Enhancing Network Attack Detection with Distributed and In-Network Data Collection System

Seyed Mohammad Mehdi Mirnajafizadeh, Ashwin Raam Sethuram

David Mohaisen, DaeHun Nyang, Rhongho Jang

WAYNE STATE UNIVERSITY

UNIVERSITY OF CENTRAL FLORIDA

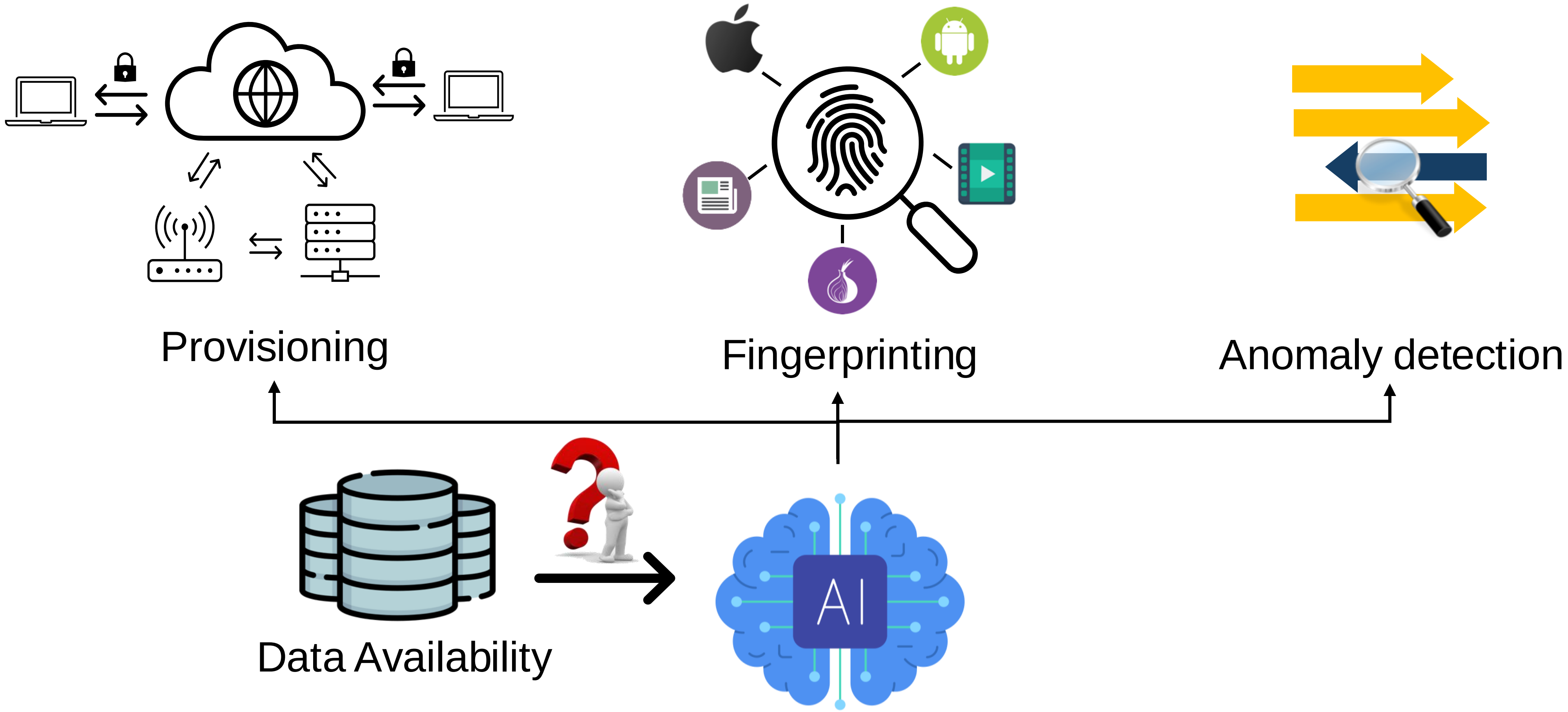
이화여자대학교 EWHA WOMANS UNIVERSITY

USENIX Security Symposium 2024

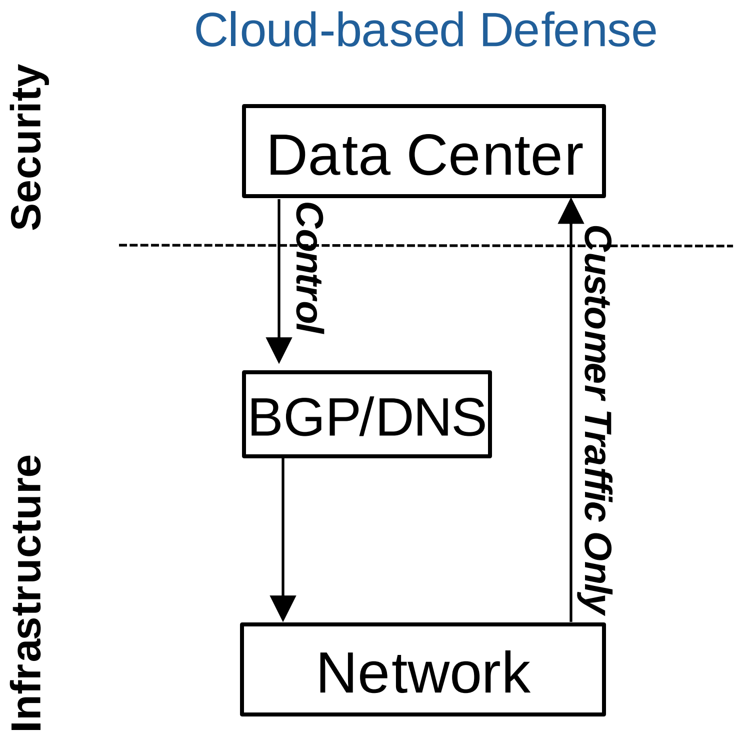
Outline

- Background - network traffic measurement
- Motivation - data availability for security
- Design goals - collaborative data collection
- Proposed system - ISDC
- Evaluation - covert channel/DDoS detection
- Conclusion

