Practical Verifiable In-network Filtering for DDoS Defense

Deli Gong, **<u>Muoi Tran</u>**, Shweta Shinde,

Hao Jin, Vyas Sekar, Prateek Saxena, Min Suk Kang

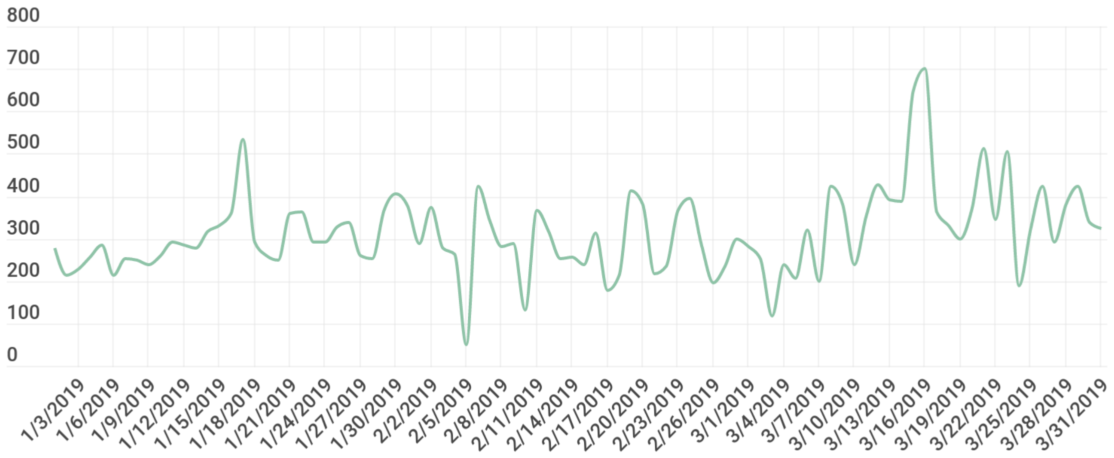
July 8, 2019 | Dallas, TX

NUS National University of Singapore
Berkeley UNIVERSITY OF CALIFORNIA
TEXAS A&M UNIVERSITY
Carnegie Mellon University

Large-scale volumetric DDoS attacks are common

(distributed denial of service)

- ***Hundreds*** DDoS attacks occur ***daily****
- Volume of DDoS traffic is ***escalating***
- New attack vectors (e.g., amplification) and attack source (e.g., botnets)



* According to Kaspersky Lab's report on DDoS attacks in Q1 2019

(2013)

COMPUTERWORLD

SECURITY IS SEXY

By Darlene Storm | Follow

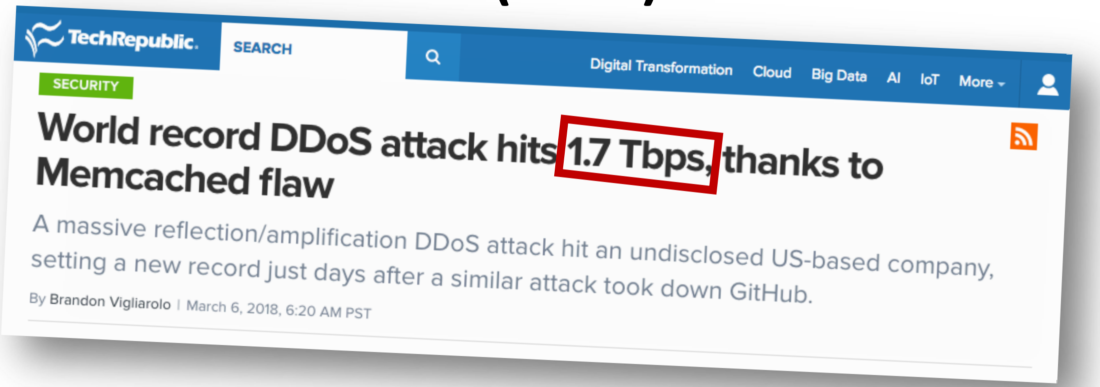
About

Most security news is about insecurity, hacking and cyber threats, bordering on scary. But when security is done right, it's a beautiful thing...sexy even. Security IS sexy.

OPINION

Biggest DDoS attack in history slows Internet, breaks record at 300 Gbps

(2018)



TechRepublic. SEARCH Digital Transformation Cloud Big Data AI IoT More

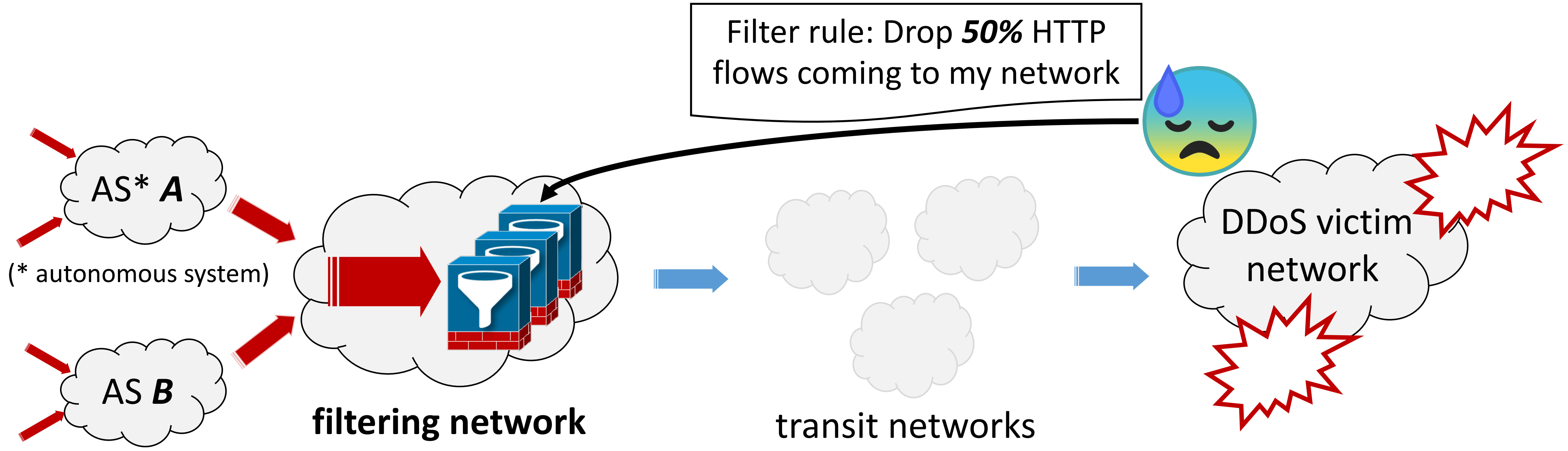
SECURITY

World record DDoS attack hits 1.7 Tbps, thanks to Memcached flaw

A massive reflection/amplification DDoS attack hit an undisclosed US-based company, setting a new record just days after a similar attack took down GitHub.

