Verify All Traffic: Towards Zero-Trust In-Network Intrusion Detection against Multipath Routing

Ziming Zhao, Zhaoxuan Li, Xiaofei Xie, Zhipeng Liu, Tingting Li, Jiongchi Yu, Fan Zhang✉, Binbin Chen

***Abstract*—With the popularity of encryption protocols, machine learning (ML)-based traffic analysis technologies have attracted widespread attention. To adapt to modern high-speed bandwidth, recent research is dedicated to advancing zero-trust intrusion detection by offloading feature extraction and model inference into the network dataplane. Especially, with the rise of programmable switches, achieving line-speed ML inference becomes promising. However, existing research only considers a single switch node as a relay to conduct evaluation. This is far from real-world deployments involving multiple switches (given that zero-trust security assumes that threats can originate from anywhere, including within the network), particularly the multipath routing phenomenon that exists in practice. In this paper, we reveal practical challenges in the context of enabling line-speed model inference in the network dataplane. Furthermore, we propose FCPlane, the forwarding and computing integrated dataplane for zero-trust intrusion detection that aims to enable efficient load balancing while providing reliable traffic analysis results, even against multipath routing. The core idea is to reconcile forwarding and computation to the flowlet level, for which a tailor-made Markov chain model is designed. Based on two public traffic datasets, we evaluate seven state-of-the-art in-network traffic analysis models deployed in four types of topologies (three with multipath routing and one without) to explore performance impact and demonstrate the effectiveness of our proposal.**

***Index Terms*—Zero-trust intrusion detection, multipath routing, programmable switches dataplane, in-network deployment**

Manuscript received 31 May 2024; revised 16 November 2024; accepted 18 February 2025. This work was supported in part by National Key R&D Program of China (2023YFB3106800), by the National Research Foundation, Singapore, and the Cyber Security Agency under its National Cybersecurity R&D Programme (NCRP25-P04-TAICeN), by National Natural Science Foundation of China (62227805, 62072398, 62172405), by the Natural Science Foundation of Jiangsu Province (BK20220075), by the Fok Ying-Tung Education Foundation for Young Teachers in the Higher Education Institutions of China (No.20193218210004), and by Key R&D Program of Zhejiang Province (2023C01039). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore and Cyber Security Agency of Singapore. (Corresponding author: Fan Zhang)

Ziming Zhao is with the School of Software Technology, Zhejiang University, Ningbo, 315100, China. E-mail: zhaoziming@zju.edu.cn.

Zhipeng Liu, Tingting Li, and Fan Zhang are with the College of Computer Science and Technology, Zhejiang University, Hangzhou, 310027, China. Fan Zhang is also with ZJU-Hangzhou Global Scientific and Technological Innovation Center, 311200, with the Key Laboratory of Blockchain and Cyberspace Governance of Zhejiang Province, 310027, with Jiaxing Research Institute, Zhejiang University, 314000, and with Zhengzhou Xinda Institute of Advanced Technology, Zhengzhou, 450001, China. He is a visiting professor at the Singapore University of Technology & Design (SUTD). E-mail: {zhipengliu, litt2020, fanzhang}@zju.edu.cn.

Zhaoxuan Li is with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, 100093, China, and also with the School of Cyber Security, UCAS, Beijing, 100049, China. E-mail: lizhaoxuan@iie.ac.cn.

Xiaofei Xie and Jiongchi Yu are with the School of Computing and Information Systems, Singapore Management University, Singapore 188065. E-mail: jcyu.2022@phdcs.smu.edu.sg, xfxie@smu.edu.sg.

Binbin Chen is with Advanced Digital Sciences Center, Singapore, Singapore, 138632, and with Singapore University of Technology and Design, Singapore, Singapore, B96049. E-mail: binbin_chen@sutd.edu.sg.

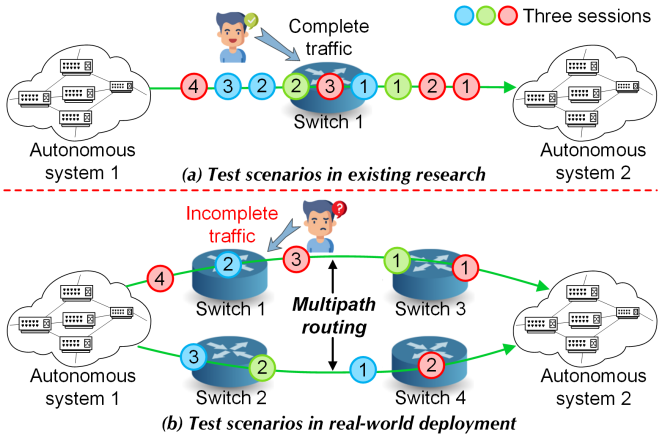


Fig. 1. Illustrative explanation of in-network traffic analysis against multipath routing. Note that the packets are simplified to unidirectional for clear presentation in the diagram, the traffic is bidirectional in the real world.

I. Introduction

Network traffic analysis is an important technology in current Internet infrastructure, which can be used for network management [1], intrusion detection system (IDS) [2], malware identification [3], *etc.* With the widespread application of encryption protocols, traditional signature-based or payload-based methods have encountered certain limitations [4]–[6]. As emerging solutions, machine learning (ML) models are gradually being used to profile and analyze encrypted traffic due to their good characterization and fitting capabilities [6]–[8]. Some representative solutions [2], [4], [9]–[12] propose marking user entities (*e.g.,* IP addresses) based on traffic detection results to form a blacklist or a whitelist. However, this paradigm does not conform to the zero-trust security concept [13], [14] for next-generation networks [15], given that one of the core principles of zero-trust security is "Never trust, always verify" [16]–[18].

In line with the zero-trust concept, network intrusion detection systems should tend to verify all traffic rather than marking priorities or privileges for certain user entities [4], [9]. To verify all traffic, the first problem should advance the *online line-speed traffic analysis in the dataplane to adapt to current high-speed bandwidth* [19], instead of *offline capture*

of traffic and execution of model inference on CPU/GPU [8]. Towards this end, a series of studies has been carried out by practitioners and communities. For example, Kitsune [8] puts lots of effort into making AutoEncoder-based detection run efficiently, while it only supports ∼112 Mbps throughput, which is a far cry from meeting the demand of modern network bandwidth (*e.g.,* 100 Gbps per port). To solve this problem, offloading feature extraction and ML model inference into the network dataplane (also as known *in-network analysis* [12]) is a promising scheme. Particularly, the emerging programmable switches allow user-customized packet processing logic in a protocol-independent manner, it has been widely studied among researchers and industry [12], [20], [21].

Although in-network ML-based traffic analysis has made extraordinary progress, existing work ignores the impact of multipath routing[1], which is very common in real-world networks [23]–[26]. Figure 1 provides an intuitive example to showcase the gap between the model development (in existing research) and the real-world system deployment. The detection scenarios in the existing research are shown in subfigure (a), where two autonomous systems (ASes) are bridged through a switch thus all traffic can be captured in switch 1 (*i.e.,* without considering multipath routing). However, the test scenarios in real-world deployment are usually as subfigure (b), several switches form a certain topology to support communication between ASes. The multipath routing could result in perceiving incomplete traffic on a switch since some traffic could pass other candidate links. Therefore, while in-network traffic analysis has been extensively researched and holds promise as a solution to advance zero-trust IDS, it still faces challenges when deployed in real-world networks, *e.g.,* the impact of multipath routing on model robustness.

In this paper, we reveal new challenges (to advance zero-trust IDS) in the context of opportunities to enable line-speed model inference in the network dataplane. Specifically, we first explore the magnitude of incomplete traffic induced by multipath routing under different topologies and assess the impact the multipath routing phenomenon has on ML model performance, to provide new directions and guidance regarding the development and deployment of ML models within networks. Furthermore, this paper presents FCPlane, a Forwarding and Computing integrated data Plane design that efficiently manages network traffic, especially in the presence of multipath routing. Our core idea is to reconcile forwarding and computation to the flowlet level, and then advance the forwarding module and the computing module designs including Flowlet Timeout Value (FTV) determination and dynamic routing strategies, as well as flowlet-level Markov model design and in-network dataplane deployment for real-time traffic analysis.

In summary, this paper makes three key contributions.

- We reveal the multipath routing challenge to advance zero-trust IDS in the context of opportunities to enable line-speed model inference in the network dataplane.
- We present FCPlane, a forwarding and computing integrated dataplane design that aims to enable efficient load balancing while providing reliable traffic analysis results (even against multipath routing), to promote zero-trust network IDS.
- We conduct a series of experiments to demonstrate the challenges encountered by the in-network model, regarding the intra-flow inconsistency of packet-based models and the performance degradation of flow-based models against multipath routing. Then, we implement the prototype of forwarding and computing integrated dataplane on our physical testbed. And we conduct extensive evaluations to demonstrate the advantages of the proposed scheme. The experiments show that our proposal realizes ∼99% F1 score for intrusion/malware detections even against multipath routing. As for the load-balancing effect, our method can achieve a good Jain's fairness index compared with existing methods.

[1]Zero-trust IDS assumes that threats can originate from anywhere, including the internal networking [14], [22]. Thus, in-network traffic analysis models should be deployed into decentralized detection points, and the multipath routing problem arises. More details are in § II and § III.

II. Background and Motivation

A. Motivation

We first introduce the research motivation in the context of zero-trust security. The high-level idea references Figure 2. It describes the tenets of zero-trust security and then explains the zero-trust views and requirements in network IDS. Subsequently, it introduces feasible schemes, *i.e.,* in-network model traffic analysis deployment strategy for verifying all traffic. Finally, it proposes the multipath routing challenges encountered by in-network models in practice.

(i) According to the Zero-Trust Security (ZTS) architecture [27] published by the National Institute of Standards and Technology (NIST), ZTS is a paradigm shift toward rethinking the network security and protection of organizational assets. ZTS involves several tenets [17], including while not limited to the following aspects:

① Zero trust assumes *there is no implicit trust* granted to assets or user accounts based solely on their physical or network location (*i.e.,* local area networks versus the internet) or based on asset ownership (enterprise or personally owned).

② Zero trust provides a set of principles and concepts around moving the Policy Decision Point (PDP) & Policy Enforcement Point (PEP) closer to the resource. The idea is to *explicitly authenticate and authorize all subjects, assets, and workflows* that make up the enterprise.

③ *Access to resources is determined by dynamic policy*, including the observable state of client identity, application/service, and the requesting asset, as well as may include other behavioral and environmental attributes.

④ For zero trust, the policy is the set of access rules based on attributes that an organization assigns to a subject, data asset, or application. These rules and attributes are based on the needs of the business process and acceptable levels of risk. *Least privilege principles are applied to restrict both visibility and accessibility*.

⑤ All resource authentication and authorization are dynamic and strictly enforced before access is allowed. This is a constant cycle of obtaining access, scanning and assessing

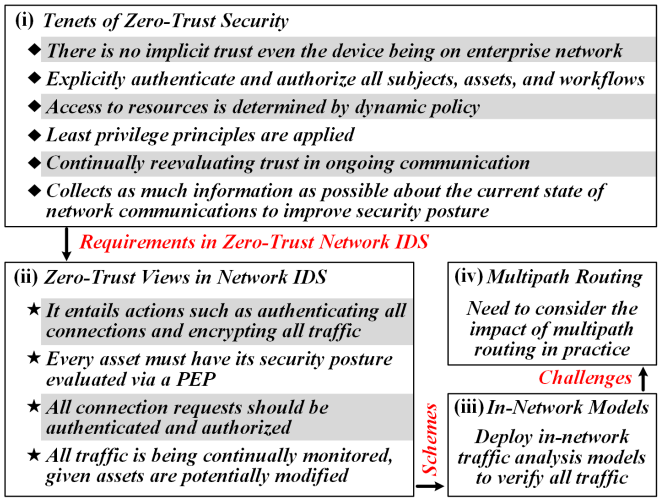


Fig. 2. Motivation for studying multipath routing with in-network models in zero-trust network IDS.

threats, adapting, and *continually reevaluating trust in ongoing communication*.