Background: Network Traffic Measurement

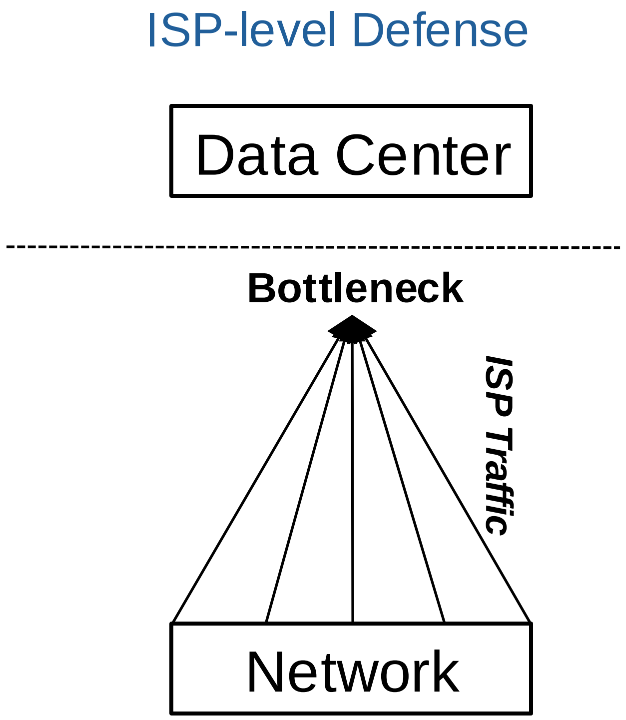


Motivation: Data Availability (DA) for Security



❌ Narrow Interest in Traffic

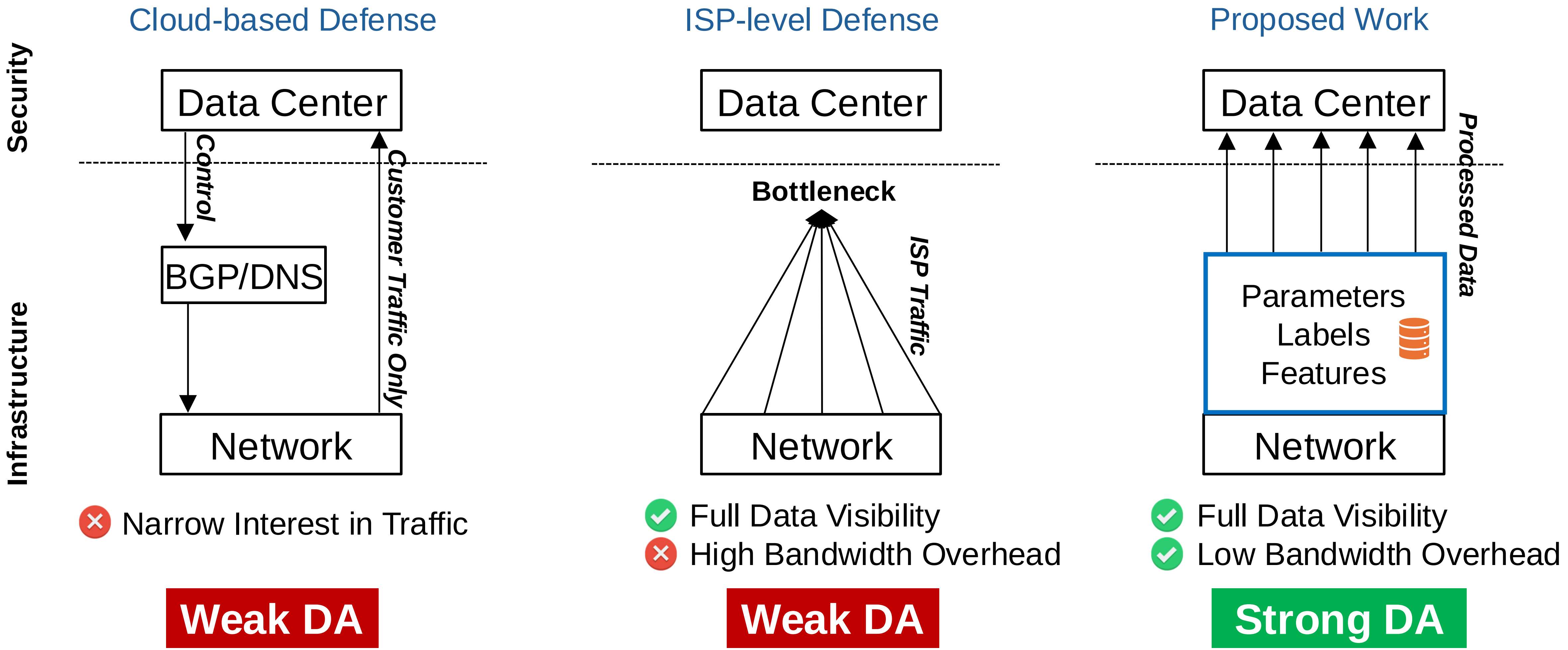
Weak DA



✅ Full Data Visibility
❌ High Bandwidth Overhead

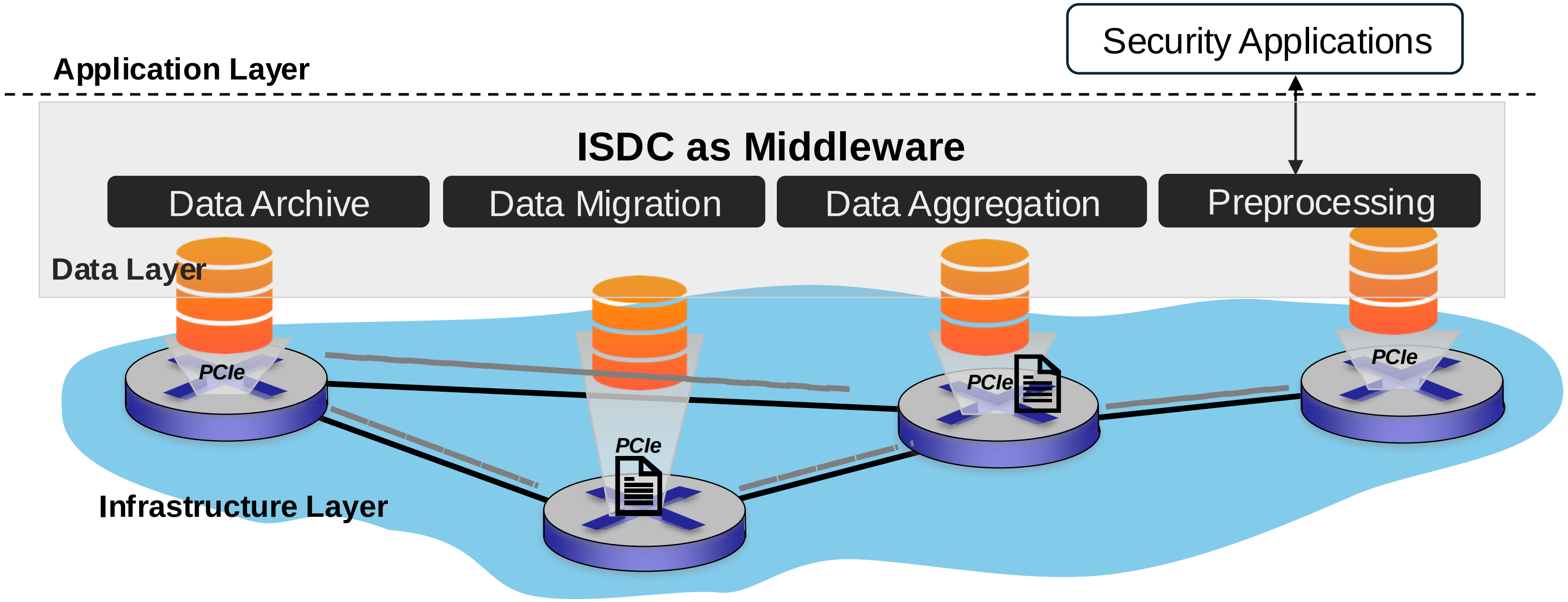
Weak DA

Motivation: Data Availability (DA) for Security



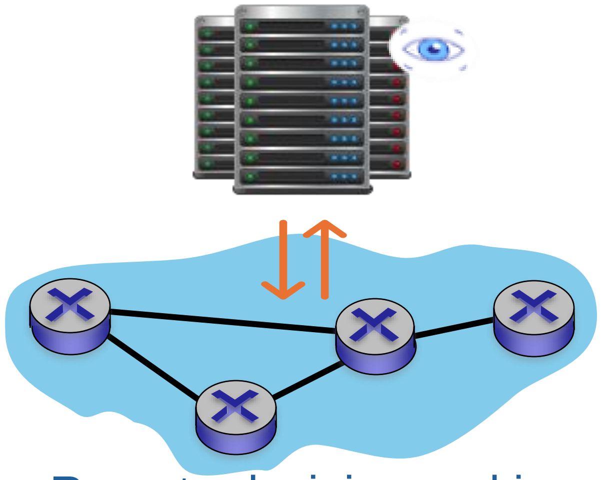
Our Goal: Collaborative Data Collection

- In-network Serverless Data Collection (ISDC)
  - Data plane collaborative network traffic measurement