By Brandon Vigliarolo | March 6, 2018, 6:20 AM PST

In-network filtering: a promising DDoS mitigation

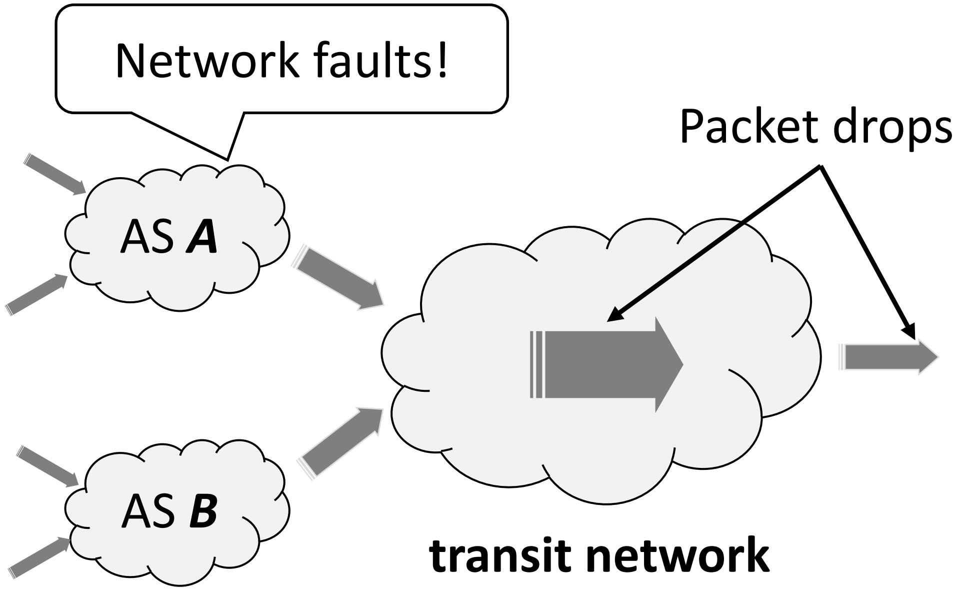


- In-network filtering
  - ✓ allows the DDoS victim to install traffic filters nearer to attack source
  - ✓ not a new idea:

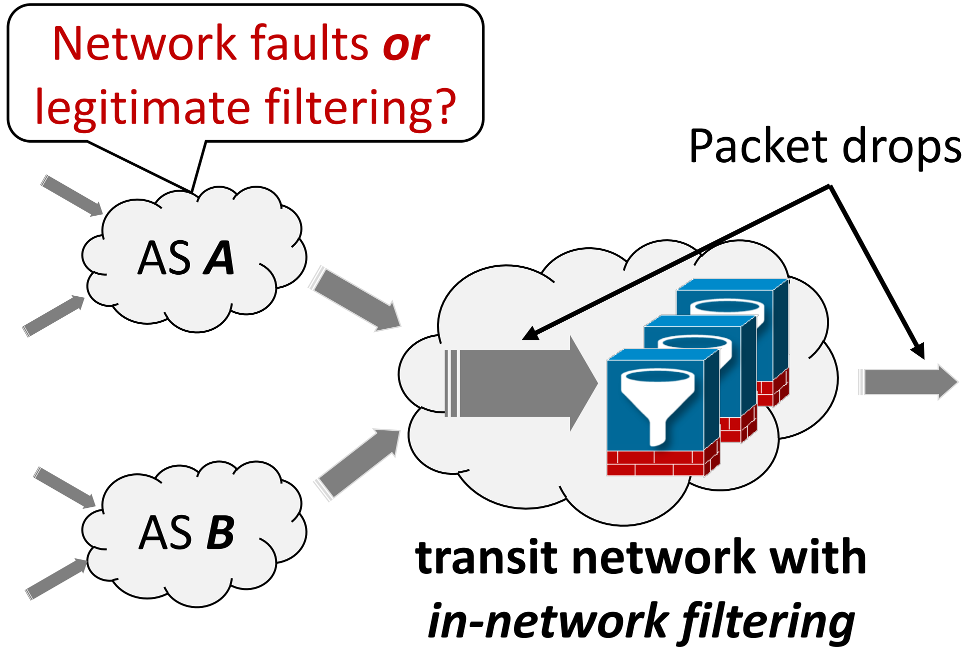
    e.g., *Pushback* [SIGCOMM'02], *D-WARD* [ICNP'02], *AITF* [USENIX ATC'05], *StopIt* [SIGCOMM'08]
  - ✓ installs at ***1% of ISPs*** can mitigate ***90% of DDoS attacks*** (*SENSS* [ACSAC'18])

One ***ignored*** problem: In-network filtering creates ***ambiguity*** about packet drops

Without in-network filtering

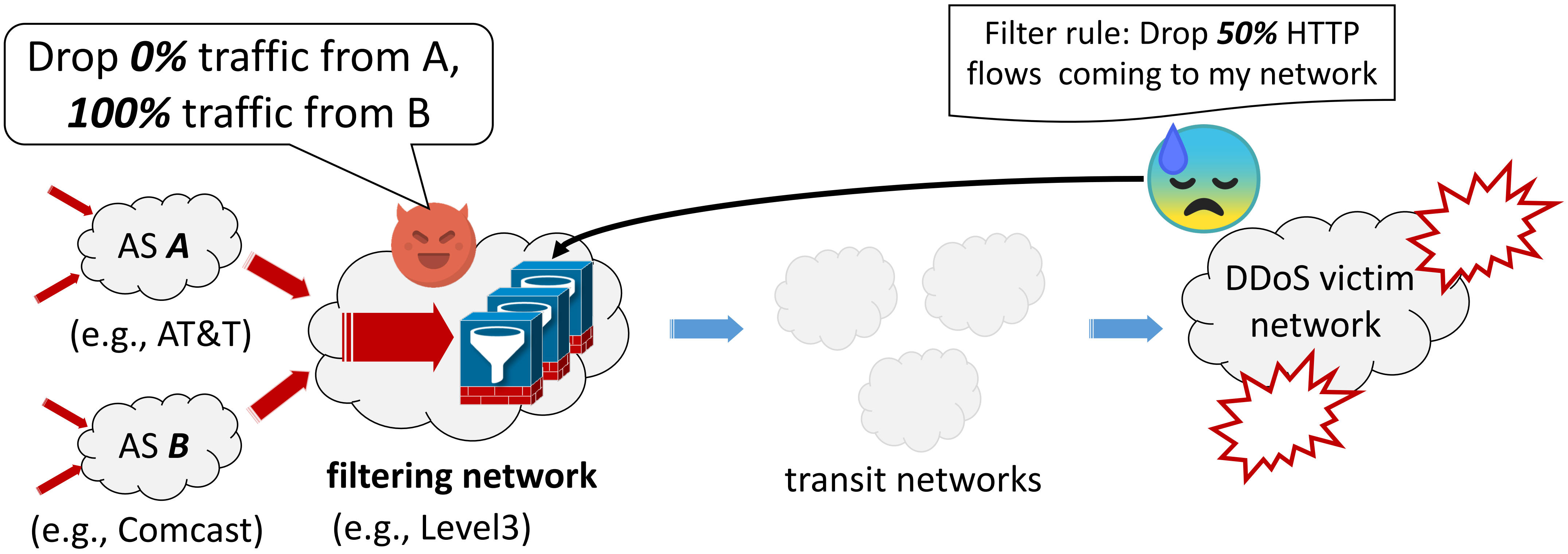


With in-network filtering

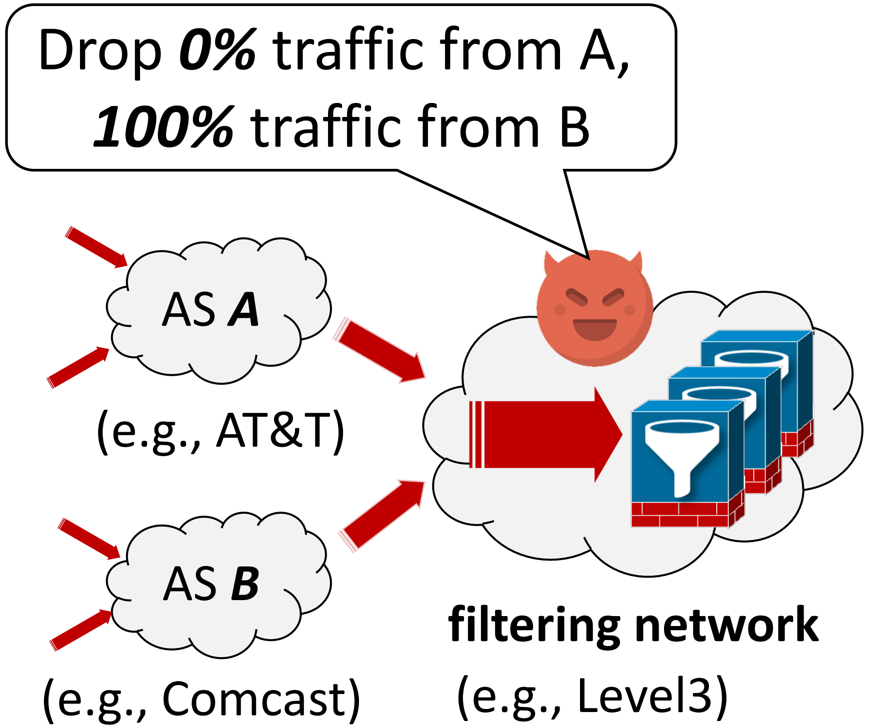


What can go wrong because of this ambiguity?