⑥ An enterprise should *collect data about asset security posture, network traffic, and access requests*, process that data, and use any insight gained to improve policy creation and enforcement.

(ii) Considering the above tenets of zero-trust security architecture, zero-trust views of a network are formed [27], which introduces a series of requirements for network IDS. Specifically, they include the following.

① The entire enterprise private network is not considered an implicit trust zone. Assets should always act as if an attacker is present on the enterprise network, and communication should be done in the most secure manner available. In other words, trust should not be automatically granted based on the device being on enterprise network infrastructure. *This entails actions such as authenticating all connections and encrypting all traffic.*

② No resource is inherently trusted. *Every asset must have its security posture evaluated via a PEP before a request is granted to an enterprise-owned resource.* This evaluation should be continual for as long as the session lasts. Subject credentials alone are insufficient for device authentication to an enterprise resource.

③ *All connection requests should be authenticated and authorized*, and all communications should be done in the most secure manner possible (*i.e.,* provide confidentiality, integrity protection, and source authentication).

④ Remote enterprise subjects and assets cannot fully trust their local network connection. Remote subjects should assume that the local (*i.e.,* nonenterprise-owned) network is hostile. *Assets should assume that all traffic is being monitored and potentially modified.*

(iii) The above zero-trust views of network IDS requires that all traffic should be inspected and verified in a real-time manner [18], rather than marking priorities or privileges for certain user entities (such as blacklist or whitelist) [4], [9]. However, practitioners and communities find that the key problem refers to *offline capture of traffic/execution of model inference cannot adapt to current high-speed bandwidth* [19]. To solve this problem, offloading feature extraction and implanting ML model inference into the network data plane (also as known *in-network deployment* [12]) is a promising and feasible scheme, as summarized in § III-B.

(iv) Although in-network traffic analysis model deployment has made extraordinary progress, existing work ignores the impact of multipath routing in practice (as discussed in § II-B) [23]–[26]. Zero-trust IDS assumes that threats can originate from anywhere, including within the network [14], [22]. Thus, in-network traffic analysis models should be deployed into decentralized detection points, and the multipath routing problem arises.

The occurrence of multipath routing can be caused by a variety of reasons, including link failures [28], network topology changes [29], [30], routing policy adjustments [31], [32], *etc.* Specifically, to cope with link failures, the scalability, control, and isolation on next-generation networks (SCION) end hosts use multi-path communication by default, thus masking link failures to an application with another working path [28]. Moreover, in real-time traffic applications, it is often unacceptable to lose connectivity because of changes in topology [30] or adjustments of routing policies [32] that render routes obsolete, multipath routing guarantees reliable end-to-end connectivity [31]. Some factors affect the occurrence probability of multipath routing, including network topology, routing protocols, load balancing strategies, *etc.* From the application layer to the link layer, the multipath characteristics of different protocols vary with respect to latency, loss sensitivity, reliability, resource reservation, and so on [23]. In addition, different multipath routing algorithms will also lead to different path allocation results due to different path selection strategies, as discussed in the prior art [23].

Therefore, multipath routing is indeed a common phenomenon, which is also an unavoidable problem in the development and deployment of in-network traffic analysis models in practice, especially considering the decentralized node deployment and detection of zero-trust IDS. To this end, this paper focuses on the robustness of in-network traffic analysis models against multipath routing, evaluating the performance of a series of representative models in various (serial and parallel) topologies. On the one hand, this paper reveals the shortcomings of existing research solutions on in-network traffic analysis models in practice (especially the robustness against multipath routing phenomena). On the other hand, we intend to propose a novel forwarding and computing integrated dataplane to enable efficient load balancing while providing reliable traffic analysis results (even against multipath routing), to promote zero-trust IDS.

B. Practical Case

To advance the development and deployment of ML-based models for zero-trust IDS in practice, we have launched an in-network traffic analysis project with the local provincial Internet Service Provider (ISP) in 2022. Its core network contains over a hundred interconnected switches (five refer to

TABLE I
EXISTING REPRESENTATIVE METHODS FOR IN-NETWORK TRAFFIC ANALYSIS.

Method	Traffic	Feature	Model	Primitive	Deployment	Bandwidth
Mousika [21]	Packet-level	Packet fields such as TTL, length	Tree	P4	Data plane	∼100 Gbps
Flowrest [19]	Flow-level	Statistical features of packet fields	RF	P4	Data plane	∼100 Gbps
NetBeacon [12]	Flow-level	Statistical features of packet fields	Tree/RF/XGB	P4	Data plane	∼100 Gbps
RIDS [11]	Flow-level	Packet length sequence	RNN	P4	Data plane	∼100 Gbps
FlowLens [20]	Flow-level	Packet size distribution histograms	NB/XGB/RF	P4	Data plane + Control plane	∼1 Gbps
DFNet [4]	Packet-level	Raw packet byte	DNN→Tree	DPDK	Data plane	∼40 Gbps
Whisper [33]	Flow-level	Frequency domain features	Clustering	DPDK	Data plane	∼13 Gbps
FCPlane (Proposed)	Flowlet-level	Packet fields such as flags, length	Markov chain	P4	Data plane	∼100 Gbps

[1] FlowLens performs model inference in control plane.
[2] Whisper leverages DPDK primitives.

programmable switches, and the others are not programmable), involving more than 2000K active hosts. Specifically, we choose the decision tree-based model [12], [19], [21] for development and deployment. Based on the pre-collected benchmark dataset (involving a series of common intrusion attacks and legitimate business traffic of protected departments), we trained the model and performed the offline test (including directly connecting the sender side and receiver side through a programmable switch) to achieve F1 score over 98%.

Subsequently, we deployed these in-network traffic analysis models into five programmable switches of the ISP's core network. Then, we replayed the benchmark test traffic on the link, while finding that the detection effect of the in-network traffic analysis model (into the programmable switches dataplane) was not satisfactory (usually below 60% F1 score). By verifying the original traffic on the sender side (by comparing the sent traffic and the traffic received on the switch node), we found that complete traffic sessions cannot be observed on these five programmable switch nodes (resulting in the calculated feature shift and thus model misclassification). This is mainly caused by the multipath routing phenomenon (common in complex network topologies), that is, multiple parts of traffic session through different links, resulting in incomplete traffic captured on the programmable switch.

III. RELATED WORKS

Based on the detailed explanation of motivation in § II, in this section, we introduce related work in terms of zero-trust security, in-network ML-based traffic analysis models, and multipath routing.

A. Zero-Trust Security

As traditional network perimeters dissolve and threats evolve, the need for robust security frameworks becomes more critical. Zero-Trust Security (ZTS) [13]–[15] is an emerging paradigm that addresses these challenges by redefining trust boundaries within a network. Zero-trust security is a strategic approach that requires strict identity verification for every user, device, and system attempting to access resources [34], [35], whether inside or outside the network perimeter [14]. The core principles [16] of ZTS include least-privilege access, never trust, always verify, *etc.*

Building on the foundations of ZTS, zero-trust network IDS takes a proactive stance in identifying and mitigating threats. On the one hand, it needs to leverage in-network model inference to verify all traffic at line speed rather than marking user entities with priorities or privileges. We will systematically introduce the related work about in-network traffic analysis in § III-B. On the other hand, zero-trust IDS operates on the assumption that threats can originate from anywhere, including within the network [14], [22]. This indicates that in-network analysis models should be deployed in decentralized detection points, and the multipath routing problem arises (details are in § III-C).

B. In-Network ML-Based Models

Regarding the ML-based approaches for in-network traffic analysis, researchers first focus on deploying decision trees/random forests within the dataplane, given that tree-based models are essentially a series of *if-else* conditional branches, which are suitable for implementation with network programming primitives. The early schemes [36]–[39] do not provide switch hardware implementation and use the BMv2 emulator [40] instead, which results in the inability to adapt to high bandwidth. The following work [12], [19], [21], [41], [42] considers switch constraints and implements tree-based model hardware deployment. Among them, Mousika [21] proposes to deploy the Binary Decision Tree on the programmable switches for packet-level classification[2]. NetBeacon [12] and Flowrest [19] enable stateful flow classification based on the multi-phase decision tree and random forests, respectively. Moreover, FlowLens [20] leverages various ML models for traffic analysis such as Naïve-Bayes (NB), XGBoost (XGB), and Random Forest (RF), while it performs model inference in the control plane (made only ∼1Gbps bandwidth can be saturated). As a recent art, RIDS [11] is the first in-network IDS implementation to deploy recurrent neural network inference directly on the programmable switch dataplane via ternary matrices, tailor-made flow state maintenance, and other hardware-friendly operations, similar to this there is INDP [43]. Furthermore, there are some methods based on the Intel DPDK [44]. DFNet [4] adopts the surrogate model to convert the deep neural network (DNN) into a decision

[2]Packet-level classification has the problem of consistency check, given traffic sessions perform field analysis and payload reorganization at the flow level. When one packet is misclassified, the effectiveness of the entire flow will be affected (could lead to unavailability of legitimate flow or defense fails), thereby making the impact of misclassification worse. We provide more details on the specific cases in § V-D.

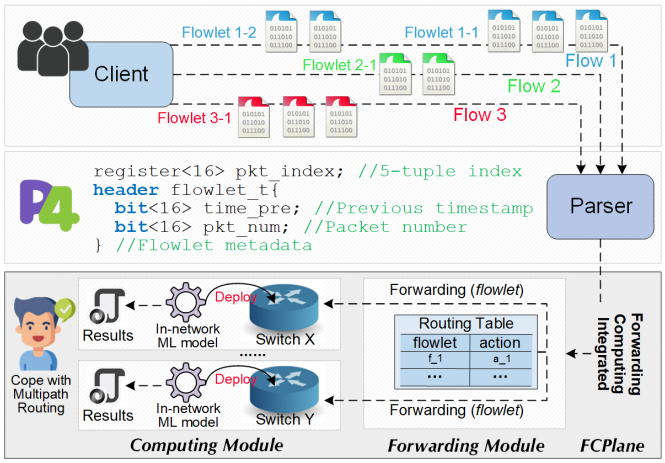


Fig. 3. The design overview of FCPlane. forwarding and computing integrated design. The forwarding module involves flowlet abstraction and routing table maintenance. The computing module is responsible for traffic representation and deployed in-network model inference to generate results.

tree, and Whisper [33] computes frequency domain features to perform clustering algorithms. In contrast, as a software-based solution, DPDK's bandwidth support capability is more limited than hardware-based programmable switches. Overall, we select 7 representative in-network traffic analysis models for evaluation, summarized in Table I.

C. Multipath Routing

Multipath routing is an avant-garde strategy that capitalizes on the redundancy present in contemporary network architectures to bolster the robustness and efficacy of data conveyance [45]. By pinpointing and harnessing multiple parallel conduits within a network, multipath routing can mitigate congestion and ameliorate load balancing [46]. This dispersion can transpire at varying granularities, such as individual packet level, by stream, or through more advanced techniques like subflow tagging. The latter, as epitomized by Flowlet [47], provides heightened adaptability by enabling the segmentation of a flow into several subflows that can be directed autonomously via divergent avenues.

Furthermore, B4 [48] combines existing routing protocols and traffic engineering services to accommodate elastic traffic demand. A custom implementation of ECMP hashing acts at the level of the flow to perform load balancing. B4 is also a hybrid strategy for centralized traffic engineering that relies on the separation between the data and control planes. HEDERA [49] monitors active flows and seeks to obtain a global view of routes and traffic to estimate the demand of flows in data centers. In summary, multipath routing is ubiquitous in the real world, so the impact of multipath routing should be considered in the design and evaluation of in-network models [50].

