A Packet Loss Monitoring System for In-Band Network Telemetry: Detection, Localization, Diagnosis and Recovery

Lizhuang Tan, *Student Member, IEEE*, Wei Su, Wei Zhang, *Member, IEEE*, Huiling Shi, Jingying Miao, and Pilar Manzanares-Lopez

***Abstract*—Network measurement provides rich data for network monitoring, control, and management. In-band network telemetry (INT) is a new network measurement technology that uses normal data packet to collect network information hop-by-hop. However, the design and implementation of INT protocol cannot do anything about packet loss: (1) The end-to-end telemetry mechanism makes INT unable to detect packet loss; (2) Since data packets may be lost due to various reasons, INT telemetry information will inevitably be lost. In summary, INT system by itself is unreliable. Incomplete telemetry data will seriously affect the performance of upper-layer network telemetry applications. In this paper, we present our successful experience in INT packet loss monitoring. We design, implement, and open source a powerful packet loss monitoring system for INT, called LossSight. The functions of LossSight include the detection of packet loss events, the deduction of the time and location of the losses, the diagnose of the root cause of the losses, and the recovery of the lost INT information. Experiment results show that LossSight provides excellent performance and extremely low overhead, including detection accuracy and diagnostic precision close to 100%, and detection latency of just milliseconds. In particular, LossSight uses a generative adversarial network to recover lost telemetry information, with excellent accuracy and reliability. LossSight has been running stably in the supercomputing interconnection environment of the National Supercomputing Center in Jinan. We suggest that all INT applications that require reliable telemetry information should be implemented based on LossSight.**

***Index Terms*—In-band network telemetry, network measurement, alternate-marking performance measurement, loss detection, loss localization, data completion, generative adversarial network.**

Manuscript received April 14, 2021; revised September 3, 2021 and October 31, 2021; accepted October 31, 2021. Date of publication November 3, 2021; date of current version December 9, 2021. The work of L. Tan, W. Su, and J. Miao was supported in part by the National Key R&D Program of China (No. 2018YFB1800305), the Fundamental Research Funds for the Central Universities of China (No. 2020YJS013), and the Open Project of Shandong Provincial Key Laboratory of Computer Networks (No. SDKLCN-2020-04). This work of W. Zhang and H. Shi was supported in part by the Pilot Project for Integrated Innovation of Science, Education and Industry of Qilu University of Technology (Shandong Academy of Sciences) (No. 2020KJC-ZD02), and the Shandong Provincial Natural Science Foundation (No. ZR2019LZH013 and No. ZR2020LZH010). The associate editor coordinating the review of this article and approving it for publication was R. Birke. *(Corresponding authors: Wei Su; Wei Zhang.)*

Lizhuang Tan, Wei Su, and Jingying Miao are with the National Engineering Laboratory for Next Generation Internet Interconnection Devices, School of Electronics and Information Engineering, Beijing Jiaotong University, Beijing 100044, China (e-mail: lzhtan@bjtu.edu.cn; wsu@bjtu.edu.cn; 19120098@bjtu.edu.cn).

Wei Zhang and Huiling Shi are with the Shandong Provincial Key Laboratory of Computer Networks, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250306, China, and also with the State Key Laboratory of High-End Server and Storage Technology, Inspur, Jinan 250014, China (e-mail: wzhang@sdas.org; shihl@sdas.org).

Pilar Manzanares-Lopez is with the Department of Information Technologies and Communications, Universidad Politécnica de Cartagena, 30202 Cartagena, Spain (e-mail: pilar.manzanares@upct.es).

Digital Object Identifier 10.1109/TNSM.2021.3125012

I. Introduction

NETWORK telemetry is a new network measurement method that can quickly collect and integrate network status data for monitoring the quality of networks and services [1]. Network telemetry can be divided into in-band network telemetry (INT) and out-band network telemetry (ONT). The main characteristic of INT can be summarized as using normal data packets to carry the status information of switching devices, instead of using specifically dedicated packets. INT allows us to obtain an accurate hop-by-hop view of the status of end-to-end network services. This potential has attracted the attention of academia and industry. There are two representative solutions to achieve in-band network telemetry, INT led by P4 [2] and IOAM led by IETF [3].

The basic operation of INT is shown in the left part of Figure 1: When a data packet enters the first switching device, this node (the INT Source Node) inserts an INT header including the INT telemetry instruction and also the collected data in the INT metadata field. Then, the packet is forwarded to the next switching device (the INT Transit Hop Node), which adds its own INT metadata according to the telemetry instruction. After passing through all the INT Transit Hop Nodes, the packet reaches the last switching device (the INT Sink Node). The INT Sink Node extracts the INT header and all INT metadata, and sends them to the Telemetry Server. In this way, the Telemetry Server can collect telemetry metadata from the data plane, including device-level ingress/egress metadata values.

Recently, numerous INT-based network measurement solutions have been proposed, offering ways of obtaining one-way delay [4], tail latency [5], available bandwidth [6], queue depth [7], network traffic [8], switch processing delay [9], QoS/SLA [10] or SFC performance [11]. In addition, advanced INT-based network controls offer network efficiency improvements, including congestion control [12], routing decisions [13], [14], abnormal detection [15], path

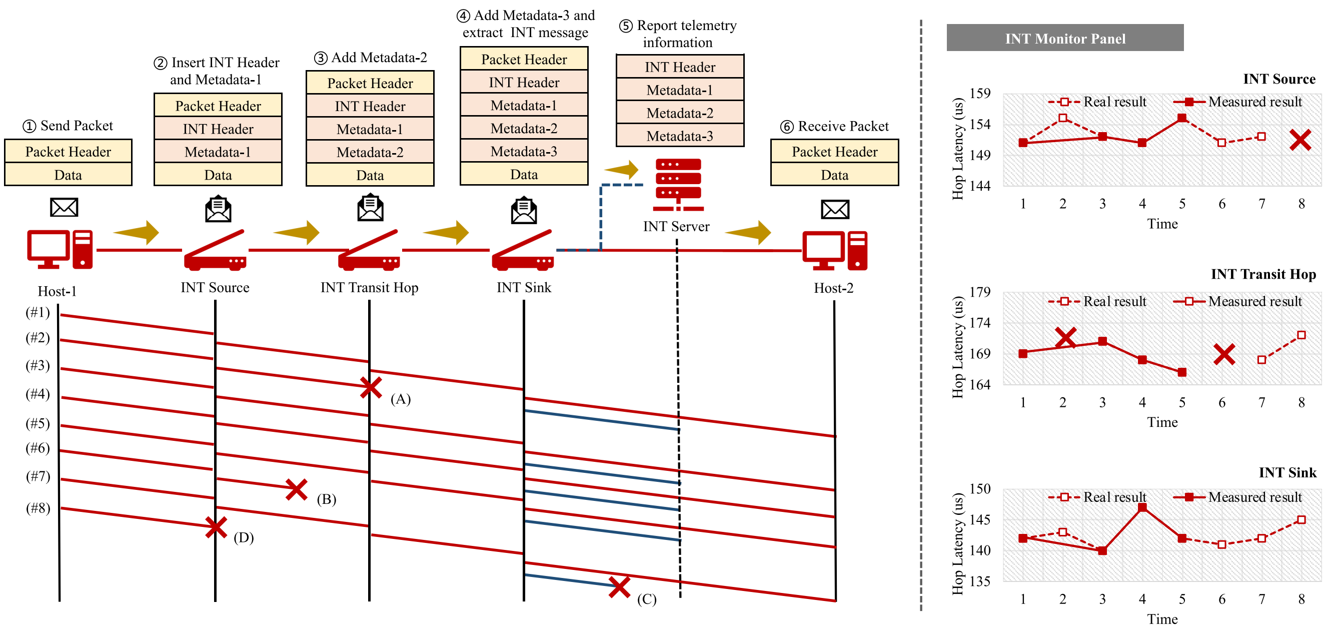


Fig. 1. In the left side, INT operation description and an example of the lost of telemetry packets. In the right side, obtained monitoring results.

tracing [5] and artificial intelligence-based network management [16], [17].

However, according to the survey [1] of above-mentioned research works and our practical experience in deploying INT, we have discovered the following facts that we would like to highlight:

1) There is no useful solution for the measurement of INT packet loss.[1] There are some high-performance packet loss measurement solutions [19], [20], [21], [22], [23], [24], [25], but none of them have been integrated into INT. It is worth noting that there is a proposal using INT to measure packet loss rate [26]. However, this solution belongs to active measurement alternatives, which are based on the injection of synthetic traffic (active probes) for monitoring tasks. Therefore, it measures the packet loss rate of probe packets instead of per-flow loss rate. It would be desirable the definition of a packet loss detection and localization solution based solely on the own INT packets.
2) Existing INT applications [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16] do not consider the impact of INT packet loss on measurement results. Although the Telemetry Server continues receiving INT reports, the telemetry information is incomplete due to the INT packet losses. Thus, telemetry information related to critical network failures is lost along with data packets. In order to prove that the loss of telemetry information does affect the performance of the upper-layer telemetry application, we evaluated the performance of anomaly detection in [15] under different packet loss rates. Results are shown in Figure 2. The Recurrent Neural Network (RNN) is implemented with Tensorflow, 128 neurons, and a single hidden layer. The mean square error is chosen for loss function, 0.001 for the learning rate and 150,000 for the iteration number. We evaluate the performance of the classification in terms of accuracy, precisions, recall and F1 score. The results show RNN model trained in an environment where loss rate is zero cannot accurately classify abnormal traffic in a lossy network. The reason is that packet loss destroys the input-data characteristics (duration, hop latency, flow latency and queue occupancy) of the RNN model. Since the INT packet losses are not reported, the RNN model is ignorant about this negative information and consequently, it cannot observe the real network status. Without any special supplementary schemes, the Telemetry Server will not be aware that telemetry flow is experiencing packet losses. Therefore, the performance of network monitoring, control, management, and optimization that use telemetry data will be affected by the incomplete telemetry information.
3) INT should have a deeper understanding of packet loss events. With the continuous improvement of the performance of network devices and links, the probability of an accidental packet loss has become very low. The occurrence of packet losses usually means that the network, and consequently the data traffic is experiencing significant problems. As shown in the right of Figure 1 (INT Monitor Panel), the Telemetry Server just receives and processes the telemetry reports of INT packets #1, #3, #4 and #5 without knowing that the other INT packets were generated but lost at some point of the

[1] The latest version of INT specification [2] divides INT into three types: INT-XD, INT-MX and INT-MD. The first two draw on the ideas of iFIT [18] and IOAM [3]. INT Metadata is reported hop-by-hop, so the collected data can be directly used for loss measurement. But for INT-MD, the working principle of reporting only by the last hop makes it uncontrollable to packet loss. In this work, we only discuss INT-MD, that is, how to measure telemetry packet loss when only Sink Node exports metadata.

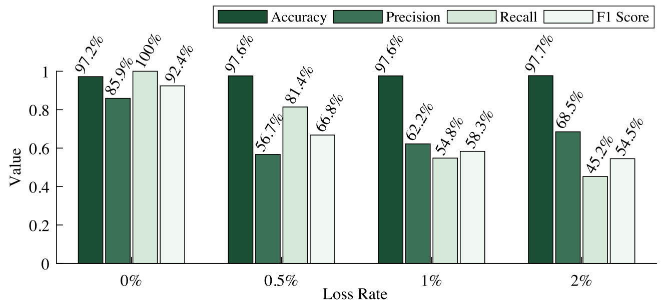


Fig. 2. Performance of the RNN model proposed in [15] under different packet loss rates.

path. Hop-by-hop information (e.g., processing delay or queue depth) and the knowledge of packet losses can be used to identify different network events, such as network congestion, network failures or attacks [12]. By accurately identifying the cause of packet loss, INT can understand network status in more detail, and adopt different processing strategies to deal with the network problems.

4) Detecting and classifying packet loss are only the basic requirements of a network monitoring system. Since telemetry information is lost along with normal data packets, an important thing for INT is to recover lost telemetry information as much as possible. If the incomplete in-band network telemetry results are recovered and fed to the upper-level applications represented by network monitoring, control and management, network administrators will get benefits that would otherwise be impossible.

In response to the above facts and needs, our goal is to make up for the deficiencies of INT in the presence of packet loss. We have customized a variety of packet loss monitoring functions for INT, including detection of packet loss events, deduction of the time and location of packet loss, analysis of the cause of packet loss, and recovery of lost INT information. These functions constitute a complete INT packet loss monitoring system, *LossSight*,[2] which helps network administrators to gain insight into packet loss events, improve network visualization and troubleshoot network failures. The detection and localization functions have been reported in previous work FindINT [28]. Compared with FindINT, LossSight improves the detection function and the localization function, and adds the diagnosis function and the recovery function.