Filtering can be used as an ***excuse*** for ***discriminating*** neighboring ASes



Filtering can be used as an ***excuse*** for ***discriminating*** neighboring ASes



(2013)

FierceTelecom

AI TELECOM TECH PLATFORMS

Telecom

Verizon blames Cogent for unbalanced peering in Netflix dispute

by Sue Marek | Jun 20, 2013 11:13am

(2014)

INTERNET

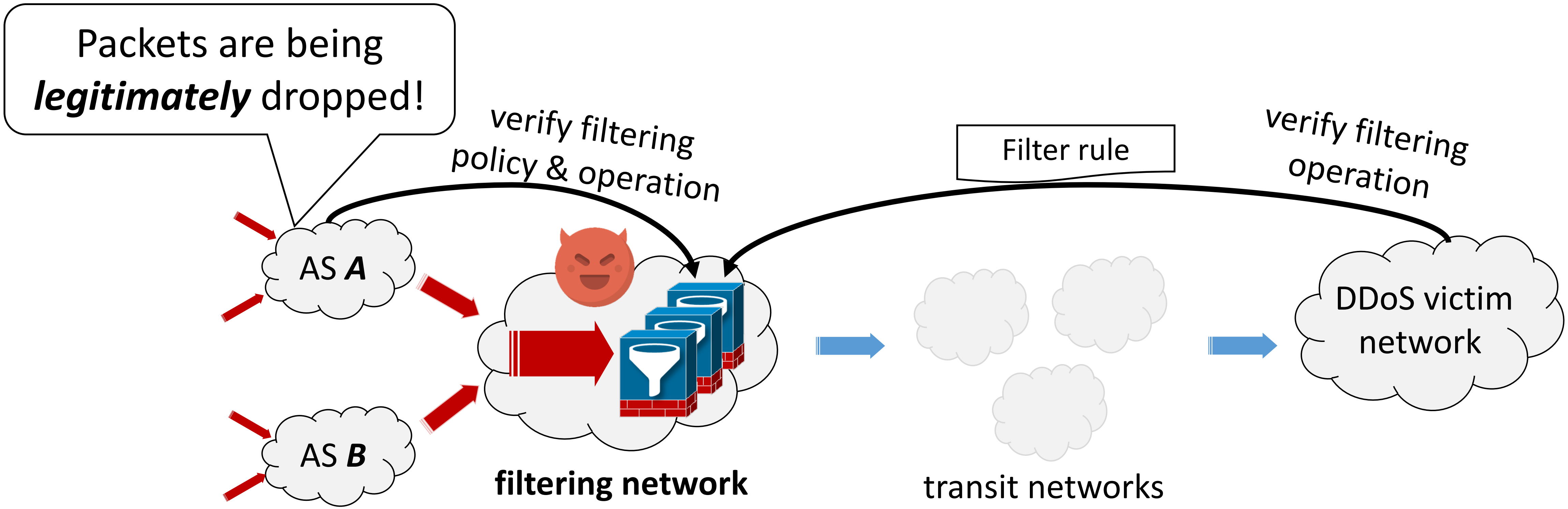
Level 3 accuses six broadband providers of degrading network traffic

Level 3 says the providers are slowing down Internet traffic to extract payment from it. But a deeper dive into the specifics reveals there may be another side to the story.

Several disputes ***already exist*** between transit networks

How to remove such an ambiguity?

Verifiability of filtering distinguishes legitimate DDoS mitigation from network faults



How to make the operations of in-network filtering ***verifiable***?

Our contributions

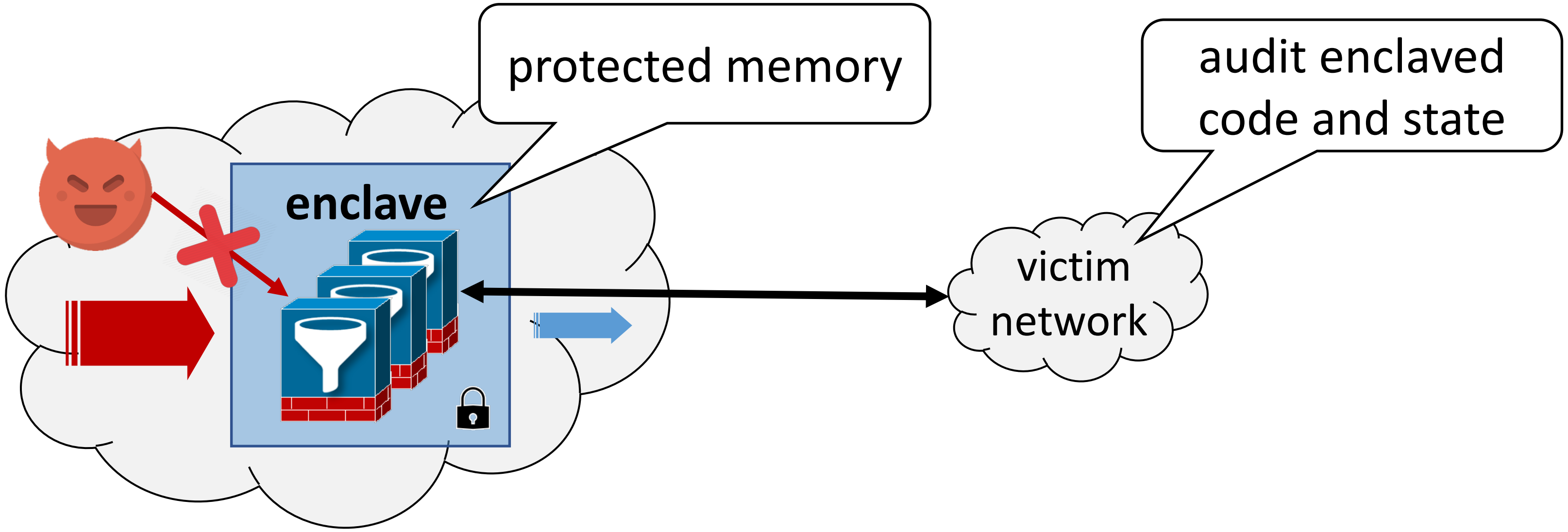
- We propose ***V*erifiable *I*n-network *F*iltering (VIF):**
  - ✓ Software networking functions with Trusted Execution Environments (e.g., Intel SGX) as root of trust.

Auditable **filter**
- ✓ uses TEEs
- ✓ is stateless
- ✓ detects bypass

Scalable **design**
- ✓ multiple filters run in parallel

Practical deployment
- ✓ at Internet Exchange Points (IXPs)