IV. Forwarding and Computing Integrated Design for Zero-Trust IDS

In this section, we first introduce the high-level workflow of forwarding and computing integrated dataplane design for zero-trust IDS. Then, we elaborate on the design details in terms of the forwarding module and computing module.

Algorithm 1 Flowlet-based Zero-Trust IDS Workflow

Input: Client traffic
Output: Analyzed and routed traffic
procedure Traffic Processing
  for all sessions s in traffic **do**
    Segment s into flowlets f
    for all flowlets f **do**
      Extract flowlet header metadata
      if time interval exceeds FTV **then**
        Link reallocation for a new flowlet
      end if
      Forward f to the integrated dataplane pipeline
    end for
  end for
  Execute **In-network model** on each switch node
  Ensure reliable results even with multipath routing
end procedure

A. Overview

The forwarding and computing integrated dataplane design overview of FCPlane is shown in Figure 3, it allows for the efficient analysis and routing of network traffic, particularly when dealing with multipath routing. In this design, the client traffic session is segmented into discrete units called flowlets. Each flowlet represents a portion of a network session and carries metadata that can be used for analysis and routing decisions. In Algorithm 1, the flowlet processing involves several key steps: (i) Segmentation. Client traffic is divided into flowlets, each containing a series of packets that are part of the same network session. (ii) Metadata extraction. For each flowlet, metadata such as the 5-tuple index (source IP, destination IP, source port, destination port, protocol), the previous timestamp, and the packet number are extracted. (iii) Flowlet maintenance. If the time interval between packets within a flowlet exceeds the predefined Flowlet Timeout Value (FTV), the flowlet is considered complete, and a new flowlet begins. (iv) Dataplane pipeline. Flowlets are then passed through the dataplane pipeline for further analysis and routing decisions.

This forwarding and computing integrated dataplane design has advantages, including efficient use of network resources through dynamic routing and in-network processing and enhanced security via real-time traffic monitoring and detection.

B. Forwarding Module

The forwarding module focuses on how to cut traffic into flowlets of appropriate sizes, and then execute routing strategies and dynamic adaptation.

Flowlet Abstraction. For flowlet-based routing, the Flowlet Timeout Value (FTV, denoted as δ) is a critical parameter that significantly influences the performance of load balancing. The determination of the FTV is essential for adapting to varying traffic loads and for preventing performance degradation due to frequent oscillations or shifts of flows between paths.

A suitable FTV threshold should meet the following conditions to ensure effective network traffic management and analysis. On the one hand, the FTV should be set in such a way that it promotes the generation of flowlets with as uniform a distribution as possible. This uniformity allows for efficient link utilization, even when employing straightforward Equal-Cost Multi-Path (ECMP) routing strategies. The goal is to balance the traffic load across different paths, minimizing the waste of bandwidth and avoiding congestion. On the other hand, the FTV must not be set too small, as this could result in overly fragmented flows and additional reordering operations. More importantly, these small flowlets might not carry enough information for subsequent in-network traffic analysis models. Taking both of the above points into consideration is an important step in achieving forwarding-computing integration.

For the first condition, we tend each flowlet to have a similar total byte size. Let F be a set of flowlets, where each flowlet $f_i \in F$ is characterized by the total byte size $B(f_i)$ of all packets within the flowlet. The objective is to minimize the variance of the total byte sizes across all flowlets, which leads to a more uniform distribution of flowlet sizes. We define the uniformity U of flowlet generation as the inverse of the variance of the total byte sizes:

$$U = \frac{1}{\text{Var}(B(F))} \tag{1}$$

where $B(F)$ is the set of total byte sizes for all flowlets in F, and Var denotes the variance. The goal is to choose an FTV that minimizes the variance of the flowlet sizes, leading to a uniform distribution:

$$\min_{\text{FTV}} \quad \text{Var}(B(F)) \tag{2}$$

This optimization problem can be approached by setting the FTV such that the expected total byte size $E[B(f_i)]$ of each flowlet f_i is approximately equal to the average total byte size $\bar{B}$ of all flowlets:

$$E[B(f_i)] \approx \bar{B} = \frac{1}{|F|} \sum_{f \in F} B(f) \tag{3}$$

where $|F|$ is the total number of flowlets.

In other words, the adjustment of the FTV aims to balance the trade-off between the granularity of flowlets and the uniformity of their sizes, ensuring that consecutive flowlets f_i and f_{i+1} have similar total byte sizes:

$$|B(f_i) - B(f_{i+1})| \leq \epsilon \tag{4}$$

where ϵ is a small threshold value that represents the allowable difference in byte sizes between consecutive flowlets.

For the second condition, that flowlet should maintain enough information for model analysis, we construct a Discrete-Time Markov Chain (DTMC) model [5]. Let $G = \{V, E\}$ denote the state diagram of the DTMC, where V is the set of states (representing the values of per-packet features) and E denotes the edges (representing transitions between states). We define $s = |V|$ as the number of distinct states, and let $W = [w_{ij}]_{s \times s}$ denote the weight matrix of G. The transition probabilities are given by a normalized weight matrix $P = [P_{ij}]$, where $P_{ij} = \frac{w_{ij}}{w_i}$ and $w_i = \sum_{j=1}^{s} w_{ij}$.

The entropy rate of the DTMC, denoted by $H[G]$, is the expected Shannon entropy increase for each step in the state transition, which quantifies the uncertainty or information content of the flowlets. The entropy rate can be calculated using the following formula:

$$H[G] = -\sum_{i=1}^{s} \sum_{j=1}^{s} w_{ij} \log(w_{ij}) + \sum_{i=1}^{s} w_i \log(w_i) \tag{5}$$

The determination of the optimal FTV can be framed as an optimization problem where the objective is to maximize the information entropy captured by the DTMC model under the limited difference in byte sizes between consecutive flowlets.

Routing Strategy. Our routing strategy leverages the elasticity of flowlets to achieve resilient load balancing. First, each switch maintains a flowlet table with entries indexed by the flow's 5-tuple, *i.e.,* $\{SourceIP,\ DestinationIP,\ Sourceport,\ Destinationport,\ Protocol\}$. Then, each entry in the flowlet table will store the previous timestamp and number of packets. Upon the arrival of a packet, the switch hashes the 5-tuple to find the corresponding flowlet table entry. If the entry is valid (*i.e.,* the time interval is active), the packet is forwarded according to the stored port. If not, a new flowlet is initiated, and the switch reselects an output port and forwarding action according to the load-balancing algorithm.

Load balancing decisions are made at the granularity of flowlets, the probability $\mathcal{P}()$ of choosing path P_j as follows,

$$\mathcal{P}(P_j) = \frac{C_j}{\sum_{i=1}^{N} C_i} \tag{6}$$

where C_j is the capacity of path P_j, and N is the total number of paths.

Dynamic Size Adaptation. The FTV for each flowlet is calculated dynamically based on the number of forwarded packets at switches. This is inspired by the fading mechanism [51], where the FTV fades (or decreases) as more packets are forwarded, allowing for timely rerouting of traffic under different workloads. The FTV is adapted using a shift operation that effectively represents the fading of the timeout value. This method is lightweight and can be operated in the dataplane of programmable switches. For a flow and an initial FTV δ_0, the adapted FTV δ can be calculated using a right shift operation as follows:

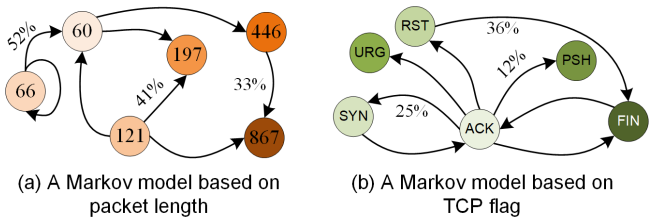
$$\delta = \delta_0 \gg 1 \tag{7}$$

where $\gg$ denotes the right shift operation, the result of the shift operation effectively reduces δ_0.

C. Computing Module

The computation module is mainly responsible for extracting field information from traffic and building a series of Markov chain models for traffic classification. We next introduce the specific model design and deployment pipeline.

Traffic Representation. In the context of encrypted traffic analysis, we can utilize various packet header fields to construct these states. For instance, we could use packet length sequences and TCP flag fields for traffic representation. (i) Packet length sequences are derived from the size of



(a) A Markov model based on packet length

(b) A Markov model based on TCP flag

Fig. 4. Illustrative explanation of Markov models.

consecutive packets within a flow. As the most representative feature in previous work [6], [7], [33], packet length sequence[3] has been proven useful for guiding traffic detection tasks. (ii) TCP flag fields, such as URG, ACK, PSH, RST, SYN, and FIN, provide critical information about the state and control aspects of a TCP connection. By analyzing the patterns and transitions of these flags within a flow, we can infer the nature of the traffic. For instance, the SYN flag typically indicates the initiation of a connection, while the FIN flag signals the termination. Meanwhile, some malicious traffic may have specific flag sequence patterns, such as TCP-based DDoS [52].

Flowlet-Level Analysis Model. Given we tend to analyze traffic at the flowlet level, the first-order Markov chain model is considered to be employed for traffic modeling. The rationale behind choosing a first-order Markov chain that lies in its future state prediction depends solely on the current state, rather than on the sequence of past events. By segmenting a session into multiple flowlets, we can capture localized patterns and behaviors within network traffic. Each flowlet retains sufficient information to allow the Markov chain model to make informed predictions about the subsequent flowlet, based on the current state.

Figure 4 provides two distinct first-order Markov chain models, each representing a probabilistic system with transitions based on specific network traffic characteristics. The left subfigure is constructed based on the sequence of packet lengths. In this model, each packet length is a state, and the transitions between states represent the probabilities of one packet length following another. For instance, a packet with a length of 66 bytes has a 52% probability of being followed by a packet with a length of 60 bytes. This particular transition could correspond to the three-way handshake process in the TCP protocol. The right subfigure is based on the TCP flags of packets. TCP flags such as SYN, ACK, PSH, FIN, RST, and URG are crucial for controlling the state of a TCP connection and communication between hosts. For example, the presence of a SYN flag may be followed by an ACK flag with a certain probability, indicating the establishment of a connection. Similarly, a FIN flag could be followed by an ACK, signaling the termination of a connection. The model helps in visualizing the typical progression of TCP flag sequences and can be used to detect potential security threats.

[3]Previous arts have shown that packet length sequences as stateful features [6], [7], [11] contain sufficient information entropy for traffic classification [33], and these features are widely applicable to protocol types [4] such as TCP, UDP, ICMP, HTTP, DNS, FTP, SSH, *etc.*

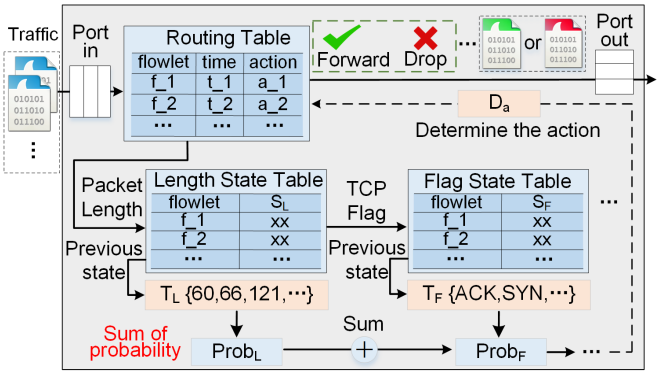


Fig. 5. The forwarding and computing integrated pipeline.

