IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 July 2024, accepted 5 September 2024, date of publication 9 September 2024, date of current version 19 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3456588

RESEARCH ARTICLE

Unveiling Latency-Induced Service Degradation: A Methodological Approach With Dataset

BALINT BICSKI[1], (Graduate Student Member, IEEE), AND ADRIAN PEKAR[1,2]
[1]Department of Networked Systems and Services, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, 1111 Budapest, Hungary
[2]HUN-REN-BME Information Systems Research Group, 1117 Budapest, Hungary

Corresponding author: Adrian Pekar (apekar@hit.bme.hu)

This work was supported in part by the János Bolyai Research Scholarship of Hungarian Academy of Sciences; in part by the New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund, under Grant ÚNKP-23-5-BME-461; in part by the Ministry of Innovation and Technology from the National Research, Development, and Innovartion (NRDI) Fund, financed through the FK_21 funding scheme under Grant 138903; and in part by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund, financed through the TKP2021-NVA funding scheme, under Grant TKP2021-NVA-02.

ABSTRACT This paper presents a comprehensive study on the identification and analysis of Service Degradation (SD) events within a university dormitory network, leveraging LAN data to develop a robust methodology applicable to diverse networking environments. Employing statistical techniques, such as Interquartile Range (IQR) and Z-score analyses, we detect significant deviations in network performance—specifically, extreme delays and jitter—that indicate potential SD. The methodology was rigorously validated in various settings, demonstrating minimal deviations in results and reinforcing the approach's consistency and reliability. Initial tests conducted in a university dormitory environment suggest the model's potential applicability in both residential and enterprise networks, thus broadening its utility. By refining the detection and understanding of SD indicators, this research contributes systematic methodological applications and a valuable annotated dataset to the field. This groundwork enables network administrators to enhance service quality preemptively, offering significant implications for future research and practical applications in network management.

INDEX TERMS Delay and jitter analysis, network performance, service degradation, quality of experience.

I. INTRODUCTION

In the era of relentless digitization, computer networks underpin our everyday digital interactions, enabling high-definition streaming, real-time gaming, and collaborative online services. As digital demands escalate, the expectation for networks to deliver uninterrupted, high-quality experiences grows concurrently. However, network service degradation (SD) [1] threatens these expectations by reducing network performance, thereby compromising the digital experience.

SD can arise from various sources such as network congestion, hardware failures, software glitches, external interference, and malicious attacks [2], [3], [4], [5], [6]. These factors contribute to a range of symptoms including increased latency, packet loss, reduced throughput, jitter, and frequent connection drops, all of which significantly impact user experience. This research specifically focuses on latency as a critical aspect of SD. The challenge lies in accurately identifying and analyzing the key characteristics of latency-related degradation within network environments.

Research efforts in detecting SD span diverse fields, including campus and residential networks [7], [8], microservice-based applications and time-sensitive networks [9], [10]. Recent studies have highlighted latency as a reliable indicator of service quality degradation, particularly in online gaming [11] and video streaming [12]. Some researchers even consider SD as a potential strategic resource management tactic [13], [14], [15], [16]. While these studies provide valuable insights, significant gaps

The associate editor coordinating the review of this manuscript and approving it for publication was Petros Nicopolitidis.

remain in accurately diagnosing SD in computer networks, especially in environments highly susceptible to degradation. Furthermore, to the best of our knowledge, there exists no publicly available dataset with SD annotations, hindering the development and validation of SD detection methods.

Building upon these foundational studies, our research employs Packet Inter-Arrival Time (PIAT) analysis in a university dormitory network setting—an environment particularly susceptible to SD—to focus on latency and jitter as a principal indicators of SD. We specifically target TCP flows and LAN segments, where PIAT, analyzed through Z-score and Interquartile Range (IQR) methods, provides critical insights into response latency without the confounding effects of end-device delays. Our methodology has proven robust when tested across various network environments, suggesting its applicability beyond the initial university dormitory setting. The identified SD events correlate strongly with potential impacts on the Quality of Experience (QoE), indicating areas for network improvement and further research. Crucially, our work also provides a dataset of real-world IP flows with labeled latency-induced SD annotations, addressing a significant gap in the field.

The main contributions of this paper are threefold:

1) We present a robust methodology for detecting SD events in network flows. By applying empirical heuristics across all application categories, we effectively identify and label these events, thereby enhancing the understanding of SD within networked environments. Corresponding digital artifacts are provided as supplementary materials to support the replicability of our analysis.
2) Our work distinguishes itself by utilizing PIAT analysis focused specifically on LAN-side service degradation. This approach isolates latency effects intrinsic to the local network segment, which are often overlooked in broader network studies, enabling more precise detection and understanding of SD causes and effects.
3) We provide a labeled dataset of network flows with identified SD events, serving as a valuable resource for further research into network service degradation. This dataset, alongside additional digital artifacts related to data analysis, offers researchers a rich base for developing and testing new hypotheses and methodologies.

This study not only enhances our understanding of SD but also offers valuable insights that can contribute to improving network performance and security. The findings are particularly beneficial for applications in constrained monitoring environments, such as ISP home routers, where IP flows are typically observed only up to a certain packet threshold before being transitioned to hardware acceleration. This process segments the flow and reduces visibility, which ultimately limits the timely observability and subsequent action taken to address critical network behaviors such as SD.

The rest of this paper is organized as follows: Section II discusses related work relevant in the context of our study. Section III provides a comprehensiveness overview of the dataset. Section IV introduces foundational principles relevant to our approach, including latency considerations, analysis of LAN and WAN segments, and the use of PIAT as the key metric, along with our data filtering strategies. Section V delves into our methodological approach, leveraging IQR and Z-Score methods to detect both singular and prolonged SD events. Section VI explores the robustness and applicability of our methods across various settings. Section VII summarizes our findings and their implications, while discusses limitations. The paper concludes with Section VIII.

II. RELATED WORK

Research efforts aimed at detecting SD have explored a diverse array of fields, though only a limited number of studies directly intersect with our specific domain of interest. Early work by Bremler-Barr et al. [5] explored the predictability of Internet SDs by analyzing round-trip time deviations, laying the groundwork for future studies in this field.

Several studies have focused on detecting and predicting SD using various network metrics. Abdelkefi et al. [6] proposed a method, which uses end-to-end delay and loss measurements to assess Internet path service quality. Their approach effectively detects abrupt changes and identifies service-level events, demonstrating the feasibility of service quality assessment based solely on end-to-end measurements. In a similar vein, Wu et al. [7] proposed a real-time packet loss monitoring system to address network quality of SD.

Machine learning techniques have been increasingly applied to network performance prediction. Hardegen et al. [8] employed Deep Neural Networks to predict throughput and duration of flows in a campus network, showcasing the potential of machine learning in network flow analysis. Our work builds on this trend, applying statistical methods in campus network to detect SD.

In service-based systems, Traini et al. [9] demonstrated how recognizing patterns in latency could be instrumental in diagnosing performance issues, while Cortellessa et al. [10] focused on detecting latency degradation patterns in microservice-based applications. Through a case study, the effectiveness of their approach in detecting artificially injected latency degradation patterns was demonstrated. In contrast, our work relies solely on passive measurement without injecting probe packets, bringing the research closer to real-world applications by analyzing unaltered network behavior.

Latency has been identified as a crucial indicator of SD, particularly in specific application domains. Amaral et al. [11] investigated the impact of network impairments on video quality in cloud gaming, developing an algorithm to predict real-time visual degradations based on accumulated latency. Similarly, Li et al. [12] introduced

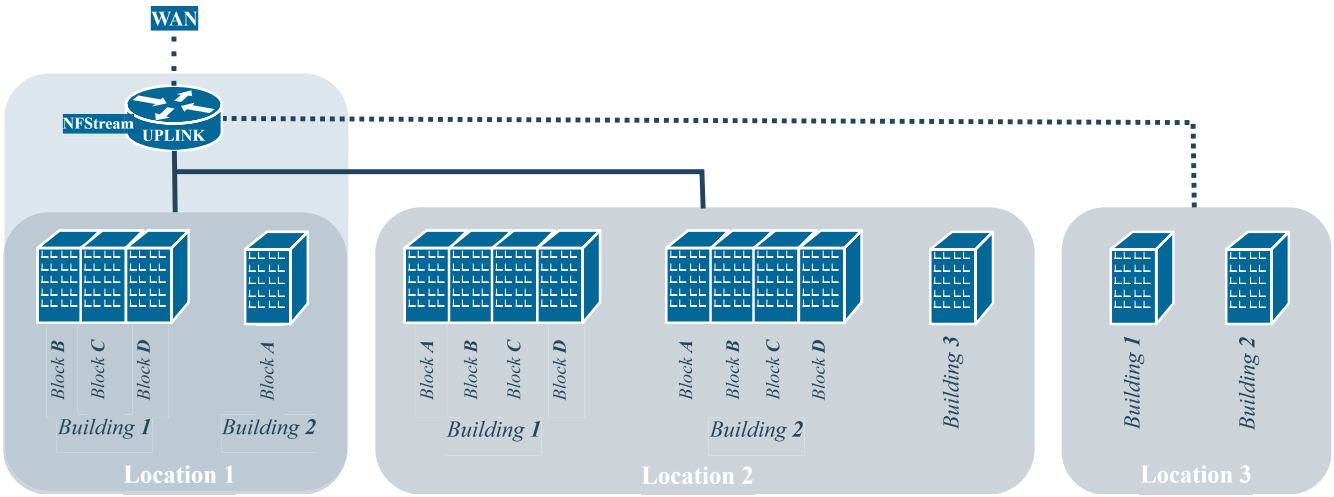


FIGURE 1. High-level topology of the measured network.

an adaptive bit rate algorithm for enhancing user QoE in low-latency live video streaming. These studies underscore the importance of latency as a key metric in SD detection, which aligns with our focus on PIAT analysis.

Some researchers have explored the concept of controlled SD as a resource management strategy. Xian et al. [13] and Hou et al. [14] proposed algorithms for predictive resource allocation in optical data center networks, using controlled SD to manage network overloads. These studies balance service quality and resource utilization, approach SD from a different perspective.

In the context of cloud computing, Pons et al. [15] introduced an approach to detect inter-VM interference and estimate performance degradation in public cloud environments. Their approach for performance estimation is tailored to cloud settings, contrasting with our methodology, which is applied to a campus network.

Our research distinguishes itself in several key aspects. Firstly, we specifically target TCP flows in LAN environments, an area that has received limited attention in previous studies. By employing PIAT analysis in a university dormitory network setting, we focus on latency and jitter as a principal indicators of SD in a context particularly susceptible to degradation. While aligning with previous studies in recognizing latency as a key metric for SD identification, our approach uniquely targets the internal network segment to provide indications of potential SD.

Moreover, our work addresses a significant gap in the field by providing a novel, annotated dataset of IP flows with labeled latency-induced SD instances. To the best of our knowledge, this is the first publicly available dataset of its kind, offering a valuable resource for future research in SD detection. This dataset, combined with our methodological approach using Z-score and IQR methods for PIAT analysis, provides a comprehensive framework for understanding and detecting SD in LAN environments.

III. DATASET

The dataset utilized in our study comprises network traffic data from a university dormitory network. The network's topology, as shown in Fig. 1, provides a high-level overview, abstracting specific details like the nature of network connections and internal building topology into simplified representations. Three primary locations within the dormitory network are depicted, each based on the physical grouping of buildings. *Location 1*, positioned on university premises, serves as the uplink router site where traffic was captured via a Switched Port Analyzer. *Location 2* is situated a few kilometers away on the city outskirts, and *Location 3* is located in a different town, connected through multiple hops as indicated by a dotted line in the figure.

A. FLOW MEASUREMENT

For network flow measurement, we employed the NFStream tool [17] configured as follows:

- We captured only IPv4 TCP traffic; the rationale for this choice is explained in Section IV-D.
- Flow expiration settings were configured to terminate all flows after 2 minutes of inactivity following the last received packet, or after 30 minutes regardless of activity.
- Packet size accounting was configured to include the IP header, but tunnel decoding was not enabled for this measurement.
- The nDPI library [18] was used to dissect up to 20 packets for Layer 7 visibility, which allowed us to identify application usage and other specific data.
- We analyzed statistical features such as packet size, PIAT, and packets with various TCP flags. Statistical measures (minimum, maximum, mean, standard deviation) were calculated for traffic in both directions—source to destination and vice versa—and combined.

TABLE 1. Characteristics of the Captured Network Flow Data.