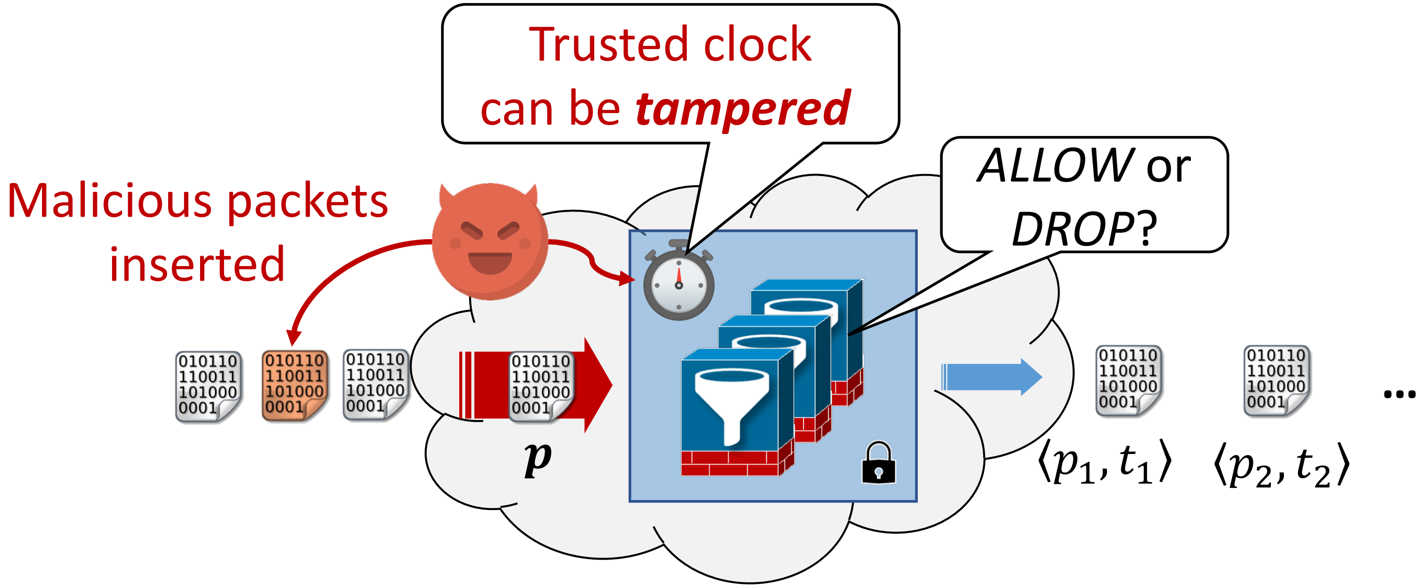
VIF design: ***auditable*** filter with TEEs



- Filtering within Trusted Execution Environments (TEEs) (e.g., Intel SGX)
  - ✓ Isolated execution
  - ✓ Remote attestation

TEEs alone is ***insufficient***
for auditable filter design!

Challenge 1: Influence from ***malicious inputs***



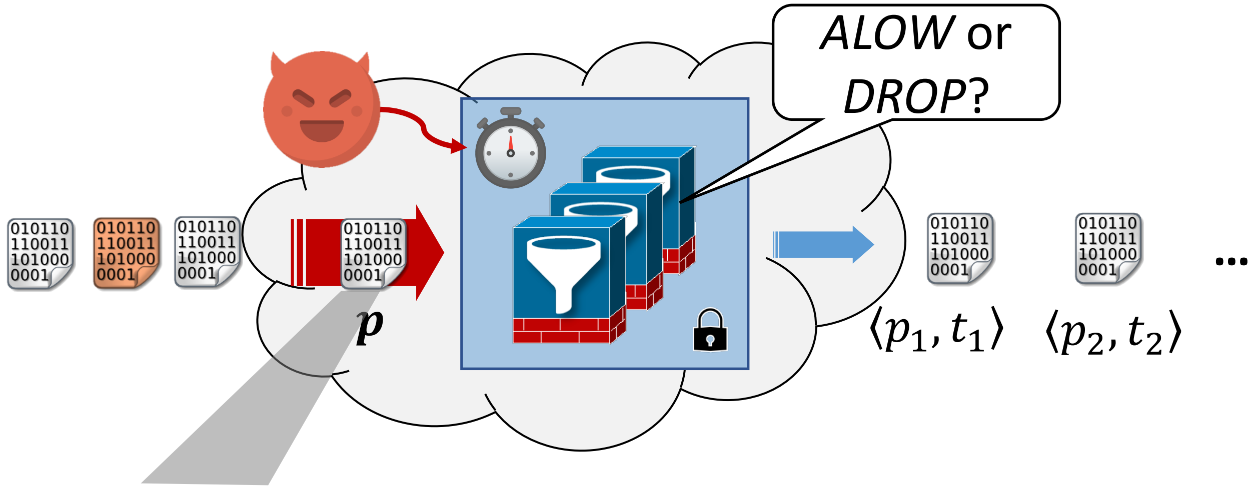
- ***Abstract model*** of the filtering function for packet $\boldsymbol{p}$*:*

$$\{ALLOW, DROP\} \leftarrow f(\langle p, t \rangle, (\langle p_1, t_1 \rangle, \langle p_2, t_2 \rangle, \langle p_3, t_3 \rangle, \dots))$$

Arrival time

Previous packets/
packet order

Solution: ***Stateless*** filter design

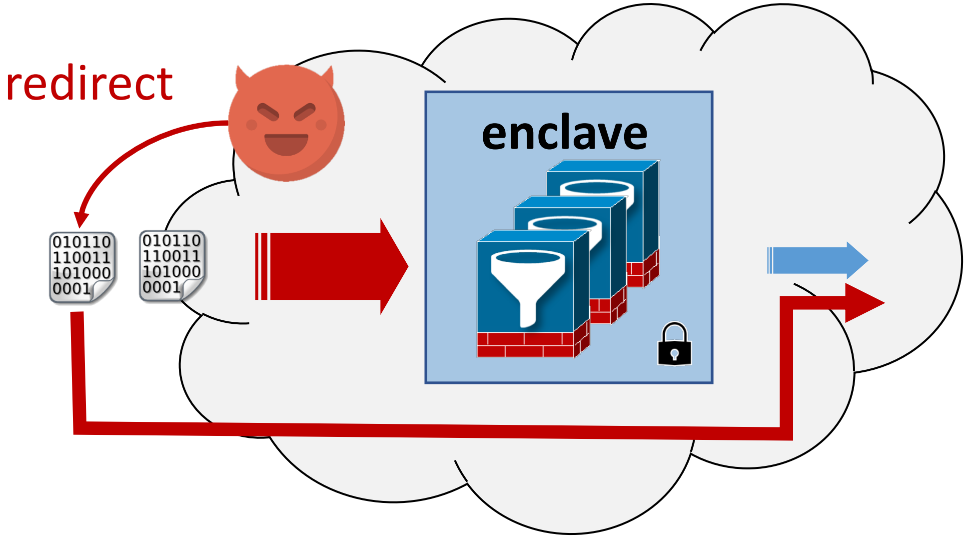


5-tuple (`srcIP, srcPort, dstIP, dstPort, protocol`)

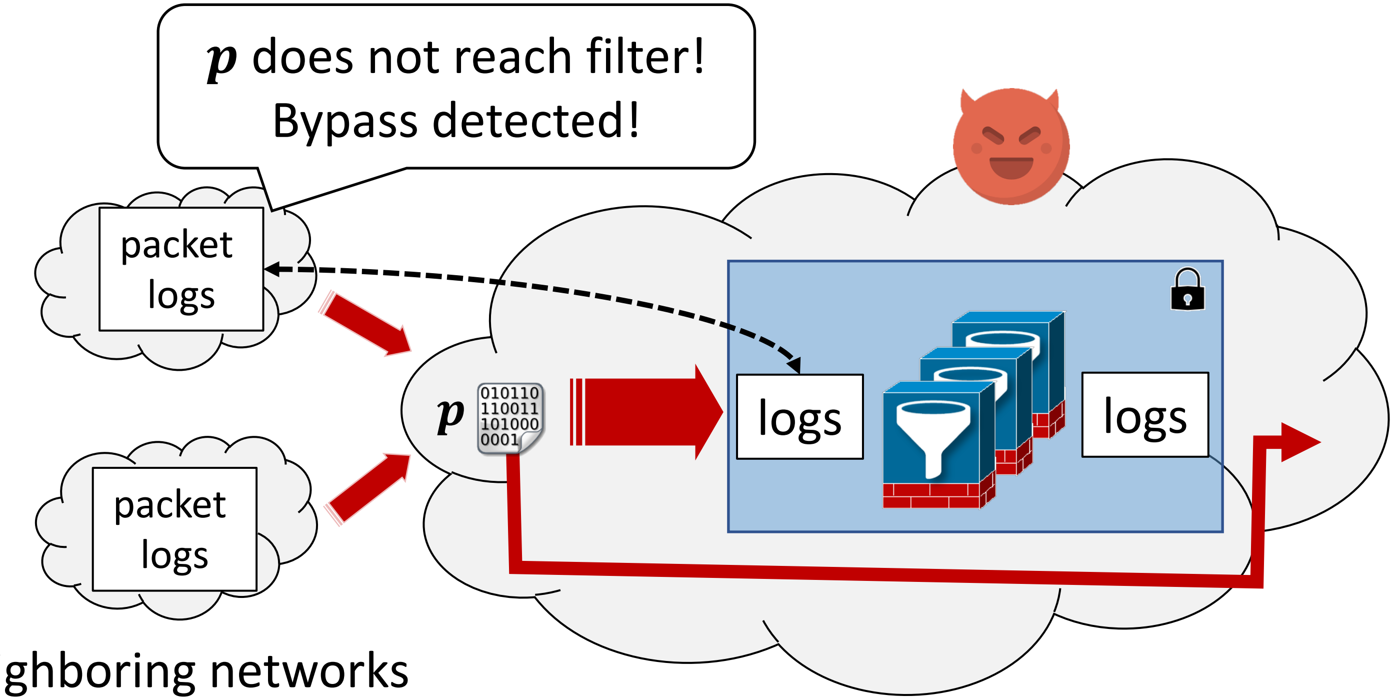
- No reliance on packet arrival time and packet order