In-Network Deployment. In Figure 5, we depict the in-network deployment for the proposed flowlet-level Markov model. The pipeline is designed to process incoming traffic and classify it based on the characteristics extracted from packet headers. The process begins with matching the traffic against table entries based on the 5-tuple fields, which determines the flowlet index. This index directs the packet flow through a series of state tables designed to extract specific features from the packet headers. The pipeline includes dedicated state tables (note that the state table and routing table can be on the same switch or different switches), each associated with a respective Markov chain model (capture the probabilistic behavior inherent in the traffic flows). The model parameters, specifically the transition probabilities, are pre-trained and scaled by a factor of 10^3 to accommodate the integer-only arithmetic of the programmable switch hardware (floating point numbers are not supported). The length state table processes the packet length sequences. For each packet, the length is extracted and used to consult the corresponding Markov model parameters. Similarly, the flag state table focuses on the TCP flag fields, such as ACK, SYN, FIN, etc. The state table maps the observed flag patterns to states in the Markov model, computing the sum of probabilities. This aggregate (sum of) probability serves as a fingerprint for the traffic flow, which is then matched against a predefined set of classification criteria to determine the flow's category. Based on the classification result, the pipeline configures the appropriate action for the flowlet (such as forwarding or dropping). Overall, it completes the integration of forwarding logic and computing analysis for traffic in the programmable switch dataplane.

V. Evaluation

In this section, we assess the multipath routing phenomenon and evaluate the impact on in-network ML-based traffic analysis models (including existing baselines and our proposed flowlet-based zero-trust IDS). To this end, we develop four topology scenarios on our physical testbed. For two public traffic datasets, we replay their traces and capture the traffic on each switch node in the topology for model performance tests. Specifically, the experiments are designed to answer the following research questions.

RQ1 (§ V-C). How do existing flow-level in-network models perform when against multipath routing?

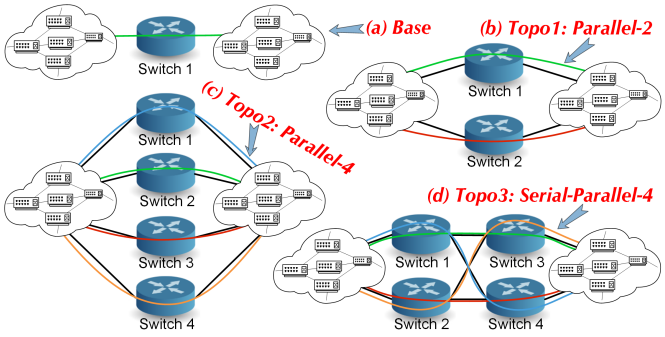


Fig. 6. The four test topology scenarios. Different colors in each topology indicate candidate links.

RQ2 (§ V-D). How do existing packet-level in-network models present in terms of result consistency?

RQ3 (§ V-E). How does the proposed flowlet-level model perform compared to the baseline?

RQ4 (§ V-F). How are the traffic distribution effect and the dataplane throughput of forwarding-computing integrated dataplane for zero-trust IDS?

A. Experimental Setup

Testbed and Tool. In our physical testbed, we build four topology scenarios as shown in Figure 6, including three (*Topo1: "Parallel-2"*, *Topo2: "Parallel-4"*, and *Topo3: "Serial-Parallel-4"*) with multipath routing in subfigure (b)-(d) and one (*Base*) without, *i.e.,* subfigure (a). *Topo2* and *Topo3* are the parallel expansion and series expansion of *Topo1*, respectively. All switch nodes reference the Tofino switch (Wedge 100BF-32X) with 32 100 Gbps ports. The programmable switches employ per-packet Equal-Cost Multi-Path (ECMP) by default [53]. The sending/receiving hosts are equipped with 100 Gbps Mellanox ConnectX-5 network cards and Ubuntu 20.04.1. MoonGen [54] is used to replay PCAP traffic. For the initial value determination of Flowlet Timeout Value (FTV) δ, we use the Z3 SMT solver [55] to maximize the optimization objective of § IV-B. FCPlane deploys dynamic size adaptation for FTV by default, as introduced in § IV-B. In addition, we also conduct the ablation experiments with different FTV settings in § V-E.

Model and Dataset Selection. As introduced in § III-B, we select seven representative in-network traffic analysis models for evaluation, summarized in Table I. Particularly, XGB is chosen for NetBeacon [12] and FlowLens [20] because XGB achieves the best performance in the original paper. For model inference, FlowLens performs in the control plane, Whisper [33] and DFNet [4] are deployed in DPDK, and the others are all executed on switches, which is consistent with their original papers.

The datasets used for evaluation are in Table II, including two intrusion detection datasets and a malware identification dataset. (i) IDS2017 and IDS2018 [56] include various intrusion attacks, and a series of legitimate interactions traffic. (ii) USTC-TFC dataset [57] contains network traffic generated by ten malware (*e.g.,* Cridex, Zeus) and ten categories of benign applications (*e.g.,* Facetime, Skype) from real-world connections. Considering that some baselines do not support multiple classifications, we perform binary classifications (*i.e.,* distinguish benign and malicious) in all models for a fair comparison. If not otherwise stated, the dataset division ratio refers to $train : test = 6 : 4$.

TABLE II
DATASETS USED IN OUR EVALUATION.

IDS 17&18	Benign	Human interaction traffic produced by B-Profile
	DDoS	UDP, TCP, HTTP
	DoS	Hulk, GoldenEye, Slowloris, Slowhttptest, Heartbleed
	Botnet	Remote shell, File, Key logging
	Patator	FTP, SSH
	Web	Brute-force, XSS, SQL inject
USTC	Benign	BitTorrent, Facetime, FTP, Gmail, MySQL, Outlook, Skype, SMB, Weibo, World of Warcraft
	Malware	Cridex, Geodo, Htbot, Miuref, Neris, Nsis-ay, Shifu, Tinba, Virut, Zeus

B. Evaluation Metrics

The evaluation metrics involve three aspects. (i) For the model classification performance, we mainly use the Accuracy Acc and F1-score $F1$. (ii) For the load balancing effect, we use Jain's fairness index to evaluate the uniformity of the traffic distribution in each switch, the details are in § V-F. (iii) We statistics the multipath routing probabilities and calculate the completeness of flow. Given a group of switches contains n equivalent candidate nodes $G_S = \{S_1, S_2 \cdots S_n\}$, *e.g.,* Switches 1 & 2 & 3 & 4 in Topo2 (Figure 6). We capture traffic of each switch $S_i, i \in [1, n]$, denoted as $\mathcal{T}_i$. For traffic $\mathcal{T}_i$, it can be divided based on the 5-tuple session index (*i.e.,* {Source IP, Source Port, Destination IP, Destination Port, Protocol}) to obtain the flow list $L_i = \{f_1^i, f_2^i \cdots f_m^i\}$ that consists m flows. For any $f_q^i \in L_i$, if its index (*i.e.,* $f_q^i.index$) appears in the traffic $\mathcal{T}_j$ of any other switch S_j (*i.e.,* $j \neq i$), we term f_q^i as the flow with multipath routing and the packets of f_q^i are packets with multipath routing[4]. Then, we can calculate the proportion of packet and flow in which multipath routing occurs. Specifically, for switch S_i, the flow ratio of multipath routing R_F is computed by

$$R_F^i = \frac{\sum_{q=1}^{len(L_i)} \{1 | \exists j, j \neq i, f_q^i \in L_i, f_q^i.index \in \mathcal{T}_j\}}{len(L_i)} \quad (8)$$

and the packet ratio of multipath routing R_P of switch S_i is

$$R_P^i = \frac{\sum_{q=1}^{len(L_i)} \{len(f_q^i) | \exists j, j \neq i, f_q^i \in L_i, f_q^i.index \in \mathcal{T}_j\}}{\sum_{q=1}^{len(L_i)} \{len(f_q^i), f_q^i \in L_i\}} \quad (9)$$

Furthermore, we define *the completeness of traffic flows* C_F as the actual received packets divided by the expected

[4]The 5-tuple session without multipath routing phenomenon refers to all its forward and backward packets traversing the same link. Therefore, if a flow appears on at least two equivalent switch nodes, multipath routing has occurred.

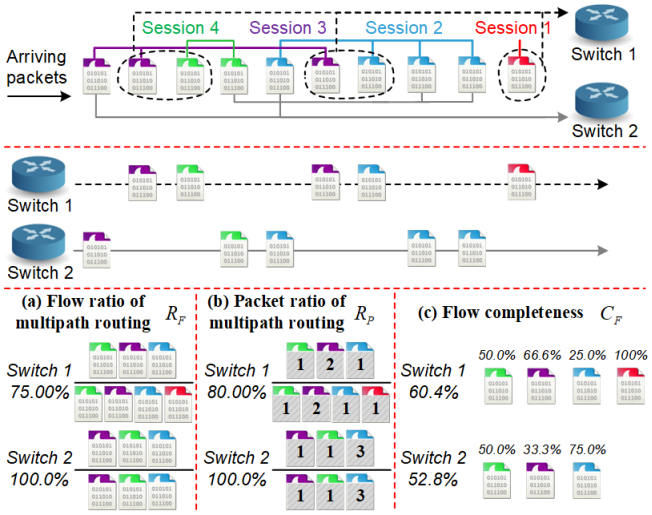


Fig. 7. Intuitive explanation of the multipath routing probabilities and the completeness of flow calculations.

received packets. Therefore, the average C_F^i of switch S_i can be calculated as

$$C_F^i = \frac{\sum_{q=1}^{len(L_i)} \{len(f_q^i)/(\sum_{j=1}^{n} \{len(f_q^j)|\forall j, I_s(f_q^i, f_q^j)\})\}}{len(L_i)} \tag{10}$$

where $I_s(f_q^i, f_q^j)$ refers to the flows possess same 5-tuple index, *i.e.,* $f_q^i.index == f_q^j.index$.

We provide intuitive illustrations to explain the computation process of multipath routing probabilities and flow completeness in Figure 7. The architecture is better to read bottom-up. Consider arriving traffic involves 4 different sessions (denoted as 4 colors, *i.e.,* red, blue, purple, and green), their packets will be forwarded to switch 1 or switch 2. The middle part of Figure 7 shows that there are 5 packets passing through switch 1 (including red, blue, purple, and green packets), and 5 packets passing through switch 2 (including blue, green, and purple packets). The bottom of the figure shows the calculation process of the three metrics. (i) According to Eq. (8), R_F calculates the ratio of flows that multipath routing occurs. The R_F in switch 1 is 75.00% (given the red session only appears in switch 1, there is no multipath routing for the red session), and in switch 2 is 100.0% (given all blue, green, and purple sessions with the multipath routing phenomenon). (ii) According to Eq. (9), R_P calculates the ratio of packets that multipath routing occurs. The R_P in switch 1 is 80.00% (given the red packet do not have multipath routing, accounting for 20.00%) and in switch 2 is 100.0% (all packets are in the multipath routing session). (iii) According to Eq. (10), C_F calculates the completeness of traffic flows. The C_F (the mean completeness of all sessions) in switch 1 is 60.4%, *i.e.,* $(50.0\% + 66.6\% + 25.0\% + 100\%) \div 4$ and in switch 2 is 52.8%, *i.e.,* $(50.0\% + 33.3\% + 75.0\%) \div 3$.

In general, the lower the flow ratio R_F, the more sessions there are without multipath routing (such as the red session in Figure 7), which will also lead to a lower packet ratio R_P, because the sessions without multipath increase the denominator in the calculation formula Eq. (9). Meanwhile, the calculation result for average flow completeness of all sessions C_F will be increased, since the flow completeness of a single session without multipath routing is 100%. The experimental results on the public dataset also indicate the relationship between these metrics, as shown in Table III. Note that the above correspondence for three metrics is not immutable, due to the calculation of multipath routing probability and flow completeness also depending on the length of each session (*i.e.,* the number of packets included). For example, in the calculation for the flow ratio R_F and the packet ratio R_P of multipath routing, the numerator and denominator correspond to flow count and packet count, respectively. Therefore, the ratio for the packet number from the session with/without multipath routing will affect the calculation results and impact the increase/decrease relationship of metrics.