	Monday	Tuesday	Wednesday	Thursday	Friday
Flow Count	3,185,436	2,513,156	3,092,647	2,703,948	2,865,281
Start of the Measurement	2024-01-15 19:24:40	2024-01-16 19:34:09	2024-01-17 19:26:35	2024-01-18 19:20:36	2024-01-19 19:20:35
End of the Measurement	2024-01-15 20:18:29	2024-01-16 20:16:14	2024-01-17 20:18:01	2024-01-18 20:15:01	2024-01-19 20:15:01
Measured Timespan	53.81 minutes	42.09 minutes	51.44 minutes	54.41 minutes	54.43 minutes
Min Flow Rate $[\frac{1}{s}]$	78	312	385	70	319
Max Flow Rate $[\frac{1}{s}]$	3469	3511	3766	3199	2651
Mean Flow Rate $[\frac{1}{s}]$	986.2	994.92	1001.83	827.91	877.04
Std. Dev. of Flow Rate $[\frac{1}{s}]$	247.48	270.75	241.2	228.51	322.27
Min Data Rate [MiBps]	0.0587	0.8091	1.2242	0.0807	0.3239
Max Data Rate [MiBps]	78,087	108,119	73,567	137,937	83,792
Mean Data Rate [MiBps]	243.49	307.21	254.69	244.64	204.68
Std. Dev. of Data Rate [MiBps]	1475.07	2322.29	1517.05	2541.15	1603.97
Flow Count *after filtering*	1,464,421	1,097,956	1,400,967	1,089,312	923,214
Flow Count *after LAN-side delay extraction*	1,356,999	1,012,526	1,282,952	1,002,830	845,631

- Our custom NFPlugin [17] managed the expiration of TCP flows based on their natural termination. Specifically, a flow is terminated after an ACK that follows two FIN packets and does not carry a FIN itself. Additionally, any flow that begins with a FIN or RST packet is also terminated, diverging from the standard TCP three-way handshake process.

B. TEMPORAL CHARACTERISTICS

Network traffic was measured over the course of one week, specifically during the evening hours. Each session lasted approximately 40-50 minutes, depending on the volume of traffic captured that day. Table 1 details the specific measurement windows and durations for each day, as well as the daily flow counts and the minimum, maximum, mean, and standard deviation for flow arrival and data throughput rates.

Fig. 2a illustrates the daily flow arrival rate per second, showing considerable variability. Generally, the flow arrival rates averaged around 900-1000 flows per second, with a standard deviation of approximately 250. Notable spikes in the data, with peaks exceeding 3500 flows per second, indicate periods of intense activity or bursts of flow arrivals. While the rate frequently remained below 1500 flows per second between these peaks, lower burst arrival rates were observed on Wednesday and Friday, though the average rates were consistent with other days. This pattern suggests intermittent periods of heightened activity amid generally steadier or lower rates of flow arrivals. Factors contributing to these fluctuations could include typical network usage patterns, scheduled events such as database updates, or anomalies within the monitored network. These peaks may highlight potential instances of SD.

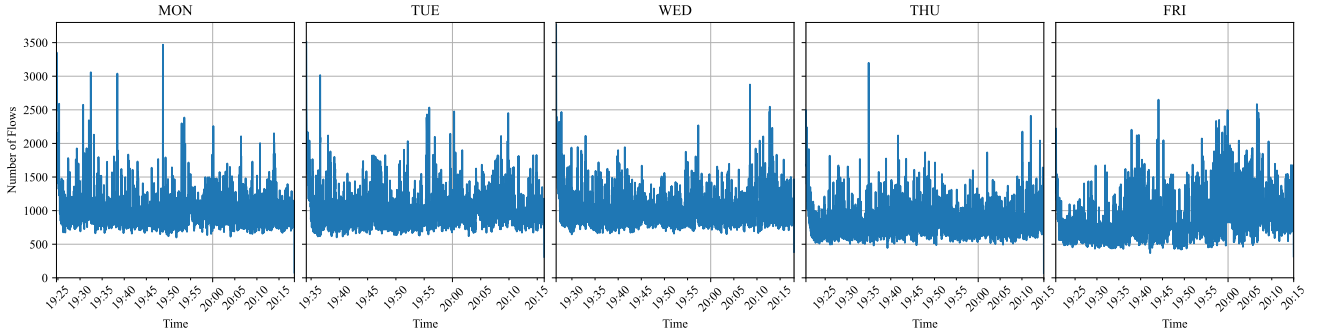
Fig. 2b illustrates the data transfer rates observed during the measurement period. Generally, rates were relatively low, often remaining below 5,000 MiBps. However, notable exceptions include spikes that exceeded 30,000 MiBps, particularly around 19:50 on Thursday and again at the same time the following day, suggesting a common cause. These spikes were abrupt and short-lived, indicating brief periods of very high data transfer activity before returning to the baseline level. On average, data rates varied between 200-300 MiBps, with a standard deviation ranging from 1400 to 2400 MiBps. The distribution and magnitude of these spikes suggest irregular and potentially unpredictable bursts in data transmission, possibly due to activities such as scheduled data transfers, network backups, or streaming of high-definition media.

Further analysis in Fig. 2c reveals that the significant spikes around 19:50 on both Thursday and Friday predominantly correspond to inbound traffic, with negligible outbound traffic. This pattern of high download activity with minimal uploads was consistent across all notable spikes, hinting at a heavy inbound data flow.

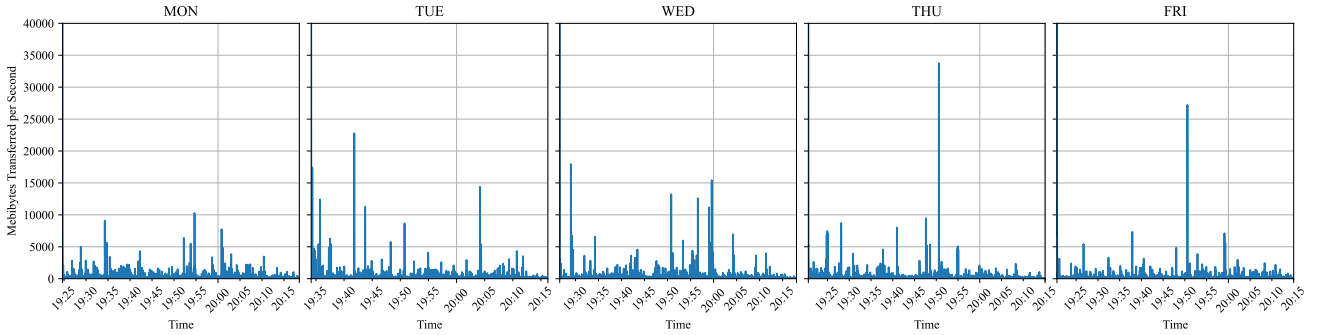
Interestingly, the flow arrival rate and the data rate do not appear closely correlated. While both metrics experience spikes, they occur at different times; the data rate peaks when the flow arrival rate is at normal levels and vice versa. This divergence could indicate scenarios where a high volume of transferred bytes accompanies a relatively low number of flows, potentially suggestive of an attack scenario.

Fig. 3 presents the Empirical Cumulative Distribution Function (ECDF) plots for three key flow features: *packet count*, *flow size* (in bytes), and *flow duration* (in milliseconds). The packet count plot reveals that over 90% of flows consist of fewer than 100 packets, predominantly resulting in shorter flows. Notably, more than a quarter of all flows contain just a single packet, and only a small fraction of flows extend to millions of packets. The flow size ECDF exhibits a similar but more gradual trend; nearly 40% of the flows transfer less than 100 bytes in total, and over 90% contain no more than 10 KB of data. In contrast, the flow duration ECDF illustrates that while many flows are short in terms of packet count and size, the duration of flows increases more gradually, with the longest flows exceeding 15 minutes.

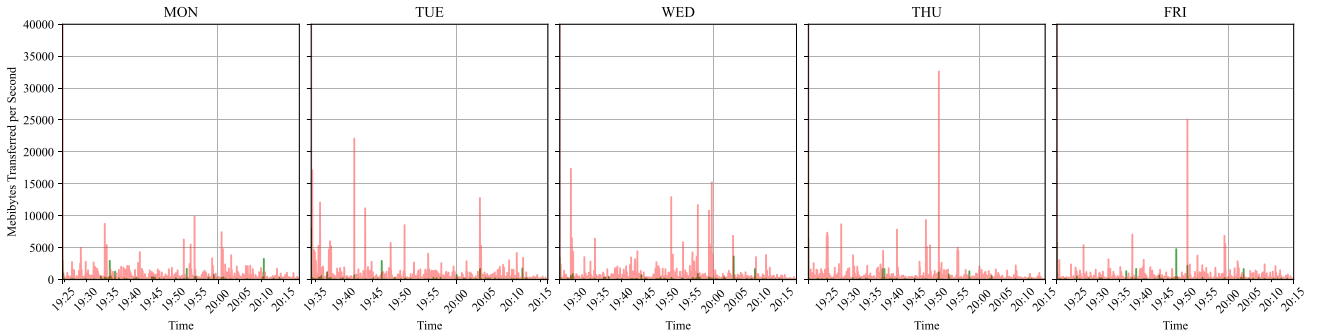
Given our focus on analyzing delays that result in SD events in flows, very short flows, specifically those with fewer than two packets (indicating no response was received or



(a) **Flow arrival rate [$\frac{1}{s}$].**



(b) **Data rate [MiBps].**



(c) **Directional data rate [MiBps]. Green: upload (outbound) data Rate. Red: download (inbound) data rate**

FIGURE 2. **Daily distribution of temporal features during the week.**

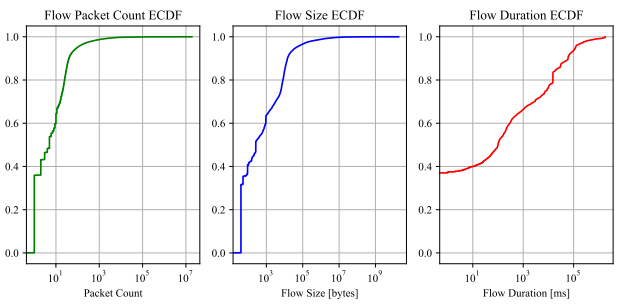
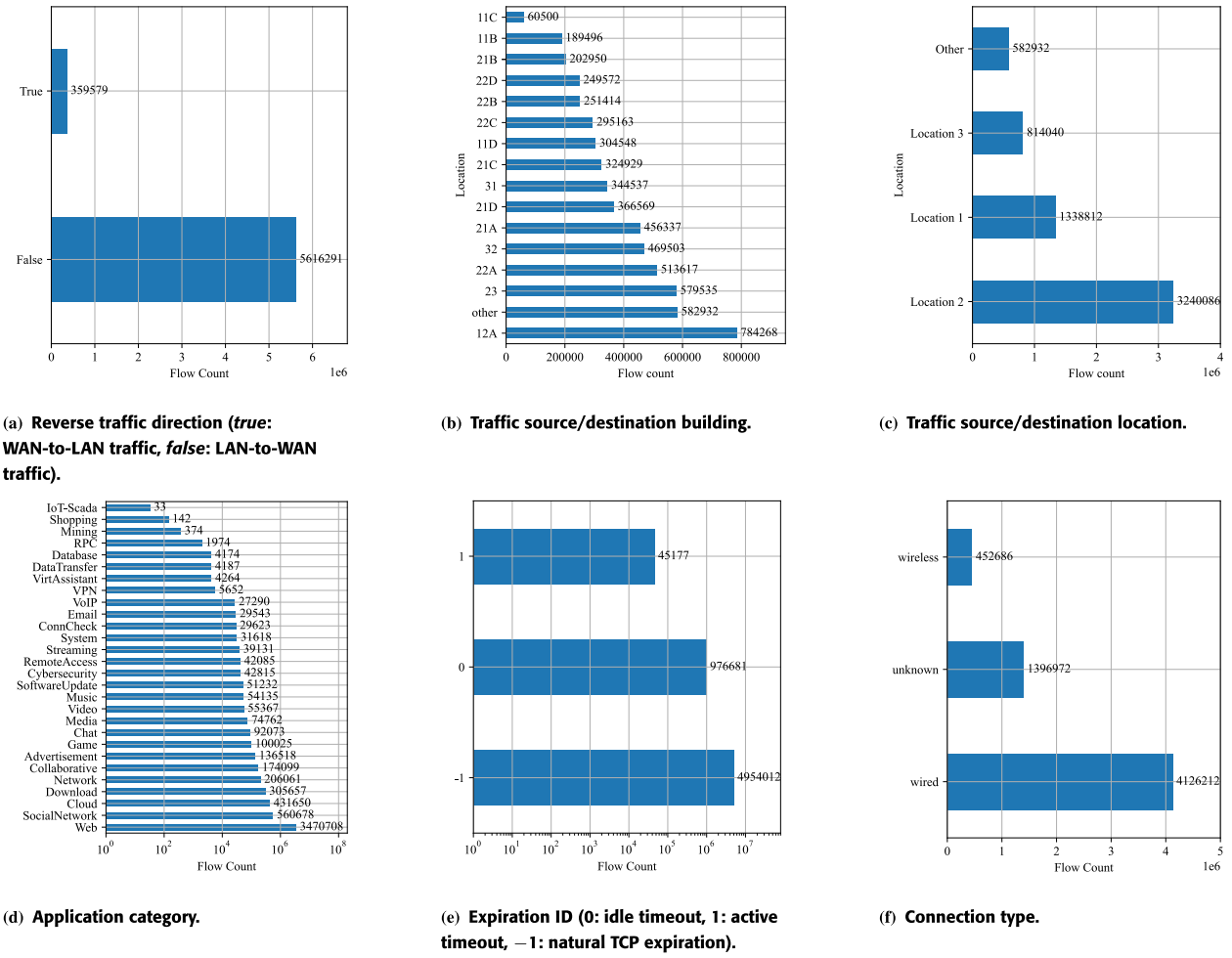


FIGURE 3. **ECDF plots for packet count, flow size and flow duration.**

sent), are excluded from our analysis. Additionally, flows were filtered based on the confidence level [18] determined by nDPI. The confidence level, indicated by a numeric value, reflects the certainty of the categorization; a higher number signifies greater confidence. Specifically, Level 6—achieved through Deep Packet Inspection rather than heuristic methods such as port-based approaches or correlations based on previous sessions—denotes the highest confidence and was the threshold for retaining data in our study. After applying these filters, approximately 40% of the flows from Monday to Thursday and 32% of Friday's traffic were retained for further analysis. The counts of flows retained post-filtering are detailed in Table 1.