$$\{ALLOW, DROP\} \leftarrow f(\langle p \rangle)$$

Challenge 2: Traffic may be redirected to bypass filter



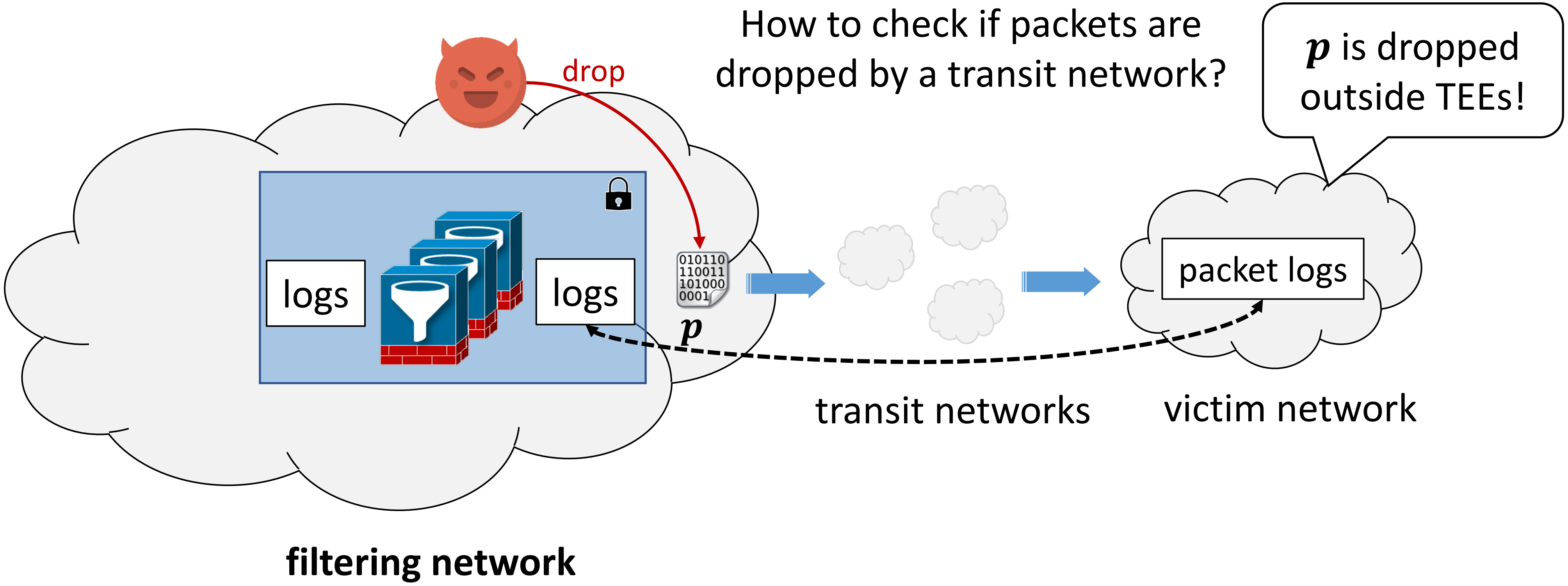
Solution to filter bypass:
Accountable logs for bypass detection



filtering network

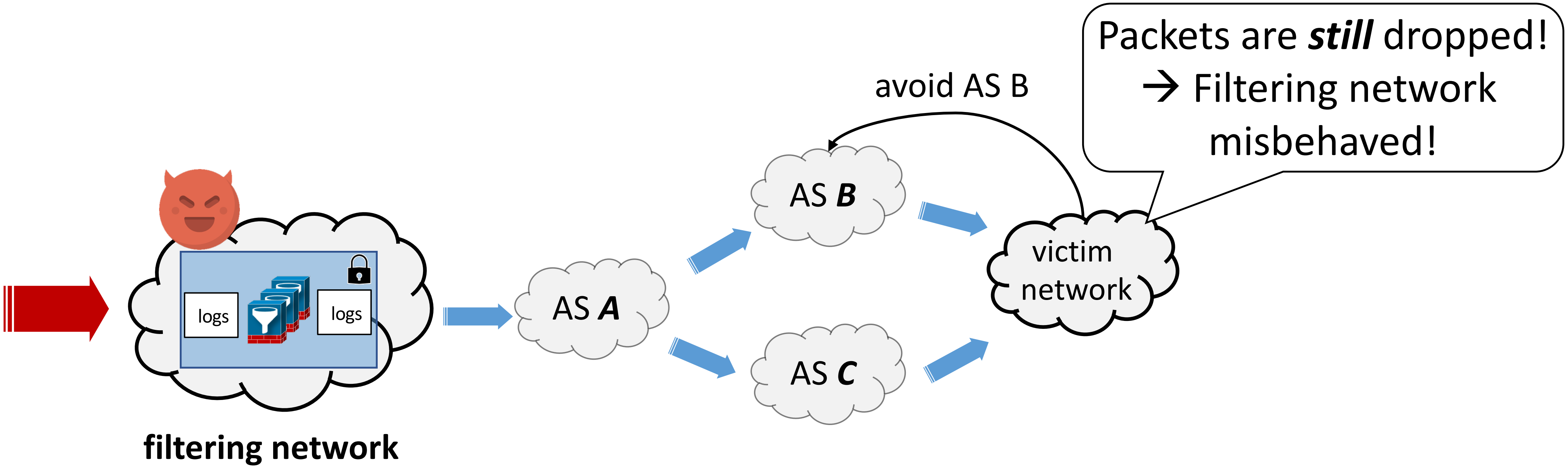
- Accountable packet logging before and after filtering
  - ✓ Compare logs to detect bypass

Solution to filter bypass:
Accountable logs for bypass detection



- Accountable packet logging before and after filtering
  - ✓ Compare logs to detect bypass

How does victim know ***who*** is dropping packets?



- Victim network ***tests*** individual intermediate ASes
  - ✓ Rerouting inbound traffic using ***BGP poisoning*** (LIFEGUARD[SIGCOMM’12])
  - ✓ Detour takes place in a ***few minutes*** and ***no*** collaboration needed (Nyx [S&P’18])