TABLE III
EVALUATE MULTIPATH ROUTING RATIOS OF THREE TOPOLOGIES. THE LOWER THE MULTIPATH RATIO (R_P AND R_F), THE HIGHER FLOW COMPLETENESS (C_F). BOLD REFERS TO THE HIGH MULTIPATH RATIO.

Scenario	Node	IDS17/18 R_P ▼	IDS17/18 R_F ▼	IDS17/18 C_F ▲	USTC-TFC R_P ▼	USTC-TFC R_F ▼	USTC-TFC C_F ▲
Topo1	Switch1	92.32%	51.22%	74.15%	**99.74%**	**98.43%**	**50.65%**
	Switch2	**93.21%**	**53.07%**	**73.72%**	99.63%	98.25%	51.01%
Topo2	Switch1	94.03%	67.40%	58.80%	99.80%	99.32%	26.49%
	Switch2	94.51%	70.75%	56.70%	**99.96%**	**99.88%**	**25.36%**
	Switch3	**95.65%**	73.13%	**47.19%**	99.84%	99.56%	26.18%
	Switch4	95.46%	**73.56%**	55.22%	99.79%	99.38%	26.41%
Topo3	Switch1	94.75%	58.93%	70.44%	99.68%	**98.49%**	51.07%
	Switch2	95.13%	61.20%	69.50%	**99.70%**	**98.49%**	**50.43%**
	Switch3	95.11%	61.90%	69.22%	98.30%	96.33%	51.78%
	Switch4	**95.17%**	**63.33%**	**68.16%**	98.52%	96.81%	51.65%

C. Flow-Level Model Evaluation (RQ1)

To test the flow-based model, we first calculate the multipath routing probabilities under various topological scenarios and then analyze the impacts on the flow-based traffic analysis baselines.

1) Multipath Routing Ratio Evaluation: First, we evaluate the ratio of multipath routing based on four topologies in Figure 6. We calculate the average completeness of traffic flows C_F, flow-level multipath ratio R_F, and the packet-level one R_P for each switch node. The results are summarized in Table III, the higher R_F and R_R mean greater multipath probability, meanwhile, the flow completeness C_F will be low. For packet-level ratio, R_R results of all switch nodes are over 90% for two datasets. For flow-level ratio, it is clear that R_F results of the IDS17/18 dataset are generally lower than that of the USTC dataset. This could be attributed to exist some short flows (the flow contains few packets) in IDS17/18, or the interval between intra-flow packets being long, resulting in not occurring multipath routing in some flows. About the completeness of flow, we observe that C_F is strongly correlated with R_F and the number of candidate switches (notated as n). Generally, the theoretical value of C_F is $1 - R_F + R_F/n$, where $1 - R_F$ corresponds to the flow without multipath routing, and R_F/n refers to flows with multipath routing equiprobability among n switches. For example, for the IDS17/18 dataset, R_F of two switches in

TABLE IV
THE ACCURACY AND F1 SCORE RESULTS (%) OF FLOW-LEVEL MODELS IN BASE AND THREE MULTI-PATH SCENARIOS.

IDS	Base		Topo1: Parallel-2				Topo2: Parallel-4								Topo3: Serial-Parallel-4							
			Switch1		Switch2		Switch1		Switch2		Switch3		Switch4		Switch1		Switch2		Switch3		Switch4	
Model	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Flowrest	99.01	99.02	91.83	91.87	92.01	91.73	82.50	82.10	81.48	81.96	80.99	80.51	80.59	80.57	84.40	84.67	83.61	84.20	80.95	81.36	80.93	80.87
NetBeacon	96.19	96.61	81.82	81.84	81.07	81.69	70.64	71.09	69.41	70.20	69.07	69.41	68.11	68.92	73.34	73.40	72.39	73.07	67.96	68.13	68.01	67.89
RIDS	99.02	99.03	90.87	90.96	90.40	90.41	78.81	79.06	78.09	78.89	77.44	77.51	76.70	77.63	81.36	82.44	81.03	81.54	77.22	78.28	76.98	77.36
FlowLens	99.13	99.35	72.26	77.50	71.26	76.25	46.12	52.55	42.73	50.23	41.19	49.15	40.45	48.75	52.69	58.50	50.69	57.34	39.68	47.58	38.22	47.43
Whisper	98.42	98.93	64.34	69.11	63.46	67.89	43.51	48.07	40.80	45.22	38.95	43.29	38.95	43.27	50.13	54.26	48.19	53.05	35.05	41.00	33.99	40.14

USTC	Base		Topo1: Parallel-2				Topo2: Parallel-4								Topo3: Serial-Parallel-4							
			Switch1		Switch2		Switch1		Switch2		Switch3		Switch4		Switch1		Switch2		Switch3		Switch4	
Model	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Flowrest	99.13	98.92	88.79	86.92	88.73	86.68	75.10	75.01	75.62	74.90	75.12	75.13	75.23	75.19	75.33	75.30	75.82	74.87	75.69	75.12	75.77	75.02
NetBeacon	98.58	98.53	81.96	81.73	81.41	82.09	66.49	66.05	65.88	65.44	66.60	65.65	65.94	66.48	66.63	66.26	66.48	66.17	66.34	66.22	66.35	65.92
RIDS	99.18	98.99	85.49	83.84	86.44	83.63	73.28	71.46	73.35	72.21	72.90	72.26	73.72	72.00	74.12	72.38	73.64	72.11	74.10	72.63	73.33	71.92
FlowLens	99.25	99.22	62.83	65.25	63.07	65.88	47.39	51.78	46.93	51.75	47.56	51.06	47.64	51.79	47.78	52.05	47.59	52.19	47.99	51.91	47.16	51.63
Whisper	94.85	93.95	77.95	74.31	77.59	74.17	60.51	53.29	60.56	53.12	60.58	52.41	60.04	53.29	60.27	53.60	60.56	53.19	60.87	53.31	61.03	53.79

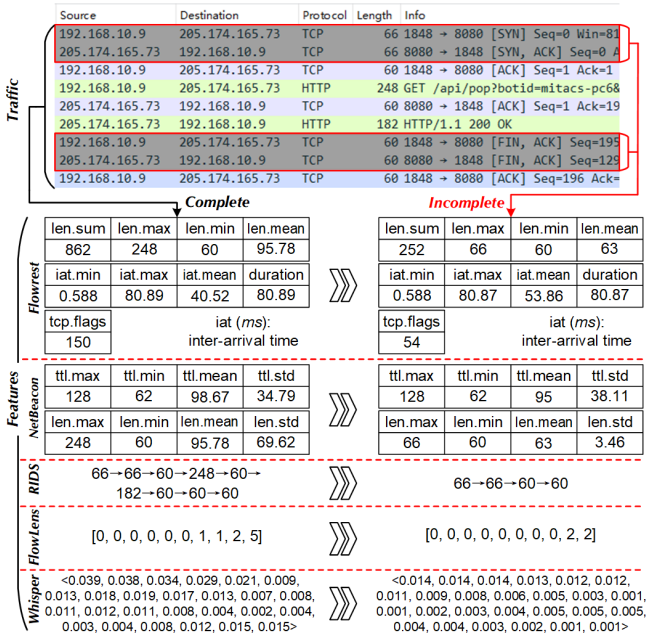


Fig. 8. A case study from the flow-based feature perspective.

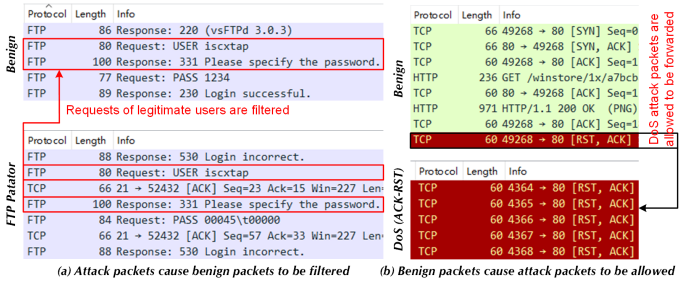


Fig. 9. Case study for packet-level models.

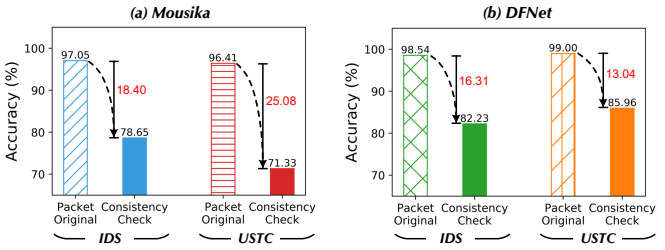


Fig. 10. Evaluation results of packet-level models.

Topo1 are ∼50%, and their C_F results are ∼75% (*i.e.,*, 1-50%+50%/2). Also, for the USTC dataset, R_F of four switches in *Topo2* are ∼100%, and their C_F results are ∼25% (*i.e.,*, 1-100%+100%/4). As expected, the greater the number of candidate switches, the greater the multipath probability. From the perspective across different topologies, the greater the number of parallel switches, R_F will increase significantly, and C_F will drop significantly (corresponding to *Topo1* and *Topo2*). Expansion of serial links almost does not affect C_F. Overall, at the packet level, the multipath routing ratio R_P generally tends to be 100% in *Topo1*-*Topo3*. Flow-level ratios vary significantly between different datasets, mainly due to the presence of short flows and intra-flow packet distribution being diverse. Furthermore, the flow completeness is strongly correlated flow-level ratio and the number of candidate switches.

2) Impact on Model Performance: As stated in Table I, the flow-level models include Flowrest, NetBeacon, RIDS, FlowLens, and Whisper. We evaluate these models in various switch nodes in different topology scenarios and without multipath routing (denoted as "Base" in Tabel IV), to explore the influence on their classification performance by multipath routing. From Tabel IV, we find that FlowLens presents the best accuracy and F1 score without multipath routing, since FlowLens only extracts features on the data plane while performing lossless model inference in the control plane (which results in only ∼1Gbps bandwidth could be saturated in Tabel I). NetBeacon and Whisper perform worst on IDS and USTC datasets, respectively.

