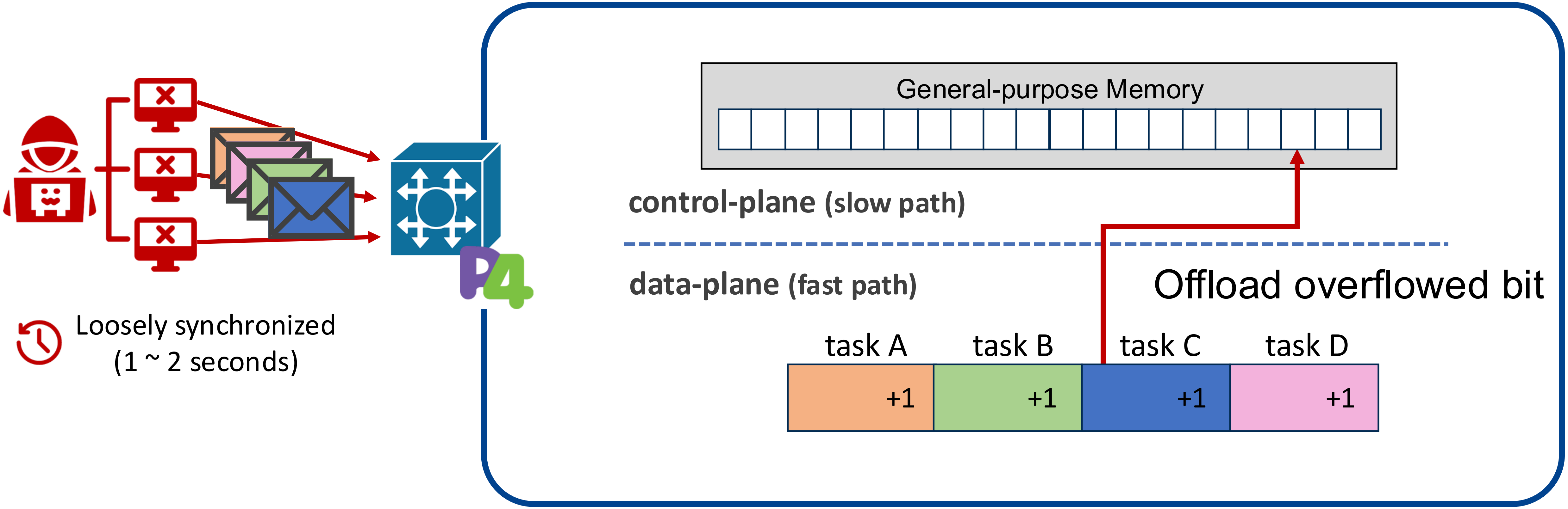
task B

task C

task D

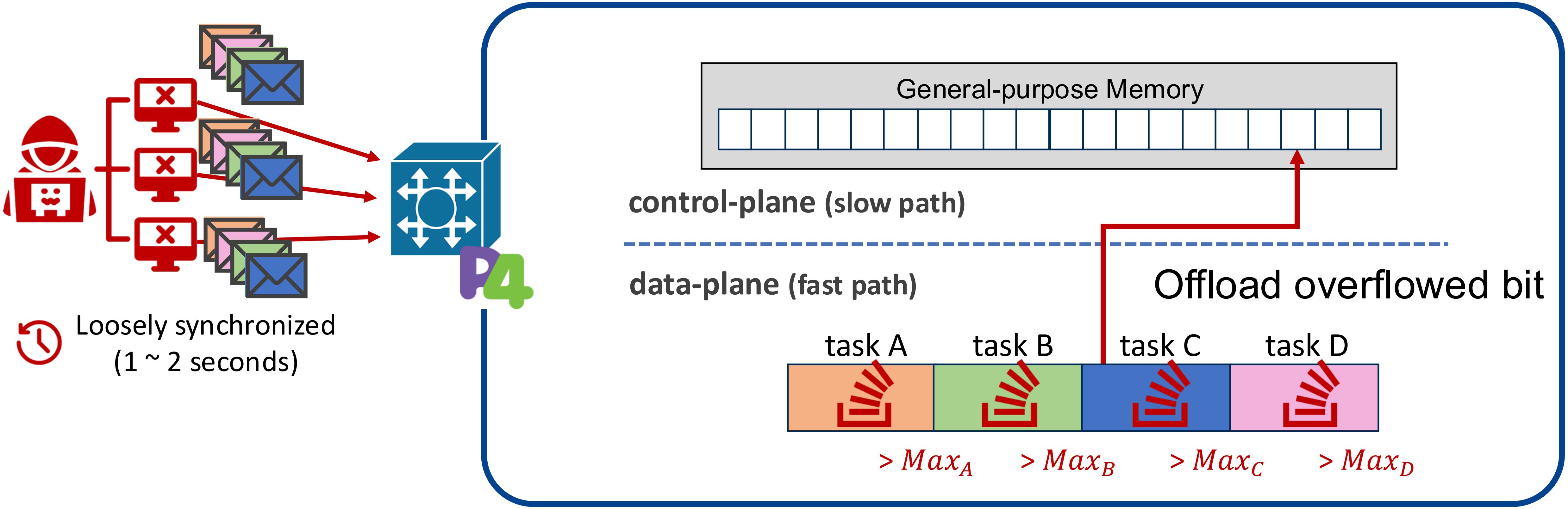
Attack strategy 1: Synchronized augmentation exploits resource sharing and augmentation

- Overwhelm ***data-to-control plane path***



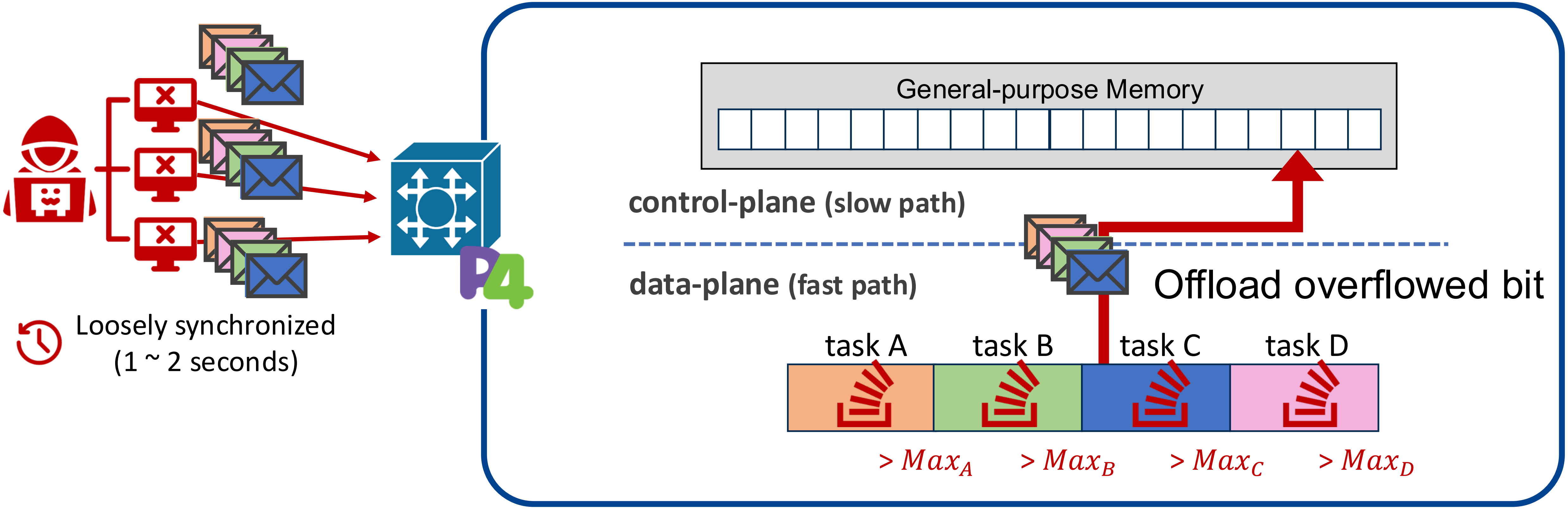
Attack strategy 1: Synchronized augmentation exploits resource sharing and augmentation

- Overwhelm ***data-to-control plane path***



Attack strategy 1: Synchronized augmentation exploits resource sharing and augmentation

- Overwhelm ***data-to-control plane path***



Loosely synchronized (1 ~ 2 seconds)

General-purpose Memory

control-plane (slow path)

data-plane (fast path)

Offload overflowed bit

task A

task B

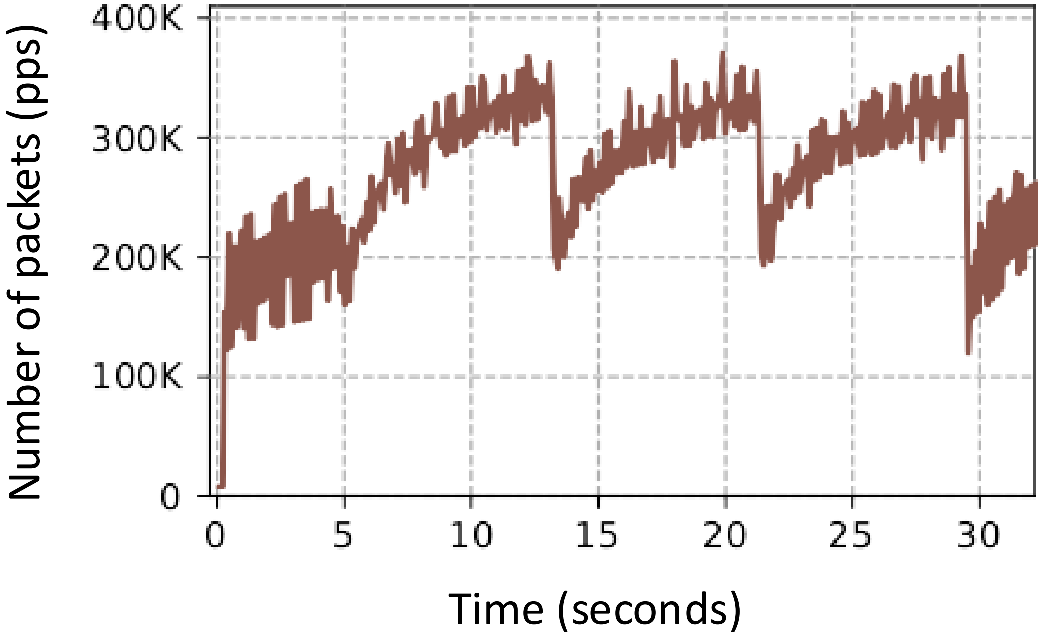
task C

task D

$> Max_A$ $> Max_B$ $> Max_C$ $> Max_D$

Attack strategy 1: Synchronized augmentation exploits resource sharing and augmentation

- Overwhelm ***data-to-control plane path***

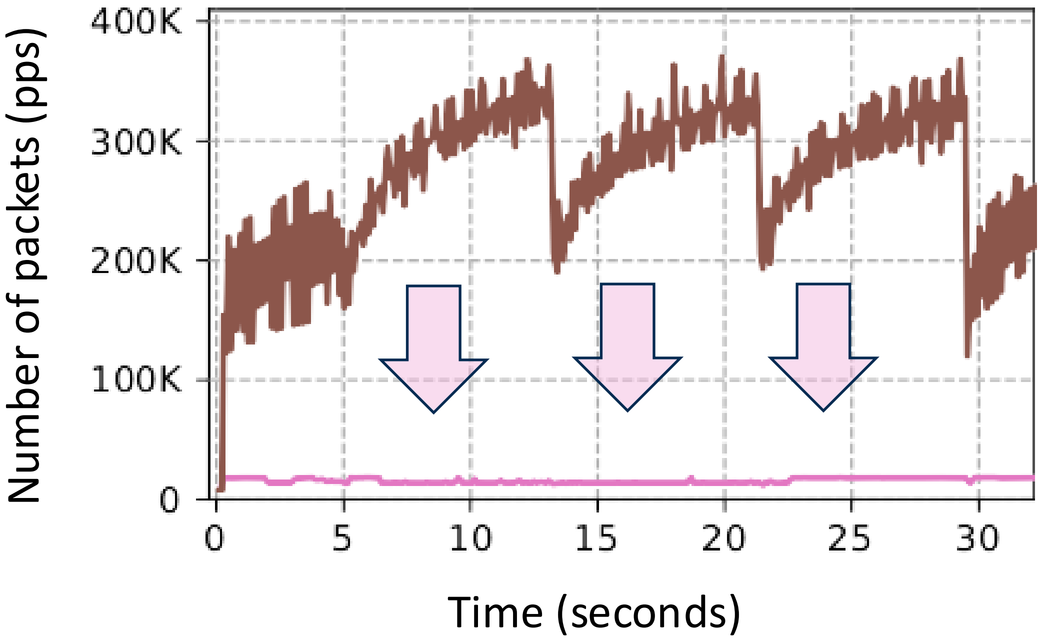


(a) Number of uploaded and processed packets

— Number of overflowed packet to control plane

Attack strategy 1: Synchronized augmentation exploits resource sharing and augmentation

- Overwhelm ***data-to-control plane path***

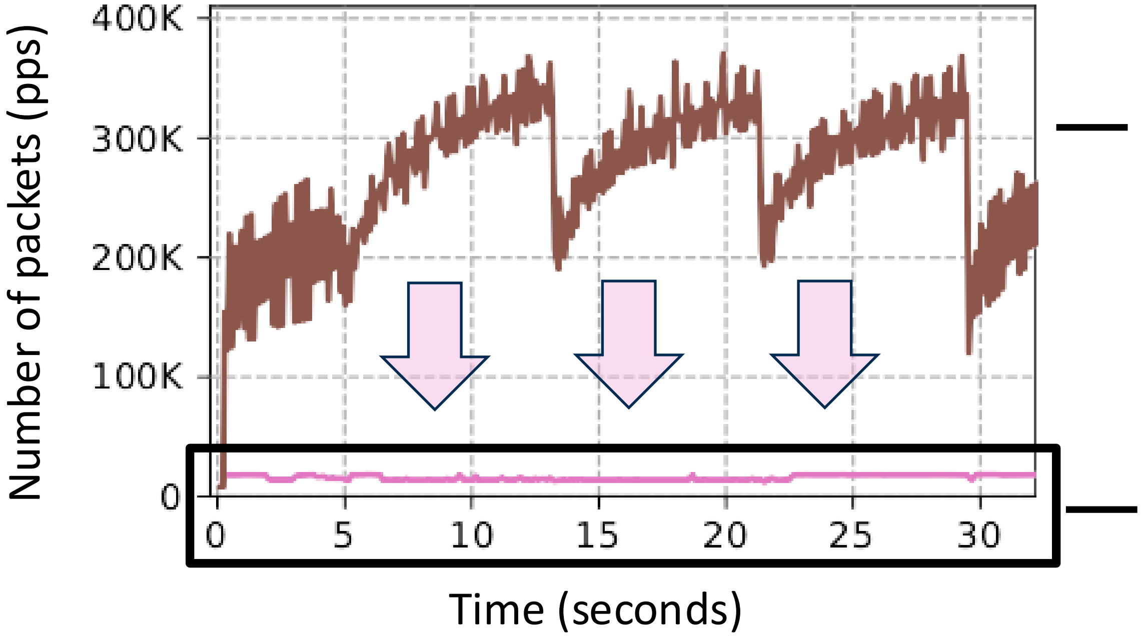


(a) Number of uploaded and processed packets

— Number of overflowed packet to control plane

Attack strategy 1: Synchronized augmentation exploits resource sharing and augmentation

