Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Review:

A survey of cloud network fault diagnostic systems and tools*

Yining QI[1], Chongrong FANG[1], Haoyu LIU[1], Daxiang KANG[2], Biao LYU[2], Peng CHENG‡[1], Jiming CHEN[1]

[1]*State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China*

[2]*Alibaba Group, Hangzhou 310024, China*

E-mail: qyning710@gmail.com; chongrongfang.zju@gmail.com; haoyu_liu@zju.edu.cn; daxiang.kdx@alibaba-inc.com; lubiao.lb@alibaba-inc.com; pcheng@iipc.zju.edu.cn; cjm@zju.edu.cn

Received Apr. 6, 2020; Revision accepted July 2, 2020; Crosschecked May 7, 2021

Abstract: Recently, cloud computing has become a vital part that supports people's normal lives and production. However, accompanied by the increasing complexity of the cloud network, failures constantly keep coming up and cause huge economic losses. Thus, to guarantee the cloud network performance and prevent execrable effects caused by failures, cloud network diagnostics has become of great interest for cloud service providers. Due to the characteristics of cloud network (e.g., virtualization and multi-tenancy), transplanting traditional network diagnostic tools to the cloud network face several difficulties. Additionally, many existing tools cannot solve problems in the cloud network. In this paper, we summarize and classify the state-of-the-art technologies of cloud diagnostics which can be used in the production cloud network according to their features. Moreover, we analyze the differences between cloud network diagnostics and traditional network diagnostics based on the characteristics of the cloud network. Considering the operation requirements of the cloud network, we propose the points that should be cared about when designing a cloud network diagnostic tool. Also, we discuss the challenges that cloud network diagnostics will face in future development.

Key words: Cloud network; Network diagnostics; Network anomaly; Network monitoring
https://doi.org/10.1631/FITEE.2000153 **CLC number:** TP306

1 Introduction

It has been a long-hold dream that since 1961, people have expected to share computing resources just as water and electricity (Garfinkel, 1999) to enjoy computing services conveniently and elastically. The dream is the embryo of cloud computing. However, it cannot be realized under the technological conditions of that era. Over the past few decades, many cloud-related technologies, such as network function virtualization and service-oriented architecture (Gong et al., 2010; Marston et al., 2011), have gradually emerged and matured. Along with the significant progress of these technologies, several cloud service providers are burgeoning and rising, such as Amazon Web Service, Microsoft Azure, Google Cloud Platform, and Alibaba Cloud. Besides, the market of the cloud has great potential for development. According to Gartner's market research report (Veloso et al., 2020), the revenue of the cloud infrastructure as a service (IaaS) market is $41.4 billion in 2019 and will increase to $81.5 billion by 2022.

Why is the market of the cloud network growing so fast? In the traditional era, building a private

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 61761136012 and 61833015), the Alibaba Innovative Research Program, China, and the Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies, China

ORCID: Yining QI, https://orcid.org/0000-0001-6817-4443; Peng CHENG, https://orcid.org/0000-0002-4221-2162

computing platform poses strict constraints on the financial and technical aspects, which is cost-prohibitive for most companies, especially some startups (Armbrust et al., 2010). Even if some of them can afford it, they will still face the problem of over-provisioning and under-provisioning of the computing platform due to the presence of business peaks and valleys. It is not a flexible or economical option for these companies. Differently, as it is defined by National Institute of Standards and Technology (Mell and Grance, 2011), the cloud computing can provide a ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources. Compared to traditional methods, cloud computing offers more cost-efficient services as well as pay-as-you-go. Benefiting from these advantages, more and more users, from companies to individuals, are willing to deploy their business on the cloud. Besides, with the development of cloud network technologies, the scope of the cloud network is expanding and the services provided are becoming diversified. For instance, a lot of traditional network services such as content delivery network (CDN) have been increasingly built on the cloud.

Despite these superiorities, cloud network still needs to overcome a number of challenges. Among them, the stability and availability (the stability emphasizes whether the network service has a stable quality of service (QoS), while the latter pays attention to whether the network service can be connected) of the cloud network are key concerns. It is known that breakdowns are unavoidable in spite of that they are rare in practice. Due to the multi-tenant nature of the cloud network system, the impact may be very serious, resulting in huge economic losses to cloud service providers and tenants. For example, in August 2013, there was a failure in Amazon, which lasted only 45 min but caused a loss of $5 million (Wang T et al., 2016). Moreover, on June 27, 2018, Alibaba Cloud had a breakdown that influenced the business of more than 1000 companies. In fact, it was caused by an operational mistake (https://www.cloudcared.cn/2219.html). Additionally, on June 3, 2019, Google's cloud service was disconnected, which affected several Google applications such as Gmail, YouTube, and a lot of third-party applications (https://tech.sina.com.cn/i/2019-06-03/doc-ihvhiqay3245348.shtml).