When multipath routing occurs, the overall robustness reference Flowrest > RIDS > NetBeacon > FlowLens ≈ Whisper (FlowLens is the worst in USTC and Whisper is the worst in IDS). We could get some deep insights from their feature calculation process. Figure 8 displays the generated feature based on the complete (black line on the left) and incomplete (caused by multipath routing, red line on the right) traffic. Some insights are summarized below. (i) Statistics-based features tend to be more robust because some feature values may not change even if the traffic is incomplete, *e.g.,* "len.min" in Flowrest and "ttl.min" in NetBeacon. (ii) Sequence-based features (for RIDS) will be affected to a

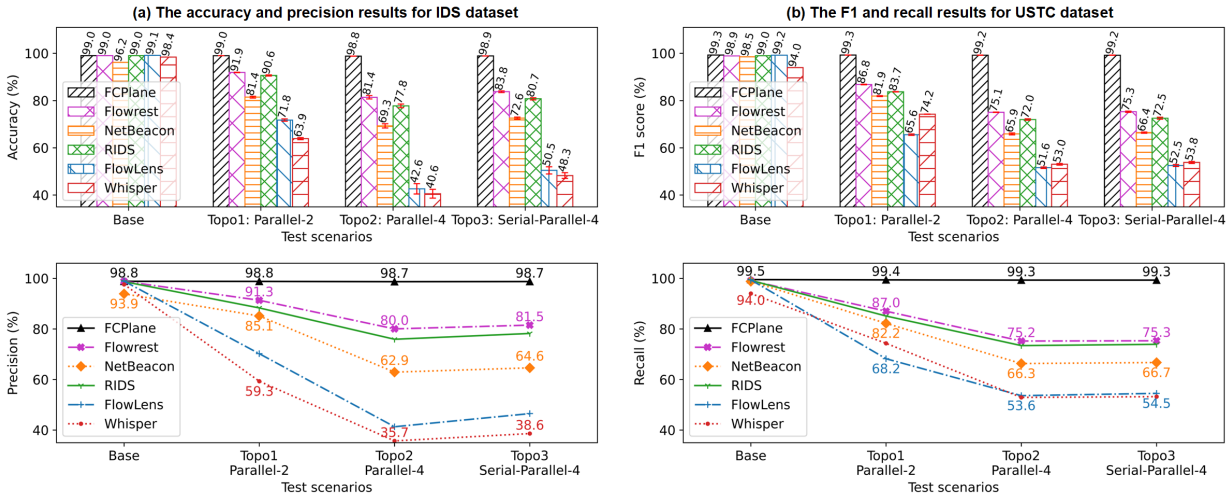


Fig. 11. The detection results (%, average by switches) of our proposed method and flow-based models in base and three multipath scenarios.

certain extent, manifested in that only some subsequences can be retained due to incomplete traffic. (iii) FlowLens uses histogram distribution as the feature, which is easily changed due to missing parts of the flow, resulting in distribution drift. (iv) Features based on the frequency domain (for Whisper) are also not robust enough because missing packets in the time domain will directly change the results of the Fourier transform. (v) Noteworthy, NetBeacon designs a multi-phase model inference, which applies different models when the number of intra-flow packets changes. Such a design will cause the model inference point [12] to shift when packets are missing, thereby introducing additional classification loss. In conclusion, for in-network traffic analysis, existing solutions have been significantly affected (decrease 20%∼50%).

D. Packet-Level Model Evaluation (RQ2)

Since packet-level models do not need to compute features based on 5-tuple flows, they are indeed not directly affected by multipath routing. However, we would like to clarify that existing packet-level approaches have practical issues in the real world. The main problem is that a false positive or false negative at the packet level may affect the availability of the entire flow or the defense fails. We illustrate in detail based on two specific examples in Figure 9. In subfigure (a), we find that the attack process of the FTP patator involves the "USER iscxtap" request and the "Please specify the password" response. However, these two packets also appear in benign interactions, filtering these two packets will cause legitimate user requests to be blocked all the time. In subfigure (b), a benign instance includes a TCP packet that carries [RST, ACK] flags. However, this [RST, ACK] packet can be used to launch the ACK-RST DoS attack [58]. This indicates some attack traffic will be allowed causing defenses to fail.

As the above practical considerations, for the packet-level model, we calculate the classification performance originally obtained according to each packet and also calculate accuracy that guarantees intra-flow consistency (the correct packets reference if all intra-flow packets are correctly classified). The results are summarized in Figure 10, it is clear that the accuracy of packet-level models drops significantly (up to 25.08%) after the intra-flow consistency check. Specifically, Mousika presents 97.05%→78.65% in IDS and 96.41%→71.33% in USTC, DFNet presents 98.54%→82.23% in IDS and 99.00%→85.96% in USTC. Therefore, packet-level models have been overestimated in previous work, with a large gap from practical applications. In real-world scenarios, the packet-level analysis model has two potential risks. On the one hand, a false positive report for a packet of the legitimate flow, affects the availability of the entire session. On the other hand, some packets of the attack session are missed (*i.e.,* the false negative), resulting in continuous defense failure.

E. Flowlet-Level Model Evaluation (RQ3)

We evaluate flowlet-based FCPlane and compare it with flow-based and packet-based models.

1) Compared with Flow-Level Models: We report the detection results for the IDS and USTC datasets, as shown in Figure 11. It is clear that the performance of FCPlane is basically unaffected on the base and three multi-path scenarios. For example, compared with "Topo2" and "Base" for IDS dataset, our accuracy is only reduced by 0.2%, while the five flow-based baselines drop by 17.6%∼57.8%. The results for the USTC dataset are similar. Particularly, the precision results of NetBeacon, FlowLens, and Whisper drop rapidly in the IDS dataset as the probability of multipath routing increases. This may be attributed to the fact that multipath routing causes their features to drift, which results in a large number of false positives in legitimate traffic, leading to lower precision scores. Overall, the forwarding and computing integrated dataplane design in FCPlane ensures that flowlets' packets can be assigned to the same switch node, thereby supporting the flowlet-level Markov chain model to accurately identify traffic. As for the slight variation in performance under different topologies, it could be due to different proportions of packet loss, retransmission, and out-of-order (as discussed in § VI-B).

TABLE V
THE PERFORMANCE RESULTS (%) OF OUR PROPOSED METHOD AND PACKET-LEVEL MODELS AFTER CONSISTENCY CHECK.

IDS	Original packet-level				Consistency check			
Method	*ACC*	*Pre*	*Rec*	*F1*	*ACC*	*Pre*	*Rec*	*F1*
Mousika	97.05	95.72	98.50	97.09	78.65	75.33	85.20	79.96
DFNet	98.54	97.97	99.15	98.56	82.23	79.28	87.25	83.08
FCPlane	**99.02**	**98.85**	**99.25**	**99.05**	**98.89**	**98.61**	**99.15**	**98.88**

USTC	Original packet-level				Consistency check			
Method	*ACC*	*Pre*	*Rec*	*F1*	*ACC*	*Pre*	*Rec*	*F1*
Mousika	96.41	95.61	97.27	96.43	71.33	69.05	77.33	72.96
DFNet	99.00	98.80	99.20	99.00	85.96	83.07	90.27	86.52
FCPlane	**99.26**	**99.07**	**99.47**	**99.27**	**99.12**	**99.07**	**99.20**	**99.13**

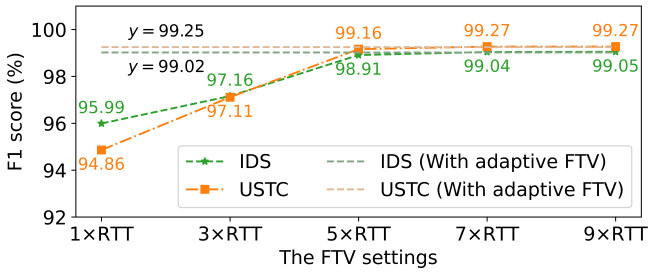


Fig. 12. The ablation experiments with different FTV settings.

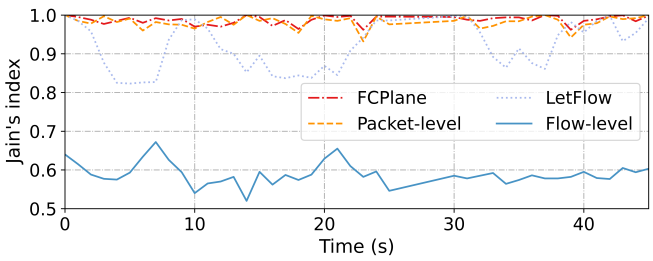


Fig. 13. The Jain's fairness index of link throughput.

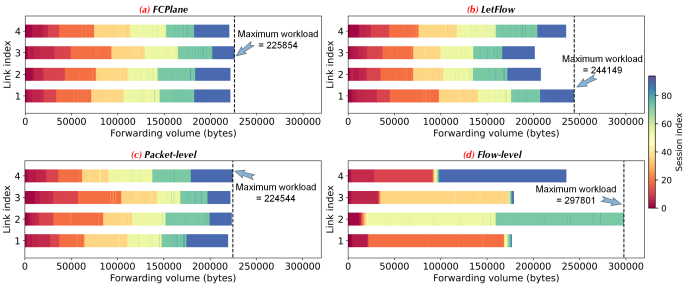


Fig. 14. Forwarding process visualization for various schemes.

2) Compared with Packet-Level Models: Then, we compare with the packet-level baseline model, the original results and those after consistency check are organized in Table V. In contrast to Mousika and DFNet, our performance degrades only slightly after the consistency check. For example, our F1 results are reduced by 0.17% and 0.14% on the IDS and USTC datasets respectively. And the precision results decrease by less than 0.25% on two datasets, which indicates that flowlet-based FCPlane does not introduce many false positives, thus supporting the availability of legitimate traffic. In general, flowlets in FCPlane have enough packet sequence information for the model to produce reliable results, compared to packet-level models that only use individual packets.

3) Ablation Experiment on FTV Setting: Next, we develop ablation experiments by changing the FTV setting. Consistent with previous work [51], we set FTV to $\{1 \times RTT, 3 \times RTT, 5 \times RTT, 7 \times RTT, 9 \times RTT\}$ respectively, where RTT refers to the Round-Trip Time. As shown in Figure 12, when FTV is set to $1 \times RTT$ and $3 \times RTT$, the performance of the Markov chain model suffers some loss. As FTV increases, the model performance gradually improves and stabilizes. This can be explained that when FTV is too small, the packet sequence contained in the flowlet becomes shorter, and thus cannot provide enough information for Markov chain model classification. Nevertheless, even with FTV set to $1 \times RTT$, our scheme still achieves 95.99% and 94.86% for IDS and USTC datasets. Furthermore, we plot the results of adaptive size adaptation (default configuration in FCPlane) as horizontal lines in Figure 12. It is clear that F1 results (for two datasets) with adaptive FTV are very close to that of $7 \times RTT$ (the difference is less than 0.03%). Particularly, adaptive FTV is a dynamic strategy that can be adjusted according to the throughput status to make flowlets more evenly distributed on multiple links. The specific cases of the forwarding process are visualized in § V-F.

F. Traffic Distribution Effect and Throughput Evaluation for Forwarding-Computing Integrated Dataplane (RQ4)

In addition, we also evaluate the load-balancing effect of the proposed data plane compared with flow-based, packet-based, and LetFlow [59] (a typical flowlet-based scheme). To carry out the experiments, we use the "Topo2: Parallel-4" in Figure 6, which involves a multipath routing consisting of 4 switch nodes in series. All approaches use the ECMP algorithm for fair comparison. In order to evaluate the load balancing of multiple candidate nodes, we leverage Jain's fairness index [51] as the metric, which is a widely used metric for quantifying the evenness of resource allocation, such as bandwidth, among a population of users or sessions in a network. Jain's fairness index F is defined as follows:

$$F = \frac{(\sum_{i=1}^{n} x_i)^2}{n \cdot \sum_{i=1}^{n} x_i^2} \tag{11}$$

where x_i represents the volume (*e.g.,* traffic bytes) allocated to the i-th switch, and n is the total number of switches. As shown in Figure 13, flow-level scheme performs the worst, with a Jain's fairness index of about 0.6, followed by LetFlow given its fixed FTV setting for all flowlets. Our approach FCPlane (with default dynamic FTV adaptation) is comparable to the packet-level one, which is expected.

Moreover, we select 100 consecutive sessions from Figure 13 to plot the traffic distribution process. The results are shown in Figure 14. Different colors correspond to different session indexes. It is clear that the flow-level approach presents the most uneven workloads on four links in subfigure (d). This can be attributed to there being some elephant flows that

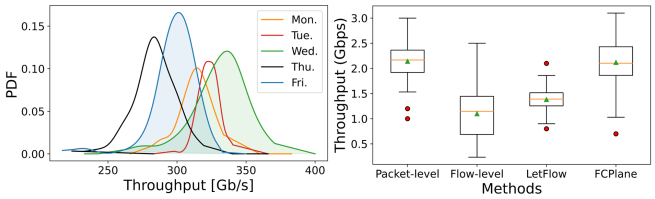


Fig. 15. Aggregated bandwidth and bandwidth distribution of flows.

include a large number of packets with heavy payloads, which is also the reason why the flow-level scheme performs the worst Jain's index result in Figure 13. The forwarding volume of the LetFlow on the four links is also somewhat uneven due to its fixed FTV setting for all flowlets, as shown in Figure 14 (b). Figures 14 (a) and (c) display that FCPlane achieves comparable results to the packet-level scheme, and their traffic distributions are relatively uniform. Overall, FCPlane could realize uniform load balancing by leveraging dynamic flowlet adaptation and maintain excellent in-network traffic analysis performance even against multipath routing.