- Overwhelm ***data-to-control plane path***



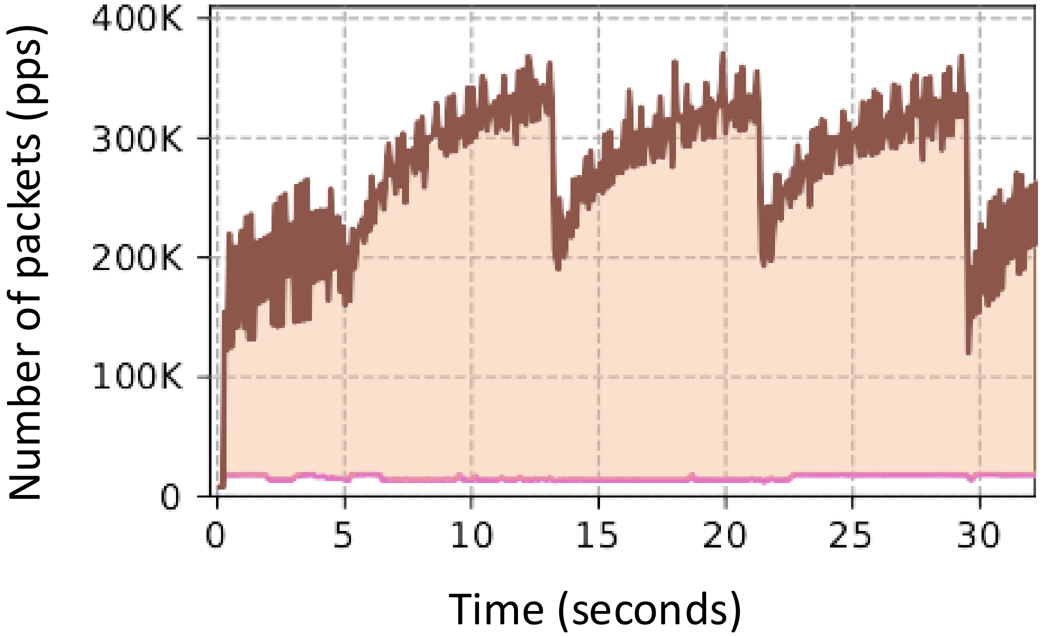
Number of overflowed packet to control plane

Number of processed packets in control plane

(a) Number of uploaded and processed packets

Attack strategy 1: Synchronized augmentation exploits resource sharing and augmentation

- Overwhelm ***data-to-control plane path***

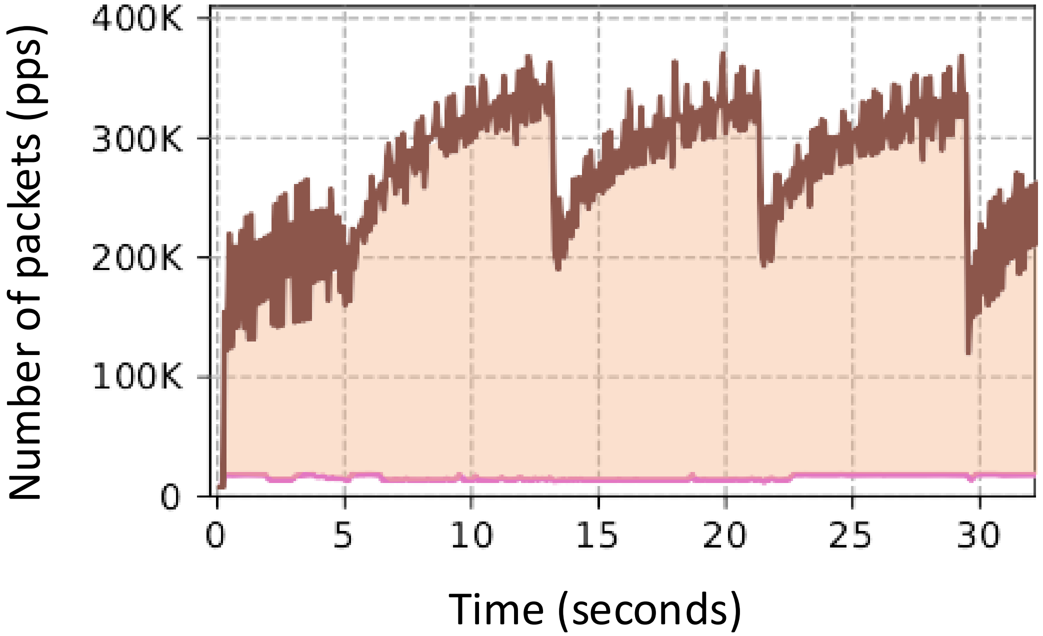


— ***Dropped***

(a) Number of uploaded and processed packets

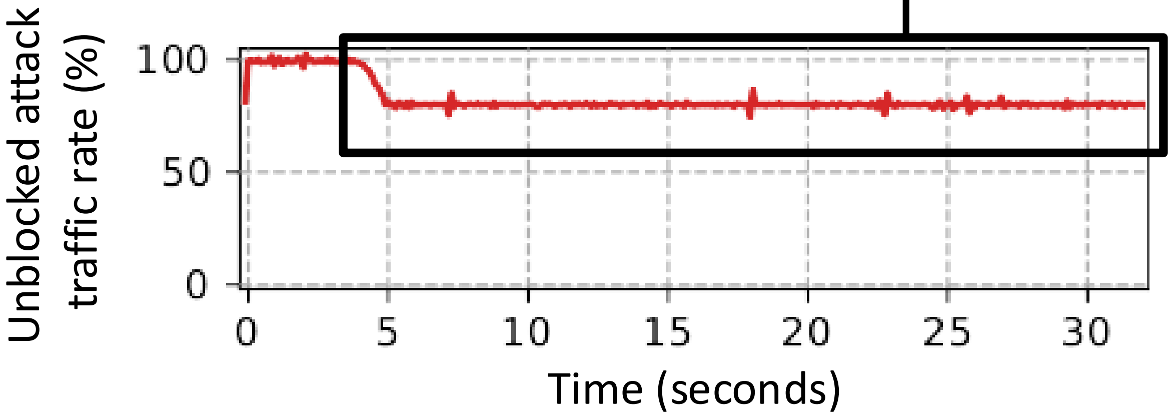
Attack strategy 1: Synchronized augmentation exploits resource sharing and augmentation

- Overwhelm ***data-to-control plane path => undercounting!***



(a) Number of uploaded and processed packets

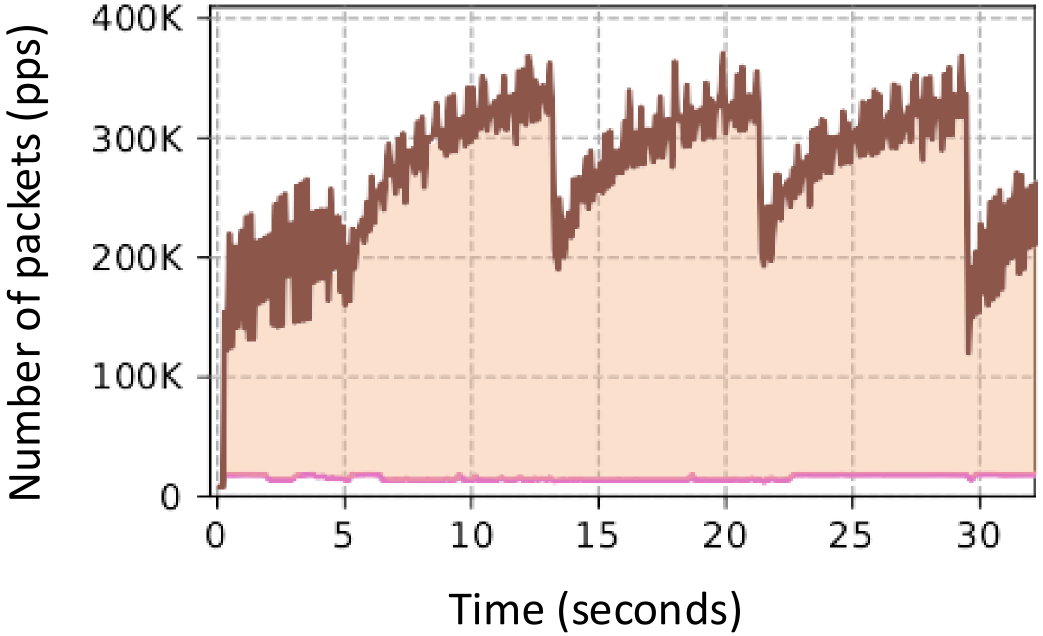
Undercounted packets causing about 78% of FNR



(b) DoS mitigation result

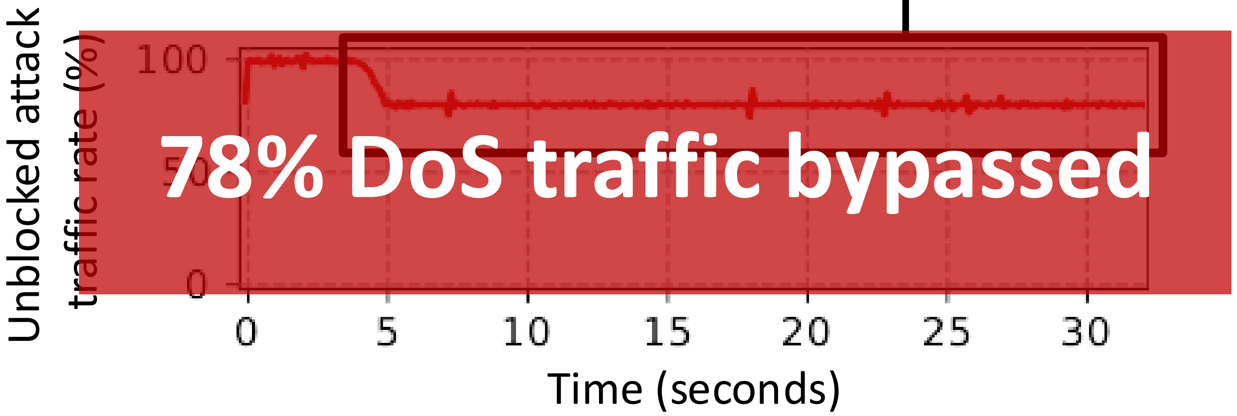
Attack strategy 1: Synchronized augmentation exploits resource sharing and augmentation

- Overwhelm ***data-to-control plane path => undercounting!***



(a) Number of uploaded and processed packets

Undercounted packets causing about 78% of FNR

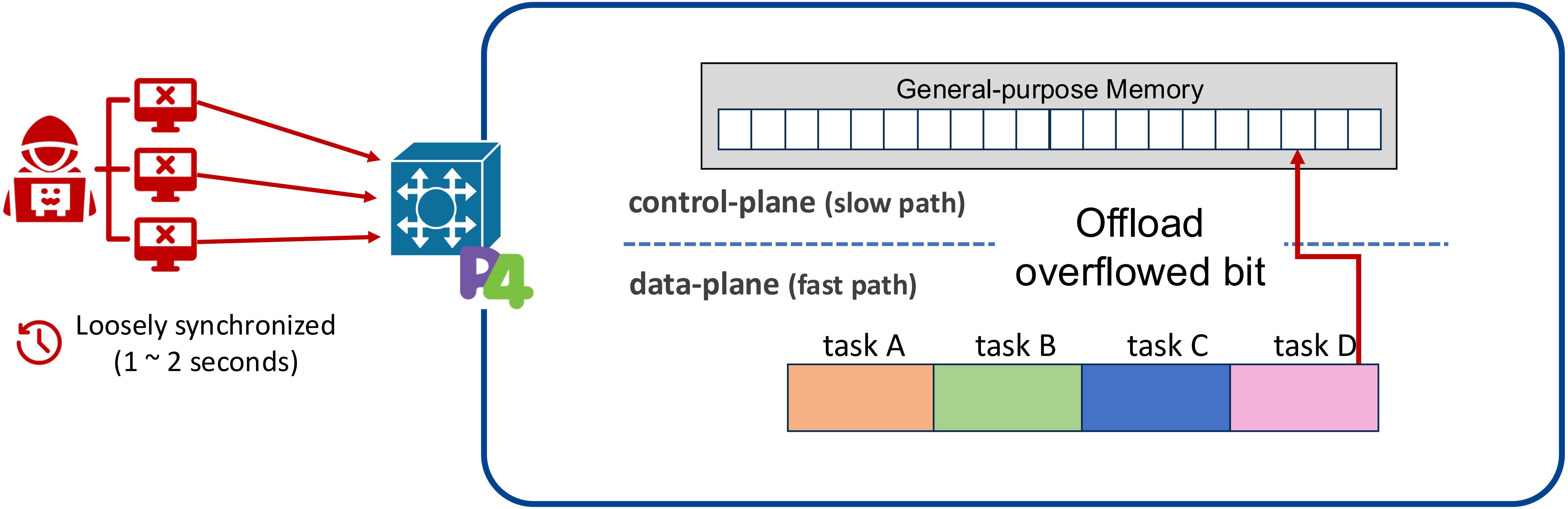


78% DoS traffic bypassed

(b) DoS mitigation result

Attack strategy 2: Memory squeezing exploits adaptive resource sharing

- Squeeze target resource ***to trigger dramatic overflow***



General-purpose Memory

control-plane (slow path)

data-plane (fast path)

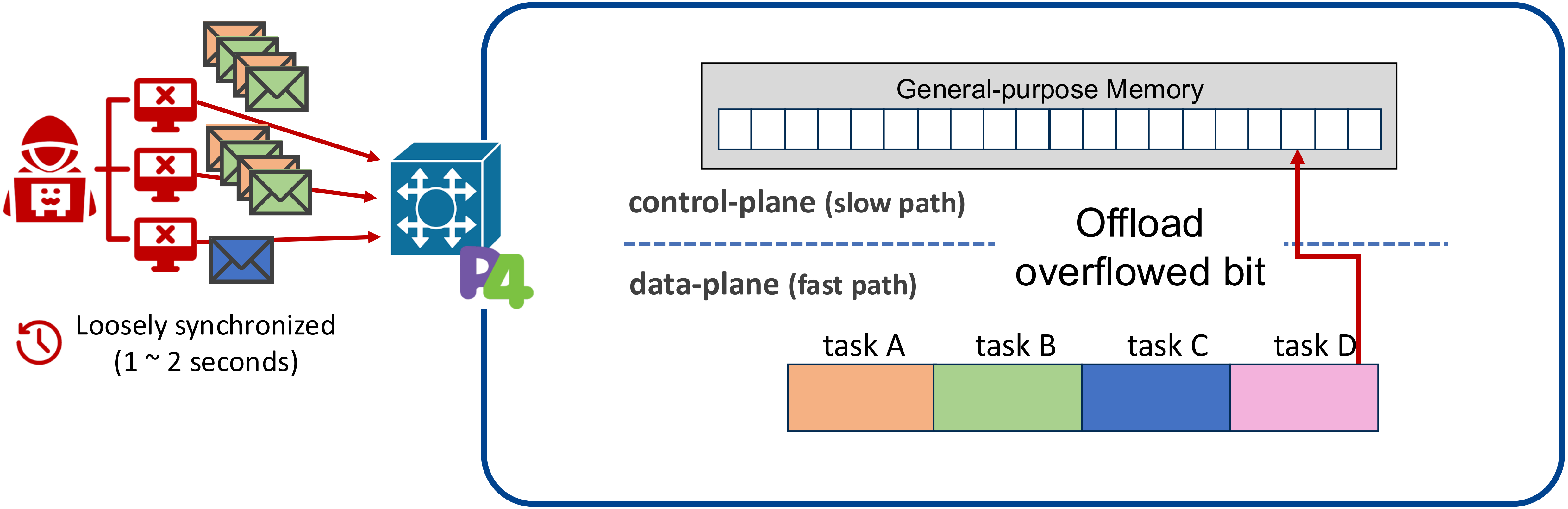
Offload overflowed bit

task A	task B	task C	task D

Loosely synchronized
(1 ~ 2 seconds)