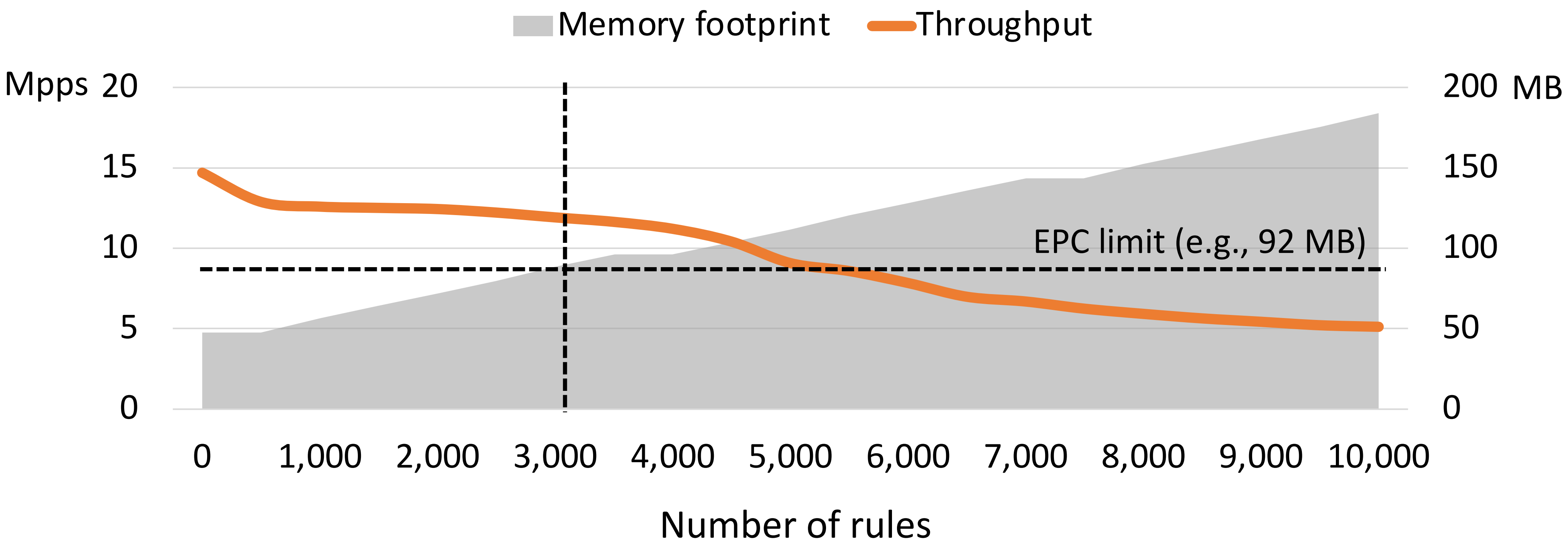
Our contributions

Auditable **filter**
- ✓ TEEs
- ✓ stateless
- ✓ bypass detection

Scalable **design**
- ✓ multiple filters run in parallel

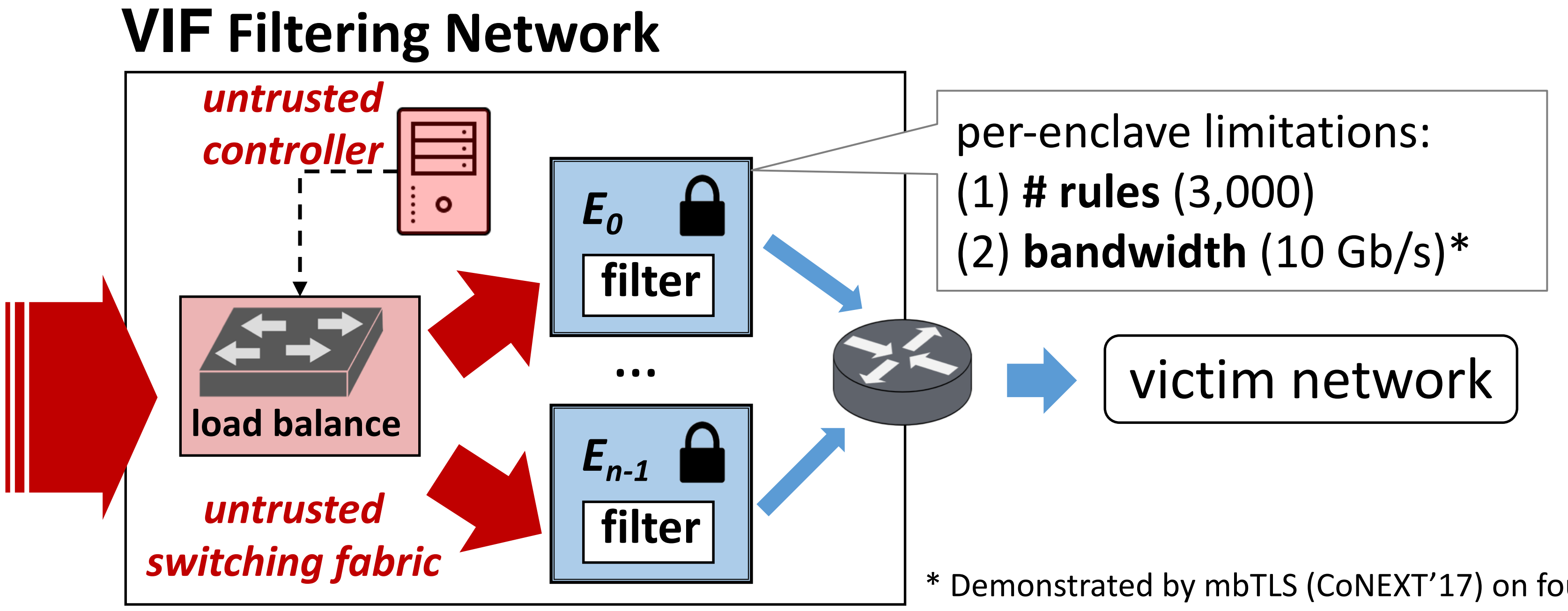
Practical deployment
- ✓ at Internet Exchange Points (IXPs)

Deployment issue: Scalability



- ***Performance issues*** when filtering within a ***single*** enclave:
  - ✓ Memory footprint grows ***linearly*** with number of rules
  - ✓ Throughput ***degrades*** when number of rules exceeds ~***3,000***

Solution to scalability issue: multiple SGX filters



* Demonstrated by mbTLS (CoNEXT'17) on four SGX-core machines.

- **More in our paper:**
  - ✓ How trusted filters detect misbehaviors from untrusted components
  - ✓ A greedy solution to calculate filter rules among filters
  - ✓ Filter rules redistribution

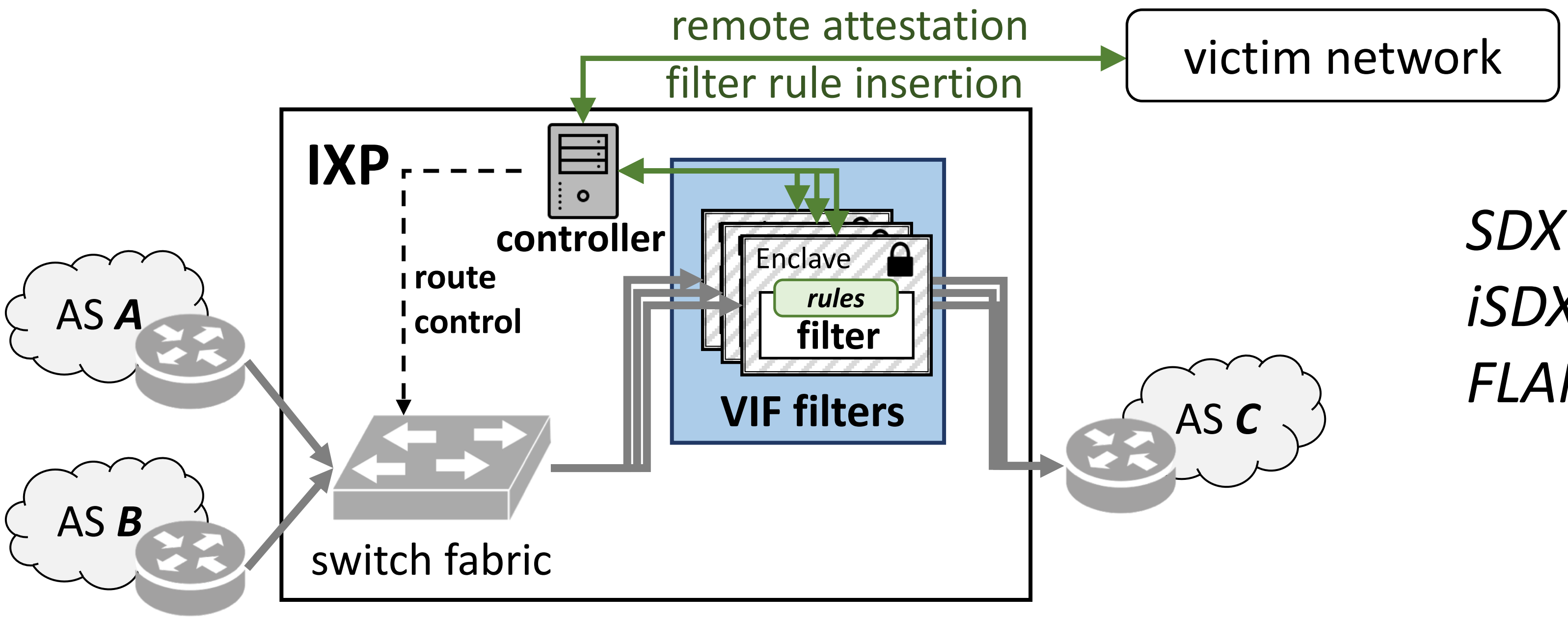
Our contributions

Auditable **filter**
- ✓ TEEs
- ✓ stateless
- ✓ bypass detection

Scalable **design**
- ✓ multiple filters run in parallel

Practical deployment
- ✓ at Internet Exchange Points (IXPs)