As far as we know, this is the first work that discusses the impact of packet loss on measurement results of an INT system and proposes a complete solution mechanism for INT.

In summary, the innovations of this paper include:

1) **We propose an AM (Alternate Marking)-based INT loss detection and localization mechanism** (see Sections IV-B and IV-C). The INT node maintains a packet counter for each flow, and alternately marks the telemetry data packet while performing INT operations. Fulfilling the available standards and Internet Drafts [2], this mechanism allows INT to perceive packet loss events in actual networks. We propose two Loss Bit coding schemes, Single-bit Alternate Marking (SAM) and Multi-bit Cycle Marking (MCM). The former requires less bandwidth, and the latter offers higher accuracy.
2) **We propose an analysis mechanism for deducing the root cause of packet loss based on collected marking results** (see Section IV-D). Combining detection and localization, this mechanism can effectively diagnose the causes of packet loss such as random, congestion, and blackhole drop.
3) **We propose a recovery method based on generative adversarial network (GAN) for lost telemetry information** (see Section IV-E). Based on the incomplete telemetry result and the diagnosis of loss pattern, GAN can learn the data characteristics of different packet loss patterns and fill in with values close to original values.
4) **We implement and open source the above mechanism on P4 switch** (code available in [29]). Experiment results demonstrate that the proposed system can measure per-flow packet loss with low overhead and high accuracy. Combined with tools such as INTCollector [30], telemetry information is more complete.
5) **We publish the first in-band network telemetry packet loss dataset captured by LossSight** [31]. This dataset details INT packet loss in a real network environment, including measured telemetry metadata, loss mark, loss time, and root cause of loss. The dataset size is 9.68 MB, covering 7 different packet loss scenarios. Each scene contains about 10,000 lines of telemetry results under real traffic that lasts about 1 hour. We have achieved high detection accuracy, diagnosis accuracy and recovery performance on this dataset.

The rest of this paper is organized as follows: Section II introduces the related work of detection and recovery of packet loss. Section III presents the mathematical description of in-band network telemetry and clarifies some concepts. Section IV describes in detail the design goals and functional components of LossSight. Section V introduces the experiment results and analyzes the performance and overhead of LossSight. Section VI discusses the research direction of LossSight. Section VII summarizes this paper.

II. Related Work

Some packet loss measurement and localization schemes have been proposed, as we summarized in Table I. These methods can be divided into active and passive approaches.

OpenNetwork [19] and FlowRadar [20] are active and centralized approaches where a SDN controller continuously monitors the flows by polling involved switches. Defining an adequate polling mechanism and selecting the polling frequency are key aspects of these solutions. LossRadar [21] also captures per-flow counters at each switch at a fine time scale (e.g., tens of milliseconds), and compares the counters at two nearby hops to detect packet loss. So, LossRadar requires

[2]The demonstration video of LossSight is available in [27].

TABLE I
SUMMARY OF SOME PACKET LOSS MEASUREMENT SOLUTIONS

Solution	Year	Type	Detection	Localization	Diagnosis	Recovery	Technical Characteristics
OpenNetMon[19]	2014	Active	✓	×	×	×	A flow state measurement scheme for OpenFlow networks, which calculates per-flow loss by polling statistics from the first and last switch of each path.
Pingmesh[32]	2015	Active	✓	✓	✓	×	A data center network monitoring tool that actively sends PING, supports multiple types of packet loss localization and cause diagnosis.
FlowRadar[20]	2016	Passive	✓	✓	×	×	A packet loss capture scheme by comparing the per-flow counters of each switch.
LossRadar[21]	2016	Passive	✓	✓	✓	×	A Bloom-filter based data structure to collect traffic digests at each switch.
NetBouncer[22]	2019	Active	✓	✓	×	×	A failure localization system that leverages the IP-in-IP technique to actively probe paths in a data center network.
RINGLM[23]	2019	Active	✓	✓	×	×	A two-way SDN link-level packet loss monitoring solution, with packet loss probe structure consisting of distributed rings.
Cociglio[24]	2019	Passive	✓	×	×	×	A multipoint passive monitoring solution, which calculates the loss rate by comparing the packet count results of different domains of backbone network.
PF-PLM[25]	2020	Passive	✓	×	×	×	A per-flow SRv6 loss measurement solution based on alternate marking.
INT_DETECT[26]	2020	Active	✓	✓	×	×	An active INT solution for data center networks, which detects gray faults (packet loss) by constructing probes.
PacketScope[33]	2020	Passive	✓	✓	×	×	A network telemetry system that lets us peek inside network switches to ask a suite of useful queries about how switches modify, drop, delay, and forward packets.
VTrace[34]	2020	Passive	✓	✓	✓	×	An automatic diagnostic system for persistent packet loss over the cloud-scale overlay network.
NTC[35]	2020	Passive	×	×	×	✓	A novel neural tensor completion scheme.
MC-GPU[36]	2020	Passive	×	×	×	✓	A GPU acceleration scheme for matrix completion.
PTPmesh[37]	2021	Active	✓	✓	×	×	A cloud network monitoring (latency and packet loss) tool which uses Precision Time Protocol.
LossSight (This paper)	2021	Passive	✓	✓	✓	✓	A complete solution combining in-band network telemetry and alternate-marking performance measurement.

* ✓ means that the solution supports the function, and × means that the solution does not support the function.

accurate time synchronization of devices across the network, accompanied by a high detection latency. NetBouncer [22] and RINGLM [23] are also active measurement solutions based on the definition and injection of specific probe packets. Since the above solutions are based on polling mechanisms injecting probe packets, their performance depends on the frequency of the queries. Consequently, the balance between the acquisition of enough monitoring information and the involved network overhead is a relevant parameter [26]. INT_DETECT [26] uses INT to measure loss by constructing probes. However, INT_DETECT can only measure the packet loss experienced by the probes, but the service. In general, the active approaches are not applicable to INT-MD. The purpose of the active approaches is to measure the packet loss rate of the probe flow, and the purpose of LossSight is to measure the packet loss rate of the telemetry flow for INT-MD. LossSight does not introduce additional traffic and does not affect the measurement results.

Cociglio [24] and PF-PLM [25] are representative solutions of the alternate-marking performance measurement (AM-PM) method. They only need a few bits to achieve loss measurement. However, their marks are updated periodically, which requires strict time synchronization too. Cociglio and PF-PLM have inspired our work: AM-PM can be applied to INT packets to know when packet loss have occurred. VTrace [34] diagnoses packet loss by installing "coloring, matching and logging" rules. It has been deployed in Alibaba Cloud for more than 20 months. However, the lightweight of VTrace in data collection is accompanied by the complexity of back-end processing. VTrace needs to reconstruct the true forwarding path of lost packets from a large number of statistical logs, which may become a performance bottleneck.

Unlike the aforementioned network-centric monitoring solutions, Pingmesh [32] is a server-centric monitoring solution. Pingmesh actively sends PING messages to detect network quality. It is used in large data center networks by Microsoft. According to the packet loss information, Pingmesh can diagnoses at least four different causes of loss. PTPmesh [37] uses Precision Time Protocol to achieve higher precision delay and packet loss measurement.

The focus of the above solutions lies in packet loss detection, localization, and root cause analysis, which are often studied independently. In the case of INT-based telemetry systems, since telemetry information is lost along with normal data packets, the recovery of lost information based on loss detection is a very important requirement.

The recovery of INT telemetry information can also be seen as a problem of lost data restoration. For INT, there is no relevant research work yet. However, there are some recovery methods in the field of network measurement in the recent years. These methods can be divided into the following four categories:

1) Statistics-based methods: These methods include Last Value Filling (LastVF), Mean Value Filling (MeanVF), Median Value Filling (MedianVF), and Common Value Filling (ComVF) [38]. However, statistical methods tend to ignore important features of telemetry information. For example, packet loss caused by micro bursts is often accompanied by abnormalities in the switch queue depth value collected by INT. The filling value of statistical methods is usually too conservative.
2) Machine Learning-based methods: These methods first use machine learning algorithms to fit the distribution of the original dataset, and then generate a customized filling value for each missing value. Common methods include K-Nearest Neighbor (KNN) [39] and Expectation Maximization (EM) [40].
3) Matrix-based methods: These methods exploit two-dimensional global information to estimate the lost measurement data. For instance, Nonnegative Matrix Factorization (NMF) [41], Sparsity Regularized SVD (SRSVD) [42], Matrix Completion (MC) [43], Low-rank Matrix Fitting algorithm (LMaFit) [44]. These methods assume a specific model of the data-generating process and the pattern of lost data.
4) Tensor-based methods: These methods are recognized as promising solutions [45]. NTC [35] is a novel neural tensor completion scheme to effectively model three-order interaction among data features with the outer product and build a 3D interaction map. MC-GPU [36] utilizes Graphics Processing Units (GPUs) to enable parallel matrix factorization for high-speed and highly accurate matrix completion. These methods are suitable for large-scale INT, where the lack of a large number of telemetry data will seriously affect the recovery performance.

In summary, according to our survey, there is currently no suitable solution suitable for the identified tasks in the field of INT monitoring. In addition, packet loss detection, localization, diagnosis, and lost information recovery need to be completed at the same time.

III. Mathematical Description of INT-Based Lossy Telemetry System

In this section, we present the mathematical description and notation of an in-band network telemetry system, and the impact of packet loss on telemetry results.

A. Notation of INT-Based Telemetry Information

We define that the start time of the telemetry process is t_1 and the end time is t_2. So from t_1 to t_2, the number of INT nodes participating in telemetry is m, and the total number of INT packets is n.

The telemetry metadata is O, and the possible values of O are $O = \{o_1, o_2, \dots, o_k\}$. As an example, queue occupancy (in bytes, cells, or packets) that the INT packet observes in the device while being forwarded. Considering the INT specification [2], queue occupancy is a 24-bit value, and consequently, $O = \{0, 1, 2, \dots, 16777215\}$.

If there is no packet loss on the network, the Telemetry Server will receive n INT reports and separate $m \times n$ telemetry metadata values. The real network state Y is a matrix with m rows and n columns:

$$Y = \begin{bmatrix} y_{11} & \cdots & y_{1n} \\ \vdots & \ddots & \vdots \\ y_{m1} & \cdots & y_{mn} \end{bmatrix} \tag{1}$$

where $y_{ij} \in X, 1 \leq i \leq m, 1 \leq j \leq n$.

B. Impact of Packet Loss on INT-Based Telemetry Information

If the Telemetry Server does not receive the telemetry report when it should have been received, an INT packet loss event has happened. We define the loss matrix $A = diag(a_1, a_2, \dots, a_i, \dots, a_n)$ where

$$a_i = \begin{cases} 1, & \text{if the packet is not lost} \\ Null, & \text{if the packet is lost} \end{cases} \tag{2}$$

If the i-th telemetry packet is lost at the j-th INT node, $a_i = Null$. i indicates the loss time and j indicates the loss location. Thus, the telemetry result observed by the Telemetry Server, if it implements a loss detection function, can be expressed as matrix Z:

$$Z = Y \times A = \begin{bmatrix} y_{11} & \cdots & Null & \cdots & y_{1n} \\ \vdots & \ddots & Null & \ddots & \vdots \\ y_{m1} & \cdots & Null & \cdots & y_{mn} \end{bmatrix} \tag{3}$$

However, if no INT loss monitoring mechanism is adopted, the telemetry system will not be able to construct the loss matrix A and evaluate the impact of packet loss on the telemetry result Z. In this paper, we introduce the use of the alternate marking method to construct the loss matrix A and calculate the telemetry result matrix Z.

C. Some Definitions

This section introduces some additional terms related to LossSight. Their relationship is shown in Figure 3.

1) **Loss Rate** [46]: The ratio of the number between telemetry reports which should have been received but which have not been received and the number of all telemetry packets.[3]

$$lossrate = \frac{n - tr(A)}{n} \tag{4}$$

2) **Loss Time**: It indicates when the loss happened, that is, the INT reports that are not received. Consequently it indicates the set of subscripts of $Null$ elements in the loss matrix A.

[3]Replace the $Null$ element with 0 in A for calculation.

