Detecting Ephemeral Optical Events with OpTel

Congcong Miao[1], Minggang Chen[1], Arpit Gupta[2], Zili Meng[3], Lianjin Ye[3], Jingyu Xiao[3],
Jie Chen[1], Zekun He[1], Xulong Luo[1], Jilong Wang [3,4,5], Heng Yu[3]
[1]*Tencent,* [2]*UC Santa Barbara,* [3]*Tsinghua University,* [4] *BNRist,* [5] *Peng Cheng Laboratory*

Abstract

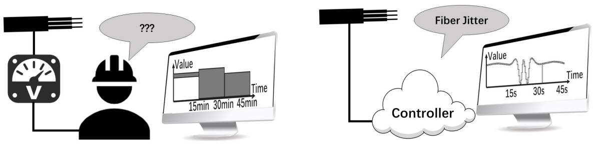
Degradation or failure events in optical backbone networks affect the service level agreements for cloud services. It is critical to detect and troubleshoot these events promptly to minimize their impact. Existing telemetry systems rely on arcane tools (e.g., SNMP) and vendor-specific controllers to collect optical data, which affects both the flexibility and scale of these systems. As a result, they fail to collect the required data on time to detect and troubleshoot degradation or failure events in a timely fashion. This paper presents the design and implementation of OpTel, an optical telemetry system that uses a centralized vendor-agnostic controller to collect optical data in a streaming fashion. More specifically, it offers flexible vendor-agnostic interfaces between the optical devices and the controller and offloads data-management tasks (e.g., creating a queryable database) from the devices to the controller. As a result, OpTel enables the collection of fine-grained optical telemetry data at the one-second granularity. It has been running in Tencent's optical backbone network for the past six months. The fine-grained data collection enables the detection of short-lived events (i.e., ephemeral events). Compared to existing telemetry systems, OpTel accurately detects 2× more optical events. It also enables troubleshooting of these optical events in a few seconds, which is orders of magnitude faster than the state-of-the-art.

1 Introduction

Cloud service providers, such as Google, Microsoft, and Tencent, have embraced the approach of setting up as many data centers as possible across metro areas [6, 20, 21, 23, 26, 38]. Such an approach enables cloud providers to physically get closer to the end-users, which in turn enables a wide range of applications with diverse bandwidth and latency requirements [29, 45]. The optical backbone network that interconnects these geographically distributed data centers is critical for ensuring reliable exchange of terabits of data every day [2, 3, 24, 25, 27]. Under the hood, the optical backbone network is composed of optical hardware (e.g., optical transponders, amplifiers, wavelength (de-)multiplexers), and fiber cables. Degradation or failure of any of these components (i.e., optical events) would degrade the inter-DC connectivity, which in turn affects the service level agreements (SLAs) for cloud services [5,18,20,49]. Therefore, to improve the reliability and availability of the optical backbone network, it is critical to promptly detect and troubleshoot optical events.

Unfortunately, existing telemetry systems are not designed for such fast-paced detection and troubleshooting of optical events. More concretely, they collect sampled or aggregated data from optical devices. Such coarse-grained data is not suited for either detecting short-lived optical events or troubleshooting related optical events to various stakeholders (i.e., application developers, data center tenants, etc.). Figure 1(a) illustrates the limitations of existing telemetry systems. Here, when a customer reports degradation in the quality of experience for video streaming service (e.g., rebuffering), attributable to a short-lived optical event lasting few tens of seconds. The network operator that looks at the telemetry data collected by the existing telemetry systems at the 15-minute granularity cannot detect or troubleshoot such a short-lived optical event. The current telemetry systems are slow in detecting and troubleshooting the more disruptive persistent events as well. Network operators need to query data from multiple vendor-specific controllers to stitch a holistic view of the underlying network, which is tedious and prone to errors. Our analysis of the trouble tickets dataset shows that it takes hours to days to troubleshoot the optical hardware failures. Though we witnessed the development of various network telemetry systems, such as Sonata [19], Marple [33], PathDump [42], OmniMon [22], etc., that offer packet-level network streaming analytics at scale, they are not suited for diagnosing degradation or failure events in optical networks.

The limitations of the existing telemetry systems are attributable to three key factors. First, the optical backbone network uses devices from multiple vendors (i.e., vendor-free optical systems in § 2.1), and the current telemetry systems develop interfaces for vendor-specific controllers to ac-



(a) Existing telemetry system

(b) OpTel

Figure 1: **(a) Existing telemetry systems fail to detect ephemeral events and are slow in detecting and troubleshooting persistent events. (b) OpTel detects and troubleshoots both ephemeral and persistent events promptly with the one-second granularity data.**

cess the optical data. Though vendor diversity is critical for cloud providers to deter vendor monopolies and avoid concurrent failures, *fragmented design* of existing telemetry systems is undesirable. It inhibits accessing optical data directly or extracting a consistent network view. Second, the existing telemetry systems rely on *arcane tool*, i.e., SNMP, to collect data from different devices. SNMP performs various data-management tasks, such as creating a local MIB database [35], supporting read and write operations to this database, etc., locally on the optical devices. Both faster reads (queries) and writes to this database will cause higher CPU usage. Given the limited resources at the device, it is not possible to query this data at higher frequencies with the SNMP protocol. Third, the vendor-specific controllers run on physical servers with fixed compute and memory resources. Such *inelastic resource allocation* for the existing telemetry pipelines creates multiple bottlenecks with the increasing number of optical devices or data-collection frequencies.

In this paper, we present the design and implementation of OpTel (Figure 1(b)), an optical telemetry system for optical networks. The proposed system offers direct access to optical data in a vendor-agnostic manner and offloads data-management tasks from the optical devices to cloud-based controllers that can easily scale with network size and collection frequency. We highlight the salient feature of the proposed system below.

Vendor-agnostic centralized control. OpTel shunts away vendor-specific controllers and replaces them with a single centralized controller that directly interfaces with optical devices in a vendor-agnostic manner. To enable such a vendor-agnostic design, we develop a standardized model for optical devices. This device-level model consists of two essential parts: logic and data model. Here, the logic model identifies key components common across devices from different vendors and standardizes their workflow. The data model specifies the configurable parameters for each component.

Streamline telemetry pipeline at optical devices. OpTel replaces SNMP (pull-based) protocol with a “push-based" telemetry pipeline. More concretely, it offloads the compute-intense data-management tasks from the optical devices to cloud-based controllers, with access to an elastic pool of resources. Such streamlining of the telemetry pipeline offloads resource-intense operations to the cloud, enabling OpTel to collect fine-grained optical data at higher frequencies from resource-constrained optical devices. The telemetry pipeline at optical devices consists of the following key parts: telemetry manager, telemetry agent, cache, and aggregator. Here, the telemetry manager interfaces with the centralized controller and is responsible for receiving configurations from the controller and configuring other parts. The telemetry agent reads data from different modules and stores them into the local cache. The aggregator is responsible for pushing the data in the cache to the centralized controller.

The rest of the paper presents the background and motivation in Section 2, details the design and implementation in Section 3. We demonstrate how OpTel enables collecting fine-grained telemetry data at the one-second granularity and how such a dataset empowers network operators to promptly detect and troubleshoot optical events, both persistent and ephemeral, in Section 4. We have been running OpTel in Tencent’s optical backbone network for the past six months. We report our experience of collecting and analyzing the telemetry data at scale. Notably, we demonstrate that access to such fine-grained data enables us to establish temporal relationships between different optical events.

2 Background and Motivation

We first provide an overview of the optical backbone network (§ 2.1). We then discuss why existing telemetry systems fail to promptly detect and troubleshoot optical events (§ 2.2).

2.1 Optical Backbone Network

The optical backbone network interconnects different data center sites, carrying terabytes of traffic each day. Figure 2 zooms-in into a specific link (i.e., an optical transport system) interconnecting two data center sites. Each link consists of an optical line system (OLS) and multiple optical transponder units (OTUs). Each OTU receives the electrical signal from the data center router (DR), and converts it into a specific wavelength, called an optical *channel*, and vice versa. When router ports have a lower capacity than the optical channel, the OTU encapsulates and multiplexes multiple router ports onto the channel.

The OLS contains two optical *segments*, one for each direction of network traffic. Each segment carries multiple optical channels, with wavelength division multiplexing/demultiplexing (MUX/DMUX) combining/splitting these channels and booster amplifier (BA) at the transmitting end and preamplifier (PA) at the receiving end. Segments also have in-line amplifiers (LAs) that amplify the signal in the optical domain to deal with long-haul transmission loss. Each part of the segment is called a *span*. As a special case, segment yields span if the OLS does not have LA. Optical Supervisory

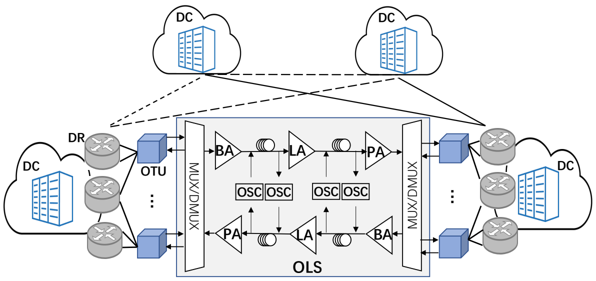


Figure 2: **An overview of the optical backbone network.**

Channel (OSC) is an additional channel that does not carry any payload traffic and monitors each span.