We also replay the five-day traffic of IDS2017 to measure the throughput of the testbed, as shown in Figure 15. The proposed design realizes the real-time traffic detection and can saturate the dataplane, *i.e.,* 400 Gbps. The pipelined nature of the switch hardware ensures the achievement of full bandwidth, that is, line-speed model inference and routing. As for the bandwidth distribution of flows in the right subfigure, the overall results refer to FCPlane (ours) $\approx$ Packet-level $>$ LetFlow $>$ Flow-level. Regarding hardware resources, our forwarding-computing integrated dataplane across 9 stages, and requires approximately 12.82% SRAM and 4.29% TCAM, which is acceptable.

VI. Discussion and Limitations

A. Dataset Quality

Given that the Markov chain model construction in the computing module of FCPlane is a data-driven process, it is related to the dataset quality. We discuss here some potential limitations when faced with low-quality datasets. Constructing a network intrusion traffic dataset that reflects the real world is not an easy task. Although IDS2017/2018 datasets are widely used to evaluate the effectiveness of network intrusion detection, previous work has revealed some of its flaws [60]–[62]. We analyze the impact of these issues on FCPlane. (i) *Flow construction.* In IDS2017/2018 datasets, the CSV files are constructed and generated by the CICFlowMeter tool [63]. Previous work has shown that there are errors in the flow construction process of the CICFlowMeter tool, such as incorrect parsing of FIN and RST packets [60], [62]. In our experiments, we do not use the CSV files of the dataset, nor do we use the CICFlowMeter tool. Instead, we use the SplitCap tool [64] to perform the 5-tuple splitting on the original PCAP traffic. (ii) *Feature extraction.* Previous work also showed that the features extracted in dataset CSV files have the problem of shortcut learning, *e.g.,* flow ID, source IP, source port, and timestamp, these attributes could lead to the IDS learning spurious correlations [65]. Consistent with their suggestion, the feature extraction in FCPlane mainly considers the packet length and TCP flag fields, without using the time-specific or host-specific information. (iii) *Labeling issues.* The dataset labels of IDS2017/2018 mainly reference the attacker/victim IP and the time window information of the attack execution. This indeed lacks the verification of the flow content and attack effect. Nevertheless, our detection method does not make any assumptions about IP addresses or time windows, and FCPlane can be combined with previous work on dataset label correction [62] (*e.g.,* using corrected/reassigned labels for training and testing).

B. Topology Scenarios

We also discuss the impact of different topology scenarios on the performance of FCPlane. On the one hand, we explain that if the dataset is collected at the sender side or receiver side, the topology of the selected dataset collection environment will hardly affect the FCPlane, even if the topology of the test environment is different from the dataset collection topology (there may be a little impact from network-induced phenomena, which will be introduced subsequently). For example, the dataset collection topology in IDS2017/2018 does not highlight multipath routing issues. It captures network traffic on each machine, meaning that each machine can observe the complete relevant session (refers to the multipath routing phenomenon that does not affect the observation of the end side, while it does not refer to there being no packet loss). Therefore, the distance between the network topology described in the chosen datasets and the proposed test topology scenarios does not impact the effect of FCPlane. On the other hand, different topologies will affect the probability of network-induced phenomena, such as packet loss, retransmission, and out-of-order in PCAP traffic captured at the receiver side [66]. These problem packets may further lead to model misclassification. To cope with this problem, FCPlane can leverage and combine existing robust model construction strategies [6].

C. Adaptive Mechanism of FTV

In § V-E and § V-F, we demonstrate the effectiveness of dynamic size adaptation in FCPlane, and we further discuss the adaptation mechanism of FTV here. On the one hand, determining an appropriate FTV value is non-trivial in practice. One possible solution is to consider more information to guide the adaptation of FVT, such as the traffic queue of ingress ports and the response delay of the egress port link. The acquisition and organization of this information can be combined with network measurement solutions. For instance, we could send probe packets and then recirculate them through each egress port to obtain the corresponding queue length [67]. On the other hand, as the number of flows increases, maintaining FTV for all flows may lead to a significant increase in resource overhead. A feasible solution is to deliver the information (such as the sent packet) to the end-side host and mark this information in the packet headers, as so saving dataplane table resources. Moreover, the fading mechanism in FTV dynamic

adaptation can be achieved by basic primitive operations (based on P4) within one clock cycle [51]. Therefore, the dynamic size adaptation in FCPlane would not become a main bottleneck under real-world scenarios.

D. Other Challenges in Zero-Trust IDS

As stated in § II, according to the tenets of zero-trust security and the derived zero-trust views and requirements in network IDS, multipath routing is indeed the practical challenge for deploying in-network analysis models to verify all traffic. In addition to multipath routing, we expand here to discuss other potential challenges for deploying zero-trust IDS. (i) *Distributed identity authentication and management.* Dynamic identity control requires rich data to support it [17]. Under the zero-trust architecture [27], the implementation of the identity management system faces challenges in combining data from different network nodes. For example, it requires uniformly managing and analyzing identity data scattered across various systems, and it is difficult to connect identity data between application systems. (ii) *Comprehensive analysis of multi-source data.* Zero-trust IDS needs to process data streams from different sources, including while not limited to network traffic, system logs, and user behavior data [27]. The complexity of these data streams implies that comprehensive analysis methods regarding multi-source data need to be explored in the future. (iii) *Continuous verification and maintenance.* The zero-trust model relies on a large network with strictly defined permissions. As a company grows, the turnover of employees is an inevitable problem [18]. In addition, issues such as network topology changes and link failures also need to be considered.

VII. Implication for Future Works

In this section, we discuss the challenge of zero-trust in-network traffic analysis model against multipath routing and summarize implications from three aspects for research communities, as shown in Figure 16.

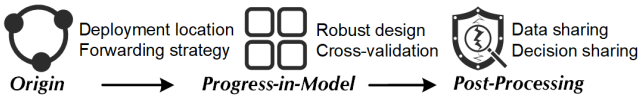


Fig. 16. The implications from three aspects.

A. Origin: Deployment Location and Forwarding Strategy

First, stemming from the source of the multipath routing phenomenon, selecting deployment nodes and configuring different load-balancing strategies will directly affect the multipath routing ratio and the completeness of traffic. *According to the results of § V-C, selecting a location with fewer parallel candidate switch nodes for deployment will theoretically obtain a lower multipath routing ratio and higher traffic completeness.* However, it needs to be considered that zero-trust security assumes that threats may originate from anywhere, including within the network. This requires that the deployed switch nodes should cover all possible attack sources. If resources permit, deploying the traffic analysis model directly at the gateway of the autonomous system will hopefully capture complete traffic, this could mean a *paradigm shift* from existing Cloud Security Service Providers [4] (CSSPs, *e.g.,* Cloudflare, Arbor, Akamai). Moreover, previous work [23] proposes that multipath provides greater stability than single-path routing (this is considered from the perspective of the single-point failure and denial of service attacks), *while our results show that the multipath routing phenomenon instead provides attackers with opportunities to bypass in-network detection.* Regarding the load-balancing algorithm, prior art [23] mentions that per-packet hashing is more uniform than flow-based hashing (thus improving overall link bandwidth), while per-packet hashing indeed exacerbates the multipath routing phenomenon, so users need to consider practical requirements.

B. Progress-in-Model: Robust Design and Validation

Another important consideration is the model design itself, which is the decision-maker for in-network traffic analysis. Admittedly, we cannot predict in advance which parts of the traffic will be missed due to multipath routing. This is like an unknown degree of packet loss that may occur in the network at any time, perhaps considering packet loss during model training is an option [6]. Nevertheless, this does not always work because the missed packet ratio can be high (depending on the number of candidate nodes in multipath routing). *According to the experimental results of flow-based baselines, the statistical features are relatively more robust*, which can be explained by the fact that statistical methods can dilute/mitigate the impact of partial packet loss on characterized outcomes. Likewise, we can also consider validation schemes such as grouping flows based on attack source and performing inter-flow cross-validation (*i.e.,* mitigating the impact of partial flow errors with attack source-level statistics). In this way, we can have more examples to judge whether there is an attack behavior (such as denial of service attacks). However, this may cause the problem of slow reaction for disconnection (since enough examples are needed to verify), so exist a trade-off between detection accuracy and response speed.

C. Post-Processing: Cooperation of Multiple Switches

Finally, maybe one of the most promising and complex solutions is cooperation between multiple switches. Essentially, in-network traffic analysis models are affected by multipath routing because only localized/partial traffic data can be captured. Therefore, if multiple switches can cooperate with each other, the impact of multipath on the model will be fundamentally solved, while this is non-trivial. We discuss some possible solutions. For one thing, traffic data across multiple switches can be shared, which improves traffic completeness and supports accurate predictions. For another, inference results output by different switch nodes can be shared, this is a similar purpose to the cross-validation. Nonetheless, whether sharing data or decision results, additional bandwidth is required for transmission, and this cost and expense need to be considered.

VIII. Conclusion

In this paper, we introduce the practical challenge and deployment observations about zero-trust in-network traffic analysis during the cooperation of the ISP. Then, we explore the probabilities of multipath routing and reveal its impact on in-network ML-based traffic analysis models. Through evaluation for 7 SOTA baselines with two public datasets, the results show that multipath routing is common and the impact on 5 flow-based models cannot be ignored. In addition, 2 packet-based methods present the problem of inconsistent results within the session. To this end, we propose a new forwarding and computing integrated dataplane and design two tightly coupled modules. With carefully designed flowlet extraction, routing strategies, and the tailor-made flowlet-level Markov chain model, we realize efficient load balancing while providing reliable traffic analysis results, even against multipath routing. Finally, we summarize implications and future directions in terms of problem origin, model design, and post-processing. Our research could advance the practice of zero-trust network IDS that robustly verify all traffic.

References

[1] Xinjie Lin et al. ET-BERT: A contextualized datagram representation with pre-training transformers for encrypted traffic classification. In *WWW*. ACM, 2022.

[2] Hongda Li et al. vnids: Towards elastic security with safe and efficient virtualization of network intrusion detection systems. In *CCS*. ACM, 2018.

[3] Karel Bartos et al. Optimized invariant representation of network traffic for detecting unseen malware variants. In *USENIX Security*, 2016.

[4] Ziming Zhao et al. Effective DDoS Mitigation via ML-Driven In-network Traffic Shaping. *IEEE TDSC*, 2024.

[5] Chuanpu Fu et al. Detecting unknown encrypted malicious traffic in real time via flow interaction graph analysis. In *NDSS*, 2023.

[6] Ziming Zhao et al. Ernn: Error-resilient rnn for encrypted traffic detection towards network-induced phenomena. *IEEE TDSC*, 2023.

[7] Chang Liu et al. Fs-net: A flow sequence network for encrypted traffic classification. In *INFOCOM*. IEEE, 2019.

[8] Yisroel Mirsky et al. Kitsune: An ensemble of autoencoders for online network intrusion detection. In *NDSS*, 2018.

[9] Menghao Zhang et al. Poseidon: Mitigating volumetric ddos attacks with programmable switches. In *NDSS*. The Internet Society, 2020.