Deployment example



SDX [SIGCOMM'15]
iSDX [NSDI'16]
FLANC [SOSR'16]

- Internet Exchange Points (IXPs) :
  - ✓ have peering relationship with ***hundreds*** ISPs
  - ✓ have flexible software-defined architecture

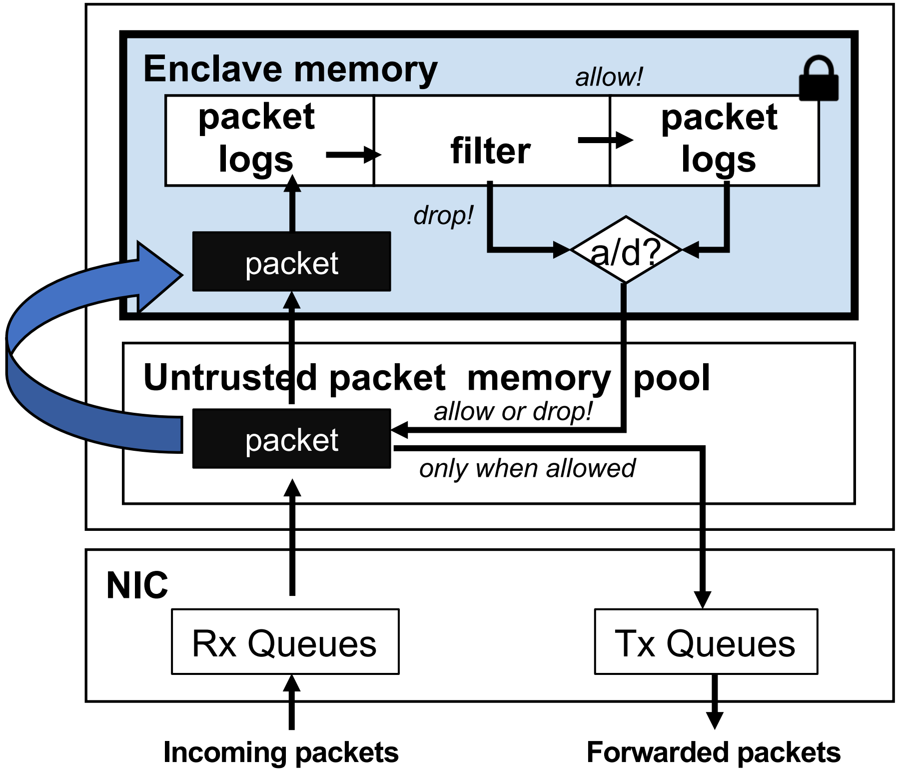
Implementation

- Overview
  - ✓ Intel SGX SDK for Linux 2.1
  - ✓ Data Plane Development Kit (DPDK) 17.05.2
- Trusted computing base:
  - ✓ modification of DPDK `ip_pipeline` (1,044 SLoC)
  - ✓ packet logging and optimizations (162 SLoC)

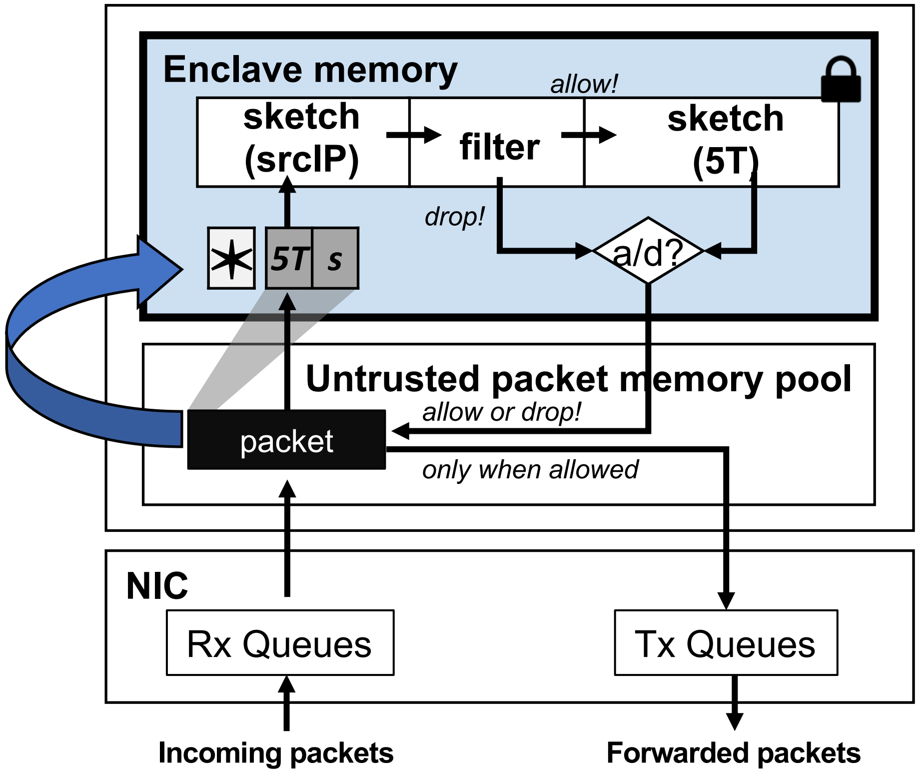
  } **1,206** SLoC
- Two optimizations:
  - ✓ Reducing context switches (more in our paper)
  - ✓ Near-zero copy approach