Most cloud providers use optical devices from multiple vendors. Vendor diversity is intentional to deter vendor monopolies and avoid concurrent failures. Typically, cloud providers, including Tencent, embrace a *vendor-free design* of the optical transport system (i.e., vendor-free optical system), where they purchase optical line systems and optical transponder units from different vendors [11].

2.2 Monitoring Optical Backbone Network

Any degradation or failure events in the optical backbone network can affect the SLAs for various cloud services. Thus, it is critical for network operators to promptly detect and diagnose such optical events, which in turn requires collecting fine-grained optical data from the underlying optical devices at high frequency. The existing telemetry systems are not designed to support such intense data-collection requirements. We identify three key factors that inhibit existing telemetry systems to scale flexible data collection.

Highly fragmented design. The control plane for most optical backbone networks is highly fragmented as it relies on vendor-specific controllers to manage individual devices. The existing telemetry systems inherited this fragmented design, where a centralized controller interfaces with vendor-specific controllers to collect the required telemetry data. Such a fragmented design inhibits flexible and direct access to fine-grained optical data at scale. Each vendor-specific subsystem implements its workflow to collect the data from individual devices, affecting how frequently each subsystem reports the telemetry data. Additionally, the data schemas across vendors are different, which further inhibits supporting a consistent representation of the collected data.

To illustrate the impact of each of these factors, we use the metric, *polling delay*, which measures the difference in time when the centralized controller sends the poll request to vendor-specific controllers and when it receives the requested data. We have performed the measurement studies of two subsystems provided by vendor 1 and vendor 2. For confidentiality, we omit the vendor name. We observe it takes about 3 minutes and 7 minutes to complete the collection of 5 indicators from vendor 1 and vendor 2 respectively. Here, each indicator represents the type of data, such as `SNR`, `Q-factor`,

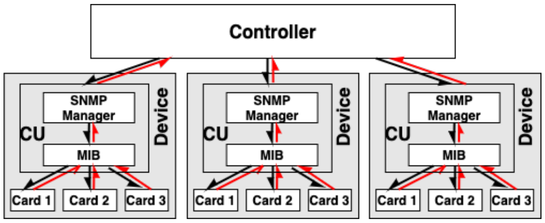
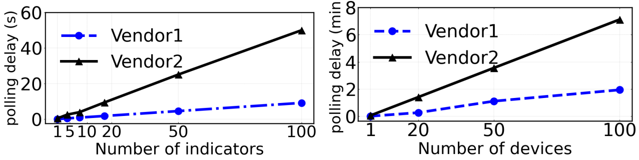


Figure 3: **SNMP's data collection workflow**



(a) Performance of devices

(b) Performance of controllers

Figure 4: **Performance of optical devices and vendor-specific controllers from two vendors.**

etc., collected from the devices. The difference in polling delay across two vendors is attributable to an artifact of different data-collection workflow each applied within its subsystem. Such high variance in polling delays across different vendors makes it hard for network operators to extract a consistent (synchronized) view of the network, affecting their ability to troubleshoot various optical events.

Reliance on arcane data-collection tools. Most existing telemetry systems for optical backbone networks rely on SNMP [10], which is not suited for high-frequency data collection. SNMP performs various data-management tasks locally on the optical device. More concretely, it creates and updates a local queryable database (MIB) on the device, and handles controller's queries. Figure 3 shows SNMP's data collection workflow. Here, to simplify exposition, we divide the optical device into control and data plane. Here, the control plane consists of SNMP manager and MIBs and the data plane comprises of multiple line cards. The black and red arrows represent the control and data flows respectively. Once the SNMP manager receives an SNMP GET request from the controller, it traverses the table in MIB database [35] one by one to get the `function` to obtain the data from the line card and then reports the requested data. This process is slow and consumes a significant number of CPU cycles, making it difficult to scale data collection frequency with SNMP.

Figure 4(a) shows how polling delay changes as the number of indicators increases. We observe that the relationship between polling delay and the number of indicators is linear. Our interactions with vendors revealed that this linear relationship is attributable to their choice of serializing read request for multiple indicators, reading only one at a time. This design choice limits SNMP's CPU usage, which competes with device's data-plane operations. Such a long polling delay with SNMP inhibits existing telemetry systems to collect fine-grained optical data at higher frequencies.

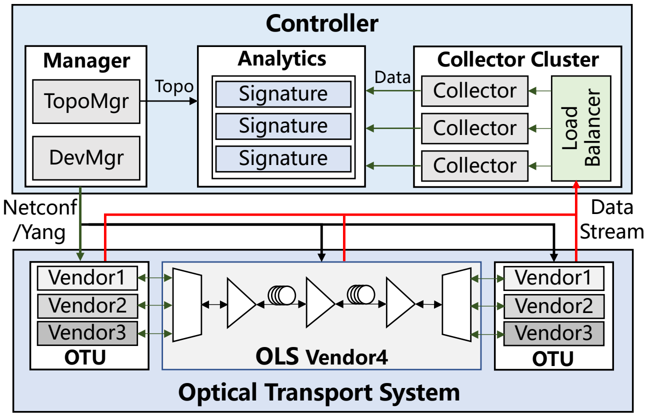


Figure 5: **Architecture of OpTel.**

Inelastic resource allocation for the telemetry pipeline. Telemetry systems need to concurrently collect data from all the underlying optical devices to ensure a consistent and fine-grained view of the network. However, we observe that the vendor-specific controllers run on physical servers with fixed compute and memory resources. Such an inelastic design makes these controllers a bottleneck in existing telemetry pipelines as the number of optical devices or the collection frequency increases.

Unsurprisingly, we observe a linear relationship between polling delay and the number of devices in Figure 4(b). This behavior is also attributable to vendors' choice to serialize requests at the controller. Such serialization ensures that the controller can handle all the incoming requests with a fixed set of resources at the cost of longer polling delays, which affects the ability to construct a consistent view of the network at fine time scales. More concretely, it is impossible to correlate the optical data across two different optical devices on a short time scale, affecting the troubleshooting capabilities of the existing telemetry systems.

3 OpTel's Design and Implementation

We now describe how the proposed system, OpTel, addresses the limitations of existing telemetry systems described above. We first state its design goals in Section § 3.1, and then describe how it achieves these goals in Section § 3.2 and § 3.3.

3.1 Design Goals

OpTel's goal is to extract multiple indicators from all the devices in the optical backbone at finer time granularities, i.e., order of seconds. Such a dataset is critical for timely detection and diagnosis of various disruptive events in the backbone network. OpTel addresses the limitations of existing telemetry systems to achieve this goal. More concretely, to address the fragmentation issue, it bypasses vendor-specific controllers to collect the telemetry data directly from the optical devices in a vendor-agnostic manner. To address the scalability issues, it streamlines the telemetry pipeline such that it performs all the complex data-management tasks to a centralized controller running in the cloud. Such a design ensures that the data collection pipeline is not bottlenecked by limited compute resources at the individual devices. The centralized controller has access to an elastic pool of resources in the cloud.

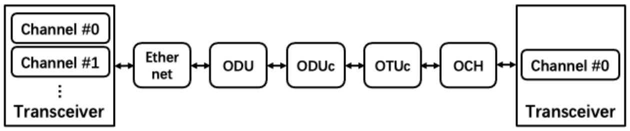


Figure 6: **The logic model of OTU.**

3.2 Vendor-agnostic Centralized Control

Figure 5 presents OpTel's architecture. Here, the centralized controller directly interfaces with the optical devices in a vendor-agnostic manner. We developed a standardized model that abstracts away the vendor-specific details for the controller. We now describe how we develop the vendor-agnostic device model and how it enables collecting data directly from the optical devices.

3.2.1 Standardized Model for Optical Devices

In vendor-free optical systems, the operation performed by different optical devices is similar at a high level, but the specific logic and workflow vary across vendors. Such heterogeneity across devices from different vendors complicates the design of vendor-agnostic interfaces. We develop a standardized model for optical devices that abstracts away the vendor-specific details to address this challenge. It consists of two parts: *logic model* and *data model*. Here, the logic model identifies key components common across devices from different vendors and standardizes their workflow. The data model specifies the configurable parameters for each component.

Logic model. The first challenge in developing vendor-agnostic interfaces is that the physical components and their workflow are proprietary to each vendor. To address this challenge, the logic model first identifies a group of logical components that are common across devices from different vendors. It then standardizes the workflow between these components. To illustrate, consider the case of optical transponder units, i.e., OTUs. Figure 6 shows OTU's logic model. Here, the logic model first identifies four logical components across all vendors: Ethernet, optical data unit (ODU, ODUc), optical transport unit (OTUc), and optical channel (OCH). Recall that an OTU encapsulates and multiplexes multiple router ports onto an optical channel. The ODUc is a high-order data unit after combining the payload data from multiple router ports. The logic model then specifies the workflow between these components. For example, the mapping between Ethernet and ODU represents an encapsulation of an Ethernet frame into an ODU frame. Such an abstraction enables the standardized representation of different optical devices.