[10] Zaoxing Liu et al. Jaqen: A high-performance switch-native approach for detecting and mitigating volumetric ddos attacks with programmable switches. In *USENIX Security*, 2021.

[11] Ziming Zhao et al. RIDS: towards advanced IDS via RNN model and programmable switches co-designed approaches. In *INFOCOM*. IEEE, 2024.

[12] Guangmeng Zhou et al. An efficient design of intelligent network data plane. In *USENIX Security*, 2023.

[13] Cornelius Itodo et al. Multivocal literature review on zero-trust security implementation. *Computers & Security*, 2024.

[14] Levente Csikor et al. Zerodns: Towards better zero trust security using dns. In *ACSAC*, 2022.

[15] Sungmin Hong et al. Sysflow: Toward a programmable zero trust framework for system security. *IEEE TIFS*, 2023.

[16] Christoph Buck et al. Never trust, always verify: A multivocal literature review on current knowledge and research gaps of zero-trust. *Computers & Security*, 2021.

[17] R Freter. Department of defense (dod) zero trust reference architecture., 2023. https://dodcio.defense.gov/Portals/0/Documents/Library/(U)ZT_RA_v2.0(U)_Sep22.pdf.

[18] USA Department of Defense. Dod zero trust strategy., 2022. https://dodcio.defense.gov/Portals/0/Documents/Library/DoD-ZTStrategy.pdf.

[19] Aristide Tanyi-Jong Akem et al. Flowrest: Practical flow-level inference in programmable switches with random forests. In *INFOCOM*. IEEE, 2023.

[20] Diogo Barradas et al. Flowlens: Enabling efficient flow classification for ml-based network security applications. In *NDSS*, 2021.

[21] Guorui Xie et al. Mousika: Enable General In-Network Intelligence in Programmable Switches by Knowledge Distillation. In *INFOCOM*. IEEE, 2022.

[22] Songpon Teerakanok et al. Migrating to zero trust architecture: Reviews and challenges. *Security and Communication Networks*, 2021.

[23] Sandeep Kumar Singh et al. A survey on internet multipath routing and provisioning. *IEEE Commun. Surv. Tutorials*, 2015.

[24] Israel Cidon et al. Analysis of multi-path routing. *IEEE/ACM Trans. Netw.*, 1999.

[25] Brice Augustin et al. Measuring multipath routing in the internet. *IEEE/ACM Trans. Netw.*, 2011.

[26] Yashar Ganjali and Abtin Keshavarzian. Load balancing in ad hoc networks: Single-path routing vs. multi-path routing. In *INFOCOM*, pages 1120–1125. IEEE, 2004.

[27] V Stafford. Zero trust architecture. *NIST special publication*, 2020.

[28] Hyok An et al. Resilience evaluation of multi-path routing against network attacks and failures. *Electronics*, 2021.

[29] Aristotelis Tsirigos et al. Multipath routing in the presence of frequent topological changes. *IEEE Commun. Mag.*, 2001.

[30] Aristotelis Tsirigos et al. Analysis of multipath routing, part 2: mitigation of the effects of frequently changing network topologies. *IEEE TWC*, 2004.

[31] Marcelo Pizzutti et al. Adaptive multipath routing based on hybrid data and control plane operation. In *INFOCOM*. IEEE, 2019.

[32] Wen Xu et al. MIRO: multi-path interdomain routing. In *SIGCOMM*. ACM, 2006.

[33] Chuanpu Fu et al. Realtime robust malicious traffic detection via frequency domain analysis. In *CCS*. ACM, 2021.

[34] Muhammad Ajmal Azad et al. Verify and trust: A multidimensional survey of zero-trust security in the age of iot. *Internet of Things*, 2024.

[35] Syed W Shah et al. Lcda: Lightweight continuous device-to-device authentication for a zero trust architecture (zta). *Computers & Security*, 2021.

[36] Coralie Busse-Grawitz et al. pforest: In-network inference with random forests. *CoRR*, 2019.

[37] Jong-Hyouk Lee et al. Switchtree: in-network computing and traffic analyses with random forests. *Neural Computing and Applications*, 2020.

[38] Bruno Missi Xavier et al. Programmable switches for in-networking classification. In *INFOCOM*. IEEE, 2021.

[39] Bruno Loureiro Coelho and Alberto Schaeffer-Filho. BACKORDERS: using random forests to detect ddos attacks in programmable data planes. In *EuroP4@CoNEXT*. ACM, 2022.

[40] The P4 Language Consortium. P4 bmv2 behavioral model. [EB/OL], 2020. https://github.com/p4lang/behavioral-model.

[41] Changgang Zheng and Noa Zilberman. Planter: seeding trees within switches. In *SIGCOMM Posters and Demos*, pages 12–14. ACM, 2021.

[42] Zhaoqi Xiong et al. Do switches dream of machine learning?: Toward in-network classification. In *HotNets*. ACM, 2019.

[43] Jinzhu Yan et al. Brain-on-switch: Towards advanced intelligent network data plane via nn-driven traffic analysis at line-speed. In *NSDI*. USENIX, 2024.

[44] D. Project. Dpdk: Data plane development kit. [EB/OL], 2010. http://dpdk.org/ Accessed November 27, 2020.

[45] Sungoh Kwon et al. Analysis of shortest path routing for large multi-hop wireless networks. *IEEE/ACM Trans. Netw.*, 2009.

[46] Jenn-Yue Teo et al. Interference-minimized multipath routing with congestion control in wireless sensor network for high-rate streaming. *IEEE Trans. Mob. Comput.*, 2008.

[47] Erico Vanini et al. Let it flow: Resilient asymmetric load balancing with flowlet switching. In *NSDI*. USENIX, 2017.

[48] Sushant Jain et al. B4: experience with a globally-deployed software defined wan. In *SIGCOMM*. ACM, 2013.

[49] Mohammad Al-Fares et al. Hedera: Dynamic flow scheduling for data center networks. In *NSDI*. USENIX, 2010.

[50] Haythem Bany Salameh et al. Cooperative adaptive spectrum sharing in cognitive radio networks. *IEEE/ACM Trans. Netw.*, 2010.

[51] Sen Liu et al. Halflife: An adaptive flowlet-based load balancer with fading timeout in data center networks. In *EuroSys*. ACM, 2024.

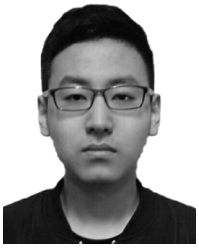
[52] Ziming Zhao et al. Ddos family: A novel perspective for massive types of ddos attacks. *Comput. Secur.*, 2024.

[53] The P4 Language Consortium. P4 tutorial. [EB/OL], 2020. https://github.com/p4lang/tutorials.

[54] Paul Emmerich et al. MoonGen: A Scriptable High-Speed Packet Generator. In *IMC*.

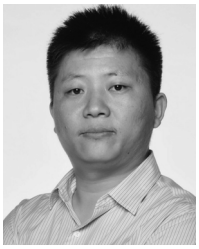
[55] Leonardo Mendonça de Moura et al. Z3: an efficient SMT solver. In *TACAS*. Springer, 2008.

[56] Canadian Institute for Cybersecurity. Cicids., 2018. https://www.unb.ca/cic/datasets/.
[57] Wei Wang et al. Malware traffic classification using convolutional neural network for representation learning. In *ICOIN*. IEEE, 2017.
[58] MAZEBOLT. Mazebolt knowledge base. https://kb.mazebolt.com/, 2016.
[59] Erico Vanini et al. Let it flow: Resilient asymmetric load balancing with flowlet switching. In *NSDI*. USENIX, 2017.
[60] Gints Engelen et al. Troubleshooting an intrusion detection dataset: the cicids2017 case study. In *IEEE SPW*, 2021.
[61] Arnaud Rosay et al. From cic-ids2017 to lycos-ids2017: A corrected dataset for better performance. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 2021.
[62] Lisa Liu et al. Error prevalence in nids datasets: A case study on cic-ids-2017 and cse-cic-ids-2018. In *IEEE CNS*, 2022.
[63] Cicflowmeter tool. [EB/OL]. https://www.unb.ca/cic/research/applications.html.
[64] Splitcap. [EB/OL]. https://www.netresec.com/?page=SplitCap.
[65] Daniel Arp et al. Dos and don'ts of machine learning in computer security. In *USENIX Security*, 2022.
[66] Renjie Xie et al. Rosetta: Enabling robust TLS encrypted traffic classification in diverse network environments with tcp-aware traffic augmentation. In *USENIX Security*, 2023.
[67] Guanyu Li et al. IMap: Fast and scalable in-network scanning with programmable switches. In *USENIX NSDI*, 2022.



Ziming Zhao received his Ph.D. degree from Zhejiang University, Hangzhou, China. He has published more than 35 papers in international journals and conference proceedings, including ToN, TIFS, TDSC, TMC, TCAD, CCS, RTSS, MobiCom, SIGCOMM, INFOCOM, WWW, TSE, ESE, ICWS, DATE, COSE, SECON, DASFAA, COMNET, ACM MM, and AAAI. His research interests include machine learning, traffic identification, AI security, and network programmable dataplane.

Zhaoxuan Li received his Ph.D. degree from State Key Laboratory of Information Security (SKLOIS), Institute of Information Engineering (IIE), Chinese Academy of Sciences (CAS), Beijing, China. He has published more than 10 papers in international journals and conference proceedings, including TSE, TIFS, TDSC, TMC, COMNETS, COSE, ESE, INFOCOM, WWW, and ICWS. His research interests include traffic identification, blockchain security, formal methods, and privacy-preserving.



Xiaofei Xie received the B.E., M.E., and Ph.D. degrees from Tianjin University. He is currently an Assistant Professor with Singapore Management University, Singapore. His research mainly focuses on program analysis, traditional software testing, and quality assurance analysis of artificial intelligence. He was a recipient of the three ACM SIGSOFT Distinguished Paper Awards in FSE'16, ASE'19, ISSTA'22 and ASE'23.

Zhipeng Liu is an MS. student in Zhejiang University, Hangzhou, China. His research interests include machine learning, Cybersecurity, and traffic identification.

Tingting Li is a Ph.D. student in Zhejiang University, Hangzhou, China. She has published more than 10 papers in international journals and conference proceedings, including HPCA, MICRO, TIFS, CCS, WWW, DATE, ACM MM, QCNC, and ICWS. Her research interests include machine learning, traffic identification, AI security, and privacy preserving.

Jiongchi Yu Jiongchi Yu is a Ph.D. candidate in Computer Science at Singapore Management University, Singapore. His research is intelligent software testing and security in cloud systems. He has published more than 10 papers in elite conference proceedings and journals, including ICSE, ISSTA, TON, TMC, COSE and Langmuir.

Fan Zhang (Member, IEEE) received his Ph.D. degree from the Department of Computer Science and Engineering, University of Connecticut, CT, USA, in 2011. He is currently a Full Professor with the College of Computer Science and Technology, Zhejiang University, Hangzhou, China, and also with the Alibaba–Zhejiang University Joint Institute of Frontier Technologies, Hangzhou. His research interests include system security, hardware security, network security, cryptography, and computer architecture.

Binbin Chen (Member, IEEE) received the B.Sc. degree in computer science from Peking University and the Ph.D. degree in computer science from the National University of Singapore. Since July 2019, he has been an Associate Professor in the Information Systems Technology and Design (ISTD) pillar, Singapore University of Technology and Design (SUTD). He currently also holds a joint appointment as Principal Research Scientist at Advanced Digital Sciences Center, which is a University of Illinois research center located in Singapore. His current research interests include wireless networks, cyber-physical systems, and cyber security for critical infrastructures. He was a recipient of the Best Paper Awards in SIGCOMM'10.