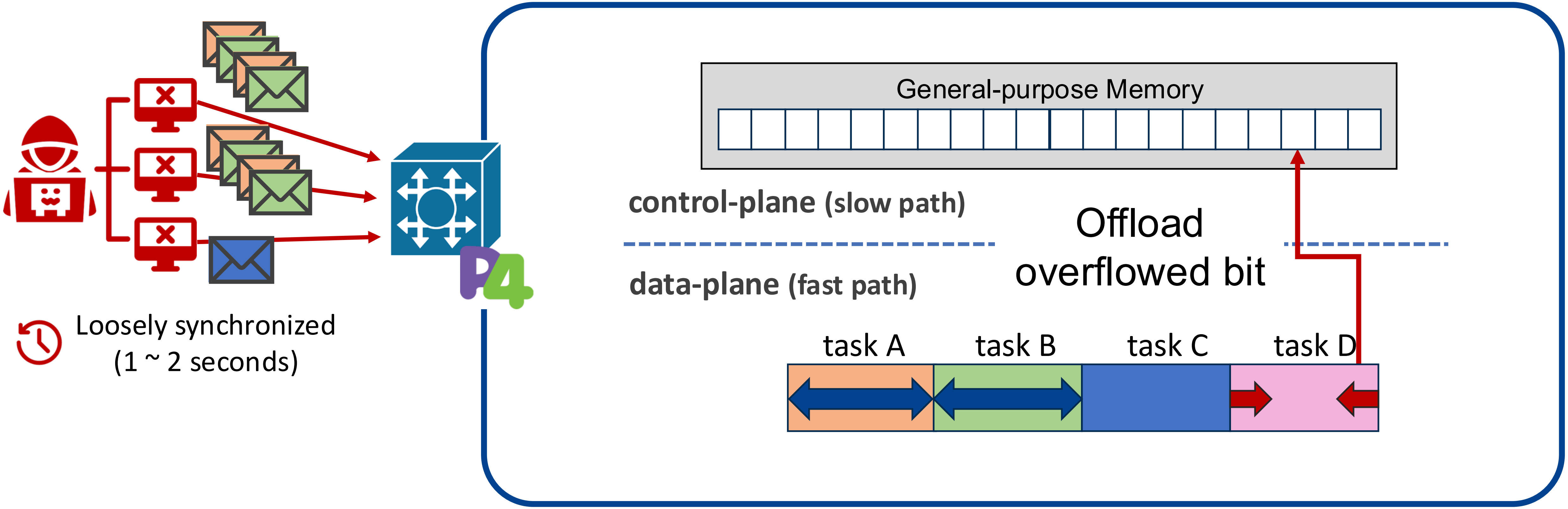
Attack strategy 2: Memory squeezing exploits adaptive resource sharing

- Squeeze target resource ***to trigger dramatic overflow***



Attack strategy 2: Memory squeezing exploits adaptive resource sharing

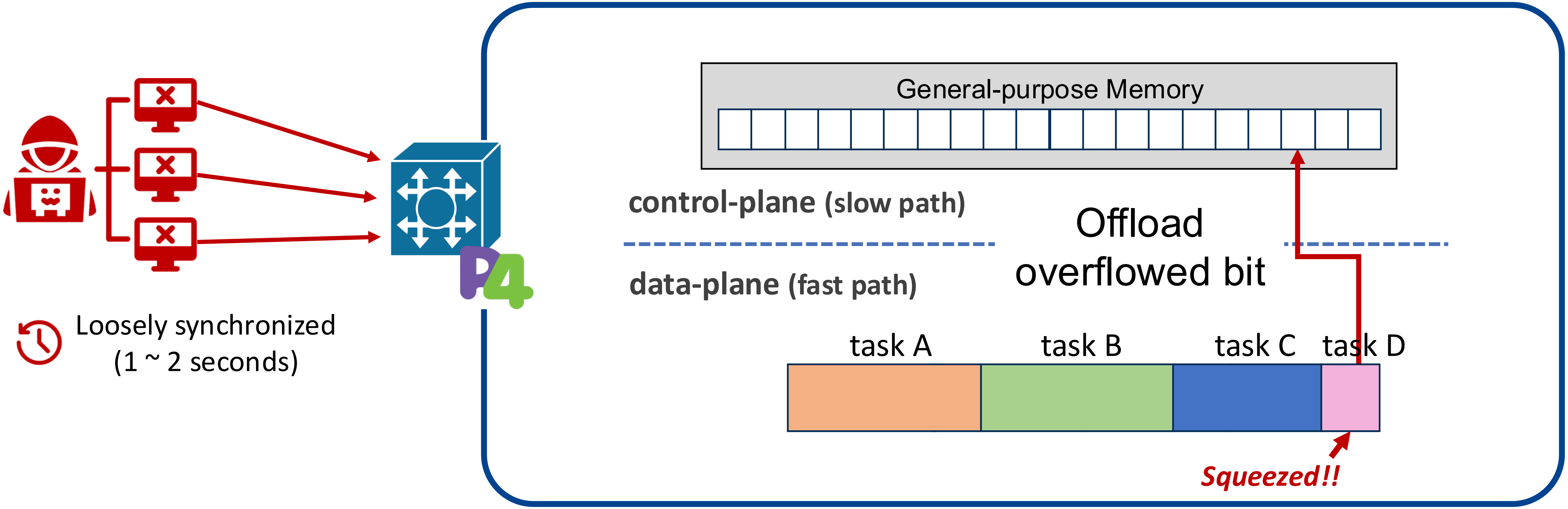
- Squeeze target resource ***to trigger dramatic overflow***



Loosely synchronized
(1 ~ 2 seconds)

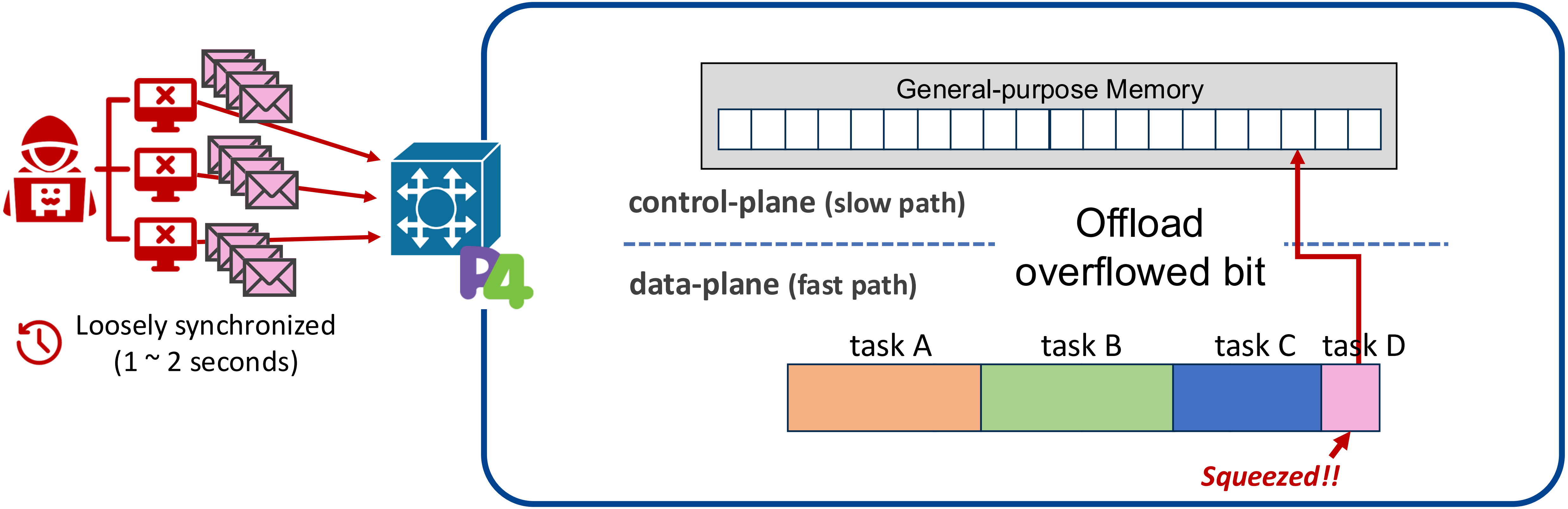
Attack strategy 2: Memory squeezing exploits adaptive resource sharing

- Squeeze target resource ***to trigger dramatic overflow***



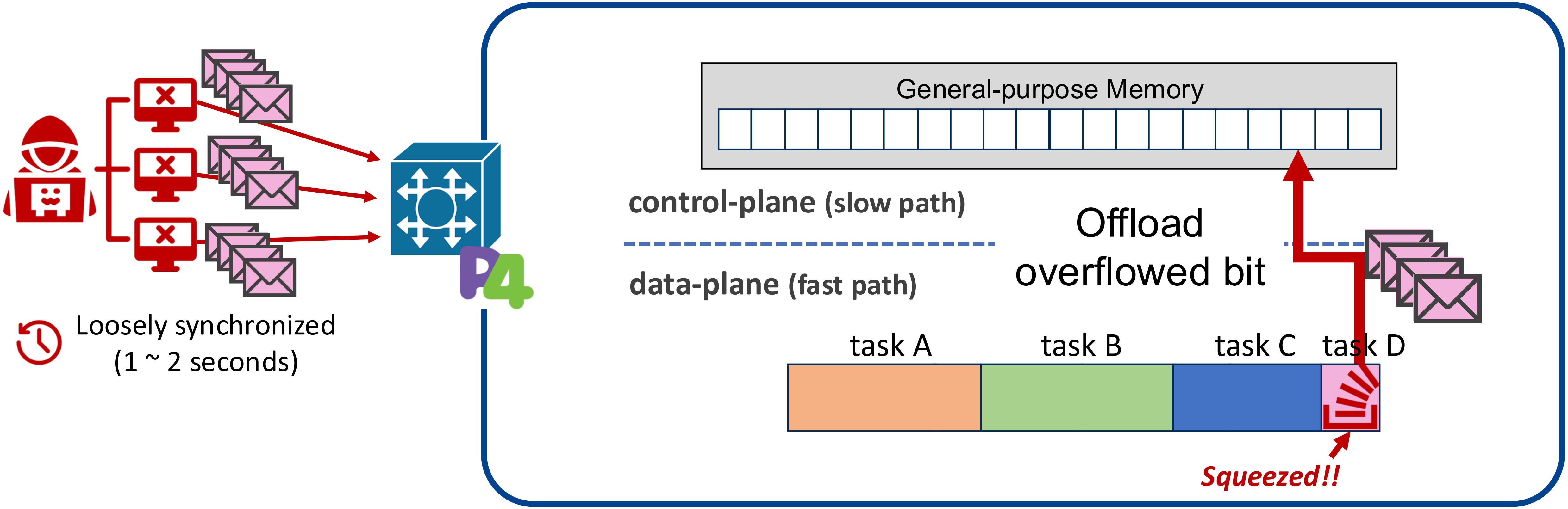
Attack strategy 2: Memory squeezing exploits adaptive resource sharing

- Squeeze target resource ***to trigger dramatic overflow***



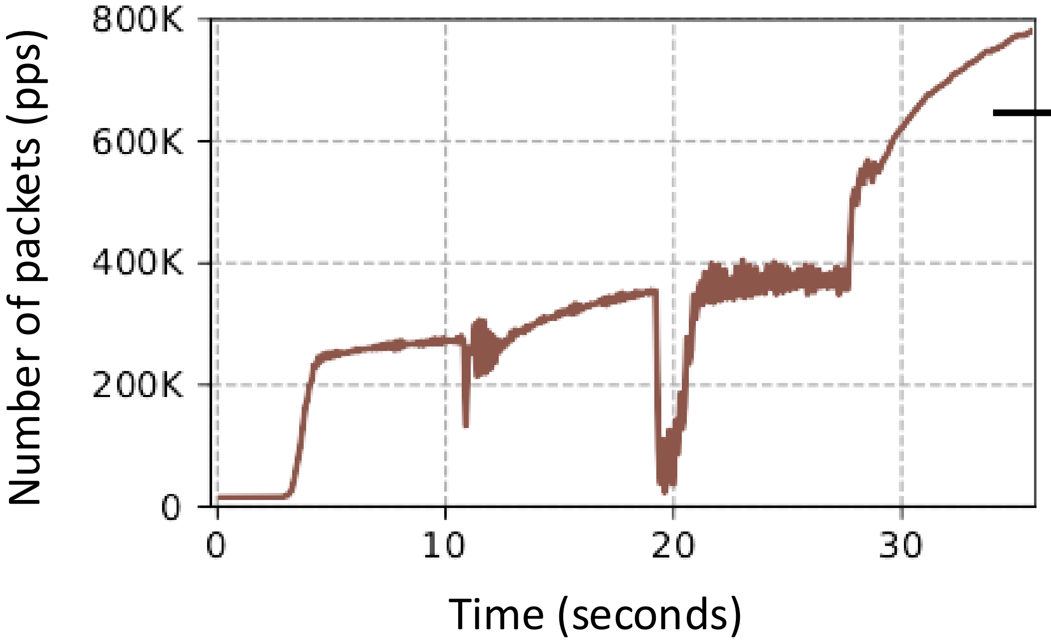
Attack strategy 2: Memory squeezing exploits adaptive resource sharing