Data model. The second challenge in enabling vendor-agnostic interfaces is that the capability of physical components inside the device is different across vendors, although their functions are the same. For example, the range of gain of an optical amplifier provided by vendor 1 is 15-25 dB, while it might be 20-30 dB from vendor 2. This heterogeneity complicates managing these devices in a vendor-agnostic manner. We design a component data model with specific descriptions of configurable parameters of each component. When each device connects to the controller, the controller obtains the specification datasheets from the device and initializes the corresponding value of configurable parameters. Such an approach simplifies the management complexity of heterogeneous devices, regardless of the capability of physical components inside the device.

We have developed a model for each device type for our optical backbone network. Our experience using these models in production settings was smooth, demonstrating their generalizability.

3.2.2 Centralized Data Collection

The standardized model allows the centralized controller to access the telemetry data directly, enabling OpTel to shunt away vendor-specific controllers. The centralized controller consists of three key modules: *global manager*, *scalable collector* and *real-time analytics*, to perform detecting and troubleshooting optical events at scale in a timely manner.

Global manager. It consists of two parts: device manager (DevMgr) and topology manager (TopoMgr). The DevMgr is responsible for configuring the underlying optical devices. For each device, it leverages the relevant standardized model to configure devices in a vendor-agnostic manner. It completes this process by issuing a Yang file [7] to the device through the vendor-neutral Netconf protocol [13]. The TopoMgr maintains a physical topology of optical devices to provide a network-wide view of the optical networks and thus helps the real-time analytics to troubleshoot the optical events at scale. To illustrate how TopoMgr aids troubleshooting, consider the case of degradation in a fiber cable. Here, as it is not possible to directly collect the data from the cable, the analytics can instead use the TopoMgr to identify the two terminal devices at each end of the cable. It can then query the transmit (Tx) and receive (Rx) power data from these devices for troubleshooting.

Scalable collector. This module is a cluster of multiple collector nodes designed to handle changes in the number of indicators, collection frequency, or the number of optical devices over time. With the aid of the cloud's elastic pool of resources, it can scale horizontally by adding (or removing) collector nodes over time. It relies on a load balancer to distribute the load among individual collectors within the cluster. It is robust against the failure of a particular collector node.

Real-time analytics. It performs the task of promptly detecting and troubleshooting optical events by combining the optical data from the collector and the topology information from the global manager. The workflow of real-time analytics consists of two parts: detection and troubleshooting. To detect degradation or failure events, it monitors the values of optical data in real-time and raises the alarm if the value exceeds a pre-specified threshold. In parallel, it starts the troubleshooting process. Rather than manually troubleshooting the optical event, it leverages the signatures of previous optical events for diagnosis. To illustrate, consider the case when the received optical power becomes zero for a device. The analytics module first raises the alarm with a message, *the receiver can not receive the light* and then begins troubleshooting. It matches the collected data with previous signatures. If it finds the match, it simply sends troubleshooting report to the operator. If the collected data does not match a pre-existing signature, it lets operators manually express their queries for troubleshooting. It automatically updates the relevant signatures for future events. Our deployment experience shows that is possible to troubleshoot most of the optical events using existing signatures.

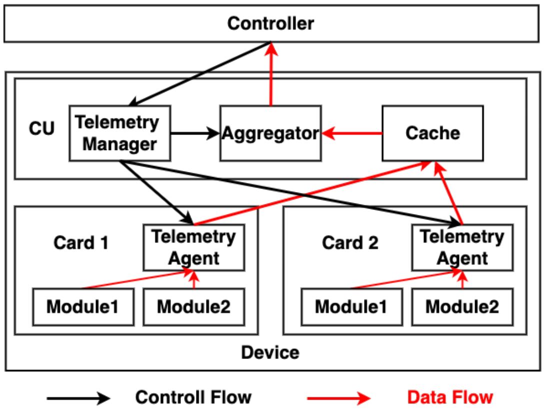


Figure 7: **Push-based optical telemetry**

3.3 Streamlined Telemetry Pipeline

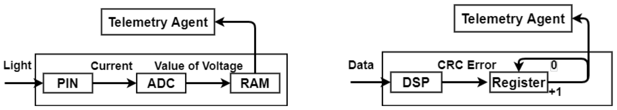
Promptly detecting and troubleshooting optical events requires collecting fine-grained data from the underlying optical devices. The widely used SNMP is flawed in performance because it performs various data-management tasks locally on the resource-constrained optical devices (Figure 3). In contrast, OpTel offloads compute-intense operations from the optical devices to the centralized controllers by push-based telemetry pipeline, enabling to collect the fine-grained optical data at higher frequencies. Figure 7 depicts the architecture of push-based optical telemetry. The telemetry pipeline at optical devices consists of the following key parts: *telemetry manager*, *cache* and *aggregator* in the control unit (CU), and *telemetry agents* in the line cards. The telemetry manager is responsible for the configurations of other parts, i.e., telemetry agent and aggregator. The telemetry agent reads data from different modules and stores them into the local cache. The aggregator is responsible for pushing the data in the cache to the centralized controller. In the following, we will describe them in detail.

Telemetry manager. The offloading of compute-intense data-management tasks from the optical devices to the controller requires preliminary configurations at the device. The telemetry manager firstly interfaces with the centralized controller to obtain the YANG file [7] and then parses the YANG file to configure the telemetry agent and aggregator. The aggregator is configured to periodically initiate a connection to push the optical data from the local cache to the controller. As for the telemetry agent in the line card, it is configured in three parts: the destination of data (i.e., cache), the source of data (i.e., modules in the line card), and the periodicity that the telemetry agent should push the data.

However, based on the real-world deployment experiences, we observed that configuring the same periodicity for pushing data at the telemetry agent and aggregator may result in the frequent data loss in the controller. This phenomenon is attributable to the different timing mechanisms. Generally, the CU always runs a Linux operating system and enables network time protocol (NTP) [30] to keep timing. However, some line cards are the embedded equipment without running a Linux operating system, resulting in it being unable to keep timing through NTP. Thus, these line cards keep timing through the crystal oscillator. The frequency deviation inside the crystal oscillator will lead to the timing inaccuracies [44]. Therefore, the performance data pushed by the telemetry agent is not strictly periodic. Slower timing will result in the data not being stored in the local cache, which in turn causes data loss in the controller. For example, assume that the controller needs to collect the data from the device at the one-second granularity. The telemetry manager configures the telemetry agent and aggregator to push the data every one second. However, the frequency deviation inside the crystal oscillator may result in the timing in the line card slower than that in the CU. It will take more than one second for the telemetry agent to push the data to the local cache. Therefore, the aggregator will push the empty data to the controller. Motivated by this, we always configure the data pushed in the telemetry agent at a higher frequency.

Telemetry agent. Once configured, the telemetry agent periodically performs the card-level data collection through the vendor-specific protocols and pushes the data to the local cache. Specifically, the values of data are generated in two ways: instant value and accumulated value.

Instant Value. It is a sampled data in a given time interval. The receiver captures the physical analog signal and then translates it into the digital value, which is further stored in the RAM. Figure 8(a) describes the process of generating instant value of the received signal in the physical layer. The PIN photodiode firstly captures the light signal and transforms it into the analog current. An analog-to-digital converter (ADC) is applied to convert the analog current into a digital value of voltage which is further stored in the RAM. The telemetry agent periodically reads RAM to collect the data through the vendor-specific protocol. Note that the value in RAM will be replaced frequently, thus enables the data to be collected at higher frequencies. In our work, the instant value records the performance in the physical layer. We use transmit/receive (Tx/Rx) power to detect optical events, and signal-to-noise ratio (SNR) and quality factor (Q-factor) to check the ability of the optical system to transmit data (Figure 9(a) and 9(b)).



(a) Instant Value (b) Accumulated Value

Figure 8: **The process of generating specific values. (a) Instant value records the performance in the physical layer; (b) Accumulated value records the performance in the data link and network layer.**

Accumulated Value. It is a counting value accumulated across the whole timeline. The digital signal processor (DSP) processes the received digital signal and counts in a certain way. Figure 8(b) describes the process of generating the accumulated value of the CRC error. The register counts the volume of CRC errors in the whole time interval. The telemetry agent periodically reads the register to collect the value. After that, the register will be reset to its initial value. The accumulated value records the performance in the data link and network layer, such as CRC error and post forward error correction (FEC). We use them to differentiate optical events according to the influence of optical events in the data link and network layer (Figure 9(c),9(d)).

Cache. The local cache serves as data storage that stores the performance data received from the telemetry agent and then bundles data at the device level. It is compatible with the performance data pushed by the different agents at different frequencies. Generally, the data for a single indicator stored in the telemetry cache is more fine-grained since the frequency of the telemetry agent pushing the data is higher than that of the aggregator reading the data. The data in the local cache will be cleaned after being read.

Aggregator. The aggregator periodically initiates a connection to get the bulked data from the local cache. Since the data provided by the cache is more fine-grained, the aggregator should merge the data to get representative statistics and push them to the controller through the gRPC protocol [1].