To guarantee the stability and availability of cloud network, both tenants and cloud service providers desire to figure out problems in the cloud network as soon as possible. It requires the assistance of cloud network diagnostic tools. Assuming that the underlying root cause is obtained by cloud diagnostic tools, the operation engineers can immediately take proper actions to fix the fault. It has been tested that the service of Alibaba Cloud can be recovered within 26 s even if half of all the fiber optic cables are known to be in trouble (https://www.infoq.cn/article/kjf2lzq0oBR11fVFahD0). This is because there are more fault-tolerant methods (e.g., rollback and reboot) in the cloud (Gong et al., 2010), which help the cloud network guarantee its availability with a high degree of confidence. Consequently, network stability and availability can be guaranteed. Recently, several related studies have been conducted. However, there is still a lack of detailed analysis and summary of cloud diagnostic systems and tools. Therefore, we investigate cloud diagnostic systems and tools which have been implemented or tested in large production cloud networks such as Azure. In this study, we present the state-of-the-art progress of this area from different aspects, such as the strength, weakness, and design focus. Also, we discuss further development directions and probable challenges in the future. The main contributions of this paper are four-fold:

1. To the best of our knowledge, this paper is the first one to conclude cloud network diagnostic systems and tools deployed in large production cloud networks.

2. In this paper, we present a detailed description and a taxonomy of state-of-the-art cloud network diagnostic systems and tools, hoping to provide a clear view of cloud diagnostics for researchers.

3. We discuss the differences between cloud diagnostic systems and tools and traditional ones. Also, we propose several insights of designing a cloud network diagnostic system or tool.

4. We discuss some future directions in the area of cloud diagnostics, expecting our discussion to be instructive.

Fig. 1 shows the structure of this paper.

2 What is cloud diagnostics

In this review, we discuss mainly how to diagnose faults and problems in the cloud network, that

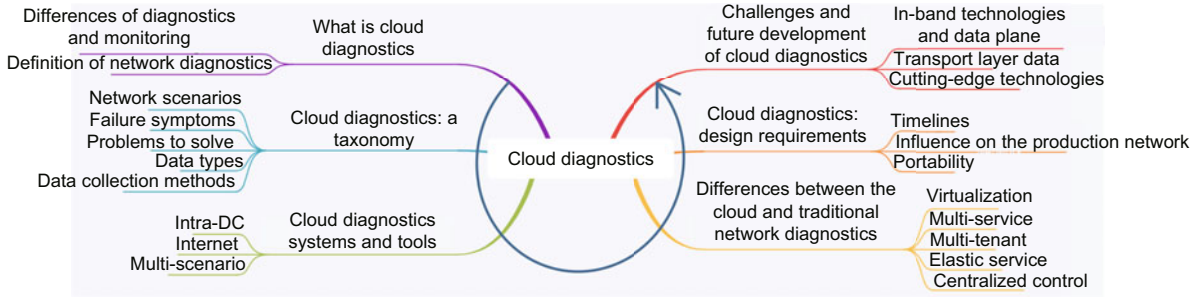

Fig. 1 Cloud diagnostics: definition, taxonomy, systems and tools, differences, design requirements, and future directions

is, cloud diagnostics. In the networking field, diagnostics is often related to monitoring. These two fields are commonly combined despite their differences. In commercial networking companies, monitoring and diagnostics are combined and named as network performance monitoring and diagnostics (NPMD) (Ganguli and Corbett, 2019). To distinguish these two fields, monitoring focuses on providing the performance measurement of networks, while diagnostics concentrates on troubleshooting the causes of faults (Aceto et al., 2013). In general, a monitoring system is always online and provides monitoring data to other systems, such as billing systems, scheduling systems, and sometimes diagnostic systems. However, most diagnostic systems work when faults are suspected to happen. Some diagnostic systems may be constructed on top of monitoring systems. For example, the packet loss rate is generally an indicator that will be continuously monitored in the network system. This is monitoring. When the packet loss rate exceeds a certain threshold, which is considered to impact the network performance, diagnostic systems are called to figure out the root cause of this problem.