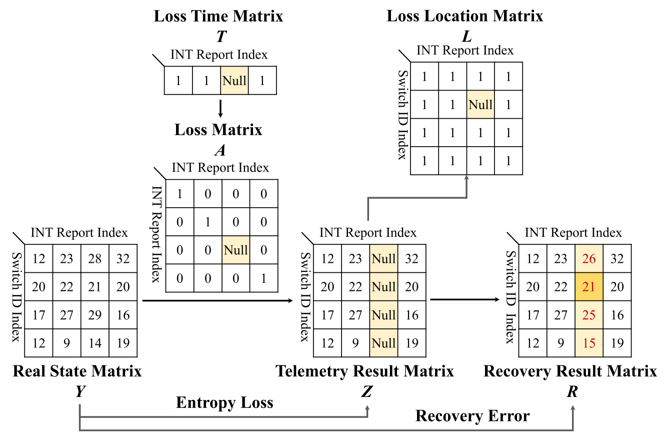


Fig. 3. Several matrices involved in LossSight and their relationships.

TABLE II
METHODS TO EVALUATE RECOVERY ERROR

Metric	Calculation
MAE	$\frac{1}{mn}\sum_{i=1,j=1}^{i=m,j=n}\lvert y_{ij}-r_{ij}\rvert$
MSE	$\frac{1}{mn}\sum_{i=1,j=1}^{i=m,j=n}(y_{ij}-r_{ij})^2$
RMSE	$\sqrt{\frac{1}{mn}\sum_{i=1,j=1}^{i=m,j=n}(y_{ij}-r_{ij})^2}$

3) **Loss Location**: Network nodes where INT packets are lost. Since the INT telemetry path goes through different nodes from the INT Source Node to the INT Sink Node, it uses the switch identifier to indicate the loss location.
4) **Loss Interval**: The difference between the sequence number of two lost packets [47]. For example, if the telemetry packet with sequence number 10 is lost, and the sequence number of the next lost telemetry packet is 30, then the packet loss interval is 20.
5) **Loss Period**: We will consider that a group of consecutively lost telemetry packets belongs to the same telemetry loss period. Assuming that P_i is the i-th telemetry packet, we define the function $f(P_i)$:

$$f(P_i) = \begin{cases} 1, & \text{if the packet is lost} \\ 0, & \text{if the packet is not lost} \end{cases} \tag{5}$$

If $f(P_i) = 1$ and $f(P_{i-1}) = 0$, it is considered that a new loss period has started; if $f(P_{i-1}) = 1$ and $f(P_i) = 0$, the loss period ends.
6) **Recovery Error**: A performance parameter to evaluate the recovery algorithm. It represents the deviation between the recovery result matrix R and the real state matrix Y. As shown in Table II, we can use Mean Absolute Error (MAE), Mean Square Error (MSE) or Root Mean Square Error (RMSE) to calculate the recovery error.
7) **Loss Pattern**: Due to different loss scenarios, such as random packet loss and non-random packet loss, congestion packet loss and non-congestion packet loss, transient packet loss and persistent packet loss, we can divide packet loss into different patterns. According to recent research, the common models for evaluating a packet loss pattern include Bernoulli Model, Gilbert Model, Markov Model and Double Regression Model [48].

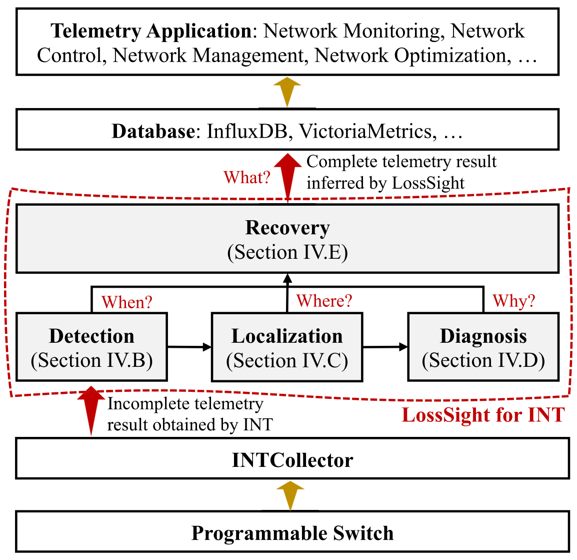

Telemetry Application: Network Monitoring, Network Control, Network Management, Network Optimization, …
Database: InfluxDB, VictoriaMetrics, …
What?
Complete telemetry result inferred by LossSight
Recovery (Section IV.E)
When?
Where?
Why?
Detection (Section IV.B)
Localization (Section IV.C)
Diagnosis (Section IV.D)
Incomplete telemetry result obtained by INT
LossSight for INT
INTCollector
Programmable Switch

Fig. 4. Four function components of LossSight and a simple process diagram.

IV. SYSTEM DESIGN OF LOSSSIGHT

In this section, we enumerate the main design goals and requirements of the LossSight system, summarize its functionality and describe the key components (detection, localization, diagnosis and recovery).

A. Design Goals and Overview

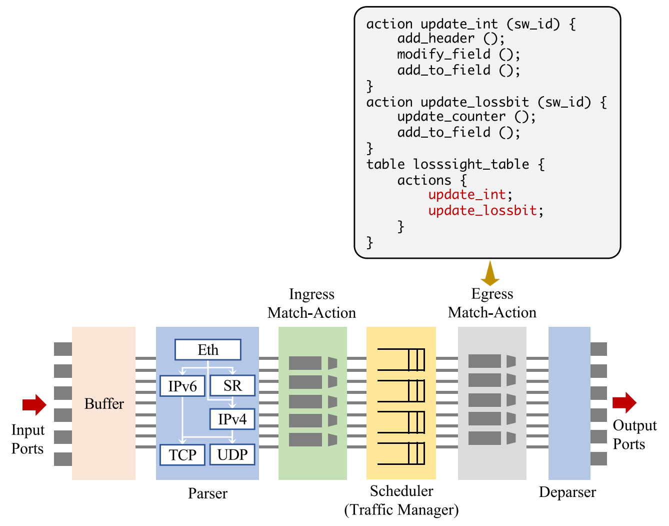
Based on the aforementioned technical requirements, LossSight system should achieve the following goals:

1) Low-overhead packet loss detection and localization: Compared with existing telemetry systems, LossSight helps INT to detect when and where packet loss has occurred.
2) Accurate analysis of the causes of packet loss: When packet loss occurs, the system should quickly find the problematic virtual or physical network device, and analyze the cause of the packet loss.
3) Lost telemetry information recovery to enhance INT measurement results as much as possible.
4) The system should be compatible with the existing INT architecture and protocols, avoiding unnecessary changes.

In order to carry out the tasks of detection and recovery of packet loss, LossSight should have the aforementioned four functions: detection, localization, diagnosis, and recovery (relating to when, where, why packet loss occurred, and what telemetry information was lost). The function components are shown in Figure 4 and described in detail in the following sections.

B. Detection Component

Alternate-marking performance measurement (AM-PM) method is described in [49] and standardized in [50]. It realizes



```
action update_int (sw_id) {
    add_header ();
    modify_field ();
    add_to_field ();
}
action update_lossbit (sw_id) {
    update_counter ();
    add_to_field ();
}
table losssight_table {
    actions {
        update_int;
        update_lossbit;
    }
}
```

Fig. 5. Processing pipeline of Protocol-Independent Switch Architecture.

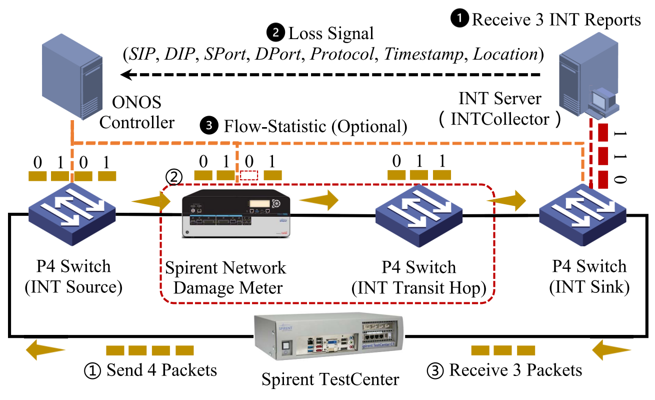


Fig. 6. The detection process of LossSight.

efficient measurement of delay and packet loss by periodically changing a 1-bit or 2-bit coloring flag. In AM-PM, every packet of the monitored flow carries 1 or 2 marking bits that are used for signaling and coordinating measurement events across the measurement points. For packet loss measurements, a periodically alternating marker, Color Bit, is dividing the traffic into consecutive blocks of data. By counting the number of packets in each block and comparing the values measured by different network devices along the path, it is possible to measure packet loss occurred in any single block between any two points [51]. Applying the AM-PM method to INT, we proposed and designed the marking protocol and two marking strategies.

Figure 5 describes the representative Protocol-Independent Switch Architecture (PISA) processing pipeline of a programmable switch [52], where the marking process is implemented as part of the egress pipeline. After inserting the telemetry header, the INT Source Node also marks this telemetry packet. As shown in Figure 6, the simplest way of marking is alternate marking: for the first packet, the INT Source Node sets the loss flag bit to 1, the second packet is set to 0, the third packet is set to 1, and so on. More reliable and efficient marking schemes are described in Section IV-B2. INT Sink Node reports the telemetry information to the Telemetry Server. The

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Ver = 2|D|E|M|L|      Reserved         | Hop ML  |RemainingHopCnt|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Instruction Bitmap       |      Domain Specific ID       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       DS Instruction          |          DS Flags             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| INT Metadata Stack (Each hop inserts Hop ML * 4B of metadata) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Last INT metadata                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fig. 7. INT-MD metadata header format.

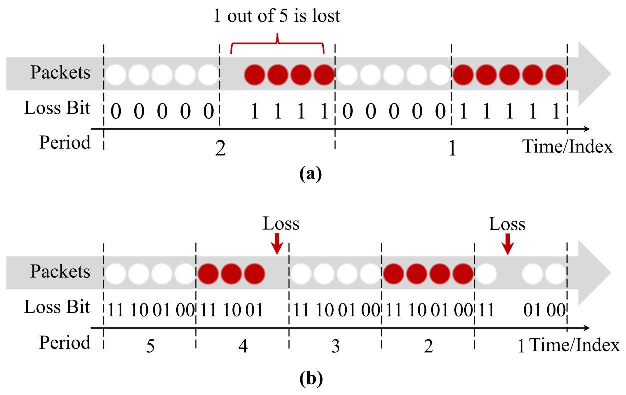


Fig. 8. Schematic diagram of two marking strategies: (a) Single-bit Alternate Marking (SAM), (b) Multi-bit Cycle Marking (MCM).

Telemetry Server (INT server or INT collector in the figure) detects the loss events and calculates the loss rate based on the successively received loss flags. If the Telemetry Server does not receive the marking information it should receive, a packet loss has occurred.

1) Protocol Design: As shown in Figure 7, LossSight can use 1 or more bits (L), in the reserved field of the INT-MD protocol header to mark a telemetry packet. We call it Loss Bit field.

2) Marking Strategies: We introduce two packet loss marking strategies, Single-bit Alternate Marking (SAM) and Multi-bit Cycle Marking (MCM).

• **Single-bit Alternate Marking (SAM)** If Loss Bit field is 1-bit long, the marking action of the INT Source Node is triggered by the arriving of a packet. The INT Source Node changes the Loss Bit from 0 to 1 or from 1 to 0 every $p/2$ telemetry packets. For example, if marking period $p = 10$ (see Figure 8 (a)), the INT Source Node marks the Loss Bit of the first 5 INT packets as 1 and then the Loss Bit of last 5 INT packets as 0.

We assume that there is no out-of-order packet in the network, and each random packet loss is independent. Under these conditions, when the packet loss probability is extremely low, the alternate marking ($p = 2$) can work normally. When the random packet loss rate on the network is high or there are congestion losses, the probability of losing 2 neighboring packets increases. In this case, alternate marking ($p = 2$) does not allow us to detect that the telemetry flow has lost two consecutive packets. This will cause the measured loss rate to be less than the true loss rate.

Assuming that network has m hops, and the random packet loss rate of each hop is $\epsilon_i, i = 1, \ldots, m$, the end-to-end random packet loss rate ϵ of telemetry packets is

$$\epsilon = 1 - \prod_{i=1}^{m} (1 - \epsilon_i). \tag{6}$$

The probability $\mathscr{P}$ of consecutive loss of p telemetry packets is

$$\mathscr{P} = \epsilon^p = \left[1 - \prod_{i=1}^{m} (1 - \epsilon_i)\right]^p. \tag{7}$$

Because SAM strategy can identify the consecutive loss of no more than p telemetry packets, a larger marking period can improve the confidence of telemetry results. According to Equation (7), when the packet loss rate is less than 20%, the probability of continuous packet losses is very low. For higher loss rates, the detection accuracy will be improved by increasing the marking period.