C. FEATURE CHARACTERISTICS

In addition to temporal features, we analyzed the distribution of various categorical and numerical attributes, as shown in Fig. 4. An examination of traffic directionality (Fig. 4a) reveals that the overwhelming majority of flows originated



(a) **Reverse traffic direction (*true*: WAN-to-LAN traffic, *false*: LAN-to-WAN traffic).**

(b) **Traffic source/destination building.**

(c) **Traffic source/destination location.**

(d) **Application category.**

(e) **Expiration ID (0: idle timeout, 1: active timeout, −1: natural TCP expiration).**

(f) **Connection type.**

FIGURE 4. **Cardinality distribution of dataset features.**

from within the LAN and were directed towards WAN, with these non-reversed flows outnumbering reversed flows by an order of magnitude. Daily analyses show that while the volume of reversed flows remained constant throughout the week, non-reversed flows varied significantly. Closer inspection of the reversed flows indicated that they predominantly consisted of TLS and RDP traffic directed towards a specific host, characterized by highly consistent packet order and sizes. Notably, the majority of these reversed flows originated from a single external host, potentially representing deliberate connections to an open RDP port, though they may also reflect typical Internet background noise.

Fig. 4b demonstrates that while the location with the highest traffic cardinality varied daily, 12A, 22A, 23, and 32 consistently reported high traffic volumes throughout the week. Here, the encoding convention follows a specific pattern: the first number represents the location, the second denotes the building, and the last letter signifies the block, as outlined in Fig. 1. For instance, 12A corresponds to *Location 1-Building 2-Block A*, while 32 corresponds to *Location 3-Building 2*. There was also a notable amount of traffic from locationally non-assigned sources. Conversely, 11C consistently recorded the lowest traffic, with only a few thousand flows each day.

When traffic is grouped by location (Fig. 4c), *Location 2* emerges as the predominant source of flows, generating over 500,000 flows daily and exceeding 3 million flows in total for the week. Given that *Location 2* houses the largest number of buildings and students, such high traffic volumes are expected. The data shows that traffic from *Location 2* consistently surpassed that from all other locations combined throughout the week.

In our examination of the application category cardinality within the captured traffic, significant variances were observed. For instance, categories such as *Web* consistently showed high activity levels, with data points exceeding 500,000 daily and peaking at 800,000 on two occasions. In stark contrast, categories like *IoT-Scada*, *Shopping*, and *Mining* registered exceedingly low activity, with fewer than 50 data points each. Intermediate levels of activity were noted in other categories, ranging from several thousand

to tens of thousands of data points. Fig. 4d illustrates the cumulative distribution of application category cardinalities across the dataset. Additionally, the application names were also analyzed, providing a more granular view of the types of flows present, which could be pivotal for more detailed investigations.

Fig. 4e presents the distribution of flow expiration types. Predominantly, flows were terminated through *natural TCP expiration* (indicated by −1 on the plot), which involves the standard FIN-ACK sequence in TCP connection terminations. Approximately 200,000 flows daily expired due to *idle timeout* (marked by 0 on the plot), occurring when no packets were received after two minutes of inactivity. *Active timeout*, which forcibly ends flows after 30 minutes regardless of activity, was a rare occurrence, affecting only a few thousand flows each day (indicated by 1 on the plot).

Finally, the distribution of connection types, as detailed in Fig. 4f, reveals that the majority of traffic originated from wired connections, accounting for over 4 million flows. Notably, around 500,000 flows were initiated by devices recognized as wireless, while a substantial number of flows lacked specific connection type information.

D. DATASET AVAILABILITY

In the spirit of fostering reproducibility and encouraging open scientific collaboration, we are making our dataset publicly available [19].

To comply with GDPR, IP addresses and MAC addresses, which could be used to identify specific students, were anonymized using the *blake2b* algorithm [20]. Nonetheless, original IP range data was retained to annotate each flow with its location within the dormitory network. This setup not only preserves privacy but also provides crucial contextual information about the flow origins and destinations. Additionally, we distinguished whether the traffic was initiated by LAN hosts or directed towards the LAN, and identified whether the traffic originated wirelessly or via a wired connection, using the applicable addressing policy. This detailed dataset facilitates a deeper understanding of network dynamics and supports robust analysis of SD and other network issues.

Furthermore, to aid in the transparency and reproducibility of our research, we are also sharing the Jupyter Notebook file [19] used for analyzing the dataset. By providing these resources, we aim to support ongoing research in network flow analysis and contribute to the broader advancement of knowledge in this field.

IV. FOUNDATIONAL CONCEPTS AND PREPARATORY CONSIDERATIONS

A. THE LATENCY-SERVICE DEGRADATION NEXUS

In today's digital landscape, where online services from entertainment streaming to business conferencing are pervasive, understanding network performance is crucial. Latency stands as a primary metric for assessing this performance and is defined mathematically as:

$$\text{Latency (L)} = \text{Time (T2)} - \text{Time (T1)}. \tag{1}$$

Here:

- T1 represents the time when an action or request is initiated.
- T2 is the time when a response is received.

Generally speaking, latency plays a crucial role in network analysis for several reasons. As a quantifiable measure of network responsiveness, latency serves as an objective metric for comparing different network configurations or tracking performance over time. Additionally, latency analysis serves as a powerful diagnostic tool, allowing network administrators to identify potential issues such as network congestion, hardware failures, or inefficient routing.

In the context of our study, we focus on latency's critical role in user experience and service quality. We utilize latency as a key indicator of service degradation. It directly impacts user experience quality, with even slight increases potentially leading to noticeable SD [11], [12]. This impact is particularly evident in applications ranging from online gaming to video conferencing, where responsiveness is key to user satisfaction.

We represent SD ΔS as a function of increased latency ΔL:

$$\Delta S = f(\Delta L), \tag{2}$$

where, f denotes the function that quantifies how service quality diminishes as latency increases. This present research aims to precisely identify this function while exploring other contributing variables.

While latency provides insight into network delay, jitter helps us understand the stability of this delay over time. Thus, complementary to latency, we also consider jitter, which is defined as the variability in latency over successive intervals. We define jitter mathematically as:

$$\text{Jitter (J)} = \text{Latency (L2)} - \text{Latency (L1)}, \tag{3}$$

where:

- L1 is the latency measured at an earlier time point.
- L2 is the latency measured at a subsequent time point.

Jitter is important in network analysis as it captures the consistency of network performance. High jitter can lead to packet loss and degraded service quality, especially in real-time applications. In our work, we examine both latency and jitter to provide a comprehensive view of network performance and to identify more severe instances of SD.

B. VERTICAL SEPARATION OF TRAFFIC INTO LAN AND WAN SEGMENTS

From the vantage point of a network edge, which serves as the observation point in our study, network flows can be clearly divided into two principal directions: towards the LAN and towards the WAN. We term this as vertical separation, which is imperative for several reasons:

- By focusing on LAN-side traffic, we can isolate and study the behavior of the local network without the

confounding effects of external factors. This allows for more precise identification of issues within the network administrator's control.
- LAN-side measurements provide a more accurate representation of the network's actual performance capabilities, as they are not influenced by internet congestion or routing inefficiencies beyond the local network.
- Separating LAN and WAN traffic enables more efficient troubleshooting. Issues identified in LAN-side traffic can be addressed directly by local network administrators, while WAN-side issues may require coordination with internet service providers.

Fig. 5 demonstrates the concept of vertical separation, with the diagram showing how traffic flows are divided. WAN components are highlighted in orange, representing delays and interactions that occur outside the local network's immediate control. Conversely, LAN components are depicted in blue, illustrating the internal network dynamics. Arrows in the figure indicate the direction of packet traffic, moving between LAN and WAN endpoints, thus providing a visual representation of how data traverses these network segments.

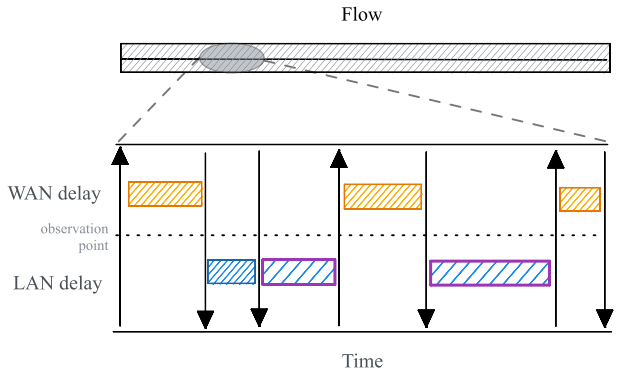


FIGURE 5. Visual representation of network traffic flow depicting the vertical separation between WAN and LAN delays. Packets are illustrated with arrows indicating direction, and specific packets meeting latency measurement conditions are highlighted with a thick purple border.

As external factors beyond the network provider's control, which occur on the WAN side, can adversely affect analysis results, we focus in our study only on LAN-side delays to understand local network conditions accurately. By employing this vertical separation in our analysis, we can develop a more nuanced understanding of network performance and service degradation, particularly within the context of LAN environments where local network conditions play a crucial role in user experience.

C. PACKET INTER-ARRIVAL TIME

PIAT quantifies the time interval between the arrivals of two consecutive packets within a network flow at the flow meter. For a sequence of packet arrival times $t_1, t_2, \ldots, t_n$, PIAT for the i-th packet is mathematically defined as:

$$\text{PIAT}_i = t_i - t_{i-1}$$

for $i > 1$, with $\text{PIAT}_1 = 0$ indicating no preceding packet for the first in the sequence.

In general, PIAT plays a crucial role in network analysis by providing detailed insights into traffic patterns. It enables the identification of both regular and irregular packet transmission behaviors, which is particularly valuable for understanding the characteristics of different applications and protocols. Fluctuations in PIAT can serve as indicators of network conditions; for instance, increases in PIAT may signal network congestion or bottlenecks, as packets experience longer queuing times under increased network load. In time-sensitive applications such as VoIP or online gaming, maintaining consistent and low PIATs is essential for ensuring high quality of service for users.

In the context of service degradation detection, PIAT analysis offers a more granular perspective on network performance compared to aggregate metrics. This fine-grained view allows for the detection of subtle degradations that might go unnoticed with coarser measurements. Our study leverages PIAT due to this precision and effectiveness. By analyzing PIAT patterns, we aim to enhance our understanding of network dynamics and improve the accuracy of service degradation detection.

D. LEVERAGING PIAT FOR LATENCY ESTIMATION

In our methodology, we utilize PIAT, measured in milliseconds, as a key metric for discerning network latency. PIAT values, along with packet direction and size measured in bytes, are captured as part of the Sub-Packet-Length-Time (SPLT) features measured by NFStream. These three features provide packet-level insights for the first n packets of each flow. For our study, we have set the recording of SPLT features to the maximum supported value of 255.

Traffic often appears as a burst of multiple packets; therefore, our analysis focuses on the time interval between the arrival of the last packet in such a burst (in the incoming direction) and the first outgoing packet that responds to this burst within the same flow. In Fig. 5, instances meeting these conditions are highlighted with a distinct purple border.