In industry, there is no unified definition of network diagnostics in the cloud. Companies give different definitions at different depths according to their business scenes, such as Techopedia, Paessler, and Solarwinds. For instance, Techopedia (https://www.techopedia.com/definition/30020/network-diagnostic-software) considers that identifying the problems in network connectivity, performance, and other related aspects is vital for cloud diagnostics. For Paessler (https://www.paessler.com/network-analyzer-diagnostics), it argues that network diagnostic tools should help users get to the roots of problems faster. However, Solarwinds (https://www.solarwinds.com/network-performance-monitor/use-cases/network-diagnostics-tool) emphasizes troubleshooting problems. In this paper, we review related papers that have analyzed data from the production cloud network and gained a certain degree of insight into the network failure, no matter the insight is coarse-grained or fine-grained.

3 Cloud diagnostics: a taxonomy

In this section, we introduce and analyze the key aspects that will be faced in designing cloud diagnostic systems and tools. It will also serve as our standard for the classification of cloud diagnostic systems and tools in Table 1. To be specific, we consider the circuit that engineers care about in different stages of designing a cloud diagnostic system. First, the birth of one cloud diagnostic system must be caused by a certain kind of cloud network problem, which happens in certain scenarios, such as inside cloud data centers. Cloud service providers would then consider what kind of data is needed and available. Finally, how to deal with the data and mine the information needed becomes the major concern. Therefore, we analyze different cloud diagnostic systems and tools into five categories based on the above procedures.

3.1 Network scenarios

The scope of cloud network has become bigger and bigger in recent years. The cloud network contains several different network scenarios. Different network scenarios will encounter different problems, and the ways to solve them are quite different as well. In this paper, we divide the network scenarios into two parts: intra-data center (Intra-DC) and Internet.

The scenario of Intra-DC focuses on the

problems inside cloud data centers. This part of network is usually controlled by cloud service providers. To improve their service quality, cloud service providers may design unique data center architectures based on their business characteristics, such as Facebook fabric data center, which uses a five-layer Clos network (Andreyev, 2014). Therefore, cloud service providers often design diagnostic systems according to their data center architectures. For example, the system in Roy et al. (2017) considers the architecture of Facebook fabric data center and installs rules in Agg switches to mark packets.

For the Internet, the scenario involves traffic, which would pass wide area network, including accesses from clients, tenant data centers, and interactions between cloud data centers. In general, the network of this scenario is not completely controlled by cloud service providers. These providers may cooperate with Internet service providers (ISPs) such as AT&T and China Mobile. Thus, when there are failures in cloud networks, the first thing that cloud service providers want to confirm is who should be responsible for the problems, themselves, ISPs, or tenants. However, it is challenging because cloud service providers usually cannot obtain complete information about ISPs' network. BlameIt (Jin et al., 2019) considers this scenario. It first distinguishes problem responsibilities and then tries to find root

Table 1 Classification of state-of-the-art cloud diagnostic systems and tools deployed and used in production networks

System/Tool	Dim 1		Dim 2				Dim 3	Dim 4			Dim 5		
	Intra-DC	Internet	Latency	Packet loss	Retransmission	Availability		Packet	Flow	Infrastructure	Active	Passive	Combination
NetPilot	✓			✓			Shorten failure time			✓		✓	
Pythia		✓	✓	✓	✓		Detect failures and diagnose pathological causes		✓		✓		
Herodotou et al. (2014)'s	✓			✓		✓	Locate failure links	✓			✓		
Everflow	✓			✓			TCAM bit errors and silent packet drops	✓					✓
NetSonar		✓	✓	✓	✓	✓	Locate failure links	✓			✓		
Pingmesh	✓	✓	✓				All types of issues in production networks	✓			✓		
NetPoirot	✓		✓	✓		✓	Detect and locate failures		✓			✓	
Trumpet	✓		✓				All types of issues in production networks		✓			✓	
CorrOpt	✓			✓			Reduce packet corruption	✓				✓	
deTector	✓			✓			Locate failure links	✓			✓		
Roy et al. (2017)'s	✓				✓		Find partial/intermittent faulty links		✓			✓	
007	✓			✓			Find packet drop link	✓			✓		
Deepview	✓					✓	VHD failure localization			✓		✓	
Odin		✓	✓			✓	CDN latency		✓		✓		
BlameIt		✓	✓				Locate failure position (in cloud, middle, or client)		✓				✓
dShark	✓	✓	✓	✓	✓	✓	Packet header	✓				✓	
NetBouncer	✓			✓			Find faulty links or devices	✓			✓		
SIMON	✓		✓				Econstruct network state variables		✓		✓		
Gandalf	✓	✓	✓	✓	✓	✓	Associate failures with software changes			✓		✓	
VTrace	✓	✓		✓			Find the root cause of persistent packet loss	✓				✓	

Dim 1: network scenario; Dim 2: failure symptom; Dim 3: problems to solve; Dim 4: data type; Dim 5: data collection method

causes. Besides, it should be noted that some diagnostic systems can be used in both kinds of scenarios.