Optimization: ***near-zero*** copy

Full packet copy



Near-zero copy

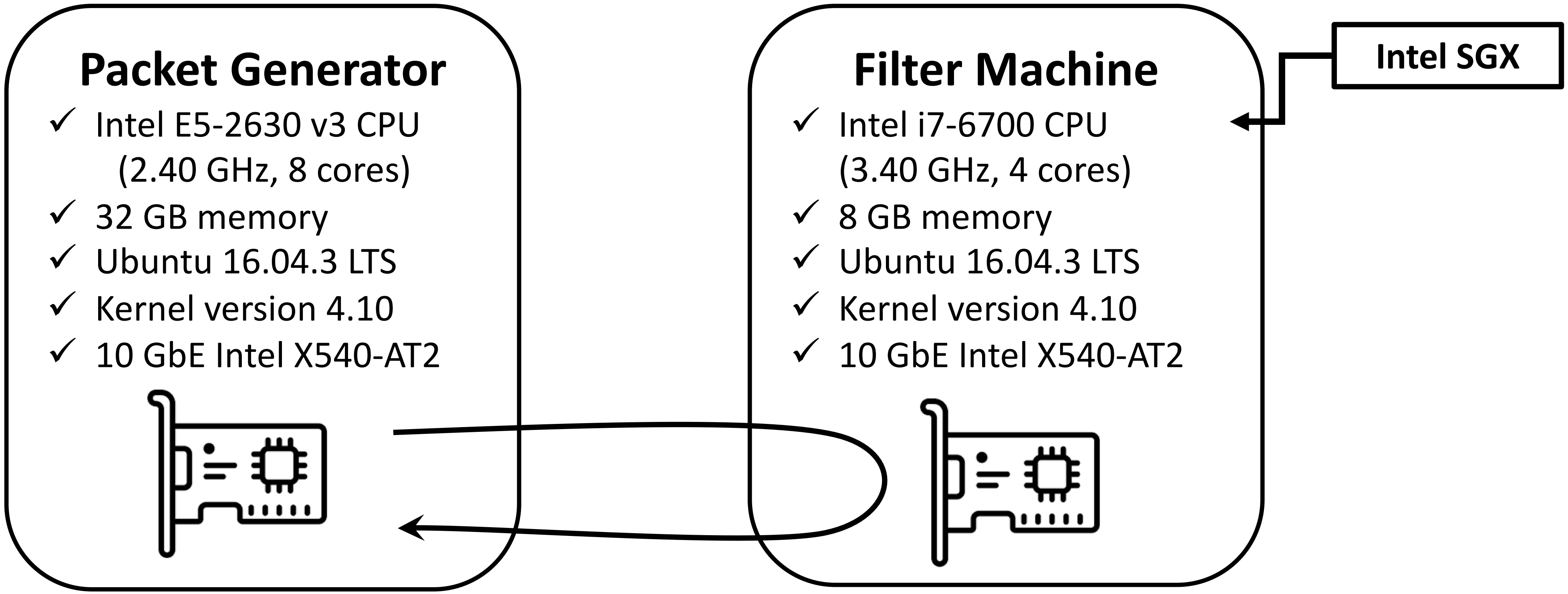


✓ low memory usage

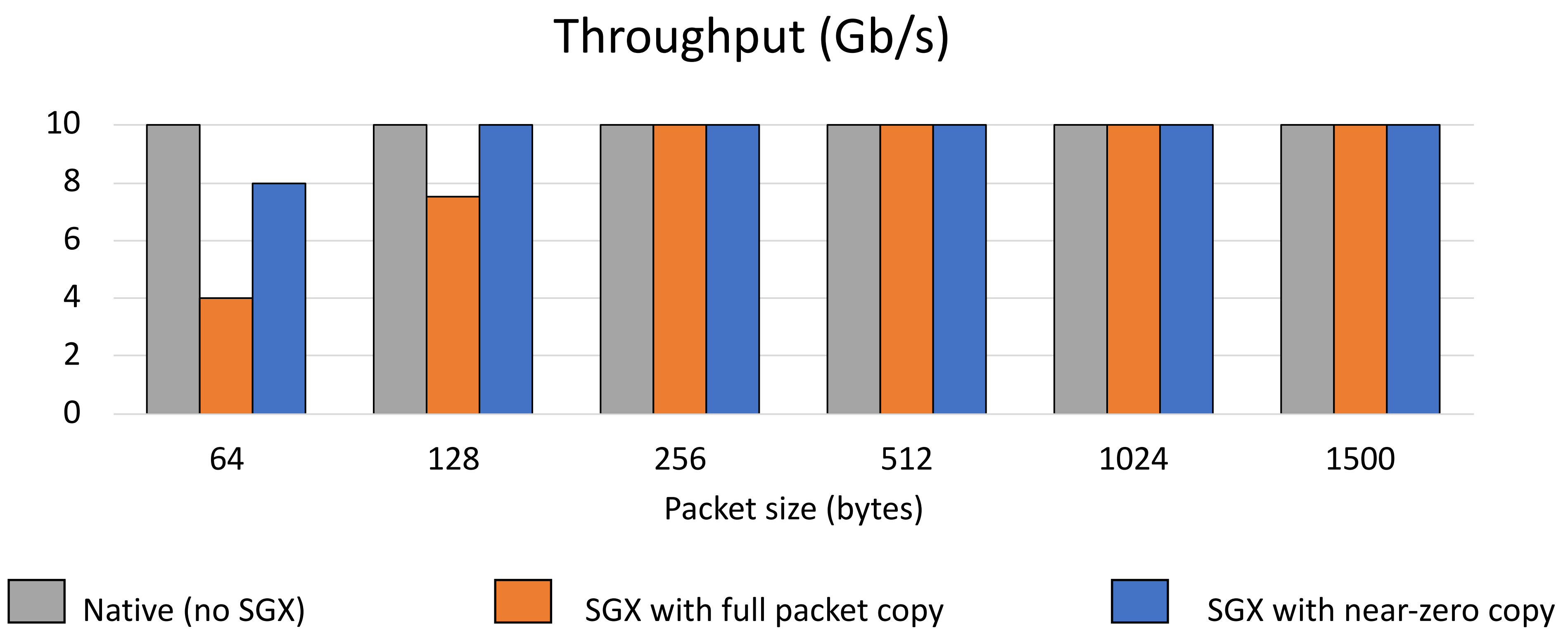
✓ low packet-logging overhead

Data-plane implementation

- **Testbed**
  - ✓ Packet generator ⟷ Filter machine
  - ✓ Measurement is done at packet generator
- **Synthetic data**
  - ✓ 3,000 random filter rules
  - ✓ 10 Gb/s traffic



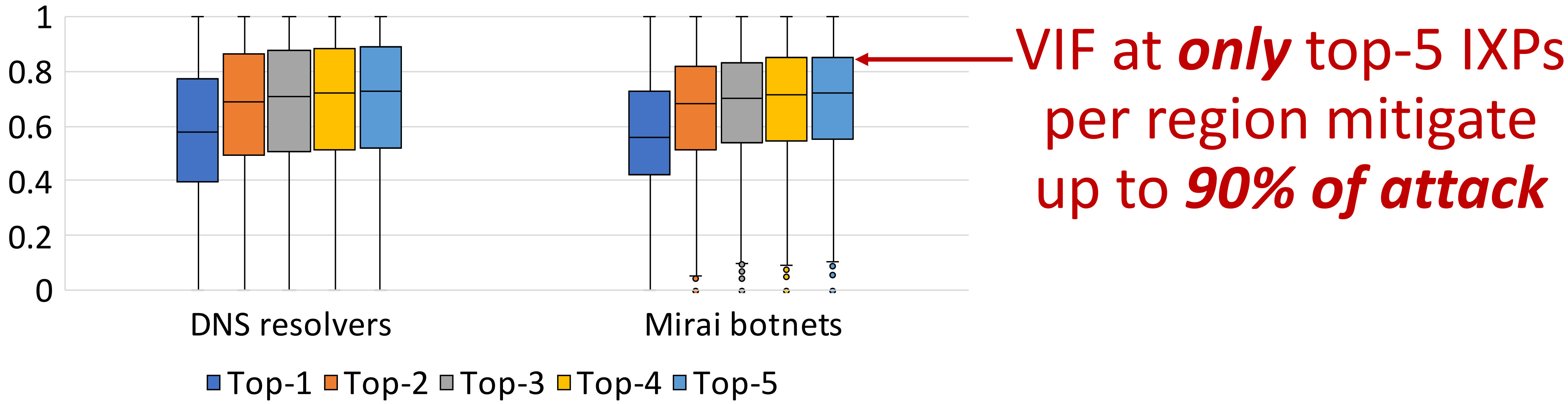
Evaluation: Data-plane performance



- **Throughput of near-zero copy:**
  - ✓ 8 Gb/s throughput even with smallest packet size (64 bytes)

Evaluation: VIF deployment at IXPs

Ratio of attack source IPs handled by top-***n*** IXPs per region



- Simulation setup:
  - ✓ Two real attack source data: ***3 millions*** DNS resolvers and ***250K*** Mirai botnets
  - ✓ CAIDA AS relationship and IXP peering for building inter-domain topology

Conclusion

- VIF addresses the ***core issue*** of in-network filtering
  - ✓ Lack of filtering verifiability → ***ambiguity*** in handling packet drops which can be exploited by malicious ISPs
- VIF: the first ***auditable*** and ***scalable*** DDoS traffic filter
- VIF takes advantages of:
  - ✓ ***Trusted execution environments*** as the root of trust
  - ✓ Software-defined, ***line-rate packet processing***
  - ✓ ***IXPs*** for practical deployment

Question?

Muoi Tran
muoitran@comp.nus.edu.sg

NUS
National University of Singapore