Our methodology leverages the unique positioning of our measurement point at the network edge, which allows us to focus specifically on LAN-side latency. We operate under the assumption that delays caused by local endpoints are negligible compared to those induced by broader network conditions. By selectively analyzing TCP flows at this edge point, we ensure the PIAT values used reflect the time difference between the receipt of the last packet from the WAN side and the transmission of the corresponding TCP acknowledgment. This duration is indicative of the time traffic spends within the local network, effectively representing LAN-side latency.

This approach provides several advantages. Firstly, by focusing on the LAN-side component of RTT, we eliminate the impact of external internet congestion and WAN-related issues that can obscure local network performance. Secondly, this localized measurement allows for a more

Algorithm 1 Identifying LAN Delay

```
procedure id_lan_delay(dir, prev_dir, reversed)
    if reversed then
        ▷ dir: dst2src (LAN2WAN),
          prev_dir: src2dst (WAN2LAN) ◁
        return dir == 0 and prev_dir == 1
    ▷ dir: src2dst (LAN2WAN),
      prev_dir: dst2src (WAN2LAN) ◁
    return dir == 1 and prev_dir == 0
```

Algorithm 2 Vertical Separation of Flows

```
procedure vert_sep(splt_dir, splt_piat)
    D ← []
    prev_dir ← −1
    for idx, dir in enumerate(splt_dir) do
        if idx == 0 then
            prev_dir ← dir
        else
            ▷ Examining the dir and prev_dir we only
              take into account the response to the last
              packet in the case of a burst ◁
            if id_lan_delay(dir, prev_dir, reversed) then
                D ← D + splt_piat[idx]
            prev_dir ← dir
    return D
```

precise evaluation of LAN performance, as it's not affected by variable delays in the broader internet. Lastly, by analyzing the packet inter-arrival times of closely succeeding outgoing and incoming packets at the network edge, we gain detailed insights into the behavior of the local network that would be difficult to discern from end-to-end measurements.

This edge-based, LAN-focused approach enables us to detect and analyze SD events with greater precision, providing a novel perspective on network performance evaluation.

Algorithms 1 and 2 illustrate the steps involved in extracting PIAT values that pertain to LAN-side delays. While a similar analysis could also be conducted for LAN-to-WAN traffic, our focus remains on the LAN side due to its potential to provide detailed insights with fewer privacy concerns within a controlled monitoring environment. Understanding these internal dynamics lays a foundation for future methodologies that could extend to monitoring WAN-side behaviors.

The identification of LAN-side delays significantly narrows down the set of PIAT values compared to the original dataset. We discard flows with fewer than two LAN-side delays from further analysis because they lack sufficient LAN-side delays to calculate jitter, which is crucial for our service degradation detection methodology. This filtering step results in a dataset that only includes flows with at least two LAN-side delays, as shown in Table 1.

The impact of this filtering on our dataset characteristics is substantial and multifaceted:

- Upon analyzing the temporal characteristics of this filtered dataset, we observed a notable decrease in flow arrival rates—from an average of approximately 1000 flows per second to around 400 flows per second–along with a more consistent distribution over time. Although some spikes in arrival rate persist, the overall data rate has slightly decreased, maintaining similar patterns to those observed prior to filtering.
- The proportion of flows with very few packets decreased significantly; initially, almost 40% of flows contained only one packet, whereas now, only 20% contain fewer than ten packets. This shift towards more substantial flows enhances the reliability of our analysis.
- The flow size has become more consolidated, with only 20% of flows containing less than 1 KB of data, down from 60% before filtering. This change provides a more representative view of typical network interactions.
- We observe fewer short-lived flows, with less than 5% lasting under 10 milliseconds compared to over 40% previously. This shift allows for more meaningful analysis of network behavior over time.

Despite these reductions, the overall pattern of the ECDF plots remains unchanged (refer to the digital artifacts at [19]), suggesting that the main characteristics of the data were retained despite the narrower dataset scope. This consistency is crucial as it indicates that while we have focused our analysis on more substantial network interactions, we have maintained the overall distribution patterns of our data.

It is important to note that even if the discarded flows had been retained, we would not have been able to utilize them effectively in our analysis. By definition (Equation 3), jitter calculation requires at least two LAN delays. Therefore, by excluding these flows, we do not lose any information that would influence our jitter-based analysis of service degradation.

The implications of this filtering on our analysis are significant:

- It improves the reliability of our jitter calculations and subsequent service degradation detection by ensuring we have sufficient data points for each flow.
- It provides a more representative view of significant network interactions, potentially leading to more accurate insights into network behavior.
- While we may miss some short-lived or small-scale network events, we believe the benefits of this approach–namely, reduced noise and more meaningful data–outweigh this limitation for our specific research goals.

This filtered dataset, with its focus on more substantial and longer-duration flows, forms the foundation of our subsequent analysis and service degradation detection methods.

E. REDUCED DATA VIEW COVERAGE

A reduction in dataset size can undeniably impact the quality of data used for SD analysis. To validate the integrity of our filtered dataset and ensure that it remains representative

of overall network behavior, we assess its coverage. High coverage would indicate that the majority of flows are captured within the initial scope of 255 packets of flows, an essential criterion given our reliance on SPLT values for PIAT-based latency analysis.

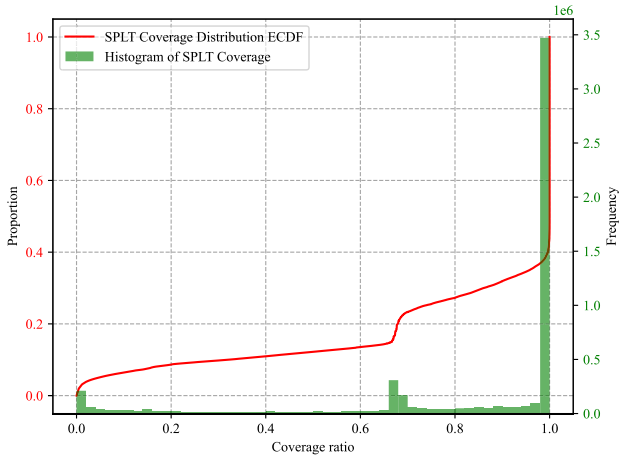


FIGURE 6. SPLT Coverage of the flow.

Fig. 6 shows that a significant majority of the flows are fully encompassed within the initial SPLT values corresponding to the first 255 packets. Nevertheless, a few thousand longer flows exceed this packet limit, indicating partial coverage. Despite this, such partially covered flows constitute less than 5% of the total. Conversely, approximately 30% of flows are partially covered, while the vast majority either enjoy full coverage or are nearly fully covered.

Another metric to consider is the ratio of LAN delay durations to the remaining WAN delays in the measured flows, as illustrated in Fig. 7. The analysis reveals that LAN delays are generally less prevalent than WAN delays, with some flows exhibiting significantly higher proportions of LAN delays. For 40% of the flows, LAN delays contribute almost nothing to their overall duration, indicating minimal LAN-side delays.

This analysis reassures us that, despite the dataset reduction, the primary characteristics and a substantial portion of the network dynamics are retained and adequately represented within the adjusted scope of our study.

V. METHODOLOGY FOR IDENTIFYING SERVICE DEGRADATION

In this section, we investigate the occurrence of SD events within the dataset, initially focusing on data from the most populous *Location 2* and limiting our analysis to the first three days to serve as a quasi-training set. This phase helps establish thresholds for classifying SD events based on Interquartile Range (IQR) and Z-score analyses, which are detailed later.

We define SD as a statistically significant deviation from typical flow latency behavior, characterized by notable increases in latency and jitter. The thresholds for identifying these deviations are tailored to the specific requirements of different application categories. For instance, even minimal increases in latency might constitute SD in delay-sensitive applications like voice calls or remote desktop interactions, where timing is critical. Conversely, for activities such as downloading, where latency sensitivity is lower, an increase in delay may not be as perceptible.

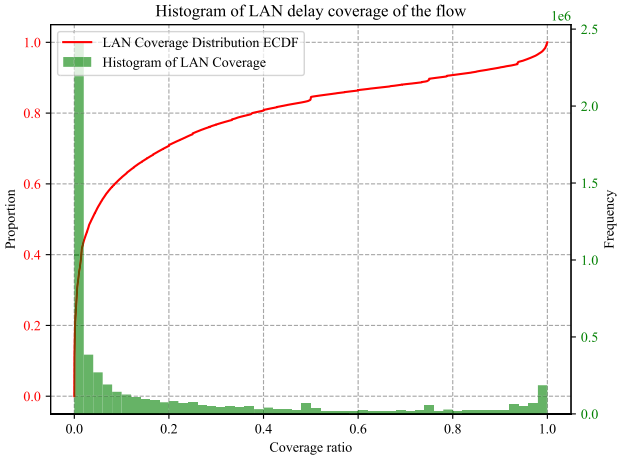


FIGURE 7. LAN delay coverage of the flow.

Additionally, we explore prolonged SD events, defined as extended periods where deviations in latency and jitter persist. An SD event in this context is identified as a contiguous series of delays, starting with an initial outlier in jitter followed by subsequent delays that also qualify as outliers. This approach seeks to capture sequences where delays not only spike unexpectedly but also remain elevated above the established anomaly threshold. We further examine scenarios where, despite significant and sustained increases in delays, the flow continues to exhibit high jitter, maintaining levels above the SD threshold.

To ensure robustness and mitigate risks such as overfitting or unreliable threshold estimations, our analysis prioritizes application categories with sufficient data volume. Specifically, we focus on categories that have recorded at least 50,000 flows at *Location 2*. This criterion coincidentally aligns with the top six application categories in the entire dataset, which include: *Web*, *Social Network*, *Download*, *Cloud*, *Network*, and *Collaborative*.

Table 2 presents the LAN delay counts following this selection step. Notably, *Web* traffic constitutes the majority of the data, overshadowing the other categories. However, the remaining categories still provide a substantial number of samples, adequate for conducting reliable statistical analysis.

A. INTERQUARTILE RANGE ANALYSIS

To evaluate the distribution of delays across various application categories, we utilized boxplots, as depicted in Fig. 8. To enhance visibility of differences across distributions, the y-axis is set to a logarithmic scale. The boxplots are ordered in descending sequence based on the volume of delay samples,

TABLE 2. Count of LAN delays per Application Category for *Location 2*.

Application Category Name	LAN Delay Count
Web	9,016,470
Social Network	1,797,216
Cloud	1,177,916
Collaborative	743,453
Download	440,421
Network	264,156
Count of all delay samples	13,439,632

and mean values are denoted by orange rectangles on each plot.

The analysis reveals distinct distribution patterns within the application categories, categorizing them into two groups based on delay characteristics. Categories such as *Download* and *Network* typically show delays not exceeding 10 milliseconds, while others like *Cloud* often exceed 100 milliseconds at their upper whisker, with delays extending into the tens of milliseconds range. Notably, *Cloud*, *Download*, and *Network* also display delays in the sub-millisecond range, with medians set at 1 millisecond. Due to limitations in NFStream's measurement capabilities, delays under one millisecond are recorded as 0 milliseconds. The mean delays for *Download* and *Network* are notably below one second, which is significantly lower compared to other categories where means reach several seconds, largely due to the presence of outliers as determined by Interquartile Range Analysis (IQR) analysis.

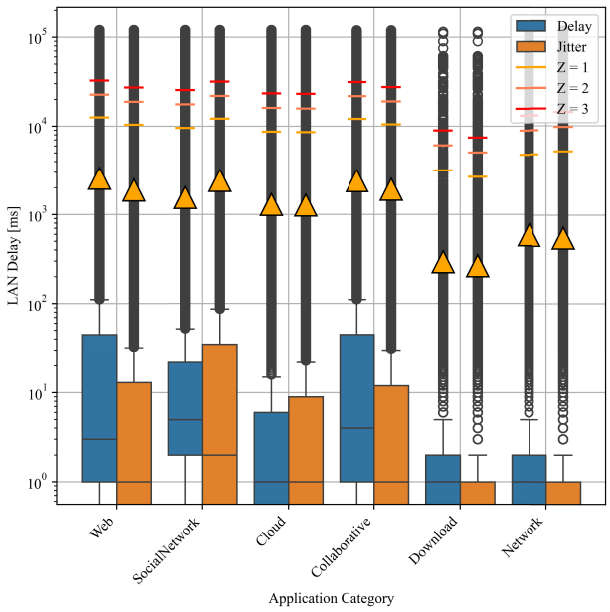


FIGURE 8. Distribution of LAN delay and jitter per application category.

Jitter distribution aligns closely with the observed delay patterns across all categories, with each showing a median jitter that matches or is lower than the median delay. The first quartile frequently registers at 0 milliseconds, indicating a significant occurrence of consistent delays (no jitter). Contrarily, *Social Network* flows display slightly longer jitters compared to delays, especially evident in the third quartile, upper whisker, and mean values.