3.2 Failure symptoms

Before finding the root cause of a network failure, what operation engineers see first is the symptom of a failure, such as high packet loss rate and high latency. The annoying thing is that different types of faults could lead to similar symptoms. For instance, packet loss can be caused by device failures or software failures. Without adequate experience of senior engineers and enough network information, it is difficult to distinguish these failures. However, it is not easy to obtain complete information needed (e.g., part of the tenants' data is not allowed to be used considering tenants' privacy) due to limitations of cloud network. Therefore, it calls cloud diagnostic systems for help to analyze the in-depth cause of problems from visible symptoms. In general, the main failure symptoms of cloud network ranging from academic papers to industrial applications are classified as latency, packet loss, retransmission, and availability according to our investigation.

3.3 Problems to solve

When operation engineers discover these failure symptoms in cloud, they want to know the causes. Usually, a certain diagnostic system is designed to diagnose a certain range of troublesome problems hidden beneath symptoms in the cloud network. Here, we will discuss what kinds of problems are the focus of diagnostic systems. We cannot exhaust the cloud problems that arise in the network due to the complex structure of the cloud network. Besides, different diagnostic systems analyze problems from different angles. For example, NetPoirot (Arzani et al., 2016) aims at discovering whether the problem has happened on the network, service, or client-side, whereas NetBouncer (Tan et al., 2019) targets at clearing faulty links or devices. Several systems concentrate on solving challenges caused by cloud network characteristics, such as dShark (Yu D et al., 2019) which focuses on problems caused by packet header conversion. Furthermore, some systems pay attention to a tricky question, e.g., silent packet drops (Zhu et al., 2015). Table 1 describes the specific problem the cloud diagnostics solves instead of giving a specific classification.

Several systems are designed to solve common problems such as locating failure areas and links, or identifying root causes. They will face some common challenges brought by the complexity of the production cloud network, such as multi-domain characteristics of cloud network (Jin et al., 2019) and the massiveness of cloud data (Zhu et al., 2015).

Meanwhile, some systems concentrate on a special problem that is hard to solve. For example, the system in Roy et al. (2017) diagnoses partial and intermittent faults. Such faults will affect network performance, but are hard to detect and diagnose. Deepview (Zhang et al., 2018) focuses on another problem called virtual hard disk (VHD) failure localization, which is a new problem in cloud network.

3.4 Data types

Thanks to the development of modern network technology, we can obtain quantities of all kinds of data for cloud diagnostics. It is divided into three categories, namely, packet-level, flow-level, and infrastructure-level. Different data should be handled differently based on their characteristics.

Packet-level data is the information of a single packet, such as the trajectory of a packet. For example, operation engineers will detect whether the packet is lost and where it is discarded to discover the failure location and the root cause. To protect the privacy of tenants in public cloud network, deep packet inspection is usually not allowed. Commonly, we usually inspect the information of the packet header if the raw packet-level data is from normal production of tenants. Another problem for packet-level data is that the path of each packet is not easy to determine. There are some diagnostic systems choosing to design clever probing methods, which inevitably leads to certain limitations. For instance, Pingmesh (Guo et al., 2015) cannot diagnose links between servers, whereas NetBouncer (Tan et al., 2019) requires that devices in the network have the function of IP-in-IP. Moreover, some diagnostic systems do not consider the specific path, but include the probability of packet paths as parameters in diagnostic algorithms, such as 007 (Arzani et al., 2018) and deTector (Peng et al., 2017). Therefore, the protocol used influences the results of these diagnostic systems, and the most frequently used protocol is equal-cost multi-path (ECMP) routing.