We use the time interval T between the occurrence of packet loss and the detection of packet loss to evaluate the detection sensitivity. Obviously, detection sensitivity is affected by different factors such as network traffic, topology size or packet loss location. At this point, we consider that the arrival of telemetry packets obeys the Poisson distribution with λ, the hop-by-hop delay is $d_i, i = 1, \ldots, m$, and the transmission and processing delay between the INT Sink Node and Telemetry Server is $d_{Sink2Server}$. Thus, T_{min} and T_{max} values can be defined as follows. The minimum detection interval T_{min} represents the detection interval of the last telemetry packet loss at the last hop before the marking strategy change. The maximum detection interval T_{max} represents the detection of the loss of the first telemetry packet at the first hop after the marking strategy change.

The distribution function of time interval t between the arrival of two adjacent packets is

$$F(t; \lambda) = 1 - e^{-\lambda t}, t \geq 0. \tag{8}$$

The expectation of T_{min} and T_{max} can be expressed as

$$\mathbb{E}\{T_{min}\} = \mathbb{E}\{T\} = \frac{1}{\lambda} \tag{9}$$

$$\begin{aligned} \mathbb{E}\{T_{max}\} &= \frac{p}{2}\mathbb{E}\{T\} + \sum_{i=1}^{n} d_i + d_{Sink2Server} \\ &= \frac{p}{2\lambda} + \sum_{i=1}^{n} d_i + d_{Sink2Server}. \end{aligned} \tag{10}$$

According to Equation (10), the larger the marking period, the lower the detection sensitivity. Therefore, although only 1-bit is used, SAM needs a suitable marking period to balance detection robustness and sensitivity.

- **Multi-bit Cycle Marking (MCM)** As shown in Figure 8 (b), LossSight can use a longer Loss Bit field in order to include sequence information in the telemetry packets. This marking strategy is called Multi-bit Cycly Marking (MCM).

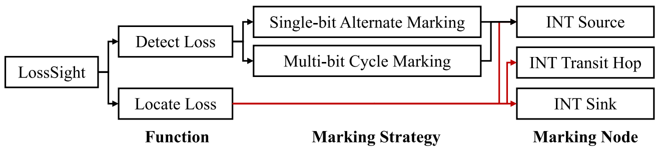
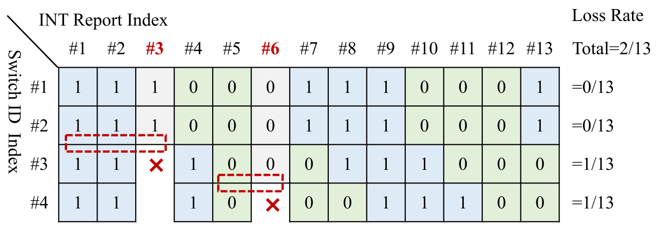
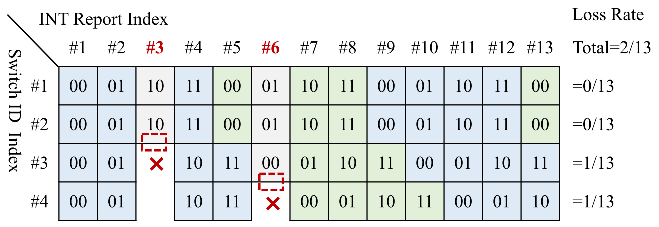


Fig. 9. The design idea of detection and localization.



Switch ID Index	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	Loss Rate Total=2/13
#1	1	1	1	0	0	0	1	1	1	0	0	0	1	=0/13
#2	1	1	1	0	0	0	1	1	1	0	0	0	1	=0/13
#3	1	1	✗	1	0	0	0	1	1	1	0	0	0	=1/13
#4	1	1		1	0	✗	0	0	1	1	1	0	0	=1/13

Fig. 10. Loss Bit encoding method (SAM) for loss localization. INT reports in red are inferred after applying loss detection and loss location algorithms.



Switch ID Index	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	Loss Rate Total=2/13
#1	00	01	10	11	00	01	10	11	00	01	10	11	00	=0/13
#2	00	01	10	11	00	01	10	11	00	01	10	11	00	=0/13
#3	00	01	✗	10	11	00	01	10	11	00	01	10	11	=1/13
#4	00	01		10	11	✗	00	01	10	11	00	01	10	=1/13

Fig. 11. Loss Bit encoding method (MCM) for loss localization. INT reports in red are inferred after applying loss detection and loss location algorithms.

Assuming that the length of the Loss Bit field of MCM is l, in a similar to Equation (7), the probability $\mathscr{P}$ of consecutive loss of 2^l telemetry packets is

$$\mathscr{P} = \epsilon^p = \left[1 - \prod_{i=1}^{m} (1 - \epsilon_i)\right]^{2^l}. \tag{11}$$

The longer the Loss Bit field, the better it can adapt to higher packet loss scenario. When the packet loss rate is less than 10%, the recognition accuracy of the five lengths are all greater than 99%. MCM strategy can reduce but cannot solve completely the detected error events caused by out-of-order packets.

Since each telemetry report carries sequence information, the Telemetry Server can deduce which telemetry packet is lost according to the received Loss Bit values. So, the expectation of detection sensitivity is $[\frac{1}{\lambda}, \frac{1}{\lambda} + \sum_{i=1}^{n} d_i + d_{Sink2Server}]$.

C. Localization Component

Localization Component applies the aforementioned detection mechanism from INT source to all INT nodes. As shown in Figure 9, all INT Nodes maintain the marking counter for each flow and add their own Loss Bit value in turn. The Loss Bit field can be specified by the Instruction Bitmap and stored in the INT metadata inserted by each node hop-by-hop in the INT Metadata Stack. Those counters are integrated into the INT operation involving little overhead.

Figures 10 and 11 illustrate the packet loss localization under the two marking strategies. The INT packets should pass through four switches, but some of them are lost at some points

TABLE III
SUMMARY OF LOSS PATTERNS

Pattern	Common Root Cause	Can be recovered?
Random	Hardware defect, etc.	✓
Congestion	Micro burst, Buffer overflow, etc.	✓
Blackhole	Flow-table failure, etc.	×
Unknown	Unknown	✓

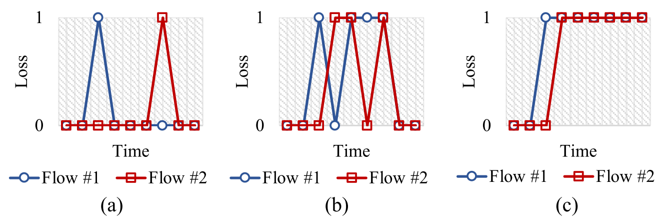


Fig. 12. Three loss patterns: (a) Random, (b) Congestion, (c) Blackhole.

of the paths. As a result, the Telemetry Server will be able to build the matrix shown in Figure 10 (SAM) and Figure 11 (MCM) according to the received telemetry reports. The principle of packet loss location can be described as follows: if the marking sequence (associated to a switch) received by the Telemetry Server does not conform to the expected marking sequence, a packet loss has occurred. As can be deduced, MCM strategy allows LossSight to obtain a more accurate measure of loss time and loss location. The dashed box indicates when the packet loss can have happened. For example, in the SAM example, the first packet loss is detected after receiving the INT report numbered in the figure as #4 (INT reports are not internally numbered), but it could have happened in any of the position marked by the dashed box. At the same time, MCM offers better localization sensitivity. Algorithm 1 describes the process of packet loss localization more generally. To infer packet loss location, Algorithm 1 traverses and checks the telemetry result matrix Z from the lower left corner to the right corner, comparing the received and expected markers. As can be seen (line 3 and line 9), different loss detection and localization algorithms are applied depending on the marking strategy (SAM or MCM). It should be noted that for the SAM marking, the algorithm may not be able to accurately locate the time and location of some losses, which is consistent with Figure 10.

Algorithm 1: Loss Localization Algorithm

```
Input: Marking Strategy: $SAM$ or $MCM$; Marking
       Period: $P$; Loss Bit Length: $l$; Number of
       Telemetry Nodes: $m$; Number of received INT
       Reports: $nn$; Number of Telemetry Packets: $n$;
       Telemetry Result Matrix (Marking Matrix): $Z$.
Output: Loss Location Matrix: $L$.
1  Function loss_localization is
       // Initialize Matrix $L$ with 1.
2      $L$ = ones(m,n) ;
       // Infer packet loss location.
3      if $marking\_strategy == SAM$ then                /* SAM */
4          for $j = 1, 2, \dots, nn$ do
5              for $i = m, \dots, 2, 1$ do
6                  if $(j+loss\_count(i,j))\%p < (\frac{p}{2}-1)$ &&
                     $Z(i,j) \mathrel{!}= 1)$ || $(j+loss\_count(i,j)\%p >$
                     $(\frac{p}{2}-1)$ && $Z(i,j) \mathrel{!}= 0)$ then
7                      $L(i,j+loss\_count(1,n)) = Null$ ;
8                      break ;
9      else
10         for $j = 1, 2, \dots, nn$ do                    /* MCM */
11             for $i = m, \dots, 2, 1$ do
12                 if $(Z(i,j+1)-Z(i,j) \mathrel{!}= 1)$ ||
                     $(Z(i,j+1)-Z(i,j) \mathrel{!}= (-2^l+1)$ then
13                     $L(i,j+loss\_count(1,n)) = Null$;
14                     break;
15     return $L$;
16 Function loss_count(i,j) is
17     int count = 0;
18     for $ii = i, \dots, m$ do
19         for $jj = 1, \dots, j$ do
20             if $L(ii,jj) == Null)$ then
21                 count++;
22     return count;
   // loss_count(i,j) is the function to
      calculate the number of historical
      packet loss from the $i$th switch to
      the $m$th switch until $j$.
```

D. Diagnosis Component

The diagnosis component of LossSight offers a classification and root cause diagnosis for single INT flows, based on different loss patterns that consider the flow-related information such as 5-tuple, flow ID, loss locations, and loss timestamps. Assuming that there are multiple INT flows in the network, the diagnosis could be done through mutual verification between them.

Based on a previous research [21], we distinguish four main types of packet loss patterns. As is shown in Table III, common packet loss can be divided into random, congestion, blackhole, and unknown [53]. Figure 12 draws an abstract description of these first three loss patterns.

Random loss is evenly distributed over time. There are two reasons for this loss pattern: hardware and software. In modern data center networks, the probability of random loss caused by hardware is lower than 10^{-6} [54]. Software bugs or faulty interfaces in switches may randomly drop some packets according to the route in the network or packet header information [55].

Congestion loss is bursty [21] and the gap between back-to-back losses is only a few microseconds [56]. Taking TCP traffic as an example, TCP sender sends a batch of packets every RTT, and the flows that experience losses will shrink their congestion windows, so in the next RTT congestion is much less likely to happen. Congestion is the most serious event that causes loss, because some telemetry information related to congestion will be lost with the packet, such as switch queue depth, and processing delay.

Blackhole loss can be divided into transient blackhole and persistent blackhole [32]. For this pattern of loss, LossSight will not receive any INT report, because all the packets go through the routing blackhole. Unlike the other cases, the detection of blackhole loss would require the implication of the SDN controller. This situation can be detected by the SDN controller comparing the matching counters of the flow tables of the switches. If the matching number of packets in the INT Source Node and the INT Sink Node is significantly different, LossSight can determine that the INT flow is experiencing routing blackhole. Although LossSight is able to detect blackhole loss with the assistance of the SDN controller, it cannot recover the telemetry information carried by packets that are lost due to this loss pattern. This is because blackholes cause a large number of continuous packet losses, similarly to an interrupted communication.

In a real network, various patterns of packet loss often occur together. Correctly distinguishing different loss patterns is very important for telemetry information recovery, because some telemetry items are related to network congestion, including switch-level metadata *Received Packet Count*, Port-level metadata *Link Utilization*, Queue-level metadata *Receive Overrun Error Count*, etc.

Algorithm 2 describes the proposed loss diagnosis algorithm. Initially, all lost packets are assigned to unknown loss patterns (lines 2 to 3). Firstly, the algorithm diagnoses congestion loss (lines 9 to 12). Secondly, it diagnoses random loss (lines 13 to 16). Finally, the algorithm diagnoses blackhole loss (lines 6 to 7, 17 to 21). Congestion loss is identified when there is continuous loss whose loss $\text{Period} >= t_{gap}$ or there is at least n_{con} consequent losses with gaps less than t_{gap} between back-to-back losses.[4] $PC_{k.num}$ is the number of consecutive lost packets due to potential congestion. $PC_{k.starttime}$ and $PC_{k.endtime}$ are the start and end times of continuous packet loss due to potential congestion. k is the number of potential congestion. Loss period was defined in Section III-C. In other words, if continuous loss or at least n_{con} loss occur within the range of t_{gap}, it is considered that congestion loss has occurred (lines 9 to 12). Among the remaining lost packets that are marked as unknown, if the loss interval is greater than R_{th}, it is marked as random packet loss (lines 13 to 16). Blackhole loss is diagnosed by comparing the counter increments at the entry and exit points of the network. If the Telemetry Server has not received an telemetry report for a long time (timeout), it is necessary to detect whether blackhole loss has occurred (lines 6 to 7). Parameters involved in Algorithm 2 and their meanings are shown in Table IV.