Outliers, critical for identifying SD events, are evident across all categories and range from a few milliseconds to over 100 seconds. Detailed quantification of these outliers, including their impact on service quality, is presented in Table 3.

TABLE 3. Outlier Statistics for *Location 2*.

Metric	Quartile Analysis	Z-score Analysis
Number of *delay* outliers	1,680,245	499,656
Number of *jitter* outliers	2,682,895	374,765
Number of *intersection* outliers	1,156,820	194,233
Total rate of *delay* outliers	12.502%	3.718%
Total rate of *jitter* outliers	19.963%	2.789%
Total rate of *intersection* outliers	8.608%	1.445%

B. Z-SCORE ANALYSIS

Next, we utilize Z-score analysis to detect outliers in our dataset, particularly focusing on unusually large delays indicative of SD. Positive Z-scores, which signify values above the mean, are of particular interest as they represent potential SD events. We adopt standard thresholds, considering Z-scores greater than 2 or 3 as significant, with the specific threshold dependent on the application category's sensitivity to delays.

Fig. 8 presents delay and jitter means marked with orange diamonds and their corresponding Z-scores depicted by horizontal lines for all studied application categories at $Z = 1$, $Z = 2$, and $Z = 3$. These illustrate the degree to which specific measures deviate from the mean in standard deviation increments. The mean and standard deviation values crucial for these calculations are detailed in Table 4, highlighting that Z-scores typically mirror the distribution of mean values. For example, more sensitive categories such as *Web*, *Social Network*, *Cloud*, and *Collaborative* have Z-score thresholds with $Z = 3$ extending beyond 20 seconds. Conversely, the *Download* category's threshold is just under 10 seconds, while *Network* exceeds this slightly in both delay and jitter.

TABLE 4. Mean and Standard Deviation Values for Each Application Category.

Metric	Web	Social Network	Cloud	Collaborative	Download	Network
μ_{delay}	2,566.86	1,559.44	1,298.99	2,403.65	294.11	592.49
σ_{delay}	10,027.89	8,011.10	7,358.91	9,668.43	2,887.64	4,166.06
μ_{jitter}	1,893.55	2,424.25	1,277.34	1,921,77	265,79	551.51
σ_{jitter}	8,464.72	9,751.16	7,279.63	8,560.04	2,389.94	4,588.31

Fig. 8 clearly demonstrates the stringent nature of Z-score analysis compared to IQR analysis. While IQR may flag a higher number of singular delay SD events (marked with black circles as outliers beyond the whiskers) due to its sensitivity to the lower limit, Z-score analysis bases its findings on deviations from the mean delay values, which are generally higher. Consequently, Z-score analysis results in fewer identified outliers. As shown in Table 3 for $Z = 3$, events exceeding this Z-score threshold are considered significant. In line with standard practices in anomaly detection, we adopt $Z = 3$ as the threshold for our Z-score analysis.

C. EXAMINING PROLONGED SD EVENTS

SD events may manifest not only as singular delay and/or jitter outliers but also as prolonged sequences of consecutive outliers. To identify these sequences, we have developed an algorithm, outlined in Algorithm 3, that groups contiguous delays into a single SD event if all delays are recognized as outliers for a period defined by *Minimum Sequence Length* (`MIN_SEQ_LEN`), with an optional consideration for jitters. The algorithm operates as follows:

(i) It iterates through all delay samples in a flow.
(ii) If the jitter and delays are high (previously identified as singular SD points), it initiates an SD sequence.
(iii) The SD sequence continues as long as the delays (`SDd`) remain high, and optionally, if `require_jitter_for_sequence` is enabled, the jitter (`SDj`) must also remain high.
(iv) The sequence concludes when the traffic returns to normal, capturing the *start* and *end indexes* of the corresponding LAN delays marking the SD event.

Fig. 9 illustrates an example of an SD event for an application category requiring a minimum sequence length (`MIN_SEQ_LEN`) of 2. While an earlier LAN delay may also be an outlier, the absence of sufficient jitter and a subsequent outlier disqualifies it as an independent SD event.

This methodology allows us to assess SD events across all application categories, utilizing outliers identified by both IQR and Z-score analyses. We explore variations with the `require_jitter_for_sequence` option both enabled and disabled. Tables 5 and 6 present the counts of SD events and the percentage of total SD event time relative to the entire duration of the flow for each category. Only sequences meeting or exceeding the stipulated minimum length are considered in the final count of SD events. Longer sequences, even though they extend beyond this threshold, are counted as a single event.

We observe a decreasing trend in the number of SD events and their duration coverage as the minimum sequence length (`MIN_SEQ_LEN`) requirement is increased. Notably, with Z-score analysis, the initial count and coverage of SD events are significantly lower compared to those identified through IQR analysis. A similar pattern of counts and coverages is reached by `MIN_SEQ_LEN` = 3 for most categories, except for the *Web* category, which shows a consistent pattern with

Algorithm 3 Identifying LAN Delay

```
procedure find_SD_sequences(SDd_list, SDj_list, MIN_SEQ_LEN, require_jitter_for_sequence)
    for i, (SDd, SDj) in enumerate(zip(SDd_list, SDj_list)) do
        if require_jitter_for_sequence then
            seq_condition ← SDd and SDj
        else
            seq_condition ← SDd
        if (sequence_length == 0 and SDj and SDd) or (sequence_length > 0 and seq_condition) then
            sequence_length ← sequence_length + 1
        else
            if sequence_length ≥ MIN_SEQ_LEN then
                start ← i - sequence_length
                end ← i
                sequences.append((start, end))
                seq_SD_list[start: end] ← [True] * (end - start)
            if sequence_length > 0 then:
                sequence_length ← 0
    ▷ If the last sequence goes until the end ◁
    if sequence_length ≥ MIN_SEQ_LEN then
        start ← len(SDd_list) - sequence_length
        end ← len(SDd_list)
        sequences.append((start, end))
        seq_SD_list[start: end] ← [True] * (end - start)
    return [sequences, seq_SD_list]
```

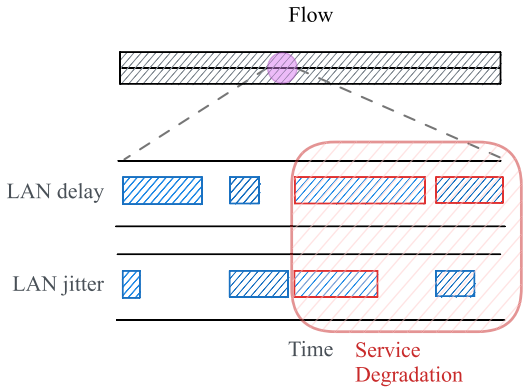


FIGURE 9. A sample SD event when `MIN_SEQ_LEN` = 2 and `require_jitter_for_sequence = False`.

IQR analysis, indicating more prolonged and extreme SD events in this category that are detectable by both methods.

The reduction in SD events is neither linear nor uniform across different application categories and the two outlier identification methods. IQR analysis generally shows a steadier decrease in SD events, whereas Z-score analysis often experiences abrupt declines in SD event counts and coverage, which then stabilizes. This behavior varies by application category at different `MIN_SEQ_LEN` values—*Download*, *Network*, and *Social Network* at 2; *Cloud* at

TABLE 5. SD Event Count and Coverage Statistics for Different Minimum Sequence Lengths with `require_jitter_for_sequence` option turned *OFF*.

MIN_SEQ_LEN	Web				Cloud				Download				Network				Collaborative				Social Network			
	Quartile		Z-Score		Quartile		Z-Score		Quartile		Z-Score		Quartile		Z-Score		Quartile		Z-Score		Quartile		Z-Score	
	count	cov.%	count	cov.%	count	cov.%	count	cov.%	count	cov.%	count	cov.%	count	cov.%	count	cov.%	count	cov.%	count	cov.%	count	cov.%	count	cov.%
1	477,758	33	*126,702*	*24*	120,813	46	*21,033*	*28*	27,830	21	*7,304*	*16*	16,658	49	*1,705*	*19*	36,110	39	*17,491*	*32*	146,444	34	*19,599*	*18*
2	128,183	25	*54,624*	*18*	29,485	34	*6,281*	*14*	5,113	10	*116*	*5*	5,523	43	*240*	*8*	12,905	27	*5,170*	*20*	24,339	12	*1,933*	*5*
3	77,573	22	*42,125*	*17*	12,630	23	*745*	*4*	1,724	7	*63*	*4*	2,848	35	*174*	*7*	6,901	20	*2,318*	*15*	11,224	9	*1,330*	*5*
4	60,582	21	*37,374*	*16*	6,302	17	*541*	*3*	870	6	*45*	*4*	1,978	31	*125*	*6*	5,351	18	*2050*	*14*	5,664	7	*1084*	*4*
5	47,025	18	*29,675*	*14*	3,723	14	*309*	*2*	593	5	*41*	*4*	1,505	29	*94*	*4*	4,422	17	*1,874*	*13*	3,604	6	*963*	*4*
6	-	-	*13,404*	*9*	-	-	*156*	*2*	-	-	*27*	*3*	-	-	*43*	*2*	-	-	*1,667*	*12*	-	-	*730*	*4*
7	-	-	*10,924*	*8*	-	-	*128*	*2*	-	-	*27*	*3*	-	-	*41*	*2*	-	-	*1,576*	*12*	-	-	*561*	*3*
8	-	-	*6,665*	*6*	-	-	*116*	*2*	-	-	*26*	*3*	-	-	*23*	*1*	-	-	*219*	*2*	-	-	*538*	*3*
9	-	-	*5,704*	*5*	-	-	*113*	*2*	-	-	*22*	*3*	-	-	*17*	*1*	-	-	*141*	*2*	-	-	*514*	*3*
10	10,465	8	*5,290*	*5*	1,097	10	*109*	*1*	255	4	*22*	*3*	686	19	*17*	*1*	805	5	*131*	*2*	763	3	*485*	*3*
15	3,881	5	-	-	614	8	-	-	146	3	-	-	156	6	-	-	229	3	-	-	63	0	-	-
20	2,033	3	-	-	468	7	-	-	99	3	-	-	70	3	-	-	153	2	-	-	20	0	-	-
25	1,603	3	-	-	402	6	-	-	70	2	-	-	46	1	-	-	117	2	-	-	10	0	-	-

TABLE 6. SD Event Count and Coverage Statistics for Different Minimum Sequence Lengths with `require_jitter_for_sequence` option turned *ON*.

MIN_SEQ_LEN	Web				Cloud				Download				Network				Collaborative				Social Network			
	Quartile		Z-Score		Quartile		Z-Score		Quartile		Z-Score		Quartile		Z-Score		Quartile		Z-Score		Quartile		Z-Score	
	count	cov.%	count	cov.%	count	cov.%	count	cov.%	count	cov.%	count	cov.%	count	cov.%	count	cov.%	count	cov.%	count	cov.%	count	cov.%	count	cov.%
1	506,144	17	*126,757*	*9*	126,392	27	*21,034*	*20*	28,619	20	*7,312*	*12*	18,664	40	*1,709*	*13*	37,910	22	*17,495*	*17*	149,818	28	*19,607*	*13*
2	74,708	6	*139*	*0*	19,636	11	*1*	*0*	4,663	9	*10*	*0*	5,956	32	*28*	*0*	7,922	5	*3*	*0*	19,629	5	*99*	*0*
3	26,105	3	*33*	*0*	6,259	5	*0*	*0*	1,300	6	*1*	*0*	2,786	22	*0*	*0*	3,925	1	*1*	*0*	6,423	2	*3*	*0*
4	15,323	3	*1*	*0*	2,652	2	*0*	*0*	664	4	*0*	*0*	1,608	16	*0*	*0*	2,841	1	*0*	*0*	2,326	1	*0*	*0*
5	10,053	2	*0*	*0*	1,351	1	*0*	*0*	414	4	*0*	*0*	1,019	12	*0*	*0*	2,072	1	*0*	*0*	1,178	1	*0*	*0*
10	2,437	2	-	-	138	0	-	-	159	4	-	-	189	4	-	-	325	0	-	-	64	0	-	-
15	1,211	2	-	-	29	0	-	-	89	2	-	-	40	2	-	-	23	0	-	-	5	0	-	-
20	865	2	-	-	13	0	-	-	64	2	-	-	13	0	-	-	4	0	-	-	1	0	-	-
25	774	1	-	-	7	0	-	-	33	1	-	-	6	0	-	-	0	0	-	-	0	0	-	-

3; *Collaborative* at 8. Interestingly, the *Web* category does not exhibit such a drastic drop, suggesting a characteristic resilience in the length of its SD events.