- Squeeze target resource ***to trigger dramatic overflow***



Attack strategy 2: Memory squeezing exploits adaptive resource sharing

- Squeeze target resource ***to trigger dramatic overflow***

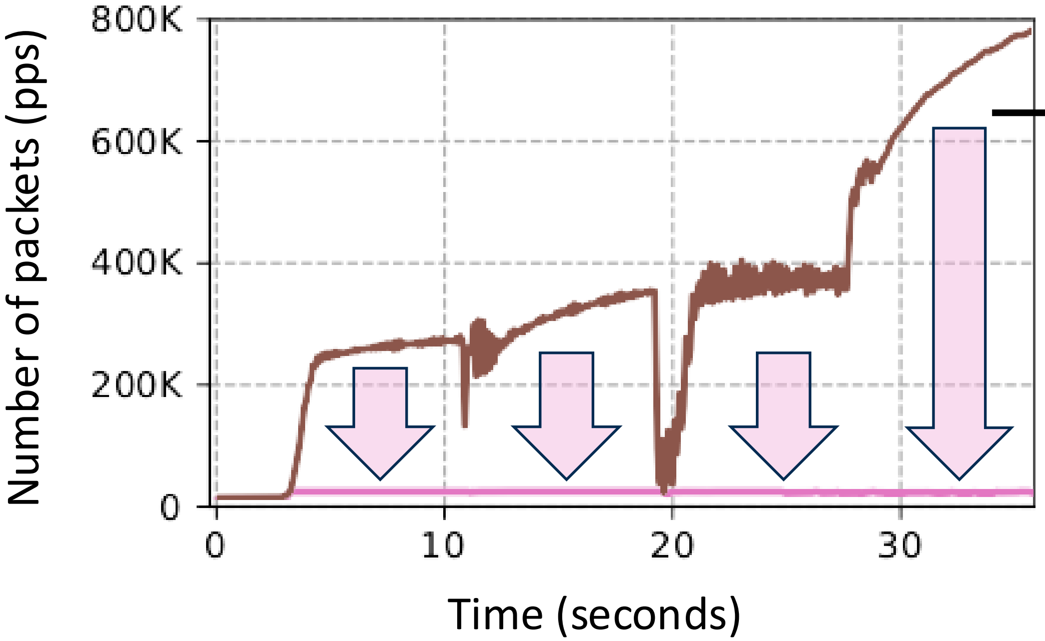


Number of overflowed packets

(a) Number of uploaded and processed packets

Attack strategy 2: Memory squeezing exploits adaptive resource sharing

- Squeeze target resource ***to trigger dramatic overflow***



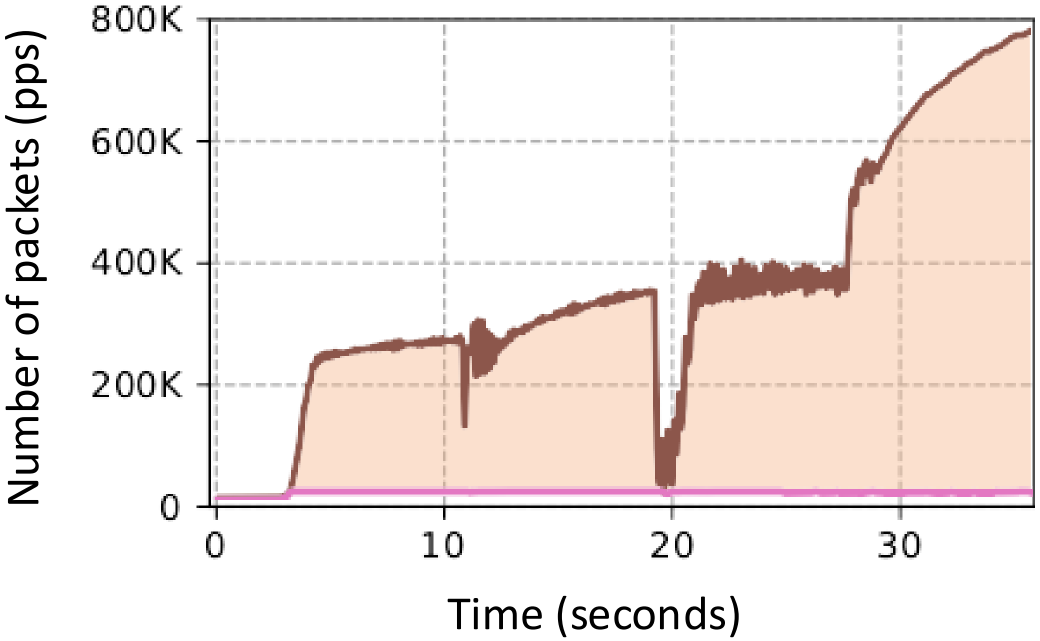
Number of overflowed packets

Number of processed packets in control plane

(a) Number of uploaded and processed packets

Attack strategy 2: Memory squeezing exploits adaptive resource sharing

- Squeeze target resource ***to trigger dramatic overflow***

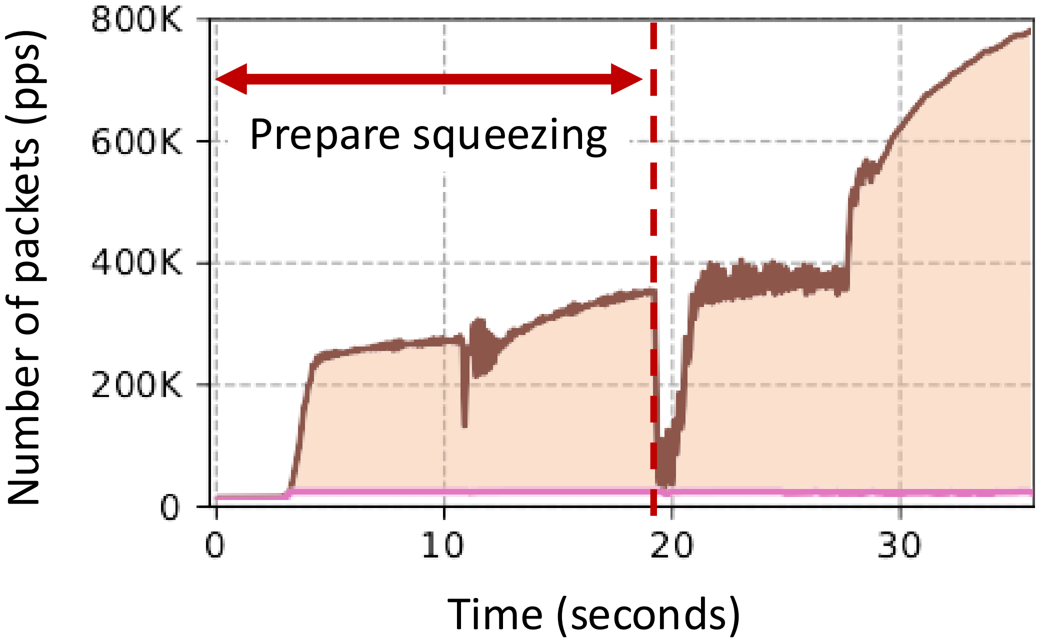


— ***Dropped***

(a) Number of uploaded and processed packets

Attack strategy 2: Memory squeezing
exploits adaptive resource sharing

- Squeeze target resource ***to trigger dramatic overflow***

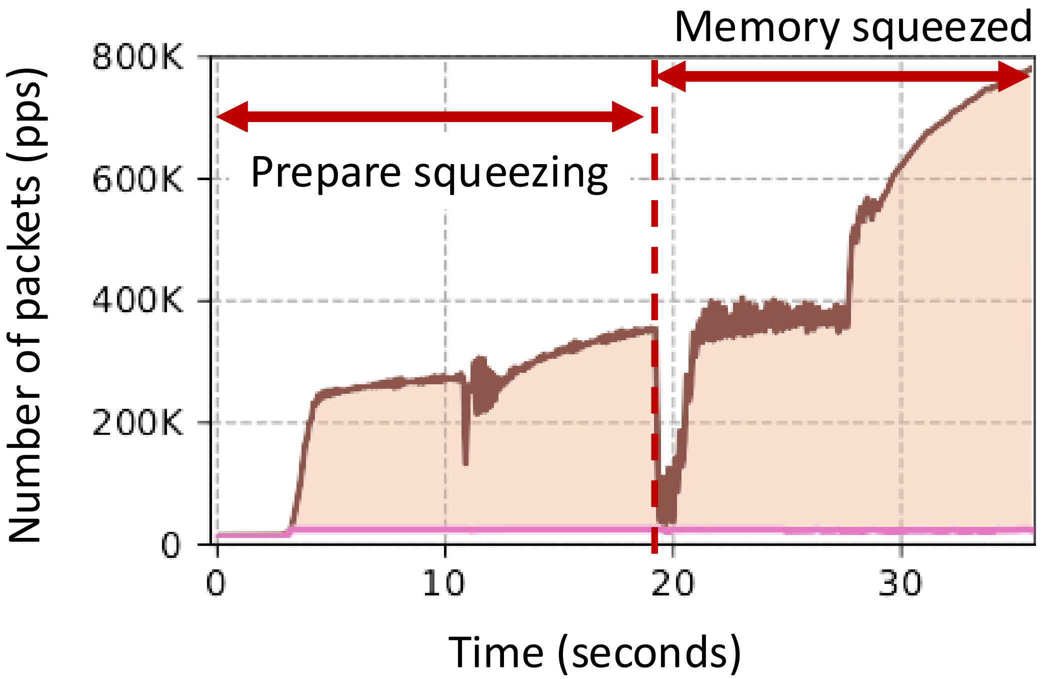


(a) Number of uploaded and processed packets

— ***Dropped***

Attack strategy 2: Memory squeezing exploits adaptive resource sharing

- Squeeze target resource ***to trigger dramatic overflow***

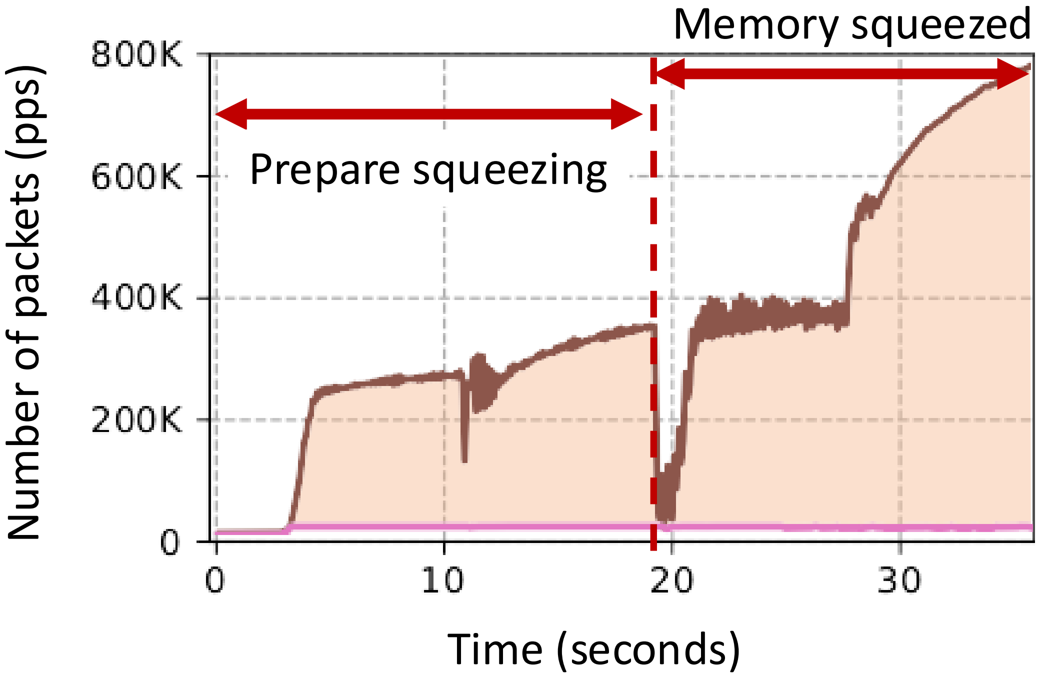


(a) Number of uploaded and processed packets

— ***Dropped***

Attack strategy 2: Memory squeezing exploits adaptive resource sharing

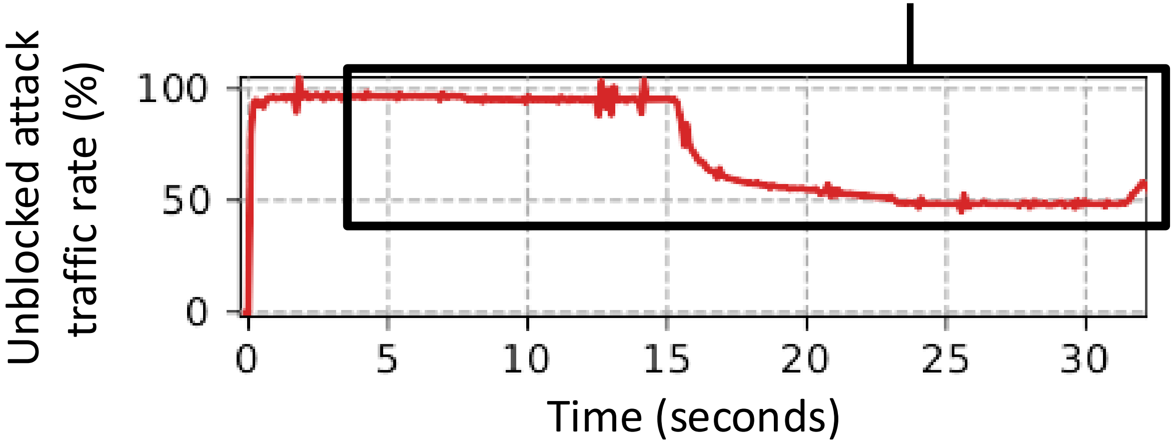
- Squeeze target resource ***to trigger dramatic overflow***



(a) Number of uploaded and processed packets

Undercounted packets causing about 50% of FNR

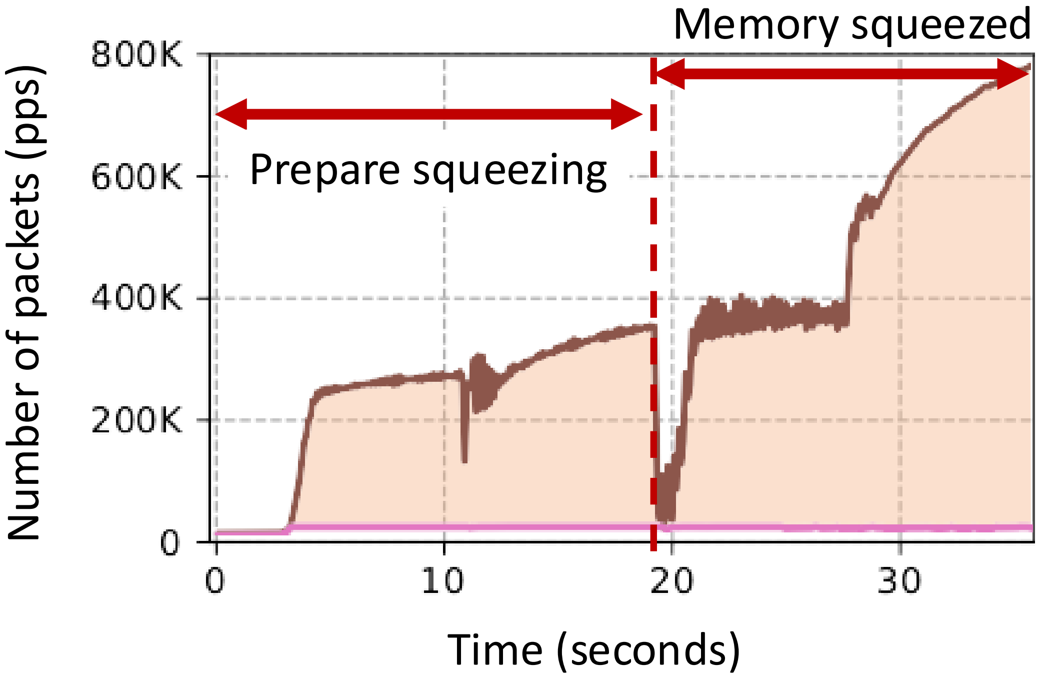
(Fail to block ~5Gbps malicious traffic)



(b) DoS mitigation result

Attack strategy 2: Memory squeezing exploits adaptive resource sharing

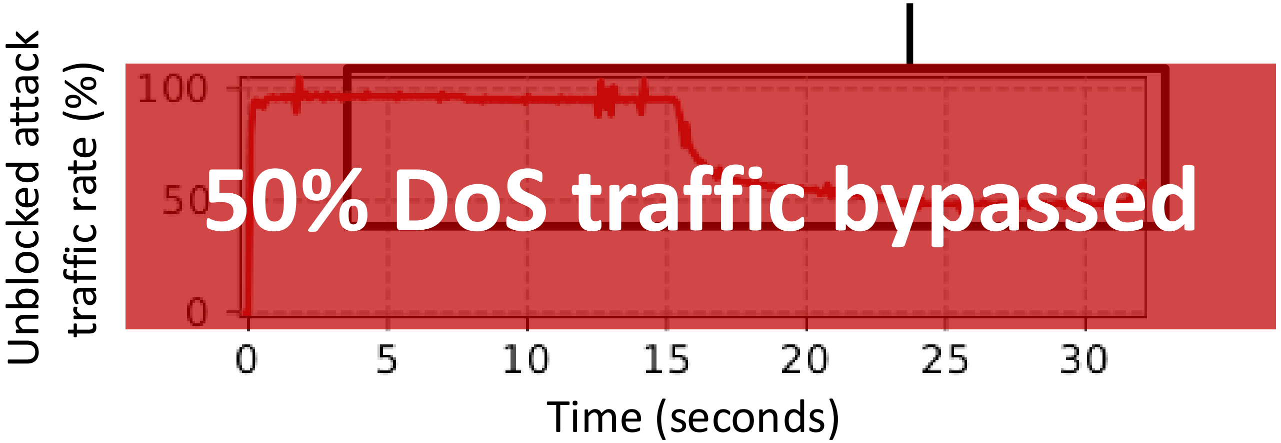
- Squeeze target resource ***to trigger dramatic overflow***



(a) Number of uploaded and processed packets

Undercounted packets causing about 50% of FNR

(Fail to block ~5Gbps malicious traffic)



(b) DoS mitigation result

Heracles exposes vulnerabilities in defenses *beyond* Cerberus

	Cerberus [SP'24]	Jaqen [Security'21]	Poseidon [NDSS'20]	Ripple [Security'21]	Mew [SP'23]
Inference Capabilities					
Inferring Detection Threshold	●	●	●	◐	◐
Inferring Windows Timing	●	●	●	◐	◐
Inferring Co-located Tasks	●	N/A	N/A	N/A	◐
Attack Feasibility					
Synchronized Augmentation	●	N/A	N/A	N/A	N/A
Memory Squeezing	●	N/A	N/A	N/A	◐
Time-window Exploitation	●	●	●	◐	◐

◐ Adversaries should send attack traffic along chosen paths across multiple switches.

Heracles exposes vulnerabilities in defenses ***beyond*** Cerberus

	Cerberus [SP’24]	Jaqen [Security’21]	Poseidon [NDSS’20]	Ripple [Security’21]	Mew [SP’23]
Inference Capabilities					
Inferring Detection Threshold	●	●	●	◐	◐
Inferring Windows Timing	●	●	●	◐	◐
Inferring Co-located Tasks	●	N/A	N/A	N/A	◐
Attack Feasibility					
Synchronized Augmentation	●	N/A	N/A	N/A	N/A
Memory Squeezing	●	N/A	N/A	N/A	◐
Time-window Exploitation	●	●	●	◐	◐

◐ Adversaries should send attack traffic along chosen paths across multiple switches.