4 Evaluation

OpTel has been running in Tencent's backbone network for the past six months, demonstrating its deployability in production settings. In this section, we show how the proposed streamlined telemetry pipeline enables collecting all possible indicators from all the optical devices in the network at the one-second frequency (Section 4.2). We then show how such fine-grained data enables the detection of ephemeral optical events (Section 4.3). We investigate how ephemeral events help predict more disruptive future events, illustrating the utility of such a fine-grained telemetry system (Section 4.4). We also demonstrate how such fine-grained data enables troubleshooting optical events in the order of few seconds, which is orders of magnitude faster than possible with the existing telemetry systems (Section 4.5).

4.1 Setup

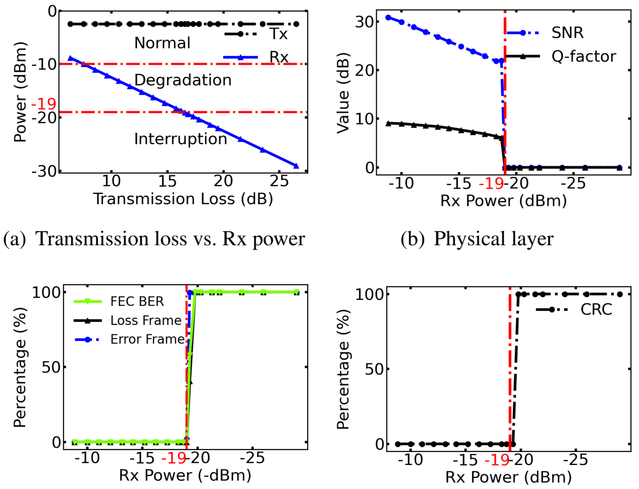
4.1.1 Dataset

We use OpTel to curate three datasets. Here, we collect the data for six months (July-December, 2020) from Tencent's optical backbone network. This backbone has O(50) links, O(100) spans, O(100) segments, O(1000) optical channels, and O(1000) optical devices from O(10) vendors. For confidentiality reasons, we do not report the exact numbers.

Optical telemetry dataset. We curate this dataset by collecting all indicators from all the optical devices at one-second granularity using OpTel. We collect the Tx/Rx power levels, SNR, and Q-factor from the physical layer. From the data link layer, we collect the Post Forward Error Correction Bit Error Rate (FEC BER) [31], loss and error frame rate (i.e., the ratio of the number of Rx vs. Tx frames and Error vs. Rx frames). We also collect cyclic redundancy check (CRC) error rate [40] from the network layer. Here, the physical layer indicators are "instant" values, and the rest are the "accumulated" values (§ 3.3). Also, note that since the OTU encapsulates and multiplexes payload from router ports, we collect the data link and network layer indicators directly from the OTU (§ 3.2.1). However, it is not efficient to only focus on the Tx/Rx power in OTU or BA/PA to troubleshoot optical events. For example, if there are several spans and LAs in a segment, and the Rx power of PA becomes 0 while the Tx power of BA does not change, we can not distinguish which span is responsible for the event. Thus, we combine the Tx/Rx power of OSC for span-level monitoring. The detailed origins of telemetry data are shown in Figure 17 in appendix A.

Location dataset. We use OpTel's TopoMgr to curate this dataset. It maintains a topology of the devices to provide a network-wide view to establish a relationship between different devices. Such relationships are critical for troubleshooting as indicators from a single device are often not enough to diagnose various optical events. For example, diagnosing degradation events in fiber cables requires data from both ends of the fiber cable.

Trouble tickets dataset. We collect this data from the network management platform at Tencent. We first filter out the events related to the optical networks (see appendix B for details) and then categorize these optical events into a small number of classes, i.e., fiber cable, hardware, and power events. Each ticket contains a timestamp recording the event's start time with detailed messages and a corresponding timestamp recording the localization of the event, i.e., event name (e.g., optical fiber jitter, amplifier instability, etc.). Note, troubleshooting optical events requires much manual effort in existing telemetry systems. We use this data to learn signatures of different optical events and show the time efficiency of OpTel on troubleshooting optical events by comparing it with the existing telemetry system.



(a) Transmission loss vs. Rx power

(b) Physical layer

(c) Data link layer

(d) Network layer

Figure 9: **An example of the physical/data link/network layer behaviors with the increase of transmission loss.**

4.1.2 Optical Events

We now present how we categorize optical events on the basis of their impact (degradation vs. interruption events) and duration (ephemeral vs. persistent events).

Interruption vs. Degradation events. To categorize optical events on the basis of their impact, we investigate the relationship between indicators at the physical, data link, and network layer (see Figure 9). Specifically, we take an optical transport system as an example, and fix the Tx power and iteratively adjust the transmission loss of optical fiber to simulate the degradation/failure event. Figure 9(a) shows a linear relationship between the transmission loss and values of Rx power in OTU based on the formula *Rx power (dBm) = Tx power (dBm) - transmission loss (dB)*. We observe similar trends in PA (not shown for brevity). After establishing the relationship between transmission loss and Rx power, we study how degradation in the power level at the receiver affects SNR and Q-factor at the physical layer (Figure 9(b)); FEC BER, loss frame and error frame rate at the data link layer (Figure 9(c)); and the CRC at the network layer (Figure 9(d)).

For higher Rx power levels (i.e., around -9 dBm), the values of SNR and Q-factor are high with SNR=31 dB and Q-factor=9 dB. The SNR and Q-factor indicate the ability of the

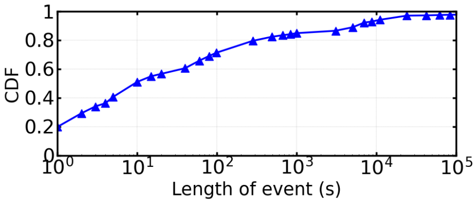


Figure 10: **The CDF of optical events' duration.**

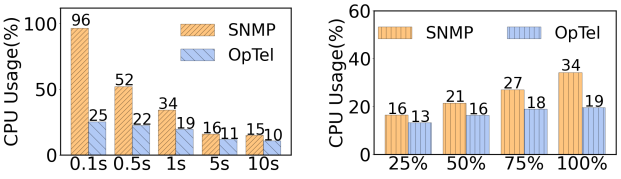
system to transmit data. Higher values for these physical layer indicators imply higher possibility of correctly decoding the transmitted '1's and '0's signal, and vice versa [18]. As Rx power decreases, SNR and Q-factor decrease linearly. But the values of data link and network layer indicators do not change as Rx power level above a specific threshold guarantees correct decoding of the transmitted signals. When the Rx power is below the threshold (i.e., -19 dBm), the Q-factor and SNR are below the sensitivity of transceiver and thus, it reports 0 to represent the abnormal state of optical system. For links with low SNR and Q-factor, the post-FEC BER increases because the number of error bits exceeds its error correction capability. Consequently, the receiver can not restore the transmitted data, resulting in nearly 100% of frame loss and error in the data link layer (Figure 9(c)) and packet loss due to CRC error in the network layer (Figure 9(d)).

Given these observations, we conclude that the Tx/Rx power level is the key indicator of optical-layer performance. This reinforces the prior study [53]. Thus, We use the changes of Tx/Rx power level to detect optical events and divide optical events into two broad categories on the basis of their impact: *degradation* and *interruption*. Generally, there is a conservative deployment of the optical transport system, with redundancy baked in at the Rx power. The degradation event occurs when the optical system transitions to an abnormal state, evident from smaller values for physical layer indicators Tx/Rx power level, Q-factor, SNR, etc. However, here such anomalies do not affect the data transmission at the data link or network layer. In contrast, the interruption events are where further degradation in the physical layer starts affecting data transmission. Note that fluctuations in Rx/Tx power levels are common in production networks. Based on the network operator's experiences, we treat any fluctuation within 1 dB range as normal.

Ephemeral vs. Persistent events. Optical events not only vary in terms of impact but also in duration. Figure 10 shows the duration of optical events (both interruption and degradation). We observe that the event duration exhibits long-tail behavior. Interestingly, we observe that 20% of events only last for one second and more than 50% of them last for less than ten seconds, indicating the prevalence of such transient optical events in the optical backbone. These observations demonstrate the utility of OpTel's ability to detect such short-lived events that go unnoticed with the existing telemetry systems. Given these observations, we divide optical events into two categories based on their duration. We call all the optical events that last less than ten seconds as *ephemeral events* and the rest as *persistent events*.

Table 1: **The proportion of four types of optical events.**

Type	P-I	P-D	E-I	E-D	Total
Percentage	44.63%	4.28%	16.85%	34.24%	100%



(a) Collection frequency

(b) Number of indicators

Figure 11: **CPU usage of the device with different collection frequencies and numbers of indicators (normalized).**

Overall, we consider four different types of optical events based on the combination of their impact and duration: *ephemeral degradation (E-D)*, *persistent degradation (P-D)*, *ephemeral interruption (E-I)* and *persistent interruption (P-I)*. Table 1 shows the prevalence of each of these event types in our dataset. For confidentiality, we do not report the exact numbers. We observe that optical events of the type P-I contribute 44.63% to the total, followed by the E-D events, which contribute about one-third to the total. The P-D events are the least prevalent, only contributing 4.28% to the total events. Note that more than 50% of optical events are ephemeral.