Additionally, when examining SD events identified by delay-jitter outliers compared to those identified by delay outliers alone, we observe similar patterns. At `MIN_SEQ_LEN` = 1, the number of SD events is slightly higher, attributed to shorter but more frequent sequences disrupted by periods of low jitter, leading to a decrease in overall SD coverage. As the required sequence length increases, the count of SD events sharply declines, necessitating longer sequences for classification as an SD event. With IQR analysis, by `MIN_SEQ_LEN` = 10, we observe coverage levels previously seen only at lengths of 25, with most categories falling below 5% except for *Network*, which maintains over 10% coverage. Conversely, for Z-score analysis incorporating jitter requirements, no SD sequences are identified in any category beyond `MIN_SEQ_LEN` = 5. Indeed, by `MIN_SEQ_LEN` = 4, only one SD event remains, and even at lengths of 2 and 3, only a handful of SD events are present. This suggests that a dual requirement for high delay and high jitter may be overly restrictive for identifying SD events in this context.

We have made the code for the detailed analysis discussed in this section, including a comprehensive set of plots, available as digital artifacts [19].

D. OUTLIER THRESHOLDS FOR RELIABLE SD DETECTION

For setting the outlier identification threshold, we opt for the Z-Score method, which provides a more conservative estimate, resulting in lower false positive rates. To determine the Minimum Sequence Length for SD events, we adopt an empirical approach, selecting a unique sequence length for each application category that reduces the SD coverage rate to below 10%. These thresholds are highlighted in Table 5 in red, indicating the first instance where coverage percentages drop below this critical threshold. The specific values for

TABLE 7. Minimum Sequence Lengths for SD Event Identification per Application Category.

Application Category	MIN_SEQ_LEN
Web	6
Cloud	3
Download	2
Network	2
Collaborative	8
Social Network	2

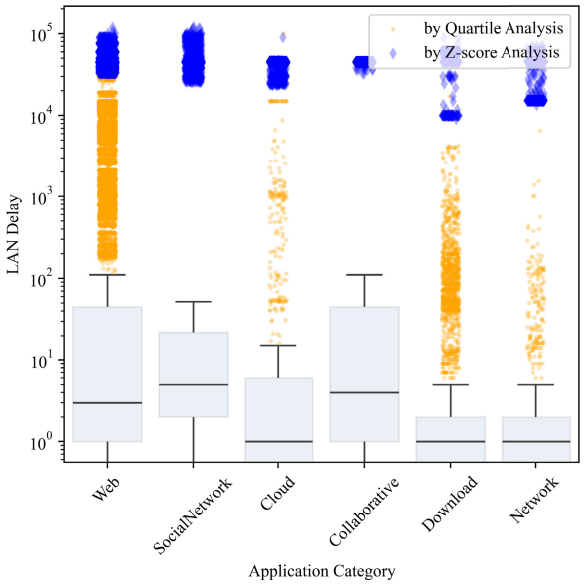


FIGURE 10. Distribution of the delay composition of SD events.

each category are summarized in Table 7, aiming to further minimize false positive SD events.

These sequence lengths correlate with the noticeable declines in SD coverage discussed earlier in Section V-C. The selected LAN delays for these thresholds are depicted in Fig. 10, which contrasts the LAN delay points identified in each application category by the most stringent IQR analysis (requiring a minimum of 25 consecutive high LAN delay and jitter) against those identified by the chosen Z-score method. This comparison illustrates that while IQR analysis tends to identify many comparatively minor delays as part of prolonged SD events, Z-Score analysis selectively identifies more extreme values by default, emphasizing its stricter criteria.

E. EFFECTIVENESS OF Z-SCORE ANALYSIS IN IDENTIFYING SD EVENTS

In this section, we analyze various statistics of the SD events identified using the chosen Z-score method. Fig. 11 presents the distribution of SD event counts across flows. The *y*-axis, represented on a logarithmic scale, indicates the number of flows, while the *x*-axis shows the number of SD events, ranging from 0 to over 14. The majority of flows either have

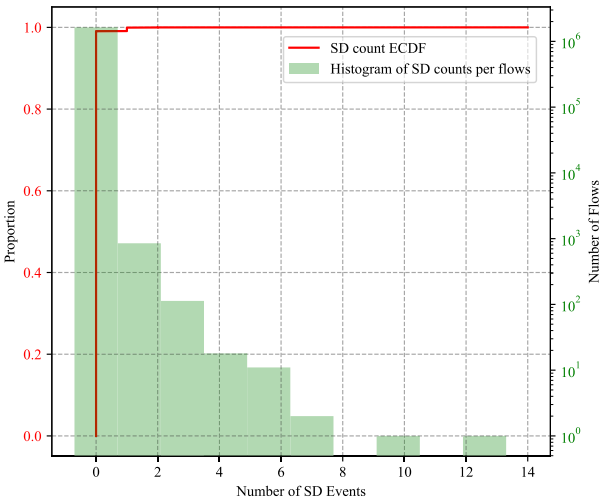


FIGURE 11. Distribution of number of SD events across flows.

no SD events or only a few, with an ECDF plot revealing that approximately 99% of flows contain at most one SD event, making multiple events within a single flow exceedingly rare.

Table 8 details the number of flows, number of SD events, and the number of flows with at least one SD event across all application categories, alongside the proportion of flows with SD events relative to the total number of flows. Consistent with Fig. 11, only a small fraction of flows contain multiple SD events, as indicated by the slight difference between *Total SD Events* and *Flows with SD*. The proportion of flows experiencing SD remains below 1% for all categories, marginally exceeding this threshold only in the *Web* category, which accounts for the majority of identified SD events.

TABLE 8. Distribution of SD Events Across Application Categories.

	Total SD Events	Flows with SD	Proportion with SD
Web	13,404	12,450	1.2434%
Social Network	1,933	1,746	0.8620%
Cloud	745	653	0.3844%
Collaborative	219	215	0.2925%
Download	240	208	0.2715%
Network	116	116	0.1062%

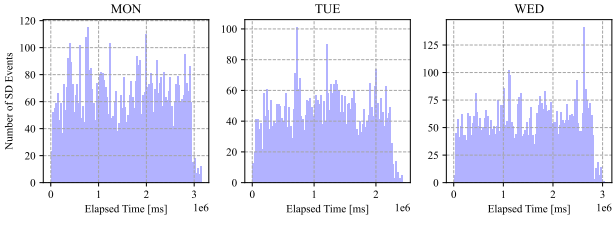


FIGURE 12. Distribution of number of SD events across flows.

Fig. 12 illustrates the temporal distribution of SD events, with the *x*-axis displaying time in milliseconds and the *y*-axis depicting the number of SD events at each time

point. This plot aims to determine whether SD events are evenly distributed over time or if certain periods contain higher concentrations of events. Notable spikes and valleys in the data suggest moments where SD event counts were significantly higher or lower, with Monday showing frequent fluctuations, but the most intense activity observed on Wednesday just before the measurement period concluded. This could indicate a specific time when network disturbances were more pronounced, affecting multiple flows. Additionally, a common trend on all days is a marked decrease in SD events towards the end of the measurement period, possibly reflecting characteristics of the measurement timeframe or the impact of the measurement process itself, such as truncating ongoing flows.

VI. GENERALIZABILITY

With the insights gained from our analysis at *Location 2*, in this section, we extend the Z-score analysis for SD identification to *Location 1* and *Location 3*. This extension aims to examine the consistency of delay characteristics and SD events across different locations, thereby assessing the generalizability of our approach. For each location, we calculate Z-scores for delay and jitter across all samples, identifying outliers where $Z > 3$. Following the established methodology from *Location 2*, we search for sequences of SD events initiated by outliers in delay and jitter, where high delay behavior persists. We then determine the smallest `MIN_SEQ_LEN` that reduces the coverage of these events below 10% for each category, comparing these values with those derived from *Location 2*. The findings and their implications for generalizability are detailed in Section VI-A.

To validate our approach, we apply the same analytical steps to the remaining holdout portion of the dataset, which includes flow data from the last two days of the experiment (Thursday and Friday). These days were chosen because they exhibited similar flow-arrival rates and data rates as observed previously (see Fig. 2). By analyzing the outcomes, we aim to confirm the consistency of the observed patterns and validate the applicability of our defined methodology during these periods. The results of this validation process are explored in Section VI-B.

TABLE 9. Count of LAN delays per Application Category for All Locations.

Application Category Name	Location 1	Location 2	Location 3
Web	3,453,780	9,016,470	2,017,416
Social Network	490,886	1,797,216	555,845
Cloud	330,038	1,177,916	197,503
Collaborative	256,976	743,453	109,425
Download	143,806	440,421	110,501
Network	109,343	264,156	38,057
Count of all delay samples	4,784,829	13,439,632	3,028,747

A. CROSS-LOCATION VALIDATION OF SERVICE DEGRADATION

To ensure consistency in our analysis, we continue to focus on the same application categories as those analyzed in data from *Location 2*. Fig. 13 compares the flow counts across the three locations, from which we find that only the *Web* and *Social Network* categories at *Location 1* and *Location 3* meet the 50,000 flow count threshold set in our original analysis. Lower flow counts in the other categories may influence SD event characteristics, skewing them toward the behavior observed in fewer flow records, which may not be as representative as those derived from higher flow counts. Despite these limitations, we proceed with the analysis for all previously chosen categories—*Web*, *Social Network*, *Cloud*, *Network*, *Download*, and *Collaborative*—keeping in mind that the results for categories with lower sample counts may be less reliable. By examining the differences in SD event requirements between categories with varying flow counts, we aim to assess whether the 50,000 flow threshold is justifiable or could potentially be lowered.

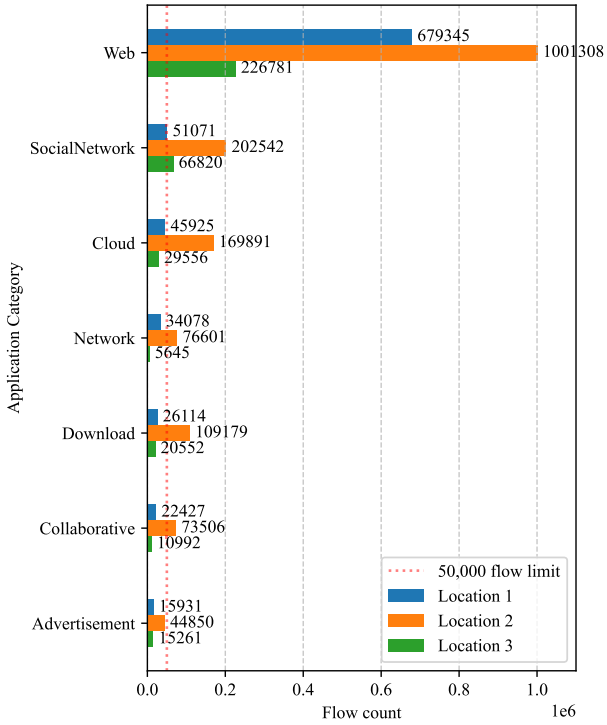
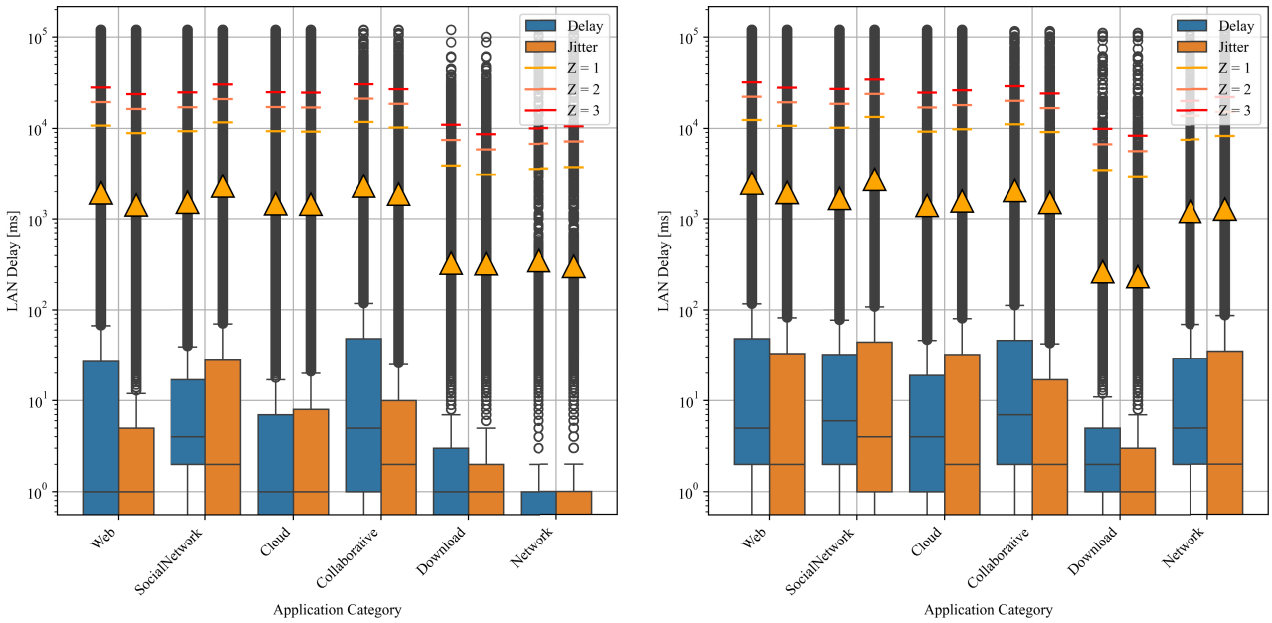


FIGURE 13. Application category cardinality for all locations.