Heracles exposes vulnerabilities in defenses *beyond* Cerberus

	Cerberus [SP’24]	Jaqen [Security’21]	Poseidon [NDSS’20]	Ripple [Security’21]	Mew [SP’23]
Inference Capabilities					
Inferring Detection Threshold	●	●	●	◐	◐
Inferring Windows Timing	●	●	●	◐	◐
Inferring Co-located Tasks	●	N/A	N/A	N/A	◐
Attack Feasibility					
Synchronized Augmentation	●	N/A	N/A	N/A	N/A
Memory Squeezing	●	N/A	N/A	N/A	◐
Time-window Exploitation	●	●	●	◐	◐

◐ Adversaries should send attack traffic along chosen paths across multiple switches.

Heracles exposes vulnerabilities in defenses *beyond* Cerberus

How can we design ***effective DoS detection*** that is ***robust against Heracles***?

Synchronized Augmentation	●	N/A	N/A	N/A	N/A
Memory Squeezing	●	N/A	N/A	N/A	◐
Time-window Exploitation	●	●	●	◐	◐

◐ Adversaries should send attack traffic along chosen paths across multiple switches.

Heracles exposes vulnerabilities in defenses *beyond* Cerberus

How can we design ***effective DoS detection*** that is ***robust against Heracles***?

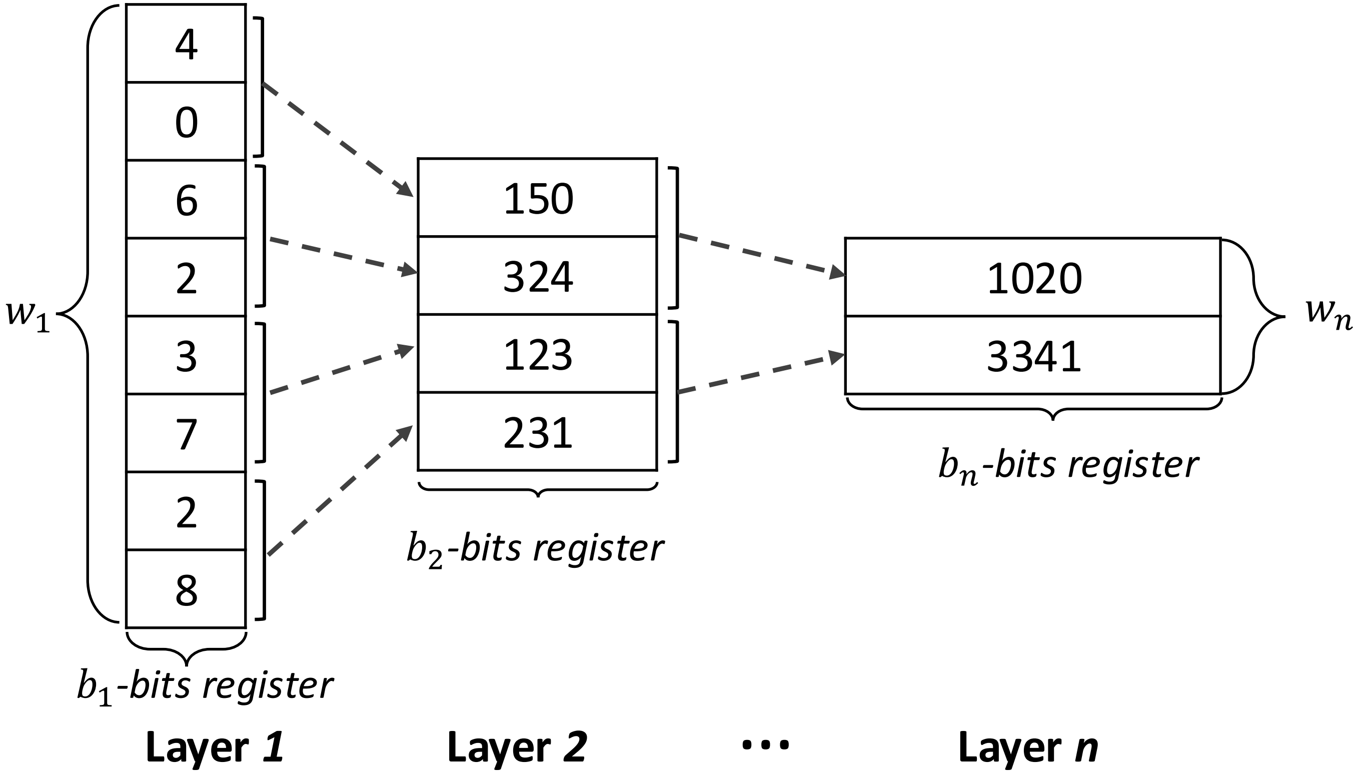
We present ***Shield***.

(Shared Hierarchical Registers for Layered Decay)

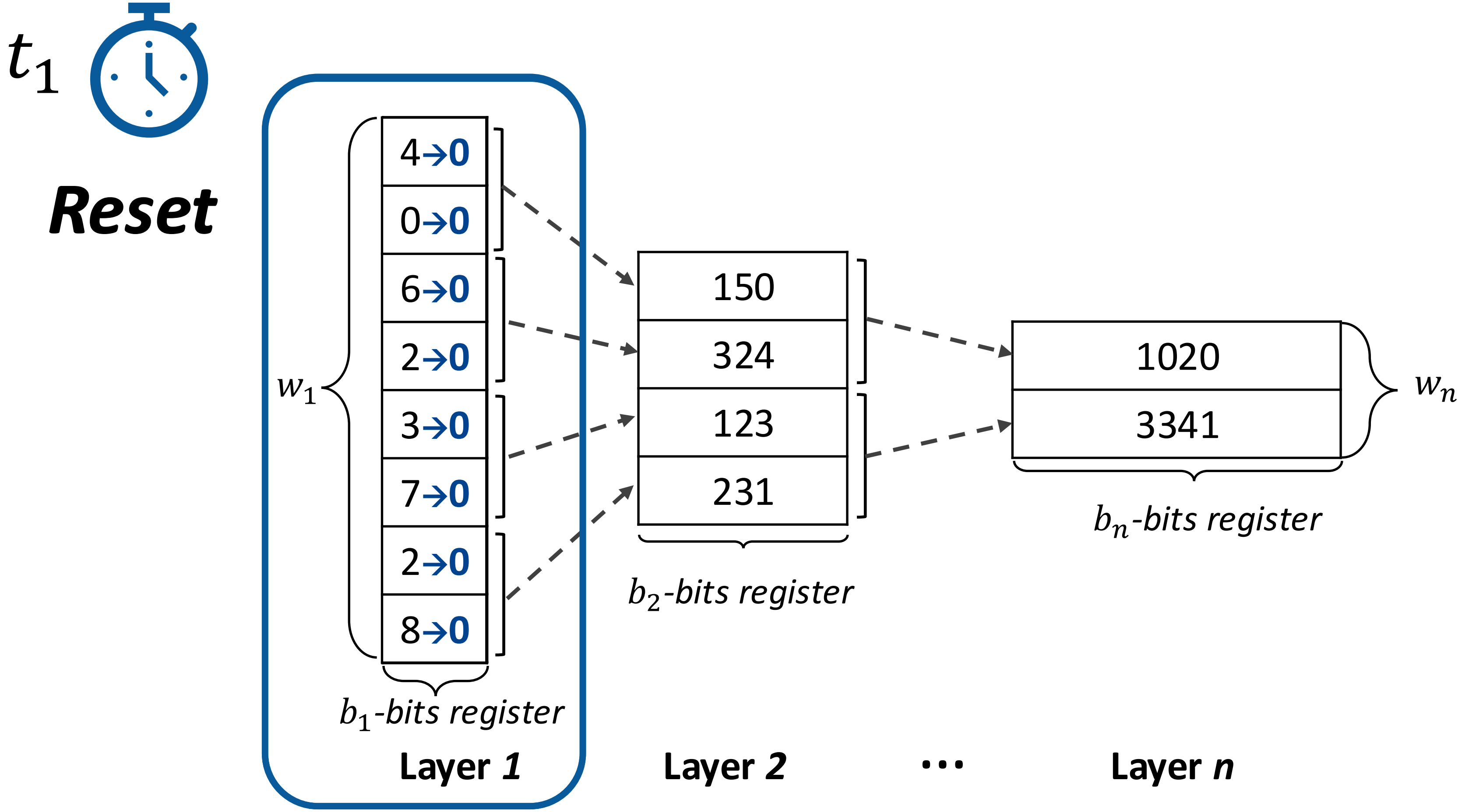
Time-window Exploitation	●	●	●	◐	◐

◐ Adversaries should send attack traffic along chosen paths across multiple switches.

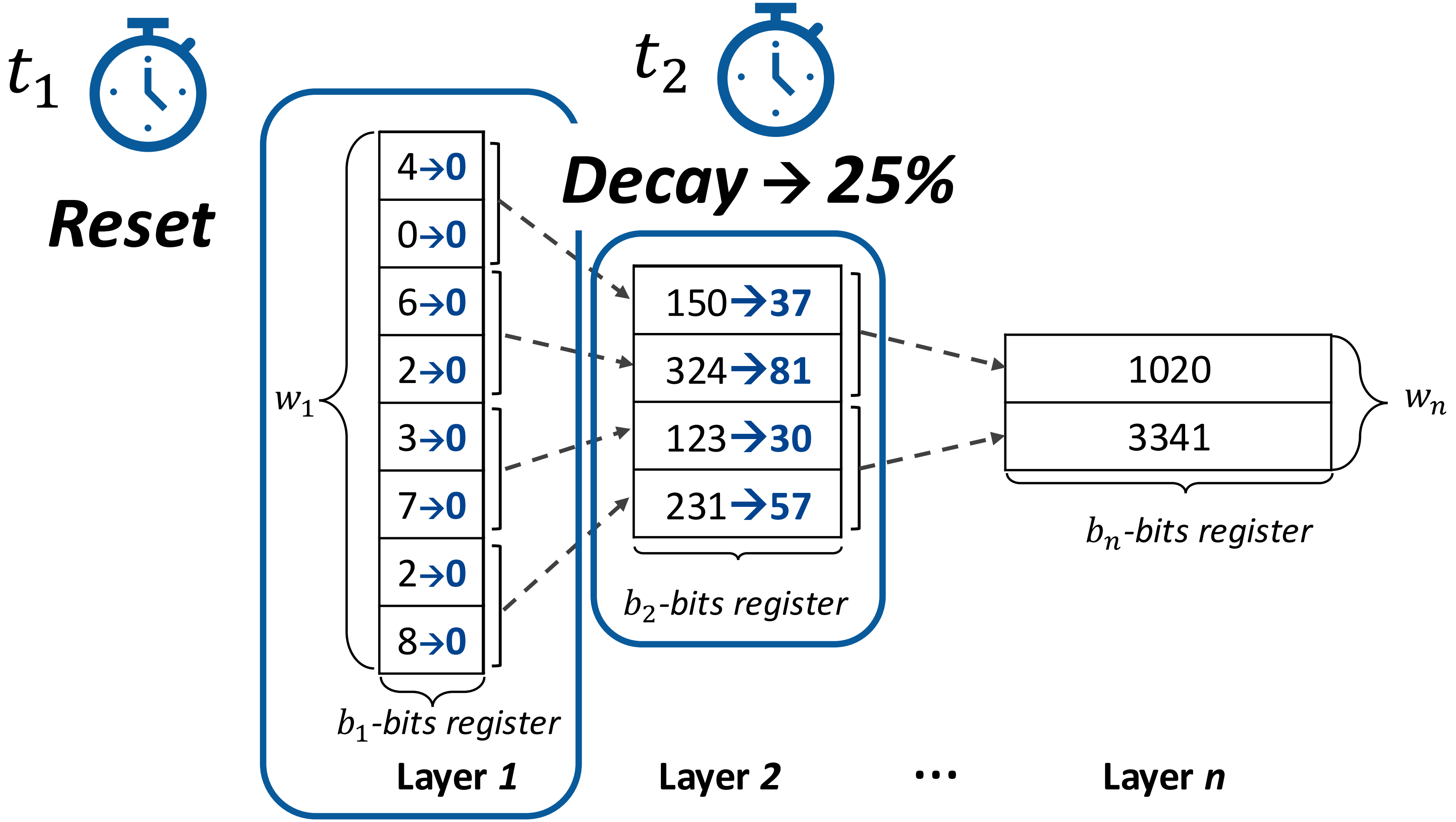
Shield decouples timing information across ***multiple hierarchical layers***