  - Control plane (local switches) data aggregation/synchronization



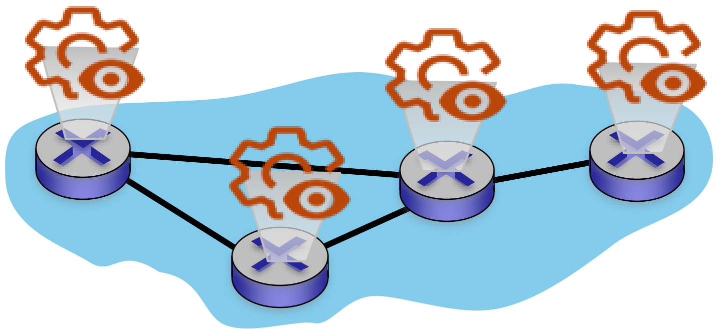
Prior Works: Resource Inefficiency during Collaboration

- Remote decision-making with global view for resource optimization [1][2]
- Onsite decision-making with local view for adaptiveness [3]



Remote decision-making

❌ Slow adaptation to dynamic shift



Onsite decision-making

❌ Duplicated task measurement

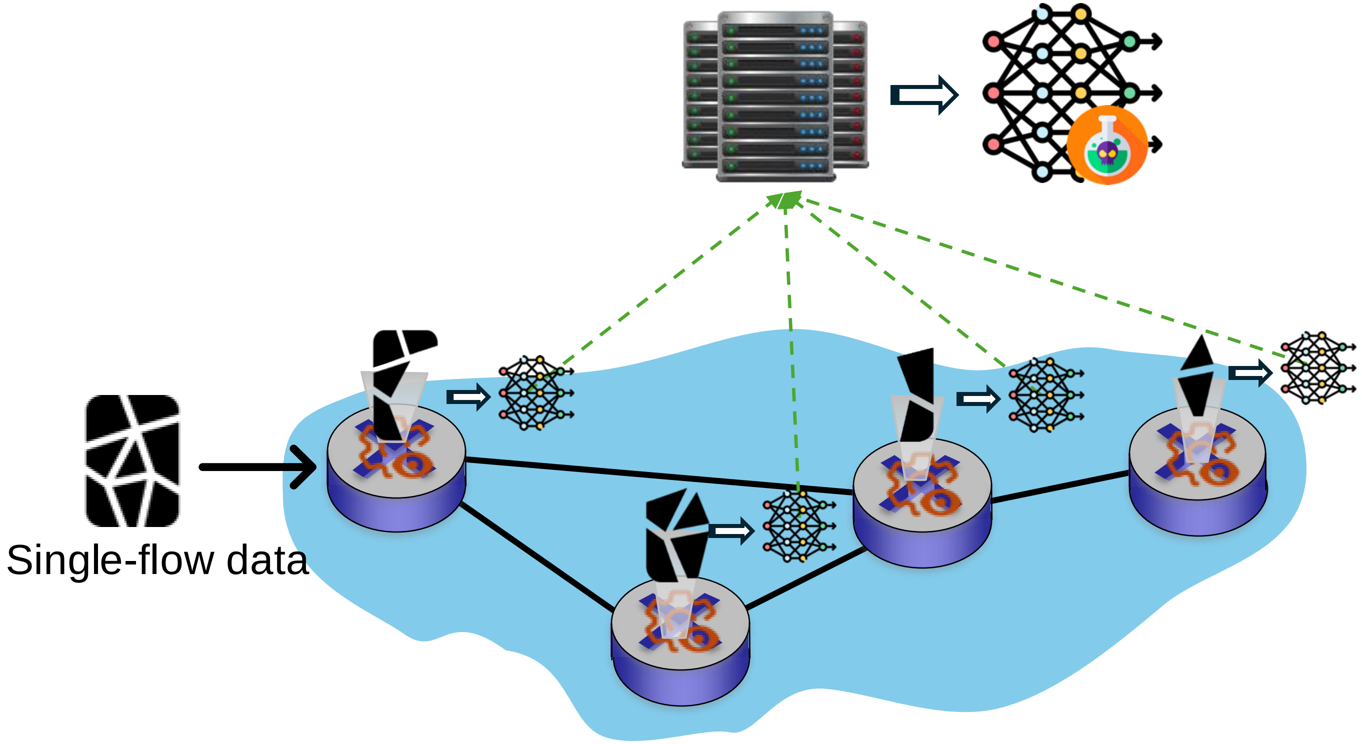
[1] Xu, Hongli, and et al. Lightweight flow distribution for collaborative traffic measurement in software defined networks. In Proc. of IEEE INFOCOM, 2019