The LAN delay counts across all application categories (see Table 9) exhibit similar patterns to the flow count statistics, albeit falling short of the delay counts observed at *Location 2*. Notably, *Location 3* shows significantly lower counts compared to *Location 1*, except for the *Social Network* category.

Fig. 14 and Table 10 depict the delay and jitter distributions for *Location 1* and *Location 3*, revealing patterns virtually identical to those at *Location 2*. The distributions across application categories maintain similar relative delays, with only minor differences: at *Location 1*, *Web* delays occasionally dip below 1 millisecond, and at *Location 3*, the



(a) **Location 1.** (b) **Location 3.**

FIGURE 14. Distribution of LAN delay and jitter for *Location 1* and *Location 3*.

TABLE 10. Mean and standard deviation values for each application category.

(a) **Location 1**

Metric	Web	Social Network	Cloud	Collaborative	Download	Network
μ_{delay}	1,945.08	1,527.17	1,475.12	2,314.52	329.43	351.44
σ_{delay}	8,761.14	9,421.93	7,836.64	9,421.93	3,551.12	3,205.05
μ_{jitter}	1,432.03	2,314.18	1,454.34	1,885.77	323,33	303.34
σ_{jitter}	7,430.50	9,335.86	7,767.23	8,353.78	2,764.91	3,382.51

(b) **Location 3**

Metric	Web	Social Network	Cloud	Collaborative	Download	Network
μ_{delay}	2,495.82	1,684.57	1,212.67	2,070.17	264.63	1,212.67
σ_{delay}	9,888.79	8,474.79	7,777.25	9,001.14	3,203.27	6,253.60
μ_{jitter}	1,969.48	2,739.77	1,580.08	1,520.29	233,12	1,286.55
σ_{jitter}	8,673.23	10,574.94	8,197.80	7,572.17	2,683.39	6,879.52

75th-percentile of *Network* delays stay below 1 ms, with only outliers exceeding this threshold. These observations suggest that while lower flow counts might slightly bias the results towards lower delays, the fundamental characteristics of delay distribution are preserved, even with significantly reduced data volumes.

Interestingly, *Location 3* experiences consistently higher delay values across all categories, with the 25th-percentile values increasing by several milliseconds, leading to a broader distribution in jitter. This pattern could reflect the unique network traffic characteristics of *Location 3*, possibly due to its more remote location compared to the other sites, which are within the same municipality. Despite these variances, the higher percentiles, mean values, and Z-scores show comparable characteristics, affirming the overall pattern consistency.

To delve deeper into longer SD events characterized by consecutive high delays, we replicate the specific scenario analyzed at *Location 2*. In this scenario, an SD event begins with an outlying delay and jitter (Z-score $>$ 3), with the high delay persisting throughout the event. We progressively increase the minimum sequence length required to qualify as an SD event, analyzing both the number and coverage percentage of these events. The results of this extended analysis are presented in Table 11.

Although the total counts of SD events decrease significantly when stricter criteria are applied, the coverage percentages exhibit a decay pattern very similar to that observed at *Location 2*, differing by only a few percentage points. Furthermore, when examining key thresholds–specifically, points where coverage falls below 10% and where significant drops in coverage occur—we identify the same critical sequence lengths as those previously established in Table 7.

This consistency across different locations not only confirms that extreme traffic patterns leading to SD events are comparable, if not identical, across the three locations,

TABLE 11. SD event count and coverage statistics for different minimum sequence lengths with `require_jitter_for_sequence` option turned *OFF* using the Z-Score identification method.

(a) Location 1

MIN_SEQ_LEN	Web		Cloud		Download		Network		Collaborative		Social Network	
	count	cov.%	count	cov.%	count	cov.%	count	cov.%	count	cov.%	count	cov.%
1	43,057	24	6,602	27	856	15	405	16	6,017	32	5,272	18
2	16,366	18	1,901	13	35	8	35	6	1,761	19	611	6
3	12,642	17	218	4	24	7	30	6	671	13	467	6
4	10,893	16	176	3	19	7	28	6	614	13	384	5
5	8,619	14	115	3	18	7	25	5	571	12	344	5
6	4,213	9	77	2	17	7	10	3	517	11	220	4
7	3,274	8	58	2	17	7	9	3	474	11	143	3
8	1,974	6	44	2	17	7	4	1	66	2	131	3
9	1,707	5	43	2	17	7	3	1	50	2	124	3
10	1,577	5	41	2	17	7	3	1	50	2	109	2

(b) Location 3

MIN_SEQ_LEN	Web		Cloud		Download		Network		Collaborative		Social Network	
	count	cov.%	count	cov.%	count	cov.%	count	cov.%	count	cov.%	count	cov.%
1	27,154	24	3,774	25	825	14	529	21	1,960	32	6,293	19
2	10,213	17	894	11	30	7	68	9	695	23	501	5
3	8,005	16	113	3	15	6	60	9	382	19	376	4
4	7,072	15	90	3	11	5	51	8	329	18	313	4
5	5,667	13	62	3	10	5	37	6	309	17	266	4
6	2,804	9	41	2	9	5	7	2	278	16	201	3
7	2,202	8	38	2	9	5	6	2	246	15	132	2
8	1,374	6	36	2	8	5	5	1	58	5	124	2
9	1,164	5	35	2	8	5	1	0	46	5	120	2
10	1,037	5	33	2	8	5	1	0	43	5	117	2

but it also validates the robustness of our analytical approach. Moreover, it suggests that this methodology can effectively be applied to application categories with significantly lower flow counts, reinforcing the reliability and appropriateness of our chosen methods for analyzing SD events.

B. SERVICE DEGRADATION VALIDATION VIA HOLDOUT SET

For our SD event analyses thus far, we have utilized data from the first three measurement days designated as training data. To extend our validation, we leverage the data from the last two days as testing data in a model training-testing scenario. This test data allows us to validate our SD event identification approach by comparing results across different days, similar to our analyses across various locations. By tallying the results and quantifying differences in thresholds and `MIN_SEQ_LEN`, we aim to confirm the robustness and consistency of our methodology.

We have chosen not to include delay and jitter distribution plots for the test data from the three locations in this document due to their high similarity to the training data distributions. However, to ensure transparency and facilitate further research, we provide the code for our detailed analysis, including a comprehensive set of plots, as digital artifacts [19]. Upon examining the delay and jitter distributions in the test data, we observe highly similar patterns to those in the training data, albeit with slight variations in median values. Notable changes between the training and testing datasets for all locations include:

- *Location 1* displays slightly higher *Web* delays paired with lower jitter, an increase in lower-end *Social Network* delays, and lower delays in the *Cloud*, *Download*, and *Network* categories, along with correspondingly lower jitter distributions.
- *Location 2* shows virtually identical distributions with a broader spread in jitter within the *Network* category.
- *Location 3* retains the higher delay and jitter distribution observed during the training phase due to its geographical distance, yet maintains highly similar patterns, with the notable exception of increased *Download* and *Network* delays and jitter.

These observations suggest that our methodology yields consistent and reliable results across different testing conditions, further validating the soundness of our chosen analytical approach.

Table 12 illustrates the relative differences (expressed as percentages) between the measured mean and standard deviation across all locations, upon which the Z-Score method bases its outlier detection. These differences are color-coded according to their magnitude, with larger deviations highlighted in darker shades of red.

Examining the mean and standard deviation statistics for delay and jitter, we observe the most significant deviation in the delay mean: a 45% decrease in the *Network* category at *Location 2*. The most substantial change in jitter mean occurred in the *Collaborative* category at *Location 1*, where the average jitter was over 75% higher compared to the training data. The greatest changes in standard deviation were 31% for delay and 42% for jitter, both observed in the *Collaborative* category at *Location 1* and *Location 3*, respectively. The *Collaborative* category generally exhibited the most variability in delay and jitter changes between the training and test data, with other notable variations in the *Social Network* and *Cloud* categories. Changes in other categories were slight or negligible.

These differences were fairly evenly distributed across the three locations, with *Location 1* experiencing the most significant changes. When testing for the optimal minimum

TABLE 12. Mean and standard deviation for each application category in the test data given as a relative percentage difference to training data.

	Metric	Web	Social Network	Cloud	Collaborative	Download	Network
Location 1	μ_{delay} (Δ%)	20.92	-9.67	-19.15	33.00	-4.63	-4.49
	σ_{delay} (Δ%)	9.41	-3.07	-4.55	30.94	-1.29	-0.08
	μ_{jitter} (Δ%)	24.28	-7.50	-27.68	78.74	-7.22	-1.49
	σ_{jitter} (Δ%)	10.51	-2.09	-7.17	52.98	-3.57	1.91
Location 2	μ_{delay} (Δ%)	7.14	44.51	14.31	-45.05	14.06	-9.12
	σ_{delay} (Δ%)	2.60	19.06	-6.62	-28.88	7.06	-6.89
	μ_{jitter} (Δ%)	-3.09	26.26	-22.06	-48.93	36.04	-8.18
	σ_{jitter} (Δ%)	-3.16	10.53	-25.57	-29.44	16.53	-6.96
Location 3	μ_{delay} (Δ%)	1.62	-29.46	-8.16	20.60	-1.69	3.61
	σ_{delay} (Δ%)	1.16	-16.57	-16.21	27.58	-1.43	2.80
	μ_{jitter} (Δ%)	2.87	-30.45	53.57	43.31	-7.71	-3.24
	σ_{jitter} (Δ%)	2.39	-17.53	15.90	41.87	-4.16	-0.14

sequence length using the same parameters, the exact same thresholds as those determined from the training data were obtained. Table 13 details the coverage percentages for all three locations and all six application categories analyzed. Despite slight variations in the exact SD event coverage percentages, the `MIN_SEQ_LEN` for all categories was consistent across all locations, with the significant decrease in coverage dipping below 10% at matching minimum sequence lengths. The distribution of these results was identical to that observed in the training data (see Table 7), confirming the reliability and consistency of our methodology across different datasets.

VII. DISCUSSION

A. CAUSES AND TRIGGERING EVENTS OF SD

SD in network traffic is influenced by a variety of factors, which can broadly be categorized into network conditions, hardware limitations, and application behaviors. Understanding these triggers is crucial for diagnosing and mitigating SD, and our PIAT analysis methodology provides valuable insights into detecting and characterizing these events.

1) NETWORK CONGESTION

One of the primary causes of SD is network congestion, where the demand for bandwidth exceeds the network's capacity to handle it. This leads to increased packet loss, higher latency, and jitter. Research has shown that congestion can be caused by both consistent high traffic volumes and sudden traffic spikes, often exacerbated by applications with high bandwidth requirements. In our PIAT analysis, congestion often manifests as increased variability in packet inter-arrival times, particularly during peak usage hours.

2) HARDWARE LIMITATIONS

Edge devices and network infrastructure often have finite processing and bandwidth capabilities. When these devices reach their performance limits, they can become bottlenocks, causing delays and packet loss. This is particularly pertinent in residential and small business networks, where the hardware might not be equipped to handle large volumes of data traffic efficiently. Our focus on LAN-side latency allows us to isolate these hardware-related issues from broader network problems, as evidenced by consistent patterns of increased PIAT values across multiple flows.

3) APPLICATION BEHAVIOR

Certain applications, especially those requiring real-time data transmission like video streaming, online gaming, and VoIP, can place significant demands on network resources. These applications are sensitive to latency and jitter, and their use can lead to observable SD patterns under suboptimal network conditions. In our labeled dataset, we have observed distinctive PIAT patterns associated with different types of applications, allowing for more nuanced SD detection.

To address these issues, network administrators and service providers must consider both the optimization of network infrastructure and the management of traffic loads. Understanding the specific causes and conditions that lead to SD allows for more targeted interventions and improved network performance. Our work contributes to this understanding by providing a detailed, empirical basis for identifying and characterizing SD events in real-world network traffic.