[4]In actual networks, packet loss caused by congestion may also show more complex characteristics, such as the congestion duration (micro burst), communication mode (Incast), the drop strategy when the port is congested, etc. Therefore, network administrators should specifically define congestion loss parameters t_{gap} and n_{con} based on their own network.

Algorithm 2: Loss Diagnosis Algorithm

```
Input: Loss Location Matrix: $L$; Loss Time Matrix: $T$;
       Number of Telemetry Packets: $n$.
Output: Loss Diagnosis Matrix: $D$.
1  Function loss_diagnosis is
       // Initialize Matrix $D$.
2      $D = L$;
       // Diagnosis part.
3      Mark the types of all losses in $D$ as unknown;
4      congestion_loss();
5      random_loss();
6      if $Time_{Now} - Time_{LastReport} \geq T_{th}$ then
7          blackhole_loss();
8      return $D$;
9  Function congestion_loss is
10     for the remaining loss marked as unknown in $D$ do
11         if (duration(Loss Period) >= $t_{gap}$) ||
           ($PC_{k.endtime} - PC_{k.starttime} < t_{gap}$ &&
           $PC_{k.num} > n_{con}$) then
12             Mark the types of all losses in this Loss
               period in $D$ as congestion;
13 Function random_loss is
14     for the remaining loss marked as unknown in $D$ do
15         if Loss Interval >= $R_{th}$ then
16             Mark the types of this loss in $D$ as random;
17 Function blackhole_loss is
18     Query the flow match counters of INT Source and
       INT Sink;
19     Calculate the counter increment of INT Source and
       INT Sink to get $\Delta C_{INTSource}$ and $\Delta C_{INTSink}$;
20     if $\Delta C_{INTSource} - \Delta C_{INTSink} \geq n_{bh}$ then
21         Append $\Delta C_{INTSource} - \Delta C_{INTSink}$ losses as
           consecutive blackholes after $D$;
```

TABLE IV
ALGORITHM 2 PARAMETERS AND THEIR MEANING

Parameter	Meaning
T_{th}	Threshold for Timeout
t_{gap}	Threshold for duration of congestion packet losses
n_{con}	Threshold for number of congestion packet losses
R_{th}	Threshold for interval of random packet losses
n_{bh}	Threshold for number of blackhole packet losses
ΔC_{Source}	INT Source's counter increment
ΔC_{Sink}	INT Sink's counter increment

E. Recovery Component

The impact of lost telemetry information on network measurement [57], network control and network management cannot be ignored. Therefore, it is necessary for LossSight to recover the lost telemetry information as much as possible, thereby complementing the tasks of detection and localization of the lost telemetry packets.

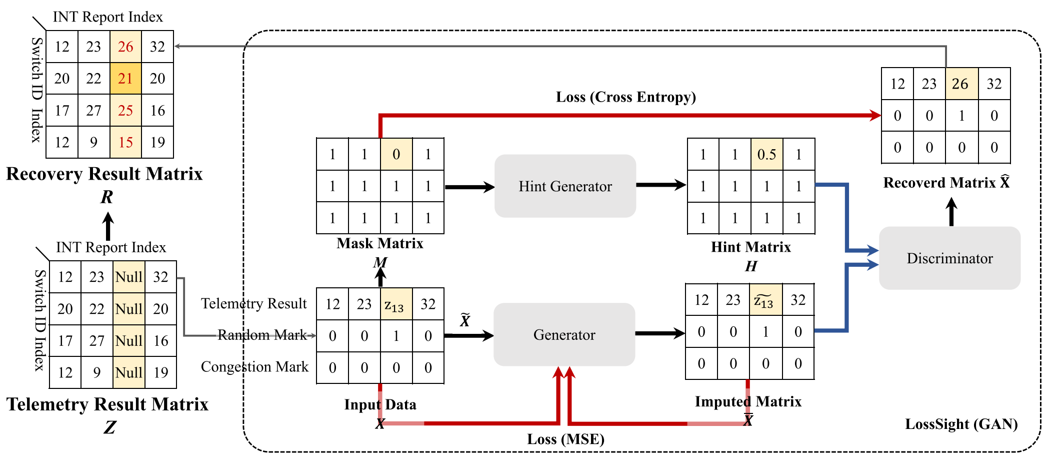


Fig. 13. Schematic diagram of recovery process.

The recovery strategy of LossSight adopts different recovery methods for the different types of lost telemetry information. Take the queue depth as an example. When the network is congested, the queue depth increases significantly. Distinguishing between random loss and congested loss is required for the recovery of lost values of queue depth. In this way, LossSight could fill in the lost queue value with probabilistically distributed values in the case of random loss and fill in with larger values in the case of queue depth losses that are not random losses (e.g., congestion loss). As commented before, for blackhole loss, which belongs to the category of network failures, LossSight does not need to recover the loss of information.

In order to automate the deployment of the above-mentioned strategies, LossSight uses the generative adversarial network (GAN) [58] to learn the essential characteristics and loss rules of lost telemetry information. The recovery model is shown in Figure 13.

The input data X of the GAN model is the incomplete telemetry data, the random loss marks, and the congestion loss marks. Each row of X can be expressed as [telemetry result, random mark, congestion mark]. The value of random mark and congestion mark is 0 or 1, and its purpose is to let GAN know the network status when the packet was lost.

The Generator fills X with a low-dimensional random variable V:

$$\bar{X} = G\left\{\widetilde{X}, M, (1-M)\odot V\right\}. \tag{12}$$

When $m_j = 1$, $\bar{X}_{ij} = \widetilde{X}_{ij}$; When $m_j = 0$, $\bar{X}_{ij}$ is the value trained by the Generator.

Therefore, in each round of training, the imputed matrix $\hat{X}$ generated by the Generator can be expressed as

$$\hat{X} = M\odot\widetilde{X} + (1-M)\odot\bar{X}. \tag{13}$$

The Generator is trained to generate plausible data. The other part of GAN is the Discriminator. The Discriminator is trained to distinguish fake values from real values. Therefore, the input of the Generator is the original data X, the mask matrix M and the random variable V. The output of the Generator is the estimate of the lost values, that is, the imputed matrix $\hat{X}$. The purpose of the Generator is to interfere with the discrimination result of the Discriminator, and to minimize the probability of the Discriminator getting the correct result (that is, judging a value as a fake value).

For loss function of the Generator, we separately discuss the two cases of X_{1j} lost and X_{1j} not lost. When X_{1j} is lost, the loss function is the probability of judging the $\hat{X_{1j}}$ generated by the Generator as loss. When X_{1j} is not missing, the loss function is the difference between the generated result and the original data X. Therefore, the loss function of the Generator is

$$\begin{aligned}\mathcal{L}_G &= \sum L_G\left(m_j, D\left(\hat{X_{1j}}, h_j, b_j\right)\right) + \alpha * L_M\left(X_{1j}, \hat{X}_{ij}\right)\\ &= -\sum_{i:b_i=0}(1-m_i)log(\hat{m_i}) + \alpha\sum_i m_i L_M\left(X_{1i}, \hat{X}_{1j}\right),\end{aligned} \tag{14}$$

where α is a hyper-parameter,

$$L_M\left(X_{1i}, \hat{X}_{1j}\right) = \begin{cases}\left(X_{1i} - \hat{X}_{1j}\right)^2, & \text{if } \mathrm{X_{1j}} \text{ is continuous}\\ -X_{1i}log\left(\hat{X}_{1j}\right), & \text{if } \mathrm{X_{1j}} \text{ is binary.}\end{cases} \tag{15}$$

We define the auxiliary variable B and H:

$$B = (b_1, \ldots, b_d) \in \{0,1\}^d, \tag{16}$$

$$H = B\odot M + 0.5(1-B). \tag{17}$$

B is a matrix composed of random 0 and 1. GAN only trains the positions where $b_i = 0$. H is the Hint Matrix, which provides the discriminator with additional information in the form of "hints." This hinting ensures that the Generator generates samples according to the true underlying data distribution.

The input of the Discriminator is the output of the Generator and the mask matrix M. The output of the Discriminator is the value of the elements in the estimated matrix M, that is, the probability that the data is lost (0 or 1), which represents whether the complete data transmitted is filled by the Generator. The output of the arbiter is expressed as

$$\hat{m_j} = D\left(\hat{X_{1j}}, m_j\right). \tag{18}$$

```
Algorithm 3: Loss Recovery Algorithm
  Input: Telemetry Result Matrix: $Z$; Number of Switches: $n$; Batch Size; Hyperparameter; Iterations.
  Output: Recovery Result Matrix: $R$.
1 Function loss_recovery is
2     for i=1, 2, ..., n do
3         $X(1,:) = Z(i,:)$;
4         Mark random $X(2,:)$ or congestion $X(3,:)$ according to loss_diagnosis($X(1,:)$);
5         while training loss has not converged do
6             (1) Discriminator optimization;
7             Draw $k_D$ samples from the dataset $\{(\widetilde{X_{1j}}, m_j)\}_{j=1}^{k_D}$;
8             Draw $k_D$ i.i.d. samples, $v_j{}_{j=1}^{k_D}$, of $V$;
9             Draw $k_D$ i.i.d. samples, $b_j{}_{j=1}^{k_D}$, of $B$;
10            for j=1, 2, ..., $k_D$ do
11                $\bar{X_{1j}} \leftarrow G(\widetilde{X_{1j}, m_j}, v_j)$;
12                $\hat{X_{1j}} \leftarrow m_j \odot \widetilde{X_{1j}} + (1 - m_j) \odot \bar{X_{1j}}$;
13                $h_j = b_j \odot m_j + 0.5(1 - b_j)$;
14            Update $D$ using stochastic gradient descent (SGD) according to Equation 19;
15            (2) Generator optimization;
16            Draw $k_G$ samples from the dataset $\{(\widetilde{X_{1j}}, m_j)\}_{j=1}^{k_D}$;
17            Draw $k_D$ i.i.d. samples, $v_j{}_{j=1}^{k_D}$, of $V$;
18            Draw $k_D$ i.i.d. samples, $b_j{}_{j=1}^{k_D}$, of $B$;
19            for j=1, 2, ..., $k_D$ do
20                $h_j = b_j \odot m_j + 0.5(1 - b_j)$;
21            Update $G$ using stochastic gradient descent (SGD) according to Equation 14;
```

The loss function of the Discriminator is

$$\mathcal{L}_D = \left[m_i log(\hat{m_i}) + \sum_{i:b_i=0} (1 - m_i) log(\hat{m_i}) \right]. \quad (19)$$

The pseudo code of the loss recovery algorithm is shown in Algorithm 3. LossSight alternately optimizes Discriminator (line 6 to 14) and Generator (line 16 to 21). In addition, we suggest that the default number of columns of Z is set to 10000 (about 0.6-15 MB throughput), which trade-off GAN performance and traffic characteristics.

V. Experiment Results

In order to verify the performance and overhead of LossSight, a testbed was built on the Supercomputing Internet Platform of National Supercomputing Center in Jinan. As shown in Figure 6, the testbed is composed of three P4 switches, a Spirent TestCenter, a Spirent Network Damage Meter, an ONOS (Open Network Operating System) controller and an INT remote server. P4 switches are OpenMesh BF-48X8Z, equipped with Barefoot Tofino

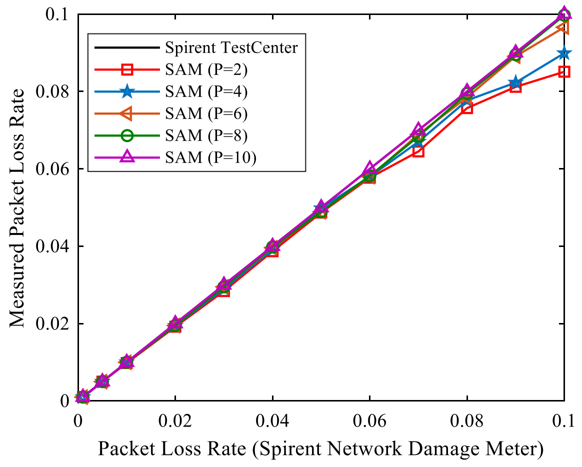


Fig. 14. Packet loss rate measurement accuracy of LossSight (using SAM).