[2] Sekar, Vyas, and et al. cSamp: A system for network-wide flow monitoring. In Proc. of USENIX NSDI, 2008

[3] Basat, Ran Ben, and et al. Cooperative network-wide flow selection. In Proc. of IEEE ICNP, 2020

New Insight: Data Fragmentation and Model Poisoning

- Local view decision-making creates fragmented data
  - Collected data is utilized as data source for distributed learning
  - Presence of fragmentation leads to model poisoning



[3] Basat, Ran Ben, and et al. Cooperative network-wide flow selection. In Proc. of IEEE ICNP, 2020

Design Goals

Goal 1: Optimize Network Resource Usage

- Effective resource utilization according to security application demands

Design Goals

Goal 1: Optimize Network Resource Usage

- Effective resource utilization according to security application demands

Goal 2: Dynamic Task Allocation

- Efficient task coordination to maximize network-wide resources

Design Goals

Goal 1: Optimize Network Resource Usage

- Effective resource utilization according to security application demands

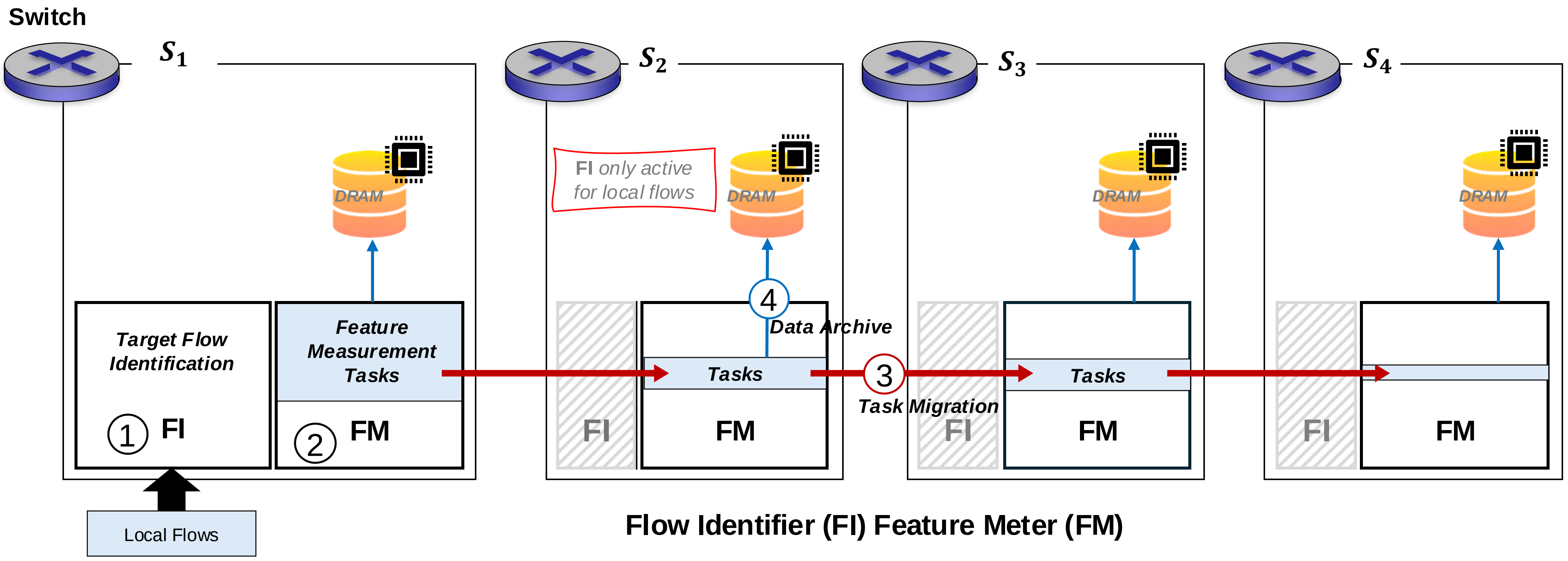
Goal 2: Dynamic Task Allocation

- Efficient task coordination to maximize network-wide resources

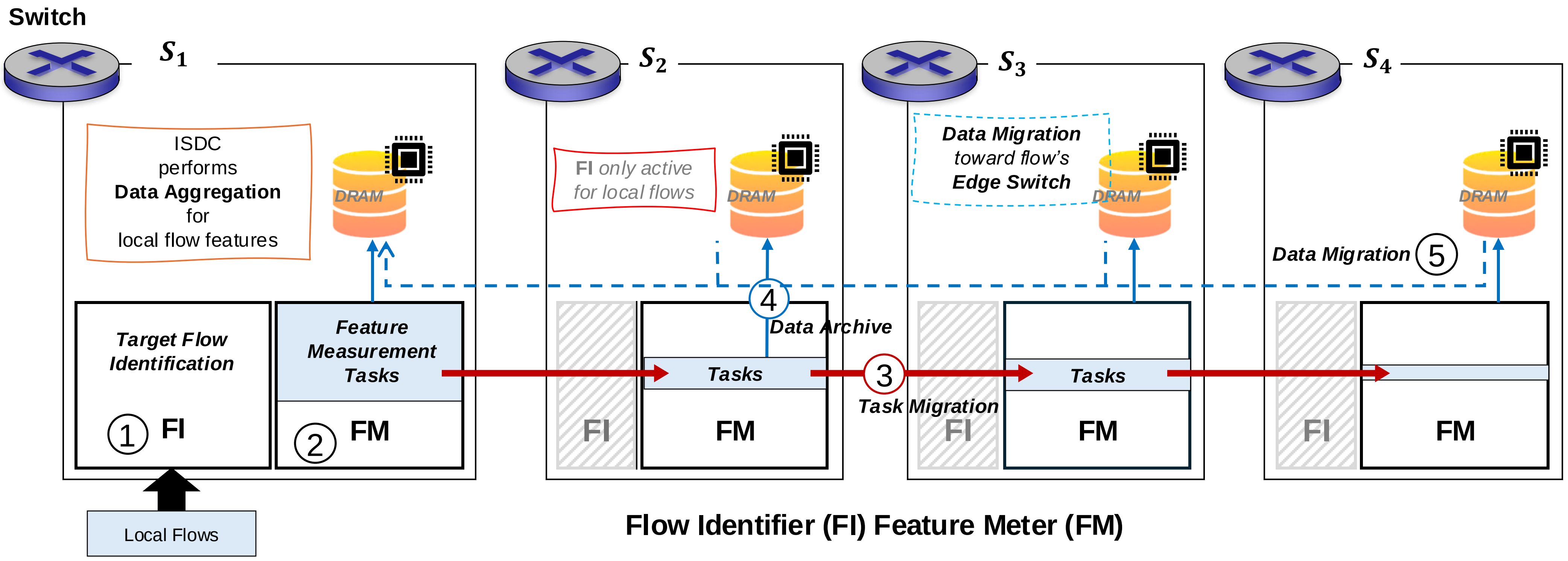
Goal 3: Reliable Data Source for Security

- Ensure data integrity to eliminate model poisoning caused by data fragmentation

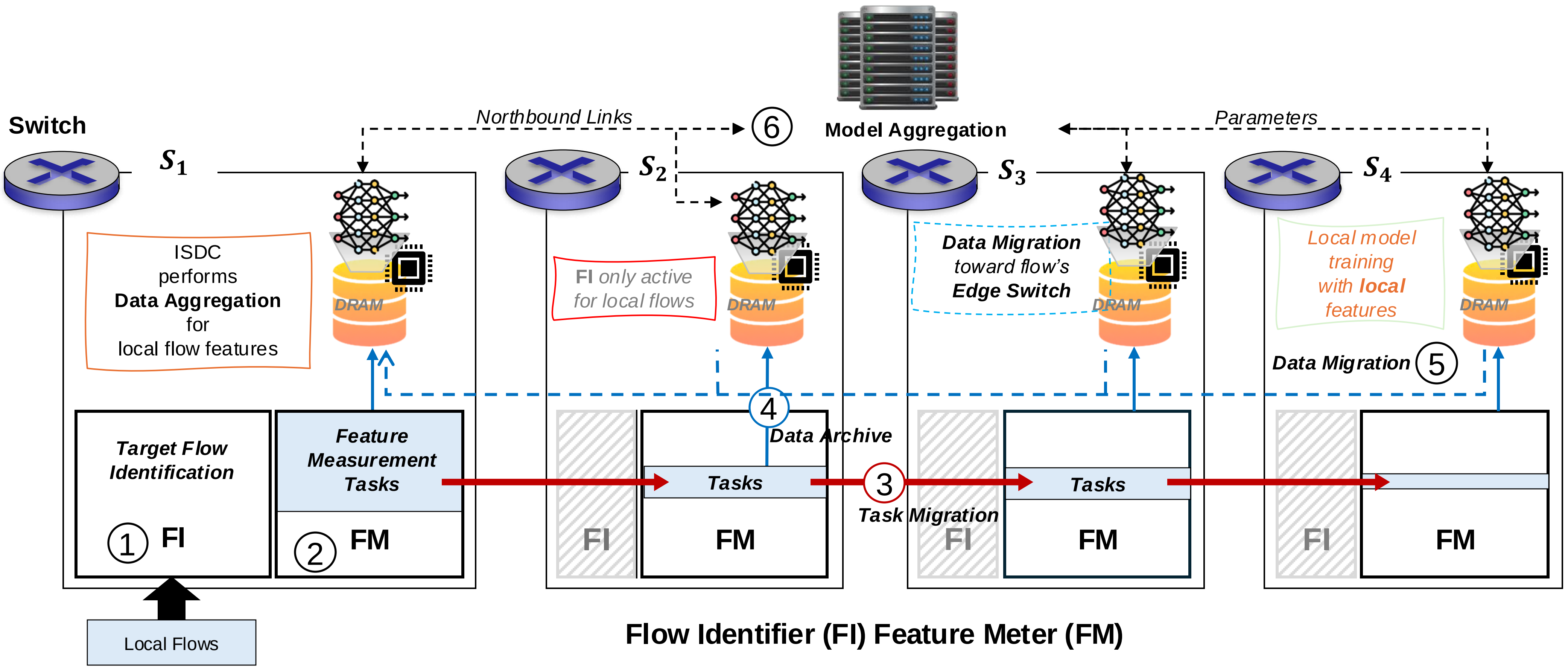
ISDC: Framework



ISDC: Framework



ISDC: Framework



Flow Identifier (FI) Feature Meter (FM)