4.2 Data Collection Overheads

Intuitively, we expect collecting optical data at higher frequencies (i.e., order of seconds) to be prohibitively expensive. We now demonstrate how OpTel's streamlined telemetry pipeline makes such high-frequency data collection feasible. We compare OpTel's overhead, quantified in terms of CPU usage at the optical devices, with existing SNMP-based telemetry systems for different collection frequencies (i.e., 0.1s, 0.5s, 1s, 5s, and 10s) and the number of indicators (i.e., 25%, 50%, 75%, and 100% of the total). We can configure the same device to either use OpTel's or conventional SNMP-based telemetry pipelines in our current deployment. Such flexibility enables us to report the CPU usage for these two different pipelines for the same set of optical devices.

Figure 11(a) shows that CPU usage increases with collection frequency for both pipeline, but the rate of change for OpTel is marginal. More specifically, the increase in collection frequency from 1 second to 0.1 seconds raises SNMP-based pipeline's CPU usage from 34% to 96%. Such high CPU usage highlights SNMP's struggles to handle polling requests at such high frequencies. In contrast, OpTel's CPU usage only increases by 6% from 19% to 25%, demonstrating its efficacy. As shown in Figure 3, the polling-based SNMP consumes a significant number of CPU cycles to collect data, including receiving the request from the controller, traversing the MIB

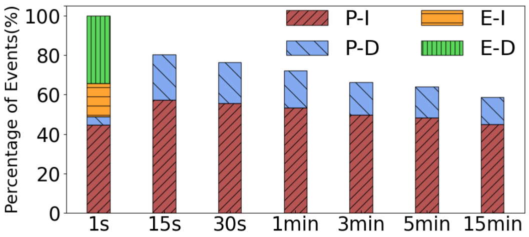


Figure 12: **The percentage of events detected with the decrease of collection frequencies (y axes is normalized by total events detected with one-second granularity data).**

database [35], and then requesting data from the line cards. The reduction in the CPU usage of OpTel is attributable to the offloading of compute-intense data-management tasks from the optical devices to the controller. As shown in Figure 7, once configured, the device only needs to periodically initiate a connection to push the data to the controller. This process does not introduce much CPU overhead on the device.

Figure 11(b) shows that CPU usage increases with the number of indicators, but the rate of change for SNMP-based pipeline is greater than OpTel's. Specifically, the CPU usage is 16% if we collected 25% of total indicators with an SNMP-based pipeline. However, the CPU usage increases to 34% if all indicators are collected. In contrast, the CPU usage is only 19% for OpTel. Recall that vendors limit SNMP's CPU usage at the cost of longer polling delays. Figure 4(a) depicts that it takes tens of seconds to complete a polling period. The high polling frequency results in a new polling request starting before the previous polling request has ended. There will be a lot of concurrent polling requests, resulting in high CPU usage. The current design choice of SNMP is not scalable to collect a large number of indicators at higher frequencies. In contrast, OpTel streamlines the telemetry pipeline by offloading resource-intense operations to the cloud. Therefore, OpTel maintains low CPU usage at the device with the increasing collection frequencies and indicators.

4.3 Detecting Optical Events with OpTel

Detecting optical events is essential for troubleshooting the related network disruptions to various stakeholders. We evaluate the efficiency and accuracy of OpTel on detecting optical events by comparing with existing telemetry systems.

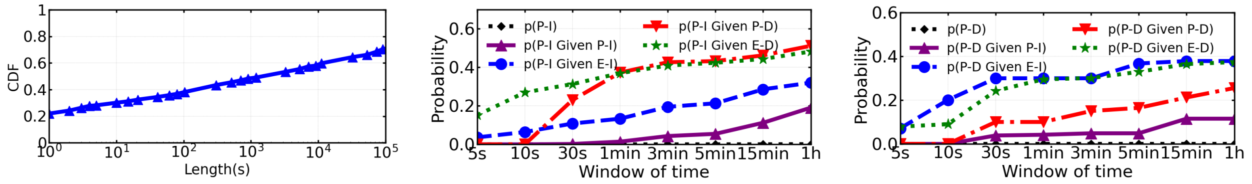
High Efficiency. We firstly study the efficiency of OpTel on detecting optical events by comparing with different collection frequencies, i.e., time intervals varying orders of minutes (1min, 3min, 5min, and 15min) and seconds (1s, 15s, and 30s). Previous works only took advantage of minute-level data to study operational optical networks [8, 18, 39, 52]. To the best of our knowledge, no prior work uses second-level data to detect optical events in operational optical networks. We introduce them in experiments to further demonstrate the relationship between the number of detected events and collection frequency. Specifically, we take the data collected at the one-second granularity as the ground truth and simulate the detection of events with different collection frequencies.

Table 2: **The comparison of detected optical events with OpTel and the existing telemetry system. *UND* means undetected optical events.**

		Existing system (15 minutes)			
		P-I	P-D	*UND*	Total
OpTel (1 second)	P-I	**33.80%**	0	10.83%	44.63%
	P-D	0	**1.92%**	2.36%	4.28%
	E-I	11.00%	0	5.85%	16.85%
	E-D	0	11.88%	22.36%	34.24%
	Total	44.80%	13.80%	41.40%	100%

Figure 12 demonstrates that OpTel achieves high efficiency on detecting optical events. For confidentiality, we do not report the detailed number of events. As the figure shows, the total number of detected events decreases when the collection frequency decreases. Specifically, OpTel outperforms the collection frequencies with 15 seconds, 1 minute, and 15 minutes by 25%, 39%, and 71%, respectively. This phenomenon proves the efficiency of OpTel to detect them. Another observation is that the collection frequencies lower than 15 seconds can not detect ephemeral optical events, and the number of persistent events (i.e., P-I and P-D) decreases when the collection frequency decreases. OpTel takes advantage of the one-second granularity data to exactly detect these ephemeral events. Surprisingly, we observe that the number of persistent events detected with the 15-second granularity data is more than that with the 1-second granularity data. This phenomenon implies that the majority of ephemeral events are wrongly identified as persistent events, i.e., E-I and P-I are wrongly identified as E-D and P-D, respectively. In other words, a portion of persistent events detected with the coarse-grained data are not actually persistent. This motivates us to learn the accuracy of detecting optical events by OpTel.

Full Accuracy. We then study the accuracy of OpTel on detecting optical events by comparing with existing telemetry systems. We take the existing system with 15-minute granularity data as an example since it is widely studied for optical layer in previous works [8, 18, 39, 52]. Similarly, we regard the one-second granularity data as the ground truth and simulate the detection of optical events. The results are shown in Table 2. For confidentiality, the number of events is normalized by the total number of optical events detected with the 1-second granularity data. Each row represents the events detected by OpTel, while each column represents the events detected by the existing telemetry system with 15-minute granularity data. We observe that the existing system can only correctly detect 35.72% of optical events (shown in **Bold**), while 41.40% of optical events are not detected (*UND* column) and 22.88% of optical events are wrongly detected (shown in underlined). Specifically, for P-D events, only less than 50% of P-D events can be accurately identified by the existing telemetry system while the rest can not be detected. As for ephemeral events, they are either identified as persistent events or not detected



(a) Inter-event length

(b) Probability of a P-I event

(c) Probability of a P-D event

Figure 13: **(a) The CDF of length of inter-events (Log-scale x-axes); (b) Probability of a P-I event in a given time window after different types of events; (c) Probability of a P-D event in a given time window after different types of events.**

by the existing telemetry system. As for E-D, 22.36% of E-D events can not be detected, occupying about two thirds of total E-D events. It can be caused by several reasons. For example, if there are several E-D events in one 15-minute time interval, only one E-D event will be identified as a P-D event, and the rest can not be identified. In general, our OpTel with 1-second granularity data accurately detects all optical events, especially for ephemeral events.

4.4 Predicting Future Optical Events

We evaluate the possibility of predicting future events based on the current event within a short time by OpTel. Figure 13(a) depicts the CDF of length of inter-events. A surprising observation is that 20% of inter-event lengths are only one second. This phenomenon suggests that these events occur in bursts and demonstrates the utility of the optical telemetry system on collecting data at the one-second granularity. Another observation is that 50% of inter-event lengths are less than 1000 seconds, suggesting a high probability of an optical event within about 15 minutes after the current event.

We focus on predicting persistent events as they represent a more prolonged impairment or loss in network capacity and are more predictable. Taking the P-I event as an example, we first compute the probability of a persistent interruption event within a window of time and call it p(P-I). For $x \in$ {P-I, E-I, P-D, E-D}, p(P-I given x) indicates the probability of observing a P-I event given a prior x event within the same window. Fig 13(b) and 13(c) depict the average probabilities across all spans as a function of window size, from 5 seconds to 1 hour. In contrast to the previous work [17], our works focus on taking advantage of one-second granularity data and the ephemeral events to achieve the short-term predictions.