Shield decouples timing information across *multiple hierarchical layers*



Shield decouples timing information across ***multiple hierarchical layers***



t_1

Reset

t_2

Decay → 25%

w_1

4→**0**

0→**0**

6→**0**

2→**0**

3→**0**

7→**0**

2→**0**

8→**0**

b_1*-bits register*

Layer *1*

150→**37**

324→**81**

123→**30**

231→**57**

b_2*-bits register*

Layer *2*

…

1020

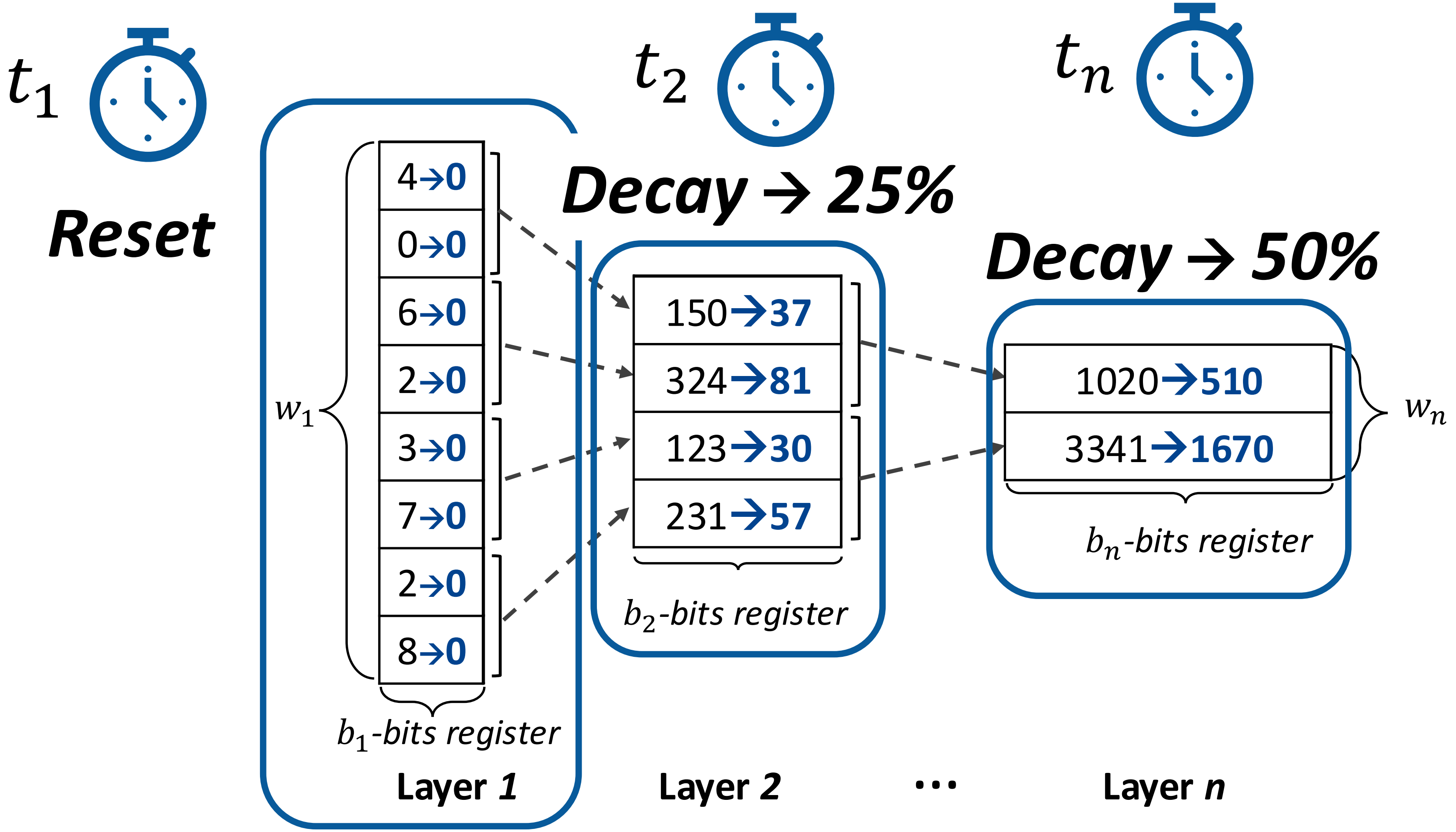
3341

w_n

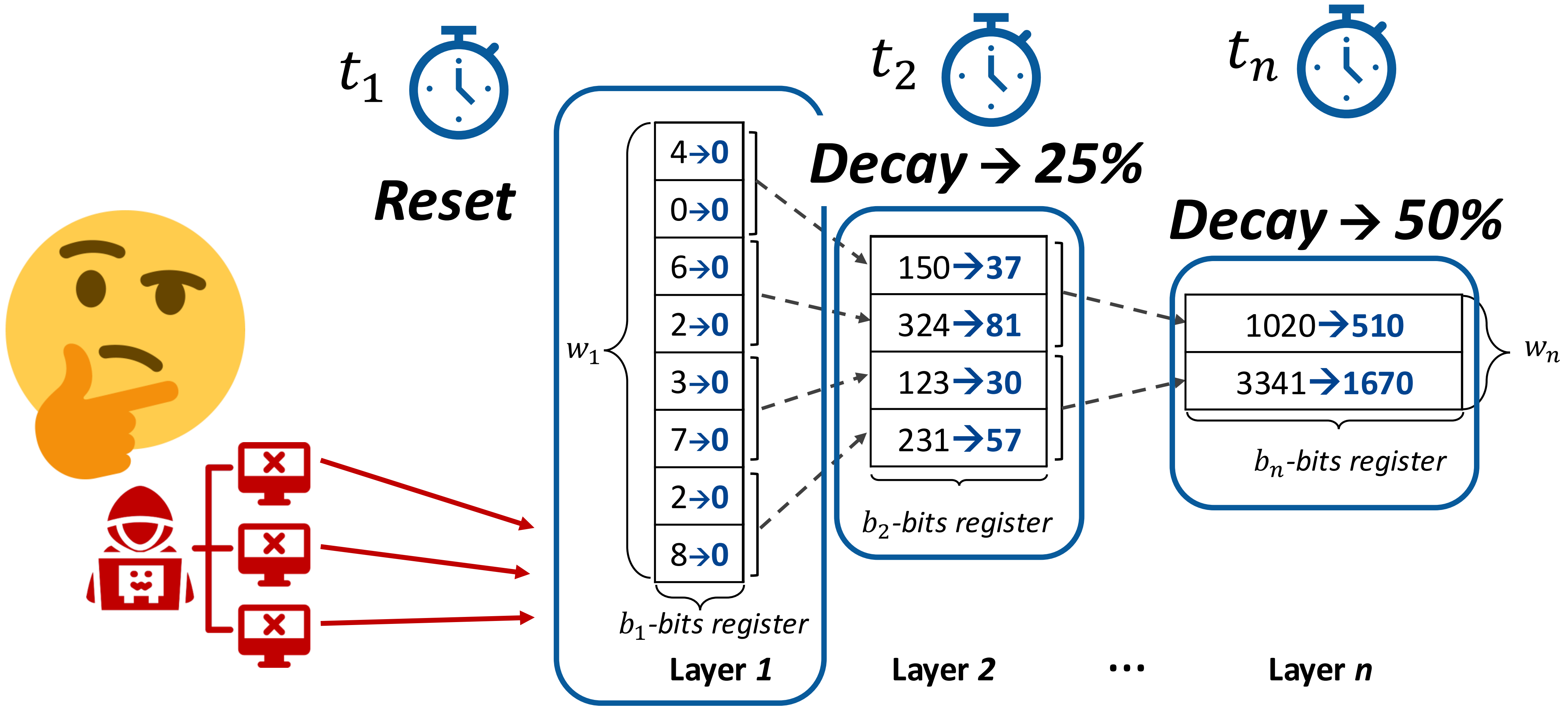
b_n*-bits register*

Layer *n*

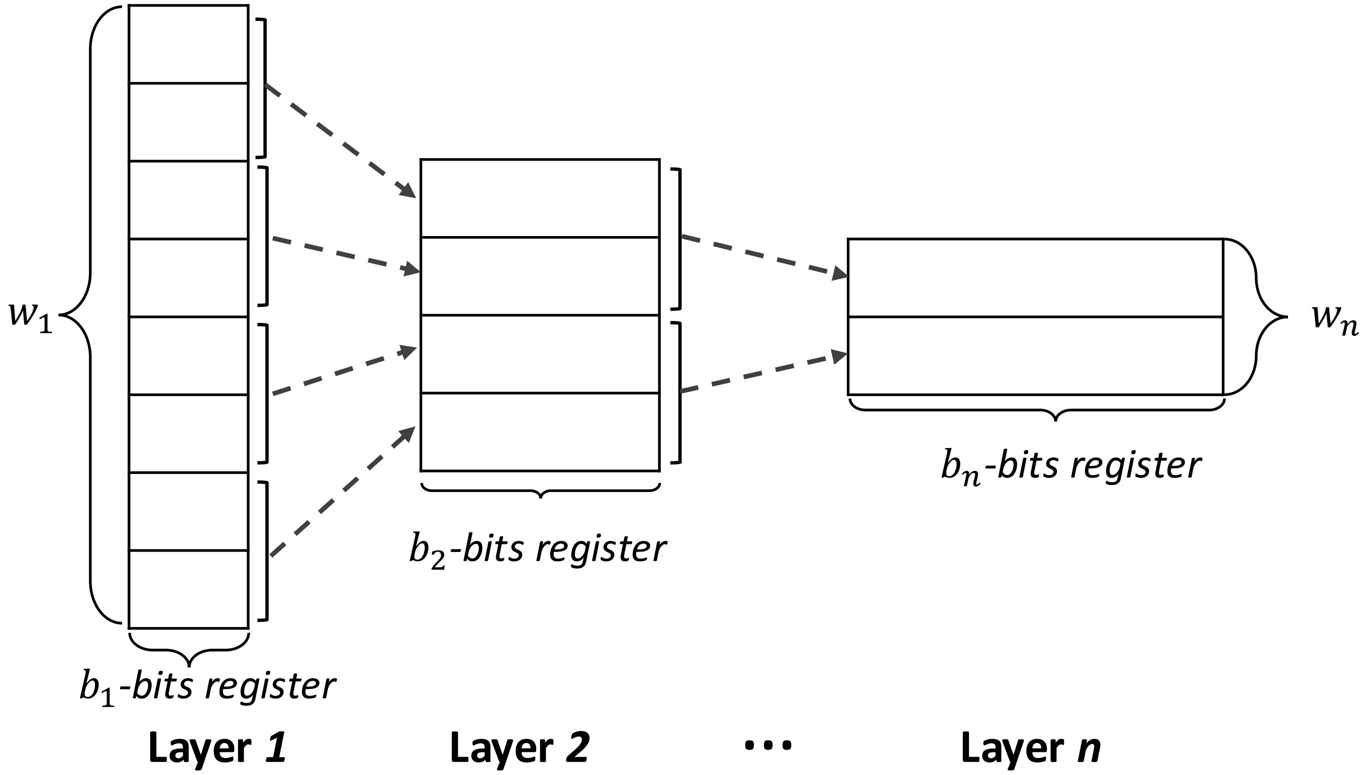
Shield decouples timing information across ***multiple hierarchical layers***



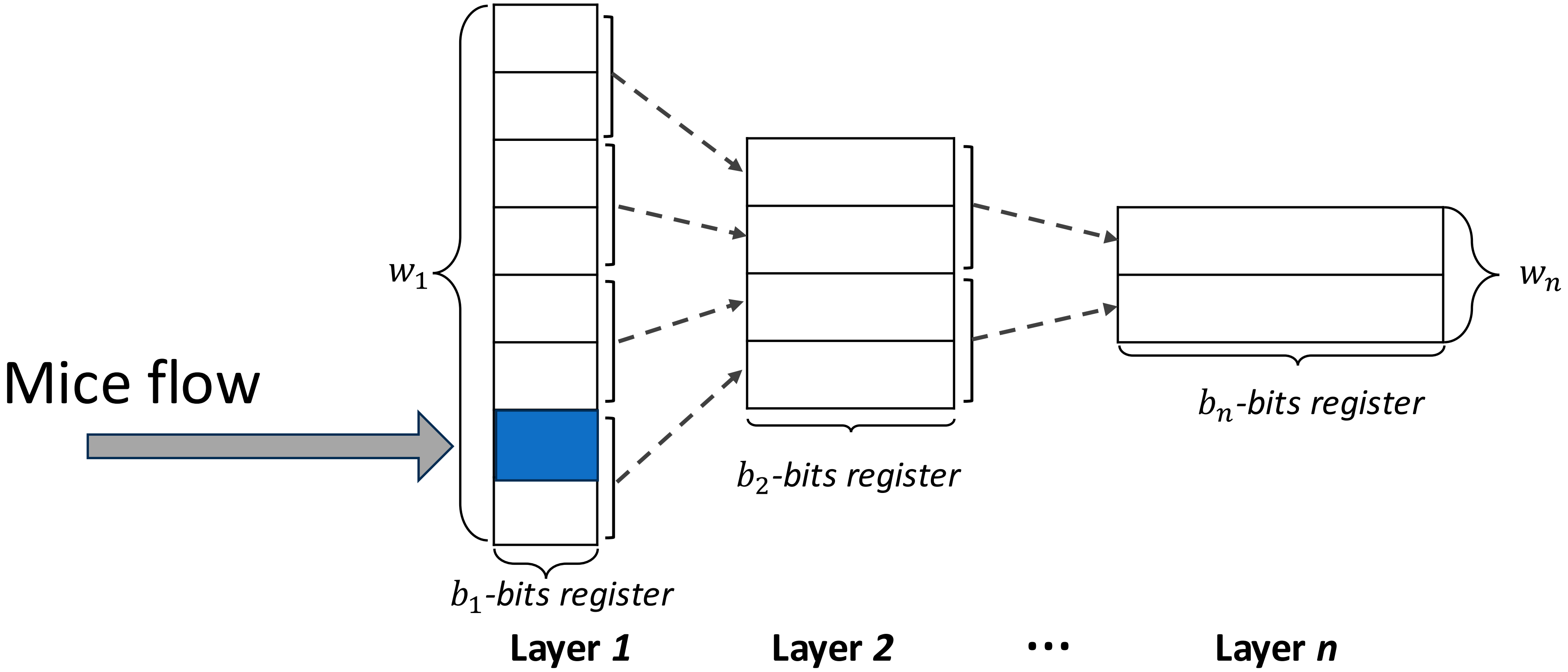
Shield decouples timing information across ***multiple hierarchical layers***



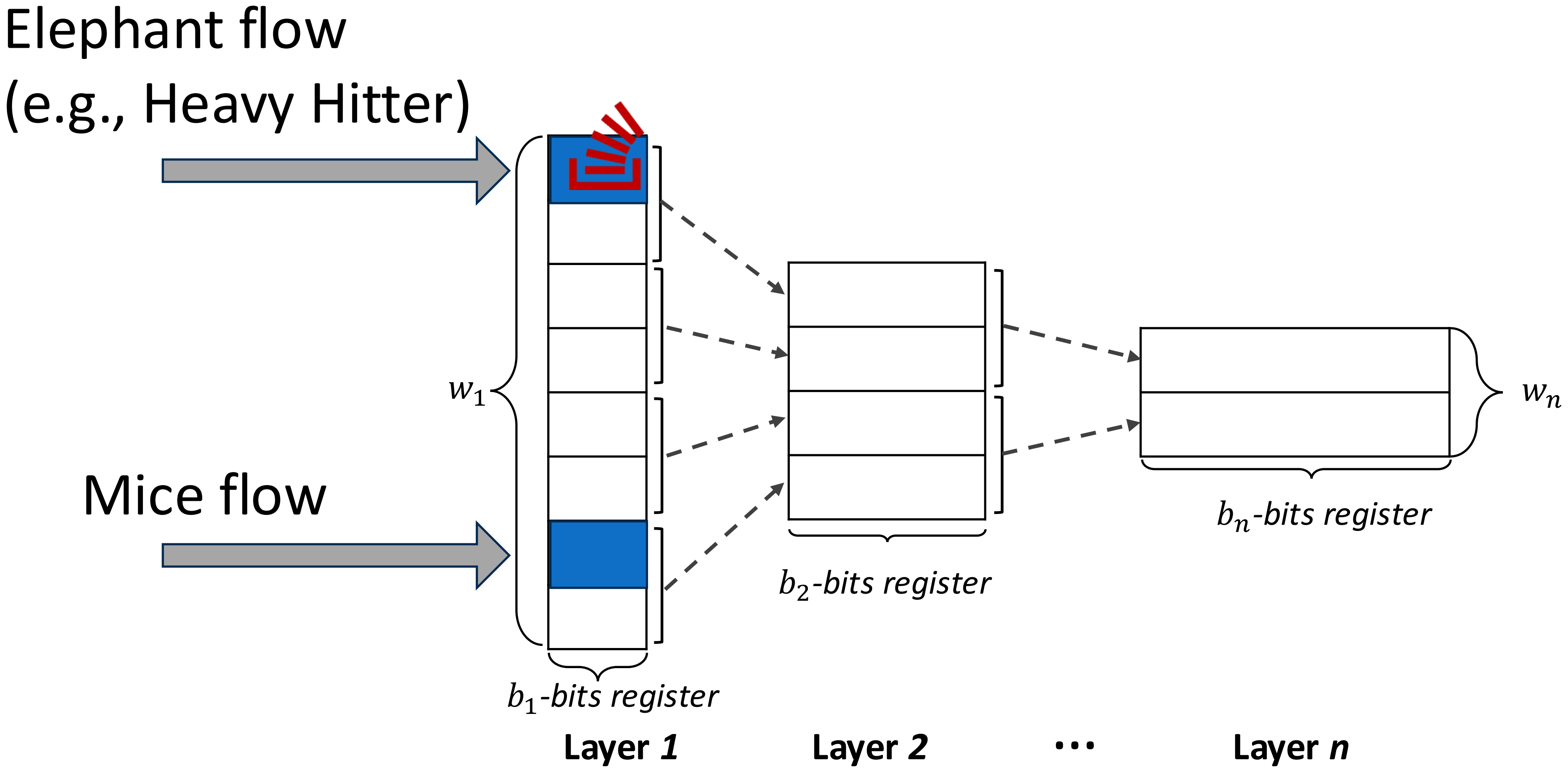
Shield organizes in-network monitoring into ***hierarchical layers***