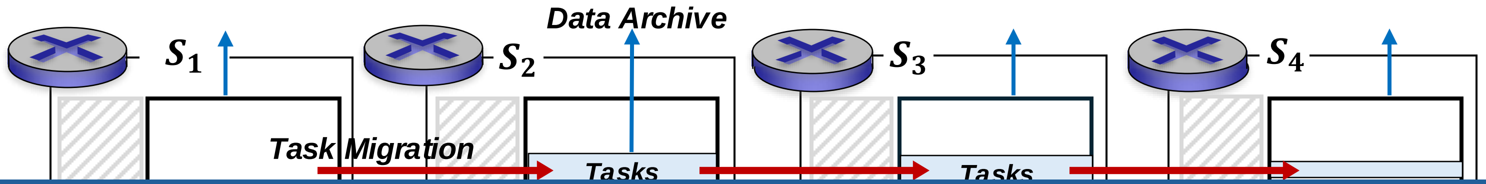
Design 1: Task Prioritization

- Challenge: Achieving full-flow coverage **is infeasible**
  - With the ever-increasing traffic volume
- Our approach: Application-focused prioritization
  - ML/DL disfavor sparse data points created by super mice flows with one or two packets
- **Flow Identifier (FI):** Real-time large flow prediction
  - Reducing memory/computational complexity from O(n) to O(1)

Reduced resource wastage in data collection for security application

Design 2: Dynamic Task Allocation

- Challenge: Lack of efficient collaboration
  - State-of-the-art onsite decision-making suffers from duplicated task measurement
- Our approach: Efficient and dynamic task collaboration
  - Having more task migration based on switch resources
  - When the task is migrated, the data is archived (task-data isolation)
- **Task migration:** A light-weight coordination protocol
  - A hybrid policy that applies two opposing strategies to maximize resource utilization and minimize task migration footprint

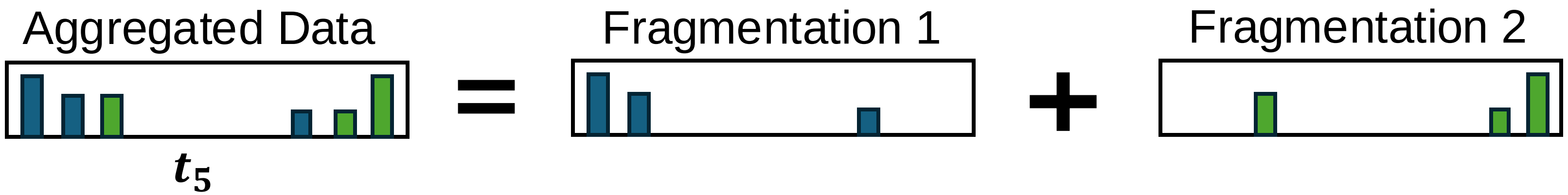


Maximize network utilization and increase flow coverage

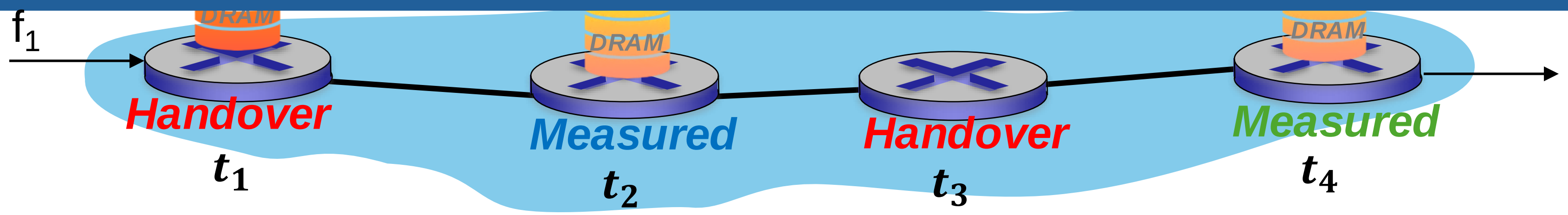
Flow Identifier (FI) Feature Meter (FM)

Design 3: Data Migration

- Challenge: Local view decision-making creates data fragmentation
  - Data fragmentation leads to model poisoning
- Our approach: In-network data aggregation
  - To enable a reliable foundation for ML/DL application
- **Data migration:** A light-weight, non-blocking protocol for data delivery/acknowledgment
  - No prior knowledge of network topology and routing path

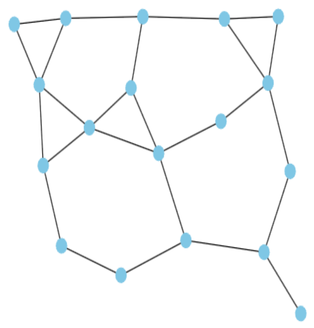


High-quality data without fragmentation via in-network data aggregation

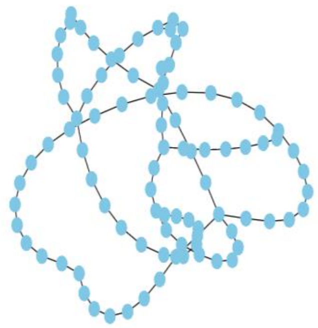


Evaluation: Setup

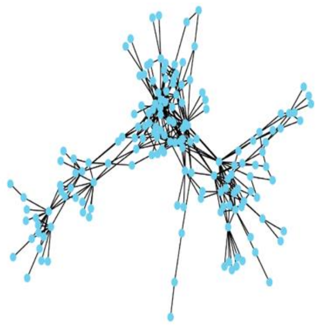
- Hardware and software implementations:
  - bmv2 P4 software switch in Mininet environment
  - Wedge 100BF-32X ASIC (Intel Tofino 1)
- Network topologies:
  - Small: 18/25 switch/links (ASN)
  - Medium: 92/96 switch/links (Vlt Wavenet)
  - Large: 161/328 switch/links (Tiscali)
- Security use cases:
  - Covert channel attack detection
  - DoS/DDoS attack detection