As expected, p(P-D) and p(P-I) increase as window size increases; the larger the window of time, the higher possibility of a persistent event occurs within that window. For a window of 1 hour, the probability of a persistent event occurrence is less than 1%. This suggests a low probability of having a persistent event in the 1-hour window. However, there is a significant jump in the probability if there has been another event in the past, e.g., E-D, E-I, and P-D. For example, for a window of 1 hour, the probability of persistent event occurrence increases to about 50% if there has been an E-D event within that window. Meanwhile, we observe that the events have a strong relation in a short time window. For example, for a window of 5 seconds in Figure 13(b), the probability of P-I occurrence increases to about 20% if there has been an E-D event within that window, while for a window of 1 minute, the probability increases to 40%. This indicates that the E-D event is strongly related to the future P-I event. As for the prediction of P-D events in Figure 13(c), the probability of P-D occurrence increases to 30% if there has been an ephemeral event (i.e., E-D and E-I) within 30 seconds, and the possibility does not increase much with a larger window of time. This suggests that the P-D event always happens after the ephemeral event within 30 seconds. Another observation is that the past P-I event is less predictive of the future persistent events, indicating that the P-I events are memoryless.

OpTel demonstrates a high possibility for predicting future events at the second-level granularity. Thus, network operators could take the fine-grained IP layer network management. First, network operators should monitor the ephemeral and degradation events and raise alarms when they occur. Then, appropriate actions should be taken since the failure probability of IP layer link will increase. For example, they could improve traffic engineering so that important traffic should be dispatched away from the corresponding link.

4.5 Troubleshooting Events with OpTel

Characterizing failure signals. We demonstrate the effectiveness of OpTel on unveiling the signatures of optical events, and thus we could quickly troubleshoot the optical events based on the observed signatures. We use the Tx/Rx power as the primary method to unveil the signatures of optical events since it is the key indicator of optical-layer performance [53]. For some optical events such as fiber events, we learn the signatures based on the network operator's experience (Figure 14). However, for some optical events such as hardware failures, we should traverse the trouble tickets dataset to learn the signatures. To locate optical events accurately, we take advantage of the centralized controller to conduct inter-device analysis by combining the Tx/Rx power from three sources, i.e., OSC, OTU, and BA/PA. For confidentiality, we do not show all signatures of optical events and take an example for each category of the events.

(a) Fiber cable: optical fiber jitter before interruption (Figure 13(b)). We firstly unveil the most frequent optical

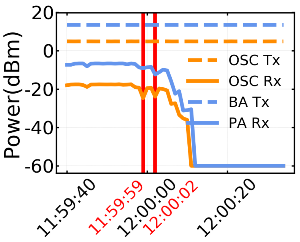


Figure 14: **Fiber**

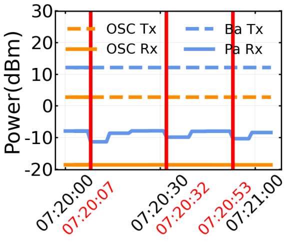
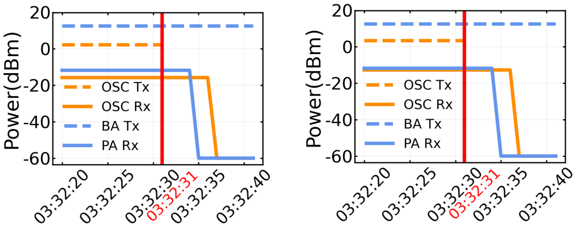


Figure 15: **Amplifier**



(a)

(b)

Figure 16: **The power outage at site of LA.**

events, i.e., fiber cable events. As shown in Figure 14, on the one hand, we observe that the Tx powers of BA and OSC remain unchanged during the timeline. However, the Rx powers at both PA and OSC is down to -60 dBm after 12:00:10 (The receiver records a predefined minimum value when the received power is below its sensitivity.). On the other hand, the timestamp of Rx power changes in OSC and PA is consistent with several ephemeral degradation events at 11:59:59 and 12:00:02 before the interruption. Thus, we can localize the optical event as an optical fiber event since the sources of the light work well and the probability of two receivers at PA and OSC having problems at the same time is relatively low.

(b) Hardware: amplifier instability. We then unveil one example of hardware failures, i.e., amplifier failure. Since the curve is quite similar, we only select a 1-minute time interval, as shown in Figure 15. We observe that the Rx power of PA changes periodically with about a 3 dB drop at 7:20:07, 7:20:32, and 7:20:53, while the indicators of the rest remain unchanged. The stable Tx/Rx values of OSC indicate a normal state of fiber cable, while the stable Tx values of BA indicate that BA works well. Thus, we can localize the optical event as the instability of PA, i.e., amplifier failure.

(c) Power: power outage at site of LA. We unveil one example of power events, i.e., the power outage at the site of the in line amplifier (LA). In a long-haul transmission system, there is a relay site containing LA that amplifies the signal to deal with long-haul transmission loss. Figure 18 in Appendix C depicts the detailed origins of Tx/Rx power in Figure 16. In Figure 16(a), We observe that the Tx power of OSC can not be collected after 03:32:31 while the Tx power of BA remains almost unchanged. Surprisingly, the Rx power values of both OSC and PA become -60 dBm after 03:32:32. Thus, we locate the power outage event at LA in the site, and the delay of Rx power is mainly due to energy storage of components in the device, such as capacitance and inductance [12]. The similar results in Figure 16(b) further prove this phenomenon. Thus, we localize the optical event as a power outage at site of LA.

These signatures of optical events present the necessity of indicators to be collected at the second-level granularity which can not be demonstrated in the existing telemetry system. OpTel unveils the signatures of optical events, which presents the superiority of optical telemetry to collect data at the one-second granularity and a centralized controller to conduct inter-device analysis in real time.

We finally evaluate the time efficiency on troubleshooting optical events by comparing OpTel with the existing telemetry system. For the existing system, it takes much manual effort to troubleshoot the optical events, and we calculate the total time of troubleshooting the optical events based on timestamps recorded in the trouble tickets dataset. As for OpTel, based on signatures learned before, OpTel conducts inter-device analysis in a centralized controller to detect and troubleshoot optical events in a timely manner. Table 3 presents the comparison of the total time of troubleshooting optical events between OpTel and the existing telemetry system across all event categories. We do not report events that have not happened in the studied dataset. There are several observations. Firstly, 89.7% of optical events (i.e., P-D, P-I, E-D, and E-I) are caused by fiber cable, including fiber cut, fiber jitter, and fiber bent/degradation. The existing telemetry system takes about 5min ~10min to troubleshoot a fiber cut event. However, it can not troubleshoot the optical events caused by fiber jitter or degradation. In contrast, OpTel only takes several seconds to troubleshoot all the events related to the fiber cable. Secondly, 7.8% of optical events are caused by hardware, and it takes quite a long time, i.e., hours ~days and much manual effort to troubleshoot them. Some hardware events, such as amplifier instability (Figure 15) can not be troubleshooted in the existing telemetry system. In contrast, OpTel only takes about 2s ~60s to troubleshoot all the events related to hardware, reducing the time by as much as two to four orders of magnitude. The total time of troubleshooting events by OpTel is related to the time length of the signature. For example, OpTel takes 2s to troubleshoot amplifier malfunction and 60s to troubleshoot amplifier instability since we need to take about 60s to get the value change patterns of Rx power in PA to troubleshoot the optical event (Figure 15). As for the power events, OpTel is efficient to troubleshoot these events within a minute. In summary, OpTel takes advantage of the one-second granularity data to learn the signatures of the optical events and thus troubleshoots optical events at scale in a timely manner.

5 Related Work

Network streaming telemetry. Previous works have extensively studied the design of network streaming telemetry systems. End-host-based network streaming telemetry sys-

Table 3: **The comparison of the total time of troubleshooting events between OpTel and the existing telemetry system.**

Event category	Percentage	Event type (Detect)	Event name (Troubleshoot)	Existing telemetry system	OpTel
Fiber cable	89.7%	PI	Fiber cut	5min~10min	10s
		EI / ED	Fiber jitter	UNK	3s
		PD	Fiber bent / degradation	UNK	10s
Hardware	7.8%	PI / ED	Amplifier malfunction / instability	hours~days / UNK	2s~60s
		PI / ED	OSC malfunction / instability	hours~days / UNK	2s~60s
		PI	OTU malfunction	hours ~days	2s~60s
Power	2.5%	PI	Power outage	hours	10s~30s
		PI	Power down	hours	10s~30s

tems [4, 16, 32, 41] performed flow-level tracking but had to deal with a limited view of the network. Switch-based network telemetry systems usually offered a coarse-grained view of the network, collecting aggregated or sampled data from the network [36, 43, 48]. Systems supporting packet-level analytics offered limited flexibility as they only supported a limited set of analytics queries [28, 33, 50, 51]. More recently, hybrid telemetry systems [19, 22, 42] struck a balance between flexibility and scale, supporting dataflow operators over packet fields at scale. Though these works enabled packet-level or flow-level network streaming analytics, they were not suited for ingesting physical, data link, and network link layer data to diagnose optical events. Previous works [34, 37] did propose a telemetry system explicitly designed for optical networks. However, they evaluated the proposed artifacts in lab environments, making it difficult to assess their performance in production settings. In contrast, OpTel demonstrates the feasibility to collect fine-grained optical telemetry data at higher frequencies (i.e., one-second granularity) by running in production at Tencent's optical backbone network for six months.