B. FACTORS IN TRANSPORT LAYER CONTRIBUTING TO SD

The Transport layer, particularly TCP, plays a critical role in ensuring reliable data transmission across networks. However, TCP's mechanisms can sometimes contribute to SD, especially under certain network conditions.

1) CONGESTION CONTROL MECHANISMS

TCP uses congestion control algorithms, such as TCP Reno and TCP Cubic, to manage network congestion by adjusting the rate of data transmission. While these algorithms are designed to prevent network collapse, they can also lead to increased latency and reduced throughput during congestion events. For instance, TCP's slow start and congestion avoidance phases can delay data transmission, especially in high-latency networks.

2) RETRANSMISSION STRATEGIES

TCP's reliance on retransmission for error recovery, using mechanisms like retransmission timeouts (RTOs) and duplicate acknowledgments, can increase latency, particularly in lossy networks. The time taken to detect packet loss and retransmit lost packets can cause delays, contributing to jitter and inconsistent throughput.

3) FLOW CONTROL

TCP's flow control mechanisms, primarily through the advertised window size, regulate the amount of data that can be sent before requiring an acknowledgment. While this prevents the sender from overwhelming the receiver, it can also limit throughput in cases where the receiver's window

size is constrained, leading to suboptimal data flow and potential SD.

Our PIAT-based approach to SD detection can help identify instances where such protocol-level issues are contributing to degraded service quality. This understanding can guide network administrators in fine-tuning protocol parameters or considering alternative protocols for specific use cases.

C. LIMITATIONS AND FUTURE WORK

While our study offers valuable insights into latency-induced SD in LAN environments, particularly within a university dormitory setting, it is important to recognize its limitations and outline areas for future research.

1) FOCUS ON LAN TRAFFIC

Our research was conducted using LAN data from a university dormitory, which may exhibit network traffic characteristics distinct from other environments such as home networks, small-office/home-office settings, or large enterprise networks. These environments often encounter different resource constraints that could affect the detection and analysis of SD events differently.

We hypothesized that our dataset provides a hybrid representation of residential and institutional network traffic–merging casual internet usage by residents with university-related activities such as accessing cloud resources or remote sessions. By leveraging data that potentially spans multiple domains, our aim was to enhance the robustness and applicability of our SD event identification method to both home and enterprise environments. However, this assumption is speculative and requires further empirical validation. Future research could therefore explore adapting our methodology to broader network contexts such as WANs or MANs, as well as testing its applicability in large enterprise settings where SD detection could yield benefits in terms of financial and resource management.

2) SIMULATION-BASED EXTENSION

While our study utilized real-world data, future research could also benefit from simulation-based studies that model various network types (*e.g.*, WAN, MAN, wireless) using protocols like TCP. These simulations could provide insights into the broader applicability of our SD detection methods across diverse network environments.

3) COMPARATIVE ANALYSIS

Due to the novelty of our approach in addressing latency-induced SD in LAN settings, we had limited opportunities for direct comparisons with existing methodologies. As the field evolves, future studies could conduct comparative analyses with emerging approaches to SD detection, enhancing the robustness and credibility of our findings.

4) TEMPORAL VARIATIONS AND DATA COLLECTION

Our study evaluated the generalizability of our findings by applying the methodology in different locations (various dormitories within the university) and at various times. Despite some variations, the data confirmed the model's validity within the current settings. Our findings also demonstrated that the methodology is effective even with considerably lower flow counts, indicating potential applicability in smaller environments.

However, to rigorously confirm the robustness of our method, future work might extend to other settings to provide a more comprehensive understanding of SD patterns across varying network loads. For example, collecting data from more diverse locations (*e.g.*, different universities, residential areas, or corporate environments) and at different times (*e.g.*, off-peak periods such as vacation time) could help validate the broader generalizability of our findings.

TABLE 13. SD event coverage statistics (*cov.%*) for different minimum sequence lengths with `require_jitter_for_sequence` option turned *OFF* using the Z-Score method run on the test data (*L1*, *L2* and *L3* stand for *Locations 1*, *2* and *3* respectively).

MIN_SEQ_LEN	Web			Cloud			Download			Network			Collaborative			Social Network		
	L1	L2	L3	L1	L2	L3	L1	L2	L3	L1	L2	L3	L1	L2	L3	L1	L2	L3
1	22	24	26	27	28	24	14	19	12	18	20	18	31	33	35	18	19	21
2	17	18	19	14	14	11	8	3	3	4	7	8	20	21	26	6	6	7
3	16	17	17	4	5	4	6	3	2	2	5	7	15	16	23	5	5	6
4	15	16	16	3	4	4	6	3	2	2	5	6	14	15	22	5	5	6
5	13	14	14	3	3	3	6	2	1	1	3	5	14	14	22	4	4	5
6	9	9	8	2	3	2	6	2	1	1	1	1	13	13	20	4	4	4
7	8	8	7	2	2	2	6	2	1	1	1	1	12	13	19	3	3	4
8	6	6	5	2	2	2	6	2	1	0	1	1	2	2	5	3	3	3
9	6	6	5	2	2	2	6	2	1	0	0	0	2	2	5	3	3	3
10	5	6	5	2	2	2	6	2	1	0	0	0	2	2	4	3	3	3

5) NETWORK TOPOLOGY CONSIDERATIONS

While our study provides valuable insights into SD in typical LAN configurations, it does not extensively explore the impact of various network topologies or imbalanced end-user connections to Layer-2 switches. Future research could investigate how different network layouts and topologies affect SD occurrences and detection, potentially leading to more nuanced and effective detection strategies.

6) IMPACT ON USER QOE

Our empirical definition of SD events is based on the occurrence of prolonged incidents characterized by statistically extreme delays and jitter, aiming to identify events that significantly impact perceived network service quality. However, we did not directly assess whether these identified SD events translate to actual impacts on user QoE. Investigating this relationship would require a more controlled experiment or the integration of expert knowledge, which remains an area for future explo-

ration. Our method provides a conservative statistical framework that operates effectively in an unsupervised manner.

These limitations highlight opportunities for future research to build upon our findings, extending the applicability of our SD detection methodology across a wider range of network environments and conditions. By addressing these limitations, future studies can enhance the robustness, applicability, and accuracy of SD detection techniques.

VIII. CONCLUSION

This study has explored the identification and analysis of service degradation events using LAN data from a university dormitory setting, aiming to develop a methodology that could be applicable to a variety of network environments. Our approach, centered on the detection of statistically significant deviations in network performance (specifically, extreme delays and jitter), has demonstrated potential applicability not only in similar institutional settings but also in residential and possibly enterprise networks.

The robustness of the methodology was validated by applying it across different locations and at different times, with results showing minimal deviation, thus underscoring the effectiveness of the SD event identification process. This consistency highlights our model's potential as a reliable tool for network administrators to preemptively address and manage network service quality issues. While our findings offer promising directions for network service management and SD detection, they also pave the way for subsequent studies to refine and expand on the groundwork laid here.

Our future work aims to validate our method's efficacy across broader contexts and network configurations, ensuring its robustness and adaptability in diverse environments.

REFERENCES

[1] P. Casas, P. Fiadino, S. Wassermann, S. Traverso, A. D'Alconzo, E. Tego, F. Matera, and M. Mellia, "Unveiling network and service performance degradation in the wild with mPlane," *IEEE Commun. Mag.*, vol. 54, no. 3, pp. 71–79, Mar. 2016.

[2] S. Kassing, V. Dukic, C. Zhang, and A. Singla, "New primitives for bounded degradation in network service," 2022, *arXiv:2208.08429*.

[3] J. van Bloem and R. Schiphorst, *Measuring the Service Level in the 2.4 GHz ISM Band*, document TR-CTIT-11-27, Centre for Telematics Inf. Technol. (CTIT), Enschede, The Netherlands, Dec. 2011.

[4] D. Canastro, R. Rocha, M. Antunes, D. Gomes, and R. L. Aguiar, "Root cause analysis in 5G/6G networks," in *Proc. 8th Int. Conf. Future Internet Things Cloud (FiCloud)*, Aug. 2021, pp. 217–224.

[5] A. Bremler-Barr, E. Cohen, H. Kaplan, and Y. Mansour, "Predicting and bypassing end-to-end Internet service degradations," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 6, pp. 961–978, Aug. 2003.

[6] A. Abdelkefi, Y. Jiang, B. E. Helvik, G. Biczók, and A. Calu, "Assessing the service quality of an Internet path through end-to-end measurement," *Comput. Netw.*, vol. 70, pp. 30–44, Sep. 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128614001698

[7] H. Wu, Y. Liu, S. Ni, G. Cheng, and X. Hu, "LossDetection: Real-time packet loss monitoring system for sampled traffic data," *IEEE Trans. Netw. Service Manage.*, vol. 20, no. 1, pp. 30–45, Mar. 2023.

[8] C. Hardegen, B. Pfülb, S. Rieger, and A. Gepperth, "Predicting network flow characteristics using deep learning and real-world network traffic," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 4, pp. 2662–2676, Dec. 2020.

[9] L. Traini and V. Cortellessa, "DeLag: Using multi-objective optimization to enhance the detection of latency degradation patterns in service-based systems," *IEEE Trans. Softw. Eng.*, vol. 49, no. 6, pp. 3554–3580, Jun. 2023.

[10] V. Cortellessa and L. Traini, "Detecting latency degradation patterns in service-based systems," in *Proc. ACM/SPEC Int. Conf. Perform. Eng.* New York, NY, USA: Association for Computing Machinery, Apr. 2020, pp. 161–172.

[11] A. Amaral, A. Armendariz, S. Erramuspe, J. Joskowicz, and M. Liu, "Prediction of video quality degradation on a cloud gaming platform," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, Jun. 2023, pp. 1–5.

[12] Y. Li, X. Zhang, C. Cui, S. Wang, and S. Ma, "Fleet: Improving quality of experience for low-latency live video streaming," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 9, pp. 5242–5256, Sep. 2023.

[13] B. Xian, W. Hou, Y. Zong, J. Wu, and L. Guo, "Prediction-based and provident service degradation in optical data center networks," in *Proc. 15th Int. Conf. Opt. Commun. Netw. (ICOCN)*, Sep. 2016, pp. 1–3.

[14] W. Hou, Z. Ning, L. Guo, and M. S. Obaidat, "Service degradability supported by forecasting system in optical data center networks," *IEEE Syst. J.*, vol. 13, no. 2, pp. 1514–1525, Jun. 2019.

[15] L. Pons, J. Feliu, J. Sahuquillo, M. E. Gómez, S. Petit, J. Pons, and C. Huang, "Cloud white: Detecting and estimating QoS degradation of latency-critical workloads in the public cloud," *Future Gener. Comput. Syst.*, vol. 138, pp. 13–25, Jan. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X22002734

[16] J. Zhu, L. Zhao, J. Zhou, W. Ye, and F. Xiao, "Service degradation-tolerated online user allocation in edge computing," *IEEE Trans. Services Comput.*, vol. 17, no. 4, pp. 1806–1819, Jul. 2024.

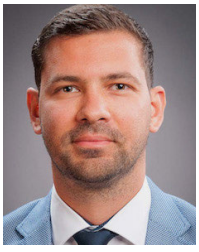
[17] Z. Aouini and A. Pekar, "NFStream: A flexible network data analysis framework," *Comput. Netw.*, vol. 204, Feb. 2022, Art. no. 108719.

[18] L. Deri, M. Martinelli, T. Bujlow, and A. Cardigliano, "NDPI: Open-source high-speed deep packet inspection," in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Aug. 2014, pp. 617–622.

[19] FlowFrontiers. (2024). *Servie Degradation Detection—Digital Artifacts*. [Online]. Available: http://github.com/FlowFrontiers/ServDeg-Dataset

[20] M.-J. O. Saarinen and J.-P. Aumasson, *The BLAKE2 Cryptographic Hash and Message Authentication Code (MAC)*, document RFC 7693, Nov. 2015. [Online]. Available: https://www.rfc-editor.org/info/rfc7693

BALINT BICSKI (Graduate Student Member, IEEE) received the M.Sc. degree from the Department of Networked Systems and Services, Budapest University of Technology and Economics, Hungary, in 2024. He is currently a Junior Researcher at CUJO AI and a Research Fellow with Budapest University of Technology and Economics. His research interests include network and service management, software-defined networking, network security, and Industry 4.0.



ADRIAN PEKAR received the Ph.D. degree from the Technical University of Kosice, Slovakia, in 2014. He is currently an Associate Professor with the Department of Networked Systems and Services, Budapest University of Technology and Economics, Hungary. Before joining academia in Hungary, he gained valuable experience through research, teaching, and engineering roles across Slovakia and New Zealand. His research interests include a wide array of topics, including network and service management, software-defined networking, network function virtualization, network programmability, machine learning, and data science.