ASN

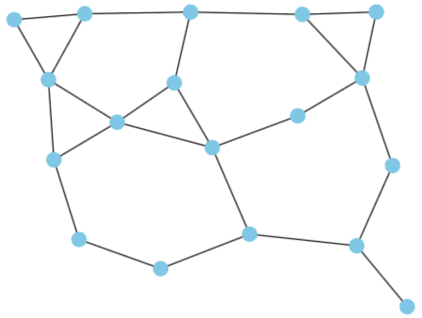


Vlt Wavenet

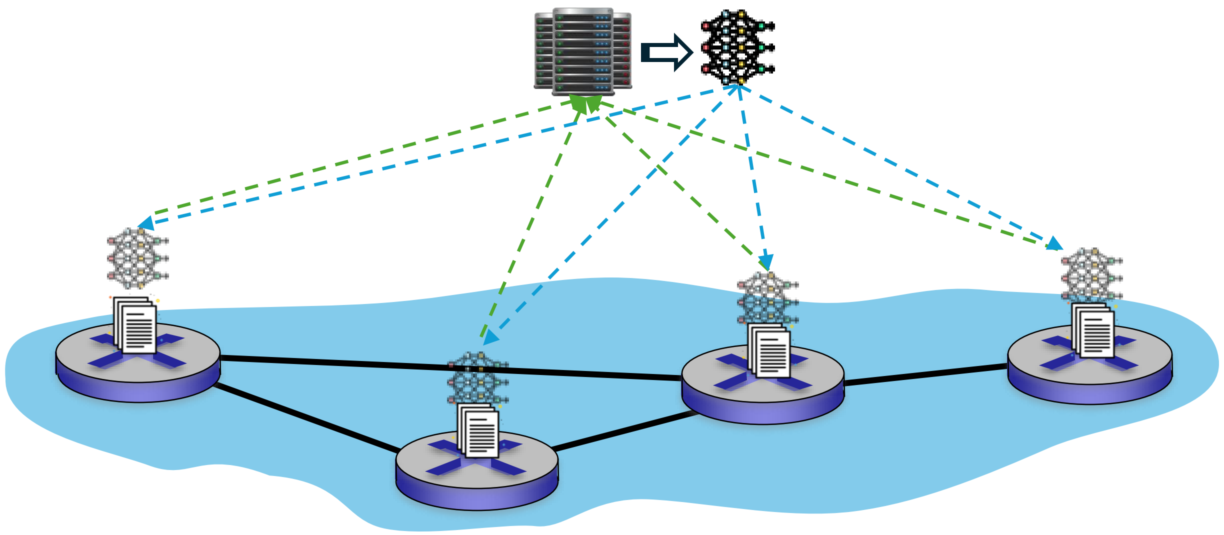


Tiscali (AS 3257)

Experimental Setting for Use Cases



- ASN topology with 18 switches and 25 links
- Collection of features from attack/benign traffic
  - Measured and aggregated in a distributed manner
- Distributed local data used for standard federated learning
- The global model is distributed to 18 switches for attack detection



Security Use Cases

#1: Covert Channel Detection

Schemes	Cov.	Frag.	Mem. Waste	Avg. WMRE	F1 1 rd.	F1 10 rd.	AUC 1 rd.	AUC 10 rd.
Strawman	30.6%	0%	52.5%	1.37	0.295	0.927	0.246	0.869
CSAMP	35.4%	0%	51.8%	1.27	0.824	0.923	0.718	0.862
NSPA	36.9%	0%	51.3%	1.25	0.816	0.927	0.709	0.868
CFS	58.1%	53%	62.1%	1.67	0.887	0.942	0.857	0.894
ISDC	**94.1%**	**0%**	**8.02%**	**0.18**	**0.960**	**0.970**	**0.938**	**0.967**

Lack of collaboration

Remote decision-making

Onsite decision-making

High flow coverage (Design 1, 2)

mentation (Design 3)

y (Design 1)

High feature accuracy (Design 3)

Enhanced ML performance for security

#2: DoS/DDoS Detection

Schemes	Cov.	Frag.	Mem. Waste	Avg. WMRE	F1 1 rd.	F1 10 rd.	AUC 1 rd.	AUC 10 rd.
CFS	49.9%	62%	67.6%	0.989	0.617	0.613	0.828	0.891
CFS-clean	49.9%	0%	67.6%	0.868	0.620	0.756	0.777	0.892
ISDC	**93.1%**	**0%**	**3.5%**	**0.297**	**0.730**	**0.809**	**0.860**	**0.945**

Security Use Cases

#1: Covert Channel Detection

Schemes	Cov.	Frag.	Mem. Waste	Avg. WMRE	F1 1 rd.	F1 10 rd.	AUC 1 rd.	AUC 10 rd.
Strawman	30.6%	0%	52.5%	1.37	0.295	0.927	0.246	0.869
CSAMP	35.4%	0%	51.8%	1.27	0.824	0.923	0.718	0.862
NSPA	36.9%	0%	51.3%	1.25	0.816	0.927	0.709	0.868
CFS	58.1%	53%	62.1%	1.67	0.887	0.942	0.857	0.894
ISDC	**94.1%**	**0%**	**8.02%**	**0.18**	**0.960**	**0.970**	**0.938**	**0.967**

#2: DoS/DDoS Detection

Schemes	Cov.	Frag.	Mem. Waste	Avg. WMRE	F1 1 rd.	F1 10 rd.	AUC 1 rd.	AUC 10 rd.
CFS	49.9%	62%	67.6%	0.989	0.617	0.613	0.828	0.891
CFS-clean	49.9%	0%	67.6%	0.868	0.620	0.756	0.777	0.892
ISDC	**93.1%**	**0%**	**3.5%**	**0.297**	**0.730**	**0.809**	**0.860**	**0.945**

+14%↑

High flow coverage (Design 1, 2)

Zero data fragmentation (Design 3)