Optical layer control. Several works have studied the control interface of optical networks. Cox [11] proposed an ultimate goal of controlling the open optical line system (i.e., vendor-free optical system) in Microsoft's optical backbone by a unified SDN controller and discussed some issues surrounding the effort. Filer et al. [15] expressed a long-term goal of unifying the optical control plane and pointed out the challenges in properly controlling the plurality of optical source and line system options. They recognized Yang model [7] and SNMP [10] as potential starting points for a standard data model and control interface. In contrast to previous works which only provided the preliminary idea, we demonstrate the feasibility of a centralized control of vendor-free optical networks by designing a standardized model for devices that abstracts away the vendor-specific details.

Optical layer characterization. Previous work [9, 14, 46, 47] characterized the dispersion (e.g., polarization mode dispersion, chromatic dispersion) of the deployed fiber cable. Our work complements these efforts by investigating similar phenomenons (and more) for a much larger deployment. Ghobadi et al. [17] reported a three-month study of Q-factor data from Microsoft's optical backbone and evaluated whether fiber segments can support higher-order modulations to increase network bandwidth. The following work RADWAN [39] provided a traffic engineering system that dynamically adapted link rates according to the SNR to enhance network throughput and availability. These works took advantage of one coarsely sampled indicator. In contrast, our work benefits from the fine-grained data and a centralized controller to support inter-device analysis to detect and troubleshoot optical events. We leave correlations of IP layer performance and optical events to future work.

Diagnosis optical events. Ghobadi et al. [18] studied Q-factor data from Microsoft's optical backbone network and observed that network outages could be predicted based on the values drops in optical signal quality. RAIL [53] regarded RxPower as a key indicator of optical layer performance and found that instances of low Rx power could cause packet corruption. CorrOpt [52] used an optical layer monitor with Tx and Rx power to help determine the root cause of packet corruption in DCNs. These previous works adopted SNMP optical MIB [35] to poll optical performance indicators spanning from 5 minutes to 15 minutes. As a result, their works were slow in detecting persistent events and not capable of detecting ephemeral events. In contrast, OpTel is an optical telemetry system that supports one-second granularity optical data collection. Meanwhile, based on the signature learned from such fine-grained data, OpTel is able to detect and troubleshoot optical events in a timely manner.

6 Conclusion

This paper presents OpTel, an optical telemetry system that uses a centralized vendor-agnostic controller to collect optical data in a streaming fashion. More specifically, it offers flexible vendor-agnostic interfaces between the devices and the controller and offloads data-management tasks from the devices to the controller. As a result, OpTel enables the collection of fine-grained optical telemetry data at the one-second granularity. It has been running in Tencent's optical backbone network for the past six months. Compared to existing telemetry systems, OpTel accurately detects 2× more optical events, half of which are ephemeral events. OpTel also enables troubleshooting of these optical events in a few seconds, which is orders of magnitude faster than the state-of-the-art.

This work does not raise any ethical issues.

Acknowledgments

We sincerely thank our shepherd Manya Ghobadi, Walter Willinger, Gilberto Mayor, Kevin Schmidt, and the anonymous reviewers for their valuable feedback on earlier versions of this paper. We also thank teams at Tencent for their contributions to the work. Zekun He and Jilong Wang are corresponding authors. This work was supported in part by the National Key Research and Development Program of China under Grant No. 2020YFE0200500. Arpit Gupta was supported by NSF/Intel Partnership on Machine Learning for Wireless Networking Program under Award 2003257, "ML-WiNS: RL-based Self-driving Wireless Network Management System for QoE Optimization".

References

[1] grpc: a high performance, open-source universal rpc framework. https://grpc.io/.

[2] The need for otn in data center interconnect (dci) transport, 2016.

[3] Inter-datacenter bulk transfers with codedbulk. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)* (Apr. 2021), USENIX Association.

[4] Alipourfard, O., Moshref, M., Zhou, Y., Yang, T., and Yu, M. A comparison of performance and accuracy of measurement algorithms in software. In *Proceedings of the Symposium on SDN Research* (2018), pp. 1–14.

[5] Alizadeh, M., Greenberg, A., Maltz, D. A., Padhye, J., Patel, P., Prabhakar, B., Sengupta, S., and Sridharan, M. Data center tcp (dctcp). In *Proceedings of the ACM SIGCOMM 2010 Conference* (2010), pp. 63–74.

[6] Arnold, T., He, J., Jiang, W., Calder, M., Cunha, I., Giotsas, V., and Katz-Bassett, E. Cloud provider connectivity in the flat internet. In *Proceedings of the ACM Internet Measurement Conference* (2020), pp. 230–246.

[7] Bjorklund, M., et al. Yang-a data modeling language for the network configuration protocol (netconf).

[8] Bogle, J., Bhatia, N., Ghobadi, M., Menache, I., Bjørner, N., Valadarsky, A., and Schapira, M. Teavar: striking the right utilization-availability balance in wan traffic engineering. In *Proceedings of the ACM Special Interest Group on Data Communication*. 2019, pp. 29–43.

[9] Bohata, J., Jaros, J., Pisarik, S., Zvanovec, S., and Komanec, M. Long-term polarization mode dispersion evolution and accelerated aging in old optical cables. *IEEE Photonics Technology Letters 29*, 6 (2017), 519–522.

[10] Case, J., Fedor, M. S., Schoffstall, M. L., and Davin, J. Simple network management protocol (snmp). *RFC 1098* (1989), 1–34.

[11] Cox, J. Sdn control of a coherent open line system. In *Optical Fiber Communication Conference* (2015), Optical Society of America, pp. M3H–4.

[12] Di Ventra, M., Pershin, Y. V., and Chua, L. O. Circuit elements with memory: memristors, memcapacitors, and meminductors. *Proceedings of the IEEE 97*, 10 (2009), 1717–1724.

[13] Enns, R., Bjorklund, M., Schoenwaelder, J., and Bierman, A. Network configuration protocol (netconf).

[14] Feuerstein, R. J. Field measurements of deployed fiber. In *National Fiber Optic Engineers Conference* (2005), Optical Society of America, p. NThC4.

[15] Filer, M., Gaudette, J., Ghobadi, M., Mahajan, R., Issenhuth, T., Klinkers, B., and Cox, J. Elastic optical networking in the microsoft cloud. *Journal of Optical Communications and Networking 8*, 7 (2016), A45–A54.

[16] Geng, Y., Liu, S., Yin, Z., Naik, A., Prabhakar, B., Rosenblum, M., and Vahdat, A. {SIMON}: A simple and scalable method for sensing, inference and measurement in data center networks. In *16th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 19)* (2019), pp. 549–564.

[17] Ghobadi, M., Gaudette, J., Mahajan, R., Phanishayee, A., Klinkers, B., and Kilper, D. Evaluation of elastic modulation gains in microsoft's optical backbone in north america. In *2016 Optical Fiber Communications Conference and Exhibition (OFC)* (2016), IEEE, pp. 1–3.

[18] Ghobadi, M., and Mahajan, R. Optical layer failures in a large backbone. In *Proceedings of the 2016 Internet Measurement Conference* (2016), pp. 461–467.

[19] Gupta, A., Harrison, R., Canini, M., Feamster, N., Rexford, J., and Willinger, W. Sonata: Query-driven streaming network telemetry. In *Proceedings of the 2018 conference of the ACM special interest group on data communication* (2018), pp. 357–371.

[20] Hong, C.-Y., Kandula, S., Mahajan, R., Zhang, M., Gill, V., Nanduri, M., and Wattenhofer, R. Achieving high utilization with software-driven wan. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM* (2013), pp. 15–26.

[21] Hong, C.-Y., Mandal, S., Al-Fares, M., Zhu, M., Alimi, R., Bhagat, C., Jain, S., Kaimal, J., Liang, S., Mendelev, K., et al. B4 and after: managing hierarchy, partitioning, and asymmetry for availability and scale in google's software-defined wan. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication* (2018), pp. 74–87.

[22] Huang, Q., Sun, H., Lee, P. P., Bai, W., Zhu, F., and Bao, Y. Omnimon: Re-architecting network telemetry with resource efficiency and full accuracy. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication* (2020), pp. 404–421.

[23] Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., Venkata, S., Wanderer, J., Zhou, J., Zhu, M., et al. B4: Experience with a globally-deployed software defined wan. *ACM SIGCOMM Computer Communication Review 43*, 4 (2013), 3–14.

[24] Jin, X., Li, Y., Wei, D., Li, S., Gao, J., Xu, L., Li, G., Xu, W., and Rexford, J. Optimizing bulk transfers with software-defined optical wan. In *Proceedings of the 2016 Conference of the ACM Special Interest Group on Data Communication* (2016), pp. 87–100.