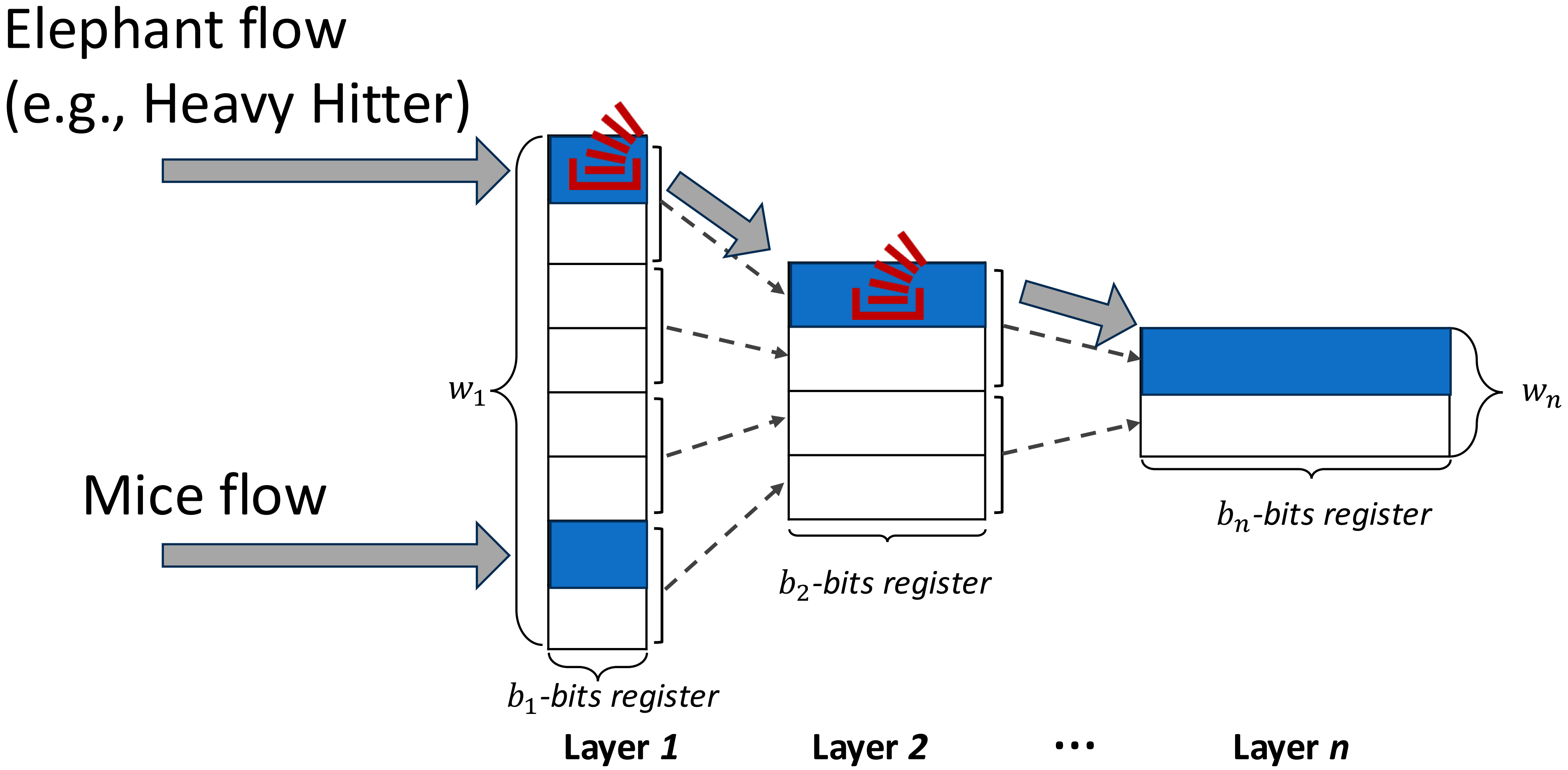
Shield organizes in-network monitoring into ***hierarchical layers***



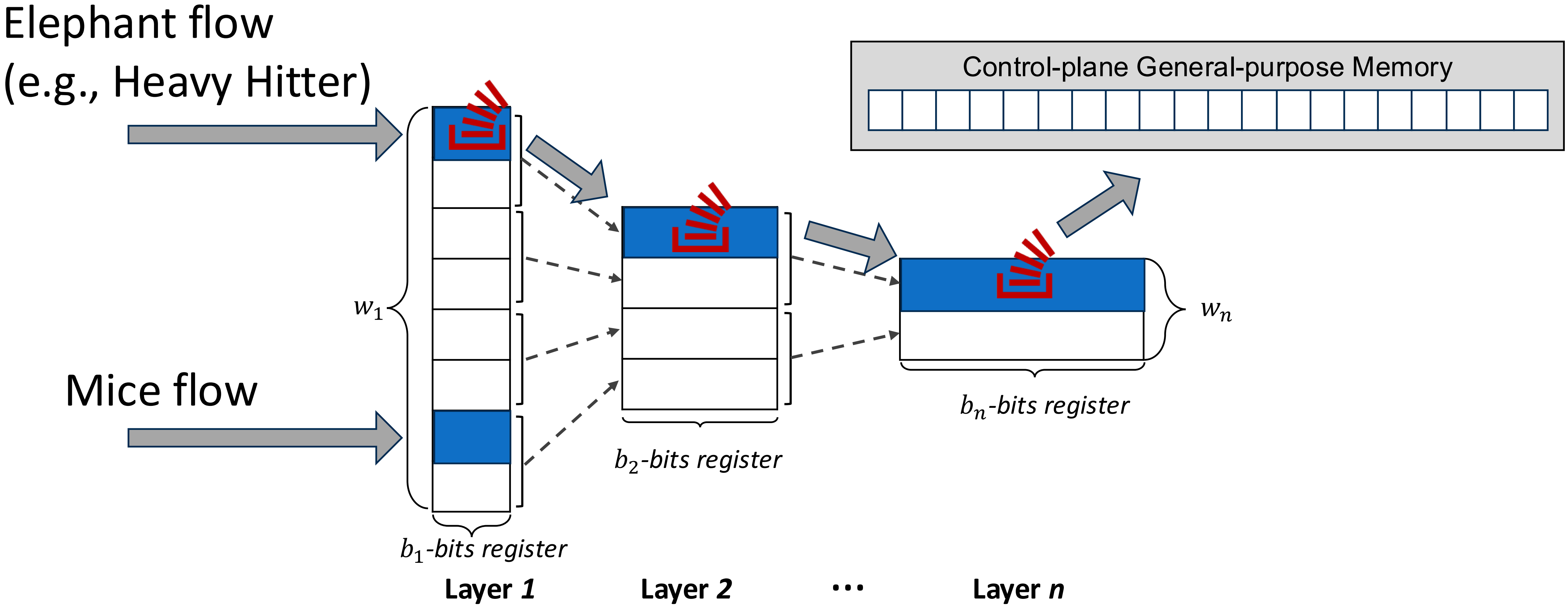
Shield organizes in-network monitoring into ***hierarchical layers***



Shield organizes in-network monitoring into ***hierarchical layers***



Shield organizes in-network monitoring into ***hierarchical layers***



Shield effectively ***mitigates*** *the* ***Heracles*** attack

- Security and hardware-constraint-aware design

Shield effectively ***mitigates** the **Heracles*** attack

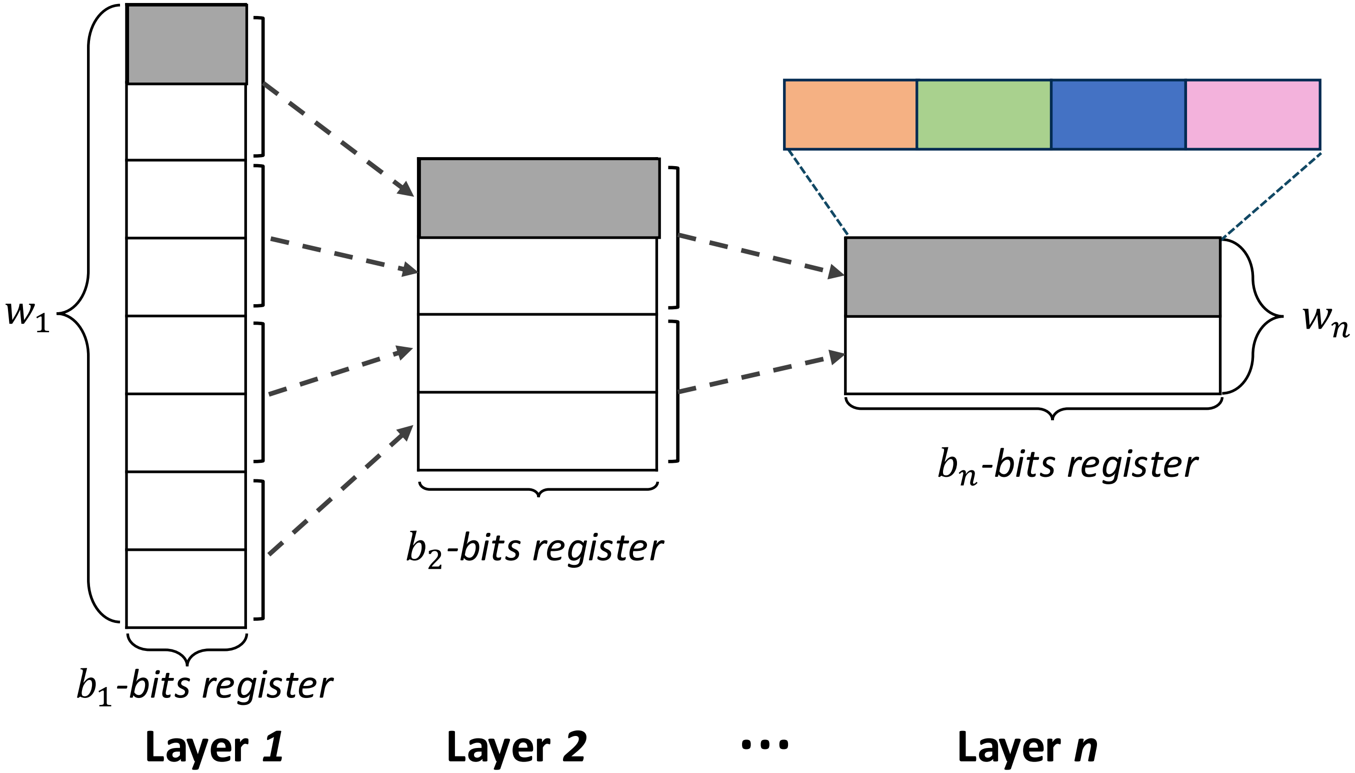
- Security and hardware-constraint-aware design

Resource consumption of 4 DoS defense tasks on Tofino switch

Resource	**Count-Min Sketch**	**FCM Sketch** **[CoNEXT’20]**	**Cerberus** **[SP’24]**	**Shield** **(Ours)**
SRAM	34.9%	41.6%	**16.7%**	**24.4%**
TCAM	0.0%	5.6%	0.0%	0.0%
SALU	37.5%	62.0%	**20.8%**	**45.8%**
Hash unit	18.9%	15.4%	14.8%	22.4%
Pipeline stages	12	12	**9**	**11**

Shield effectively ***mitigates*** *the* ***Heracles*** attack

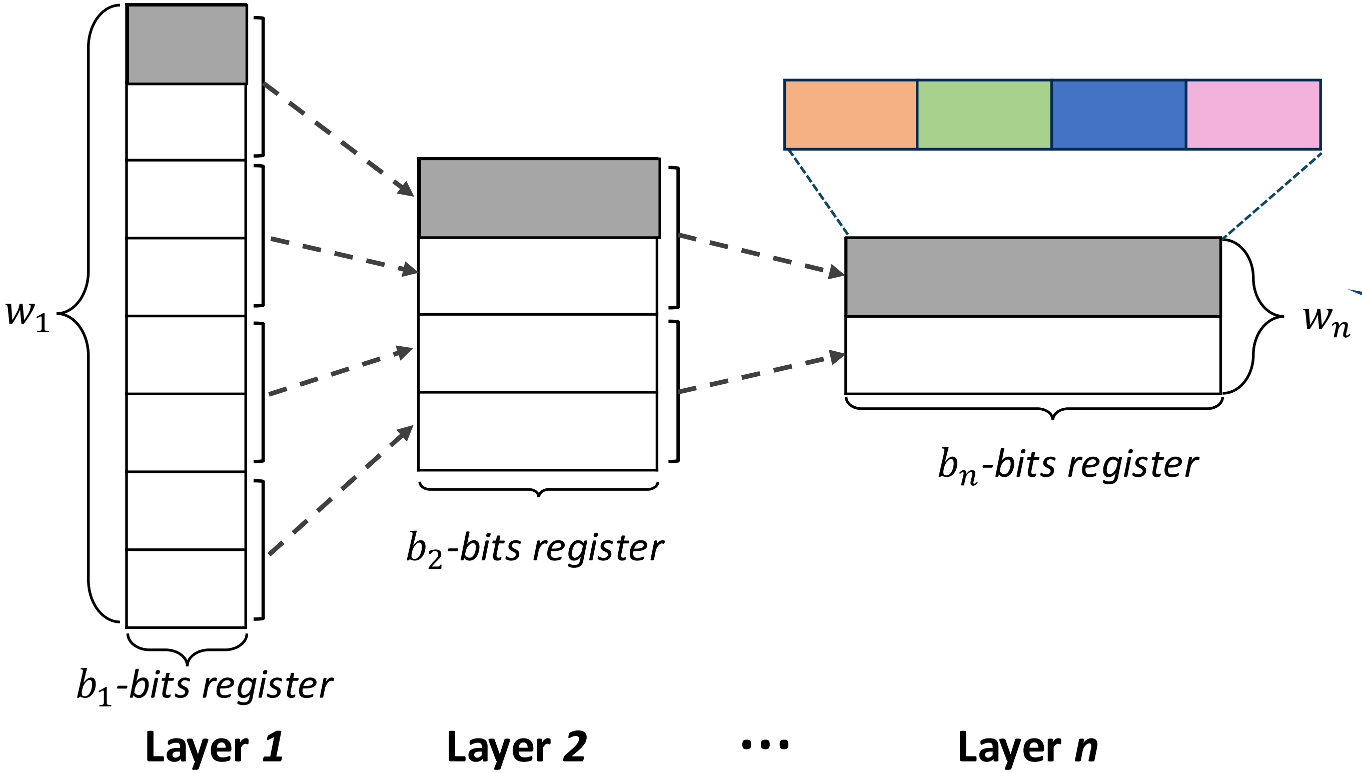
- Security and hardware-constraint-aware design