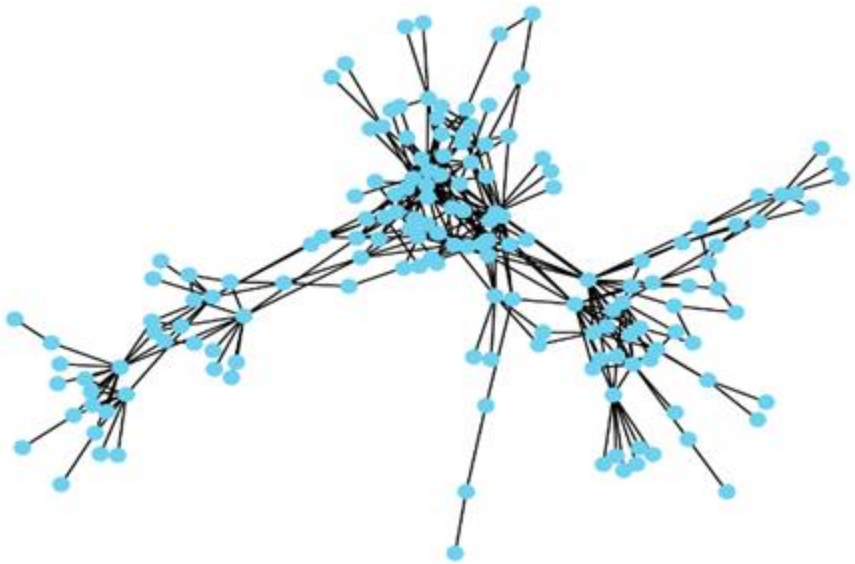
High memory efficiency (Design 1)

High feature accuracy (Design 3)

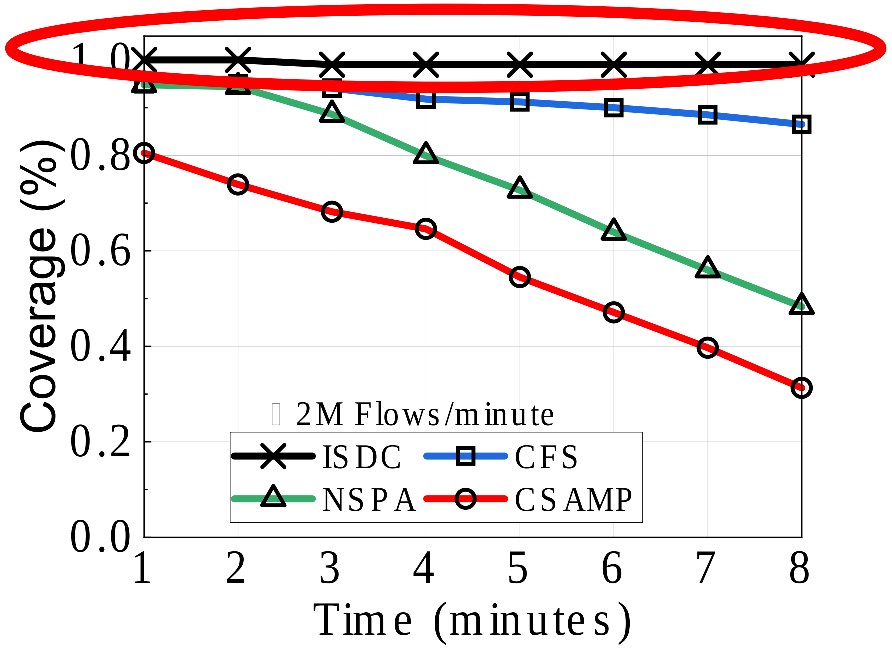
Enhanced ML performance for security

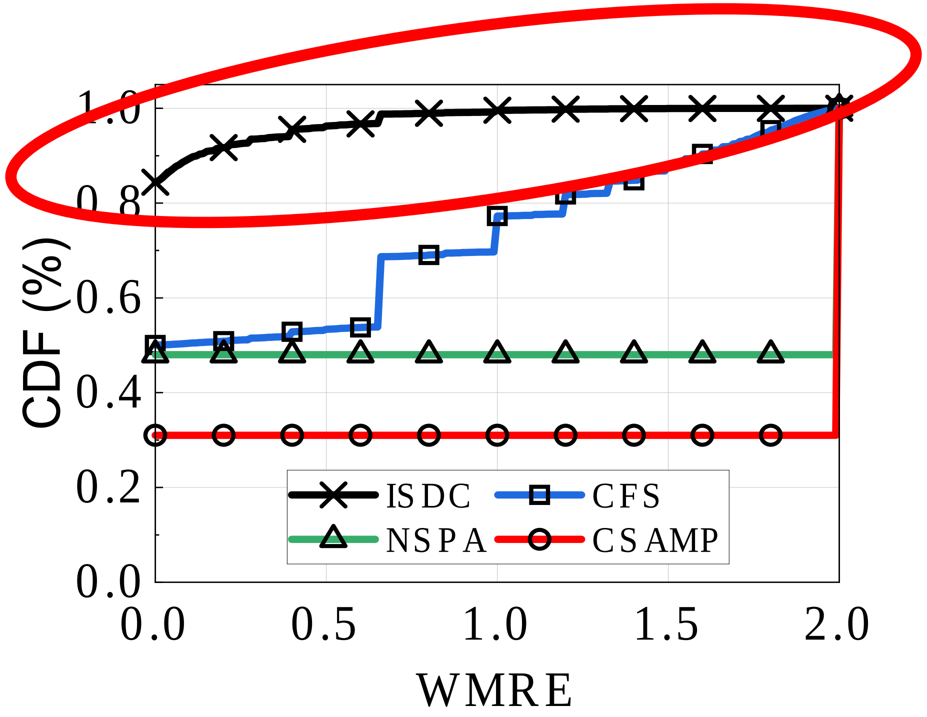
Experimental Setting for System Evaluation

- **Large-scale** real-world topology with **161** switches and **328** links
- Used **eight**-minute CAIDA traffic, a total of **14.5/250** million flows/packets
- Metric
  - Flow coverage (%): higher is better
  - Feature quality (WMRE): smaller is better



System Performance: Data Collection





- Consistent delivery of full data coverage for top-500k flows
- Delivery of high-quality data, with **95%** of collected features have WMRE of less than **0.5**

Conclusion

- Limitation of existing collaborative framework
  - Resource wastage
  - Fragmentation caused model poisoning
- **ISDC**
  - Effective resource usage
  - Efficient resource allocation
  - Light-weight in-network data aggregation
- Achieved goal
  1. High **coverage** and **quality** data collection
  2. Enhanced **data availability** for ML/DL security application
- Source code: https://github.com/NIDS-LAB/ISDC

Q & A

Thank You!