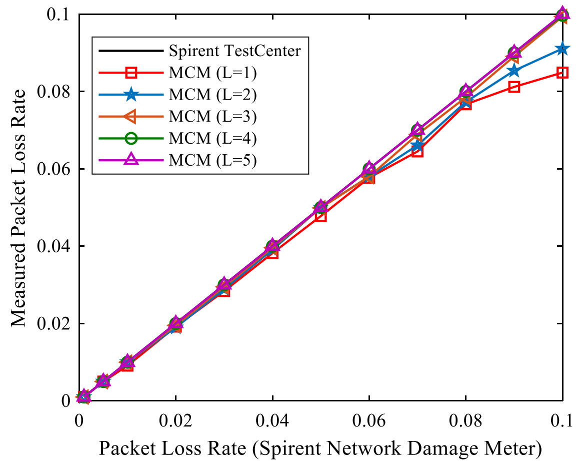


Fig. 15. Packet loss rate measurement accuracy of LossSight (using MCM).

programmable switching chip (1.8 Tbps switching capacity) and 48*10GbE (SFP+)/25GbE (SFP28) interfaces, 8*40GbE (QSFP+)/100GbE (QSFP28) interfaces. The Spirent TestCenter sends and receives packets, and counts the loss rate as a reference. The frame size is 1024 bytes. The Spirent Network Damage Meter achieves random packet loss. The ONOS controller version is 2.2 (Sparrow). The source code of LossSight is open source [29], and it also supports Mininet environment. The original data of telemetry experiments have been published [31].

A. Performance of Detection and Localization

Figure 14 and Figure 15 evaluate the accuracy of the two marking strategies: SAM and MCM. In general, as P (the marking period of SAM) and L (the Loss Bit field of MCM) increase, the loss rate measured by LossSight is very close to the real loss rate. Firstly, the average measurement deviation between the measured values using SAM and MCM and the report values of the Spirent TestCenter is 3.95% and 1.28%, respectively. As the loss rate increases, SAM and MCM produce larger detection errors, especially when P = {2, 4} and L = {1, 2}. The reason is that the probability of continuous

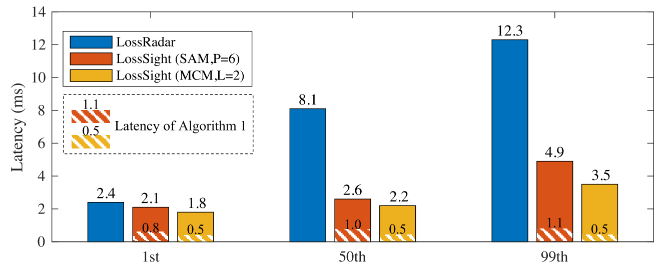


Fig. 16. Comparison of detection latency between LossSight and LossRadar.

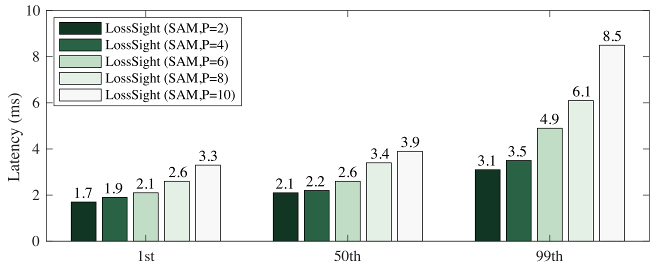


Fig. 17. Detection latency of LossSight (using SAM).

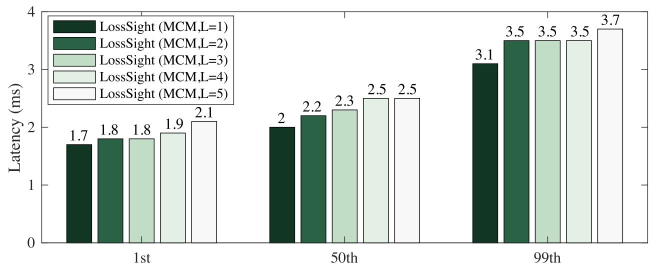


Fig. 18. Detection latency of LossSight (using MCM).

loss increases, and it is difficult for LossSight to detect packet loss within a marking period. As P and L increase, this error decreases. Secondly, the detection accuracy of SAM (P = 2) and MCM (L = 1) is very close because they essentially implement the same marking strategy.

For packet loss rates of 0% and 10%, LossSight locates the position of 100% of the packet losses using P > 8 (SAM) and L > 4 (MCM). It is worth noting that LossSight measures the actual packet loss experienced by an INT flow, so this deviation can be regarded as a probabilistic deviation.

For all this, taking into account the accuracy and cost of telemetry, we recommend that the parameters should be set to P = 6 (SAM) and L = 3 (MCM).

B. Detection Latency

The detection latency is affected by the path length, the network rate, the packet interval, the marking period and the Loss Bit field length. Telemetry Server can only confirm packet loss after eliminating the influence of out-of-order packet. As is shown in Figure 16, we compared the detection latency of LossSight and LossRadar [21]. The highlighted part is the latency of the detection algorithm. The remainder of the total latency is affected by the location of the loss packet. Two conclusions can be drawn. (1) Triggering by packet loss events (LossSight) has an inherent advantage over triggering by polling (LossRadar). It can be observed that LossSight using SAM or MCM offers more stable values. Loss detection in LossSight is triggered by the continuous Loss Bit filed values, so there is no need for an excessive waiting time. Since the confirmation latency of MCM (L = 2) is shorter than SAM (P = 6), the detection speed of MCM is faster than SAM. (2) The average latency of detection algorithm is affected by the packet arrival interval. In this experiment, SAM is approximately 2X that of MCM in terms of the average latency of detection algorithm. Moreover, LossRadar detection delay depends on the query period, and the query period of LossRadar is usually 10 ms.

The detection latency of SAM and MCM for different values of P and L is evaluated in Figure 17 and Figure 18, respectively. As can be observed, the greater the parameter P or L, the greater the detection latency. In the case of SAM, because the INT packets do not carry sequence information, the detection latency is worse than MCM, also presenting a long-tail effect. In both cases, most of the loss detection can be completed in a few milliseconds. In addition, in the case of MCM, the main contribution to the detection latency is the network latency.

TABLE V
ALGORITHM 2 PARAMETER SETTINGS

	T_{th} **(s)**	t_{gap} **(ms)**	n_{con}	R_{th} **(ms)**	n_{bh}
Range	1	5/50/100	3/5/10	5/10/50	100

C. Performance of Diagnosis

Judging the accuracy of diagnosis is a difficult work, because, currently, there is no an accurate definition of random loss and congestion loss. Most existing works use their own evaluation criteria as the diagnostic reference on their private dataset, e.g., [32] and [21]. In order to objectively conclude the diagnostic performance, we evaluated the performance of the proposed loss diagnosis algorithm (Algorithm 2) when there is only a single loss pattern, and when there are mixed loss patterns. In the first case, we consider only random loss pattern (Figure 19) or congestion loss pattern (Figure 20). For the second case, Figure 21 shows the results when both random loss and congestion loss patterns are present. The diagnosis parameters are detailed in Table V.

The following metrics are obtained:

Accuracy = (TP+TN)/(TP+TN+FP+FN),

Precision = TP/(TP+FP),

Recall = TP/(TP+FN),

where TP: True Positive, TN: True Negative, FP: False Positive, FN: False Negative.

For the telemetry results with only random losses, we analyzed the influence of different R_{th} values on the diagnosis results under different random loss rates {0.1%, 0.5%, 1%, 5%, and 10%} (see Figure 19). When the packet loss rate is low (<0.5%), the obtained values for accuracy, precision, and recall of Algorithm 2 are 100%. That is, Algorithm 2

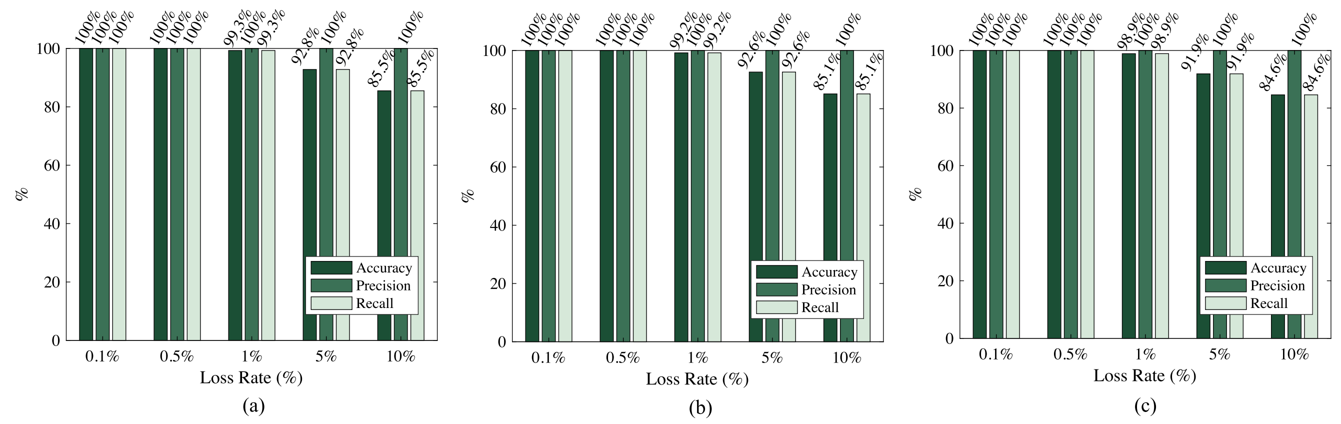


Fig. 19. Diagnostic results if single loss pattern (random loss): (a) $R_{th} = 5$ ms, (b) $R_{th} = 10$ ms, (c) $R_{th} = 50$ ms.

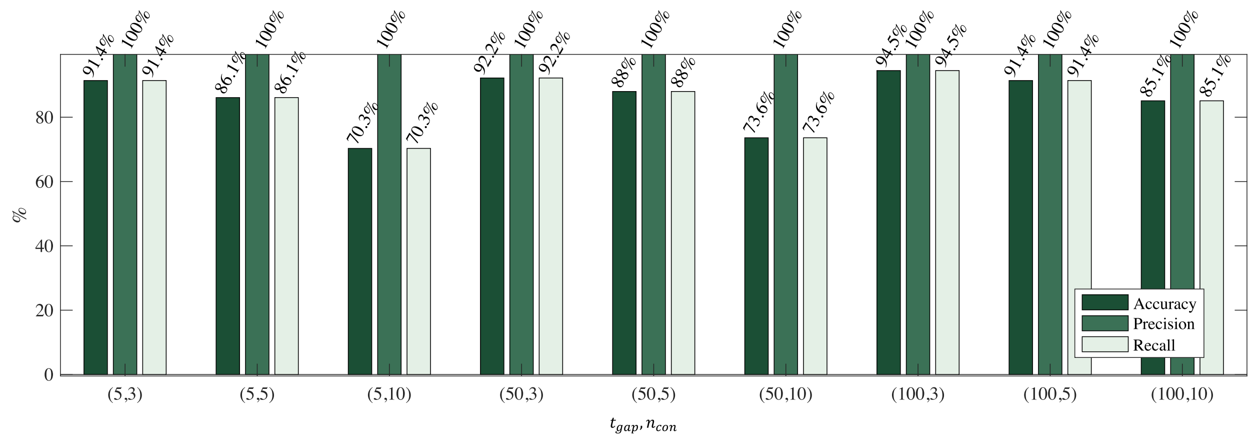


Fig. 20. Diagnostic results if single loss pattern (congestion loss).

can correctly diagnose random losses. When the loss rate is high, the accuracy and recall values obtained by Algorithm 2 decrease, but they are still higher than 84.6%. The reason is that as the loss rate and R_{th} increase, LossSight classifies some random losses as congestion losses. After analyzing the obtained results, we suggest that when LossSight is deployed, R_{th} can be set as 1.5 * the average random packet loss interval.

For telemetry results only with congestion losses, we analyzed the influence of different t_{gap} and n_{con} values on the diagnosis results (see Figure 20). As t_{gap} increases or n_{con} decreases, the accuracy and recall rates of the diagnostic algorithm increase, but precision rate is always 100%. The values of t_{gap} and n_{con} have a greater impact on diagnosis performance. Therefore, we suggest that in the actual deployment of LossSight, t_{gap} and n_{con} should be tuned to achieve better diagnostic performance.