[25] Kachris, C., and Tomkos, I. A survey on optical interconnects for data centers. *IEEE Communications Surveys & Tutorials 14*, 4 (2012), 1021–1036.

[26] Labovitz, C., Iekel-Johnson, S., McPherson, D., Oberheide, J., and Jahanian, F. Internet inter-domain traffic. *ACM SIGCOMM Computer Communication Review 40*, 4 (2010), 75–86.

[27] Laoutaris, N., Sirivianos, M., Yang, X., and Rodriguez, P. Inter-datacenter bulk transfers with netstitcher. In *Proceedings of the ACM SIGCOMM 2011 Conference* (2011), pp. 74–85.

[28] Liu, Z., Manousis, A., Vorsanger, G., Sekar, V., and Braverman, V. One sketch to rule them all: Rethinking network flow monitoring with univmon. In *Proceedings of the 2016 ACM SIGCOMM Conference* (2016), pp. 101–114.

[29] McQuistin, S., Uppu, S. P., and Flores, M. Taming anycast in the wild internet. In *Proceedings of the Internet Measurement Conference* (2019), pp. 165–178.

[30] Mills, D. *RFC1305: Network Time Protocol (Version 3) Specification, Implementation*. RFC Editor, 1992.

[31] MIZUOCHI, T. Recent progress in forward error correction and its interplay with transmission impairments. *IEEE Journal of Selected Topics in Quantum Electronics 12*, 4 (2006), 544–554.

[32] MOSHREF, M., YU, M., GOVINDAN, R., AND VAHDAT, A. Trumpet: Timely and precise triggers in data centers. In *Proceedings of the 2016 ACM SIGCOMM Conference* (2016), pp. 129–143.

[33] NARAYANA, S., SIVARAMAN, A., NATHAN, V., GOYAL, P., ARUN, V., ALIZADEH, M., JEYAKUMAR, V., AND KIM, C. Language-directed hardware design for network performance monitoring. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication* (2017), pp. 85–98.

[34] PAOLUCCI, F., SGAMBELLURI, A., CUGINI, F., AND CASTOLDI, P. Network telemetry streaming services in sdn-based disaggregated optical networks. *Journal of Lightwave Technology 36*, 15 (2018), 3142–3149.

[35] PRESUHN, R., CASE, J., MCCLOGHRIE, K., ROSE, M., AND WALDBUSSER, S. Management information base (mib) for the simple network management protocol (snmp). Tech. rep., STD 62, RFC 3418, December, 2002.

[36] RASLEY, J., STEPHENS, B., DIXON, C., ROZNER, E., FELTER, W., AGARWAL, K., CARTER, J., AND FONSECA, R. Planck: Millisecond-scale monitoring and control for commodity networks. *ACM SIGCOMM Computer Communication Review 44*, 4 (2014), 407–418.

[37] SADASIVARAO, A., JAIN, S., SYED, S., PITHEWAN, K., KANTAK, P., LU, B., AND PARASCHIS, L. High performance streaming telemetry in optical transport networks. In *Optical Fiber Communication Conference* (2018), Optical Society of America, pp. Tu3D–3.

[38] SAEED, A., GUPTA, V., GOYAL, P., SHARIF, M., PAN, R., AMMAR, M., ZEGURA, E., JANG, K., ALIZADEH, M., KABBANI, A., ET AL. Annulus: A dual congestion control loop for datacenter and wan traffic aggregates. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication* (2020), pp. 735–749.

[39] SINGH, R., GHOBADI, M., FOERSTER, K.-T., FILER, M., AND GILL, P. Radwan: rate adaptive wide area network. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication* (2018), pp. 547–560.

[40] STONE, J., AND PARTRIDGE, C. When the crc and tcp checksum disagree. *ACM SIGCOMM computer communication review 30*, 4 (2000), 309–319.

[41] TAMMANA, P., AGARWAL, R., AND LEE, M. Simplifying datacenter network debugging with pathdump. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)* (2016), pp. 233–248.

[42] TAMMANA, P., AGARWAL, R., AND LEE, M. Distributed network monitoring and debugging with switchpointer. In *15th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 18)* (2018), pp. 453–456.

[43] TILMANS, O., BÜHLER, T., POESE, I., VISSICCHIO, S., AND VANBEVER, L. Stroboscope: Declarative traffic mirroring on a budget. In *Proc. of NSDI* (2018).

[44] WALLS, F. L., AND VIG, J. R. Fundamental limits on the frequency stabilities of crystal oscillators. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control 42*, 4 (1995), 576–589.

[45] WOHLFART, F., CHATZIS, N., DABANOGLU, C., CARLE, G., AND WILLINGER, W. Leveraging interconnections for performance: the serving infrastructure of a large cdn. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication* (2018), pp. 206–220.

[46] WOODWARD, S., NELSON, L., FEUER, M., ZHOU, X., MAGILL, P., FOO, S., HANSON, D., SUN, H., MOYER, M., AND O'SULLIVAN, M. Characterization of real-time pmd and chromatic dispersion monitoring in a high-pmd 46-gb/s transmission system. *IEEE Photonics Technology Letters 20*, 24 (2008), 2048–2050.

[47] WOODWARD, S. L., NELSON, L. E., SCHNEIDER, C. R., KNOX, L. A., O'SULLIVAN, M., LAPERLE, C., MOYER, M., AND FOO, S. Long-term observation of pmd and sop on installed fiber routes. *IEEE Photonics Technology Letters 26*, 3 (2013), 213–216.

[48] YU, D., ZHU, Y., ARZANI, B., FONSECA, R., ZHANG, T., DENG, K., AND YUAN, L. dshark: A general, easy to program and scalable framework for analyzing in-network packet traces. In *16th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 19)* (2019), pp. 207–220.

[49] ZHONG, Z., GHOBADI, M., KHADDAJ, A., LEACH, J., XIA, Y., AND ZHANG, Y. Arrow: Restoration-aware traffic engineering.

[50] ZHOU, Y., SUN, C., LIU, H. H., MIAO, R., BAI, S., LI, B., ZHENG, Z., ZHU, L., SHEN, Z., XI, Y., ET AL. Flow event telemetry on programmable data plane. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication* (2020), pp. 76–89.

[51] ZHU, Y., KANG, N., CAO, J., GREENBERG, A., LU, G., MAHAJAN, R., MALTZ, D., YUAN, L., ZHANG, M., ZHAO, B. Y., ET AL. Packet-level telemetry in large datacenter networks. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication* (2015), pp. 479–491.

[52] ZHUO, D., GHOBADI, M., MAHAJAN, R., FÖRSTER, K.-T., KRISHNAMURTHY, A., AND ANDERSON, T. Understanding and mitigating packet corruption in data center networks. In *Proceedings of the 2017 Conference of the ACM Special Interest Group on Data Communication* (2017), pp. 362–375.

[53] ZHUO, D., GHOBADI, M., MAHAJAN, R., PHANISHAYEE, A., ZOU, X. K., GUAN, H., KRISHNAMURTHY, A., AND ANDERSON, T. {RAIL}: A case for redundant arrays of inexpensive links in data center networks. In *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)* (2017), pp. 561–576.

A The origins of telemetry data collected from optical device

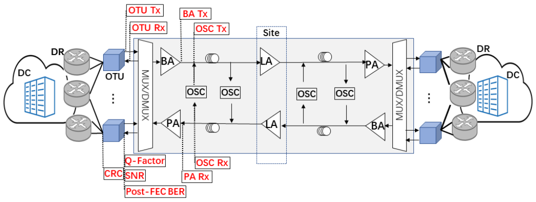


Figure 17: The origins of telemetry data collected from optical device

B Filtering out the network events from trouble tickets dataset related to optical events

Since the tickets in trouble ticket dataset describe whole network events, each ticket contains a timestamp that records the start time of the network event and the detailed alarm message and corresponding a timestamp recording the end time of the event with the causes of the failures. After manually reviewing a number of tickets, we observed that most optical events had been saliently described in the trouble tickets. Filtering out and grouping these tickets requires a lot of effort. We design a two-layer filtration. Specifically, in the first layer, we filter out the trouble tickets related to the optical backbone network by matching keywords, phrases, and regular expressions to get a set of optical trouble tickets. In the second layer, by manually reviewing the optical trouble tickets, we observe that the optical events can be categorized into a small number of classes, i.e., fiber cable, hardware and power event. We classify these tickets based on matching keywords or phrases. In some instances, there may be multiple tickets pertaining to the same failure event. Grouping these multiple tickets into a single event requires some piece of information to be repeated in each ticket.

C Data collection point of power event.

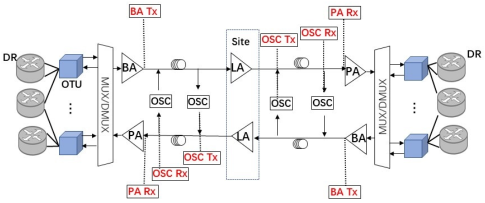


Figure 18: Schematic diagram of data collection

The performance data shown in 16(a) is collected from the top part of Figure 18, and the performance data shown in 16(b) is collected from the bottom part of Figure 18. Note, the LAs in the site share the electrical power sources.