- ***Fixed-size memory slicing***
  → No memory squeezing attack

Shield effectively ***mitigates*** *the* ***Heracles*** attack

- Security and hardware-constraint-aware design

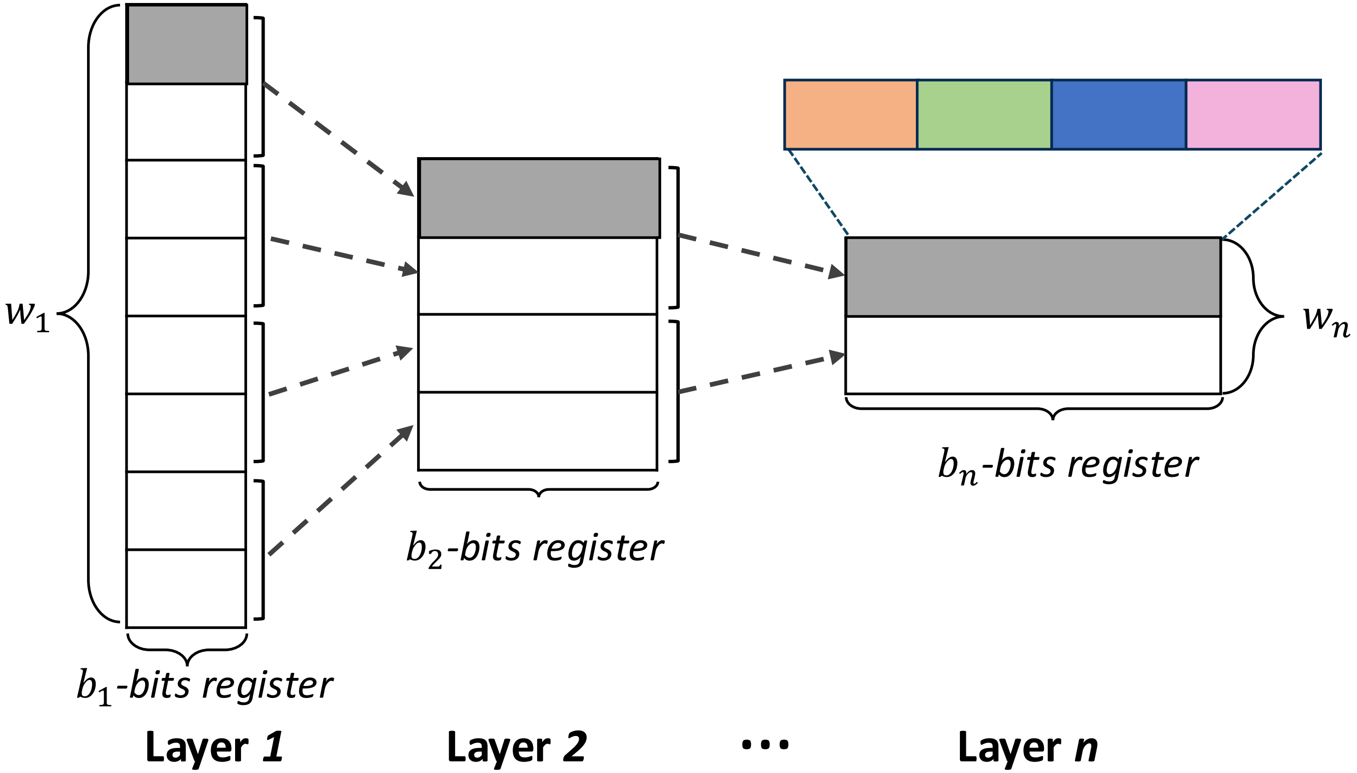


- ***Fixed-size memory slicing***
  → No memory squeezing attack

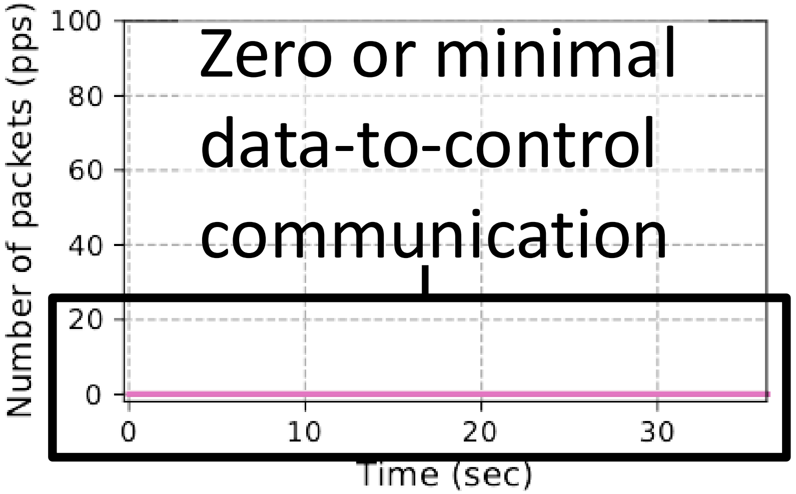
Illusion of huge counter bucket

Shield effectively ***mitigates*** *the* ***Heracles*** attack

- Security and hardware-constraint-aware design



- ***Fixed-size memory slicing***
  → No memory squeezing attack



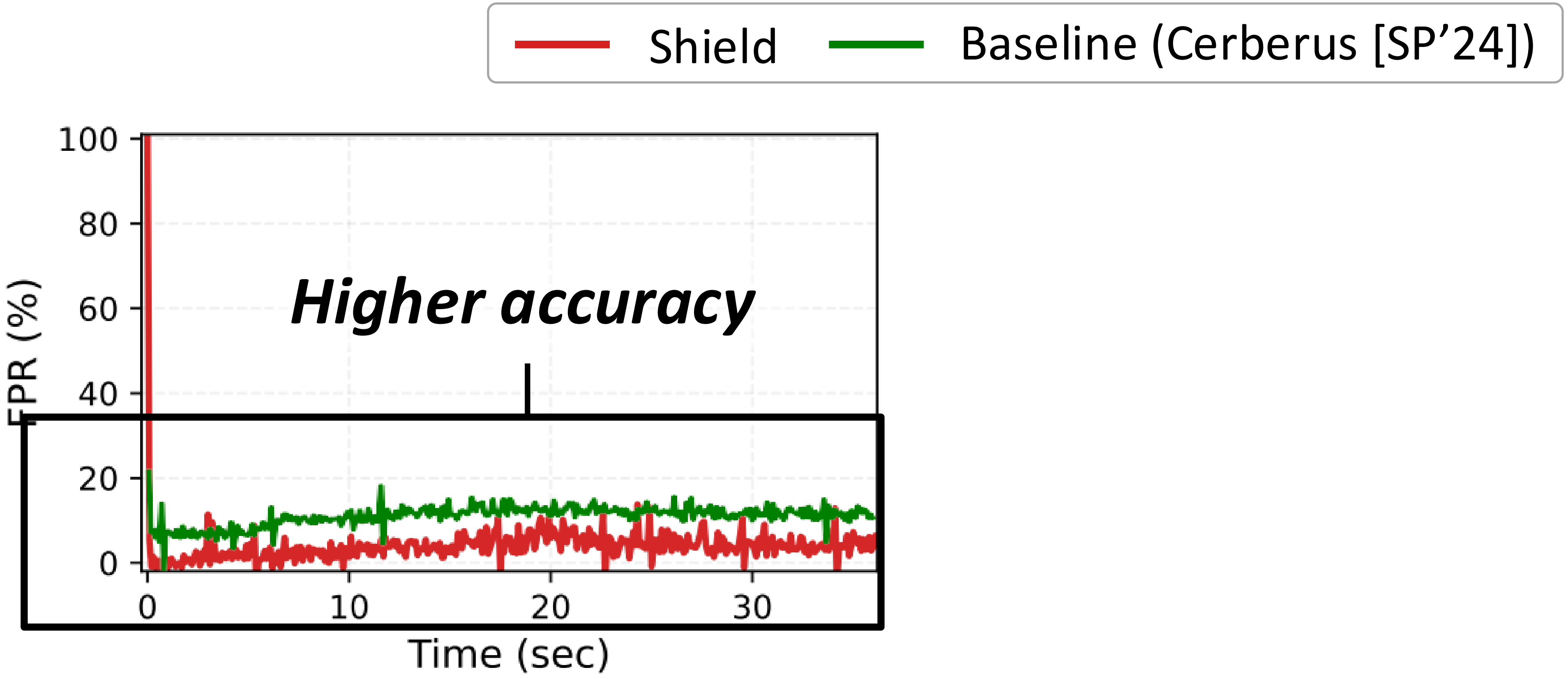
Zero or minimal data-to-control communication

Number of uploaded packets

Shield effectively ***mitigates*** *the* ***Heracles*** attack

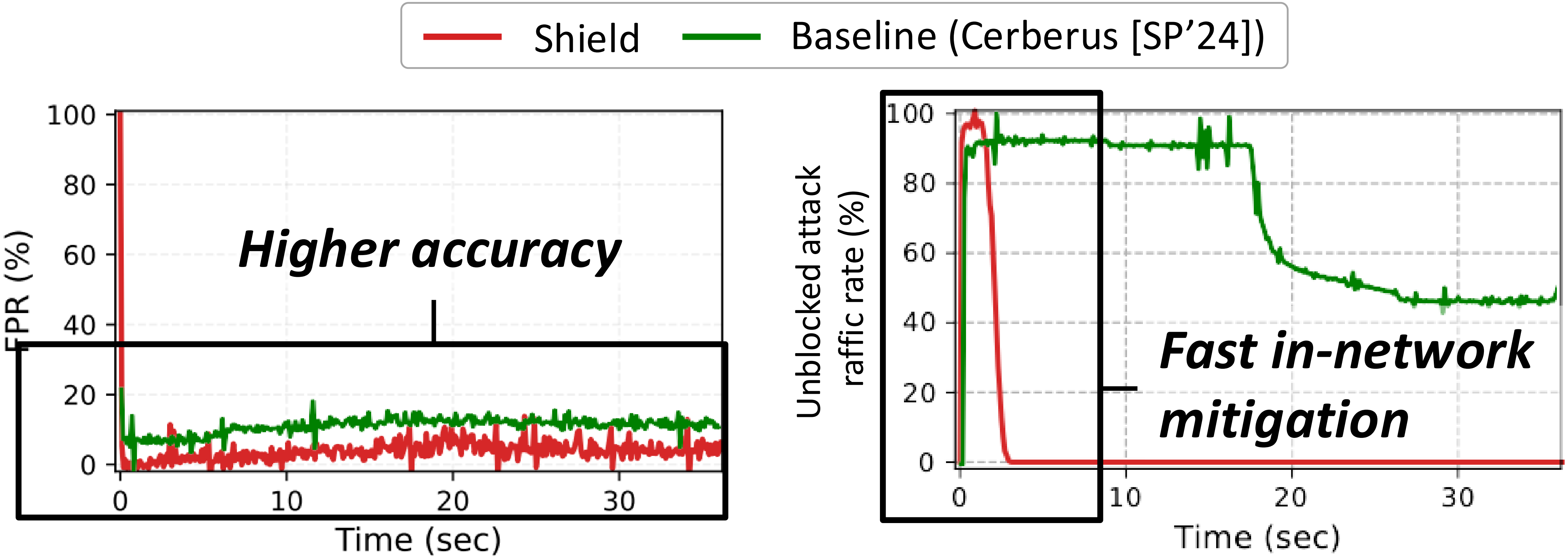
🟥 Shield 🟩 Baseline (Cerberus [SP’24])

Shield effectively ***mitigates*** *the* ***Heracles*** attack



(a) Detection accuracy

Shield effectively ***mitigates*** *the* ***Heracles*** attack



Higher accuracy

Fast in-network mitigation

(a) Detection accuracy

(b) DoS mitigation result

Takeaway

Takeaway

- Adaptability under hardware constraints is a ***security trade-off***

Takeaway

- Adaptability under hardware constraints is a ***security trade-off***
  - Efficiency-driven design leak critical control information
  - Excess flexibility enables adversarial resource manipulation
  - Cross-plane interaction creates structural bottlenecks

Takeaway

- Adaptability under hardware constraints is a ***security trade-off***
  - Efficiency-driven design leak critical control information
  - Excess flexibility enables adversarial resource manipulation
  - Cross-plane interaction creates structural bottlenecks
- ***Heracles*** reveals that these ***risks are structural***
- ***Shield*** demonstrates how they can be ***systematically eliminated***

Takeaway

- Adaptability under hardware constraints is a ***security trade-off***
  - Efficiency-driven design leak critical control information
  - Excess flexibility enables adversarial resource manipulation
  - Cross-plane interaction creates structural bottlenecks
- ***Heracles*** reveals that these ***risks are structural***
- ***Shield*** demonstrates how they can be ***systematically eliminated***

A first step toward ***proactively exposing structural weaknesses*** for ***designing robust data-plane defenses***

Takeaway

https://github.com/NetSP-KAIST/shield

Artifact Evaluated

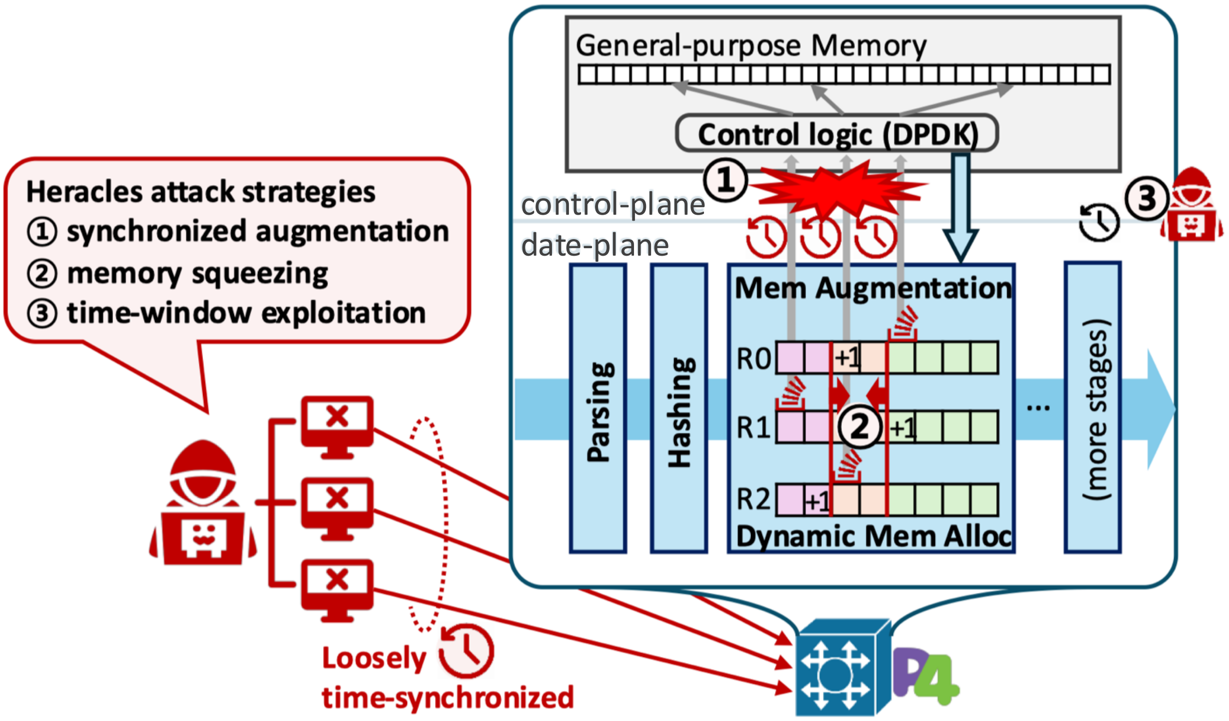
NDSS SYMPOSIUM

Available

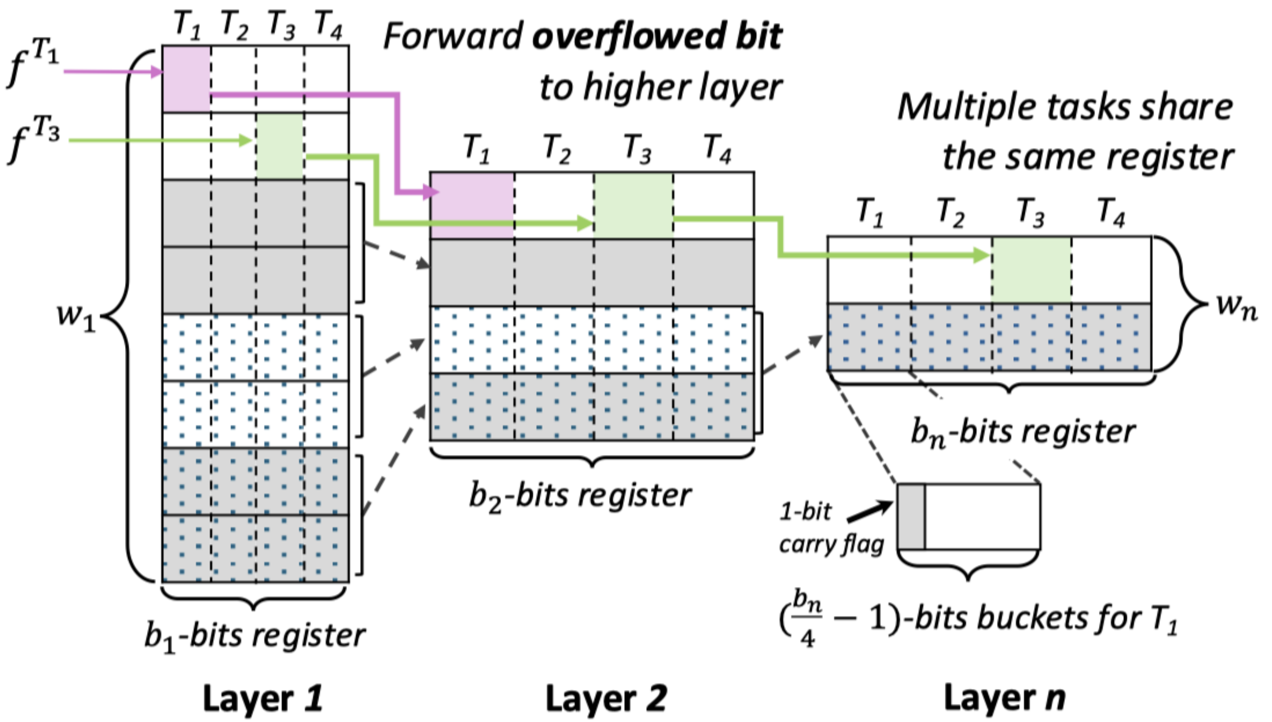
Functional

A first step toward ***proactively exposing structural weaknesses*** for ***designing robust data-plane defenses***

The *Heracles* attack



Shield



Thank You

Hocheol Nam

hcnam@kaist.ac.kr

KAIST

NetS&P
Research Lab @ KAIST

TEXAS A&M
UNIVERSITY

SUCCESS lab
Secure Communication and Computer Systems