For telemetry results with mixed loss patterns, we compared the diagnostic performance of LossSight and LossRadar (see Figure 21). The adjustable parameters are set to: $R_{th} = 10$, $t_{gap} = 50$ and $n_{con} = 5$. The probability of random loss is 0.1%. The result is shown in Figure 21. In general, LossSight relies on marking information for loss diagnosis, while LossRadar relies on the packet header information reported by switches (which offers a more adequate diagnosis basis). LossRadar first detects burst losses and then diagnoses all root causes. The algorithm is more complex and the

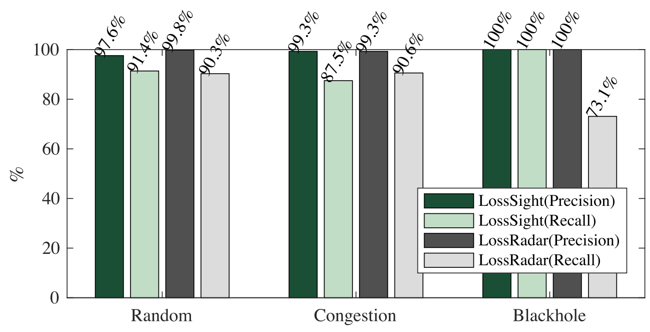


Fig. 21. Diagnostic results if mixed loss patterns.

results are more accurate. LossSight only relies on marking and timestamps, so the precision is slightly lower. However, for the blackhole pattern, LossSight outperforms LossRadar.

In summary, we recommend that network administrators apply LossSight to diagnose single loss pattern and, then, they should obtain and adjust the optimal values of R_{th}, t_{gap} and n_{con} consequently.

D. Performance of Recovery

We compared the performance of LossSight and some existing recovery algorithms. Statistics-based methods include LastVF, MeanVF, MedianVF and ComVF [38]. Machine learning-based methods include KNN [39] and EM [40].

TABLE VI
COMPARISON OF RECOVERY PERFORMANCE (MSE) OF DIFFERENT RECOVERY SCHEMES

Dataset	LastVF	MeanVF	MedianVF	ComVF	KNN	EM	LossSight(GAN)
0.1% random loss	61.2712	55.1476	**53.5213**	53.7812	67.3466	76.4198	56.6328
0.5% random loss	60.8823	**55.3132**	55.3471	56.5277	66.4182	77.1980	55.4311
1% random loss	72.9128	57.3349	57.4612	56.2714	59.6438	67.4418	**55.9142**
5% random loss	**45.3981**	60.6639	61.4516	61.4429	66.9019	64.4222	61.6637
10% random loss	**41.4581**	61.6103	66.8712	65.9634	71.7288	72.6532	67.4451
congestion loss	22.4341	65.6176	72.7981	72.7981	39.3343	47.2979	**20.7981**
0.1% random loss + congestion loss	**43.1533**	68.9104	76.7932	76.7932	75.4487	80.8119	43.7617
0.5% random loss + congestion loss	44.8419	69.9083	77.7814	77.8452	79.9631	66.4844	**39.9105**
1% random loss + congestion loss	39.7088	70.7647	74.7781	74.7781	68.9053	67.6904	**35.8933**
5% random loss + congestion loss	39.0980	64.8303	74.9091	74.5313	79.8983	57.0884	**33.8129**
10% random loss + congestion loss	25.8923	68.7305	74.7495	74.7495	26.9123	71.6636	**25.2312**

Matrix-based and tensor-based recovery algorithms are not suitable for INT systems. Some parameters of LossSight (GAN) are: Batch Size = 128, Hyperparameter = 100, Iterations = 10000.

We use MSE as the indicator of the performance of the compared recovery algorithms under different packet loss patterns [31]. The results are shown in Table VI. On the whole, LossSight (GAN) and LastVF have significant advantages. In 6 of the 11 evaluated datasets, LossSight (GAN) achieved the best results. For the mixed-pattern loss, LossSight has high stability. This shows that the strategy of LossSight of recovering by distinguishing different types of packet loss is effective. In addition, the distribution of packet loss patterns affects recovery performance. Different recovery algorithms have different recovery performances for different packet loss patterns. The recovery performance of LastVF and LossSight to congestion loss is better than random loss. The recovery performance of MeanVF, MedianVF and ComVF to random loss is better than that of congestion loss.

E. Performance Improvement for In-Band Network Telemetry

The switch queue depth is an important indicator for network congestion monitoring [59] or congestion control [12]. When network is severely congested, packet loss will cause that queue depth telemetry results are not available and, consequently, network administrators will misunderstand that network is in good condition. In this section, we evaluated the use of LossSight for monitoring the loss of INT packets carrying switch queue depth information.

The switch queue depth is 64. The switch adopts weighted random early detection (WRED) mechanism to avoid congestion. The lower threshold is 48 and the upper threshold is 64. When the queue depth is below the lower threshold, the drop probability is 0%. When the queue depth is between the lower and the upper thresholds, the drop probability is 50%. When the queue depth is higher than the upper threshold, the drop probability is 100%. The reference telemetry results are subject to the log records of P4 switch.

Figure 22 compares the queue depth information obtained from the sending of 1200 INT telemetry packets, with and without using LossSight, and considering 0.1% random loss + congestion loss. First of all, the INT telemetry results

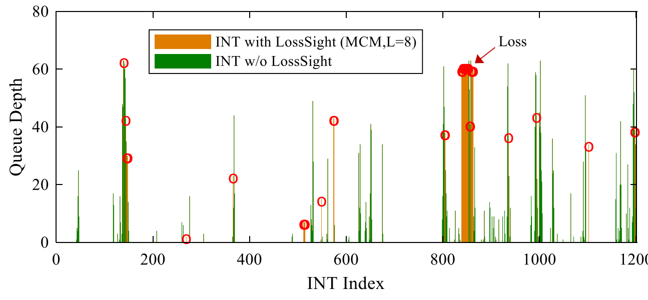


Fig. 22. The intuitive impact of LossSight on INT.

without LossSight are not comprehensive. The Telemetry Server receives 1,155 telemetry reports (96.25%). 45 telemetry reports (3.75%) are lost. Using LossSight, all INT packet losses are detected. The result of the diagnosis algorithm is 2 random losses (0.17%) and 43 congestion losses (3.58%). Secondly, for congestion loss (INT indices from 839 to 865 and a duration about 300 ms), the recovered values by LossSight are very close to the reference telemetry result. MSE is 21.3458.

In summary, LossSight can make up for the shortcomings of the existing INT system that cannot sense the loss of telemetry packets. On the basis of detecting and locating loss, LossSight can learn the feature distribution of telemetry results, and make the telemetry results close to the real ones by recovering.

F. Overhead

We analyzed the overhead in communication, memory and forwarding caused by LossSight and other existing solutions.

Compared with OpenNetMon [19], FlowRadar [20], LossRadar [21] and INT_DETECT [26], the overhead of LossSight is almost negligible. Because LossSight does not poll switches or construct probes, it only needs a few bits of INT telemetry packets to implement the loss detection. Compared with INT-XD and INT-MX, the number of INT packets reported by LossSight is only $1/n$, where n is the number of switches. Therefore, LossSight overhead is extremely low.

Using LossSight, each node maintains a counter for each flow, and the number of bits of the counter depends on the

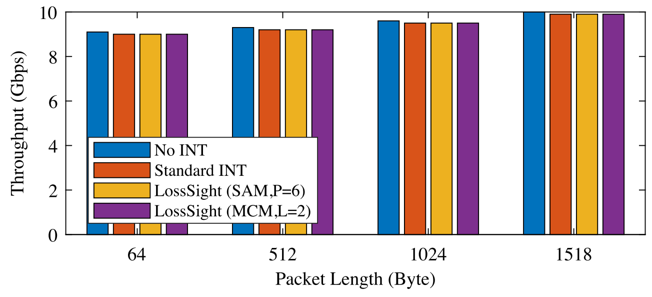


Fig. 23. Throughput measurement results.

period of SAM and MCM to perform 100% accuracy statistics [60]. Taking SAM as an example, when period is 8, the length of each counter is 3 bits. Therefore, 1 KB of memory supports 2730 flows and 4 KB memory supports 10922 flows. The memory requirement on an INT node only grows linearly with the number of telemetry flows instead with the number of telemetry packets.

As shown in Figure 23, we evaluated the effects of non-use of INT, standard INT, and LossSight on the packet processing capability of switches. The packet lengths were 64 bytes, 512 bytes, 1024 bytes, and 1518 bytes, respectively. We found that INT has very little impact on the packet forwarding capability. Compared with INT, LossSight has almost no impact on packet processing capabilities.

VI. Discussion

In this section, we would like to remark the idea that in-band network telemetry does not offer telemetry information through an isolated channel, but INT is affected by real network behavior. For that reason, researchers must pay attention to the impact of potential packet loss on network telemetry results.

After evaluating the proposed LossSight system, we identify three main aspects that will focus on our future work in order to improve LossSight potential.

Firstly, important characteristics of AM-PM method are its simplicity and efficiency to obtain measurements of network delay and packet loss. Making full use of coding rules of Loss Bit field to enable LossSight to carry more abundant measurement information is an interesting direction that should be studied.

In addition, how to recover the lost information more accurately is still an important aspect worthy of research for LossSight. It is necessary to investigate the deep-seated causes of the packet loss and analyze the correlation between the packet loss and the telemetry information to achieve an accurate recovery of the lost information. A precise knowledge of network topology and traffic characteristics is also relevant in this task.

Finally, it is necessary to expand the application scenarios of LossSight. A future line of work is the integration of LossSight in existing advanced network telemetry applications, including network measurement, network closed-loop control, and network management. It is important to further evaluate the performance improvement that LossSight system offers to these network telemetry applications. Since some network telemetry applications are not open source, we can only evaluate a few applications.

VII. Conclusion

In this paper, we have proposed, implemented, and evaluated the LossSight system, a packet loss monitoring system for in-band network telemetry. LossSight includes the functions of packet loss detection, location, diagnosis and recovery required for in-band network telemetry. LossSight solves the detection and measurement of INT packet loss by alternate-marking telemetry packets. From incomplete in-band network telemetry data, LossSight can automatically deduce and fill the lost telemetry information and output the complete telemetry information. Experiment results show that LossSight provides high detection and recovery accuracy with extremely low overhead, and can further improve the performance of network monitoring, control and management.

Acknowledgment

The authors are grateful to the editor and anonymous reviewers for helpful comments and suggestions that helped them improve the technical and presentation quality of the paper significantly.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] L. Tan *et al.*, “In-band network telemetry: A survey,” *Comput. Netw.*, vol. 186, Feb. 2021, Art. no. 107763.
[2] “In-Band Network Telemetry (INT) Dataplane Specification V2.1.” P4Lang/P4-Applications. 2020. [Online]. Available: https://github.com/p4lang/p4-applications/blob/master/docs
[3] “IOAM linkage solution for the protection cases of 5G bearer network,” Internet Eng. Task Force, Fremont, CA, USA, Internet-Draft draft-li-ioam-path-protection-00, 2020. [Online]. Available: https://datatracker.ietf.org/wg/ioam
[4] Y. Lin *et al.*, “NetView: Towards on-demand network-wide telemetry in the data center,” *Comput. Netw.*, vol. 180, Oct. 2020, Art. no. 107386.
[5] R. B. Basat, S. Ramanathan, Y. Li, G. Antichi, M. Yu, and M. Mitzenmacher, “PINT: Probabilistic in-band network telemetry,” in *Proc. SIGCOMM Virtual Event*, Jul. 2020, pp. 662–680.
[6] N. Kagami, R. I. T. da Costa Filho, and L. P. Gaspary, “CAPEST: Offloading network capacity and available bandwidth estimation to programmable data planes,” *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 1, pp. 175–189, Mar. 2020.
[7] T. Pan *et al.*, “INT-path: Towards optimal path planning for in-band network-wide telemetry,” in *Proc. INFOCOM*, Paris, France, Apr. 2019, pp. 487–495.
[8] T. Pan, E. Song, C. Jia, W. Cao, T. Huang, and B. Liu, “Lightweight network-wide telemetry without explicitly using probe packets,” in *Proc. INFOCOM WKSHPS*, Toronto, ON, Canada, Jul. 2020, pp. 1354–1355.
[9] N. Van Tu, J. Hyun, and J. W.-K. Hong, “Towards ONOS-based SDN monitoring using in-band network telemetry,” in *Proc. APNOMS*, 2017, pp. 76–81.
[10] N. Choi *et al.*, “Run-time performance monitoring, verification, and healing of end-to-end services,” in *Proc. NetSoft*, Paris, France, Aug. 2019, pp. 30–35.
[11] R. Hohemberger, A. F. Lorenzon, F. Rossi, and M. C. Luizelli, “Optimizing distributed network monitoring for NFV service chains,” *IEEE Commun. Lett.*, vol. 23, no. 8, pp. 1332–1336, Aug. 2019.
[12] Y. Li *et al.*, “HPCC: High precision congestion control,” in *Proc. SIGCOMM*, Beijing, China, Aug. 2019, pp. 44–58.

[13] N. Katta *et al.*, "Clove: Congestion-aware load balancing at the virtual edge," in *Proc. CoNEXT*, Dec. 2017, pp. 323–335.

[14] A. Karaagac, E. De Poorter, and J. Hoebeke, "Alternate marking-based network telemetry for industrial WSNs," in *Proc. WFCS*, Porto, Portugal, Apr. 2020, pp. 1–8.

[15] S. Nam, J. Lim, J.-H. Yoo, and J. W.-K. Hong, "Network anomaly detection based on in-band network telemetry with RNN," in *Proc. ICCE-Asia*, Nov. 2020, pp. 1–5.

[16] J. Hyun, N. Van Tu, and J. W.-K. Hong, "Towards knowledge-defined networking using in-band network telemetry," in *Proc. NOMS*, Apr. 2018, pp. 1–7.

[17] S. Tang, J. Kong, B. Niu, and Z. Zhu, "Programmable multilayer INT: An enabler for AI-assisted network automation," *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 26–32, Jan. 2020.

[18] B. Lu *et al.*, "iFIT: Intelligent flow information telemetry," in *Proc. SIGCOMM Posters Demos*, Beijing, China, Aug. 2019, pp. 15–17.

[19] N. L. Van Adrichem, C. Doerr, and F. A. Kuipers, "OpenNetMon: Network monitoring in OpenFlow software-defined networks," in *Proc. NOMS*, Krakow, Poland, May 2014, pp. 1–8.

[20] Y. Li, R. Miao, C. Kim, and M. Yu, "FlowRadar: A better NetFlow for data centers," in *Proc. NSDI*, Mar. 2016, pp. 311–324. [Online]. Available: https://www.usenix.org/conference/nsdi16/technical-sessions/presentation/li-yuliang

[21] Y. Li, R. Miao, C. Kim, and M. Yu, "LossRadar: Fast detection of lost packets in data center networks," in *Proc. CoNEXT*, 2016, pp. 481–495.

[22] C. Tan *et al.*, "NetBouncer: Active device and link failure localization in data center networks," in *Proc. NSDI*, Feb. 2019, pp. 599–614. [Online]. Available: https://www.usenix.org/conference/nsdi19/presentation/tan

[23] X. Zhang, Y. Wang, J. Zhang, L. Wang, and Y. Zhao, "RINGLM: A link-level packet loss monitoring solution for software-defined networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 8, pp. 1703–1720, Aug. 2019.

[24] M. Cociglio, G. Fioccola, G. Marchetto, A. Sapio, and R. Sisto, "Multipoint passive monitoring in packet networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 6, pp. 2377–2390, Dec. 2019.

[25] P. Loreti, A. Mayer, P. Lungaroni, S. Salsano, R. Gandhi, and C. Filsfils, "Implementation of accurate per-flow packet loss monitoring in segment routing over IPv6 networks," in *Proc. HPSR*, Newark, NJ, USA, May 2020, pp. 1–8.

[26] C. Jia *et al.*, "Rapid detection and localization of gray failures in data centers via in-band network telemetry," in *Proc. NOMS*, Budapest, Hungary, 2020, pp. 1–9.

[27] "The Demonstration Video of LossSight." 2020. [Online]. Available: https://www.bilibili.com/video/BV1CV411j7AX

[28] L. Tan, W. Su, J. Miao, and W. Zhang, "FindINT: Detect and locate the lost in-band network telemetry packet," *IEEE Netw. Lett.*, early access, Mar. 19, 2021, doi: 10.1109/LNET.2021.3067343.

[29] "LossSight." 2020. [Online]. Available: https://github.com/lzhtan/LossSight

[30] N. Van Tu, J. Hyun, G. Y. Kim, J.-H. Yoo, and J. W.-K. Hong, "Intcollector: A high-performance collector for in-band network telemetry," in *Proc. CNSM*, Rome, Italy, Nov. 2018, pp. 10–18.

[31] L. Tan, "LossSight Dataset." 2020. [Online]. Available: https://github.com/lzhtan/LossSight-Dataset

[32] C. Guo *et al.*, "Pingmesh: A large-scale system for data center network latency measurement and analysis," in *Proc. SIGCOMM*, London, U.K., Aug. 2015, pp. 139–152.

[33] R. Teixeira, R. Harrison, A. Gupta, and J. Rexford, "PacketScope: Monitoring the packet lifecycle inside a switch," in *Proc. SOSR*, San Jose, CA, USA, 2020, pp. 76–82.

[34] C. Fang *et al.*, "VTrace: Automatic diagnostic system for persistent packet loss in cloud-scale overlay network," in *Proc. SIGCOMM*, Aug. 2020, pp. 31–43.

[35] K. Xie *et al.*, "Neural tensor completion for accurate network monitoring," in *Proc. INFOCOM*, 2020, pp. 1688–1697.

[36] K. Xie *et al.*, "Accurate and fast recovery of network monitoring data: A GPU accelerated matrix completion," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 958–971, Jun. 2020.

[37] D. A. Popescu and A. W. Moore, "Measuring network conditions in data centers using the precision time protocol," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 3, pp. 3753–3770, Sep. 2021.

[38] W.-C. Lin and C.-F. Tsai, "Missing value imputation: A review and analysis of the literature (2006–2017)," *Artif. Intell. Rev.*, vol. 53, no. 2, pp. 1487–1509, 2020.

[39] G. E. Batista and M. C. Monard, "An analysis of four missing data treatment methods for supervised learning," *Appl. Artif. Intell.*, vol. 17, nos. 5–6, pp. 519–533, 2003.

[40] G. Folino and F. S. Pisani, "Evolving meta-ensemble of classifiers for handling incomplete and unbalanced datasets in the cyber security domain," *Appl. Soft Comput.*, vol. 47, pp. 179–190, Oct. 2016.

[41] C. Févotte and J. Idier, "Algorithms for nonnegative matrix factorization with the β-divergence," *Neural Comput.*, vol. 23, no. 9, pp. 2421–2456, Sep. 2011.

[42] M. Roughan, Y. Zhang, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and Internet traffic matrices," *IEEE/ACM Trans. Netw.*, vol. 20, no. 3, pp. 662–676, Jun. 2012.

[43] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–772, 2009.

[44] Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Math. Program. Comput.*, vol. 4, no. 4, pp. 333–361, 2012.

[45] Z. Zhang, C. Ling, H. He, and L. Qi, "A tensor train approach for Internet traffic data completion," *Ann. Oper. Res.*, pp. 1–19, Jun. 2021. [Online]. Available: https://link.springer.com/article/10.1007/s10479-021-04147-4

[46] R. Koodli and R. Ravikanth, "One-way loss pattern sample metrics," Internet Eng. Task Force, Fremont, CA, USA, RFC 3357, 2002.

[47] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and modelling of the temporal dependence in packet loss," in *Proc. INFOCOM*, vol. 1, Mar. 1999, pp. 345–352.

[48] S. Y. Han, N. B. Abu-Ghazaleh, and D. Lee, "Efficient and consistent path loss model for mobile network simulation," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1774–1786, Jun. 2016.

[49] A. Riesenberg, Y. Kirzon, M. Bunin, E. Galili, G. Navon, and T. Mizrahi, "Time-multiplexed parsing in marking-based network telemetry," in *Proc. SYSTOR*, 2019, pp. 80–85.

[50] G. Fioccola *et al.*, "Alternate-marking method for passive and hybrid performance monitoring," Internet Eng. Task Force, Fremont, CA, USA, RFC 8321, 2018.

[51] T. Mizrahi, G. Navon, G. Fioccola, M. Cociglio, M. Chen, and G. Mirsky, "AM-PM: Efficient network telemetry using alternate marking," *IEEE Netw.*, vol. 33, no. 4, pp. 155–161, Jul./Aug. 2019.

[52] S. Tang, D. Li, B. Niu, J. Peng, and Z. Zhu, "Sel-INT: A runtime-programmable selective in-band network telemetry system," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 2, pp. 708–721, Jun. 2020.

[53] P. Tammana, R. Agarwal, and M. Lee, "Simplifying datacenter network debugging with pathdump," in *Proc. NSDI*, Savannah, GA, USA, 2016, pp. 233–248.

[54] T. Benson, A. Anand, A. Akella, and M. Zhang, "Understanding data center traffic characteristics," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 1, pp. 92–99, 2010.

[55] Y. Zhu *et al.*, "Packet-level telemetry in large datacenter networks," in *Proc. SIGCOMM*, Aug. 2015, pp. 479–491.

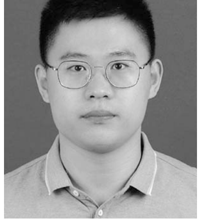
[56] D. Shan, F. Ren, P. Cheng, R. Shu, and C. Guo, "Observing and mitigating micro-burst traffic in data center networks," *IEEE/ACM Trans. Netw.*, vol. 28, no. 1, pp. 98–111, Feb. 2020.

[57] Q. Zheng, S. Tang, B. Chen, and Z. Zhu, "Highly-efficient and adaptive network monitoring: When INT meets segment routing," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 3, pp. 2587–2597, Sep. 2021.

[58] J. Yoon, J. Jordon, and M. van der Schaar, "GAIN: Missing data imputation using generative adversarial nets," in *Proc. ICML*, Jul. 2018, pp. 5689–5698. [Online]. Available: http://proceedings.mlr.press/v80/yoon18a.html

[59] D. Shan, F. Ren, P. Cheng, R. Shu, and C. Guo, "Micro-burst in data centers: Observations, analysis, and mitigations," in *Proc. ICNP*, Cambridge, U.K., Sep. 2018, pp. 88–98.

[60] Q. Huang, H. Sun, P. P. C. Lee, W. Bai, F. Zhu, and Y. Bao, "Omnimon: Re-architecting network telemetry with resource efficiency and full accuracy," in *Proc. SIGCOMM*, 2020, pp. 404–421.



Lizhuang Tan (Student Member, IEEE) received the B.S. degree in communication engineering from the College of Information Science and Engineering, Shandong Normal University, Jinan, China, in 2017. He is currently pursuing the Ph.D. degree with the National Engineering Laboratory for Next Generation Internet Interconnection Devices, Beijing Jiaotong University, China. His research interests include software-defined networking and deterministic network. He is a Student Member of CCF.

Wei Su received the B.S., M.S., and Ph.D. degrees in communication and information systems from Beijing Jiaotong University in 2001, 2004, and 2008, respectively, where he is a Full Professor with the School of Electronic and Information Engineering. He has studied Internet for more than 20 years. He was selected for the Beijing Young Talents Program in 2013. He was a Visiting Scholar with Future University, Hakodate, Japan in 2015. He has published more than 50 research papers and four monographs in areas of communications and computer networks. His research interests are next generation network and mobile Internet. He won the Second Prize of the National Technology Invention Award in 2014.

Wei Zhang (Member, IEEE) received the B.E. degree from Zhejiang University in 2004, the M.S. degree from Liaoning University in 2008, and the Ph.D. degree from the Shandong University of Science and Technology in 2018. He is currently a Professor with the Shandong Provincial Key Laboratory of Computer Networks, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan, China. His research interests include future generation network architectures, edge computing, and edge intelligence.

Huiling Shi received the M.S. degree from Shandong University in 2004. She is currently an Associate Researcher with the Shandong Provincial Key Laboratory of Computer Networks, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan, China. Her research interests include network architectures and edge computing.

Jingying Miao received the B.S. degree in communication engineering from Beijing Jiaotong University, China, in 2019. She is currently pursuing the M.S. degree with the National Engineering Laboratory for Next Generation Internet Interconnection Devices, Beijing Jiaotong University. Her research interests are software-defined networking and data plane programmable technology.

Pilar Manzanares-Lopez received the M.S. degree in telecommunication engineering from the Universidad Politécnica de Valencia, Spain, in 2001, and the Ph.D. degree in telecommunication engineering from the Universidad Politécnica de Cartagena (UPCT), Spain, in 2006. She started working as an Assistant Professor with UPCT, in 2001, where she currently works as an Associate Professor with the Department of Information and Communication Technologies. Her research interests include software-defined networking, programmable data planes, P2P networks, and distributed systems.