Unlike packet-level data, flow-level data means

statistics data of packets, flows, or sessions, which must continue for some time. It includes transmission control protocol (TCP) statistics such as latency, packet loss rate, and traffic data. TCP statistics is a promising class of data because it comes from the transport layer and can reveal network quality without involving user privacy. A lot of statistical algorithms and machine learning methods (Widanapathirana et al., 2011; Arzani et al., 2016; Roy et al., 2017) can be used to analyze flow-level data.

Infrastructure-level data indicates the status of network devices, e.g., CPU usage. In the cloud network, benefiting from the virtualization technology, engineers can obtain in-depth data conveniently and flexibly. Infrastructure-level data has been used as a main metric in several cloud diagnostic systems, such as NetPilot (Wu et al., 2012), Deepview (Zhang et al., 2018), SNAP (Yu ML et al., 2011), and Sherlock (Bahl et al., 2007).

3.5 Data collection methods

There are two ways to obtain raw data of cloud diagnostics, namely, active probing and passive collection. The former method means actively sending packets to interesting locations or along with target paths and collecting corresponding information according to the specific rules in the cloud network, while the latter one means passively collecting and analyzing tenants' daily production data.

Pingmesh (Guo et al., 2015) and 007 (Arzani et al., 2018) are active methods. Through active probing, operation engineers can customize a probing plan for a certain kind of failure and specific network structure. Moreover, once there is active probing in the production network, it may affect network performance, and sometimes will lead to degradation of service quality, which could be sensed by tenants. Thus, the burden caused by active probing is a main concern of diagnostic systems. Also, active probing cannot be guaranteed to be in-band, which implies that it may miss some information because the paths of packets it sends are not the same as those of production packets.

The passive collection has also been investigated, such as Trumpet (Moshref et al., 2016) and CorrOpt (Zhuo et al., 2017). It must be in-band and reflect forwarding conditions of production packets. However, the amount of data is too large to store and analyze in traditional ways. Engineers must try their best to filter or compress data to reduce the cost of computing and storage. Privacy is another problem to consider, especially for public cloud networks. Deep packet inspection is not allowed, and the administrative rights of tenants' cloud systems are limited in plenty of situations. For example, the data collection progress may be dropped by the security policies of tenants (Calder et al., 2018). Hence, operation engineers cannot always obtain the data they want.

There are also diagnostic systems that incorporate both two methods. In these systems, passive detection is used to obtain a rough result, and active probing then is employed to clarify the specific reason based on the rough results. For instance, Everflow (Zhu et al., 2015) mirrors all flows to find the packet loss point and uses a guided probe to reproduce the behavior of problematic packets and figure out the reason.

4 Cloud diagnostic systems and tools

In traditional networks, there are a lot of mature and influential network diagnostic tools. Most of them are still used in the cloud network, such as Netflow (Claise et al., 2004), sFlow (Wang M et al., 2004), and SNAP (Yu ML et al., 2011). However, due to the unique properties of cloud networks (e.g., multi-tenancy), many diagnostic systems oriented to production cloud networks are emerging. Some of these systems are designed based on or inspired by traditional methods. To illustrate the current research progress in the cloud diagnostics, we summarize typical cloud network diagnostic systems that have been tested or deployed in a real-world production network. In the following, the specific design and implementation of each system are provided in detail by scenarios. We classify these systems in Table 1 based on the taxonomy mentioned in Section 3.

4.1 Intra-DC

In this subsection, we introduce state-of-the-art related works which can run or have been tested in production cloud networks' data centers. They may be based on the specific architecture of a certain cloud network, but are also references for future research and industrial practice.

NetPilot (Wu et al., 2012) is a failure diagnostic platform to detect failures and shorten the failure recovery time. In NetPilot, the data for analysis comes from simple network management protocol (SNMP) traps, switch and port counters, and syslogs. With this data, NetPilot can identify a series of components that are likely to cause a problem before in-depth investigation of operations engineers. Therefore, problems can be eased with the help of data center redundancy. It can reduce diagnostic time from hours to minutes, and automatically mitigate data center network failures.

The system in Herodotou et al. (2014) realizes a new method to locate data center network failures. With the prior knowledge of the network topology, it can calculate failure probability of every link using ping data. To establish the probabilistic model, it figures out the most likely ping path based on the network protocol used (e.g., ECMP), and generates a ranked list of links and devices associated with calculated failure scores after filtering noises.

Everflow (Zhu et al., 2015) is a packet-level network telemetry system for large data center networks, which mirrors packet headers from switches based on certain pre-defined rules. The mirrored information will be centralized and used for in-depth analysis, to identify whether there is a network problem. However, the specific reasons for network problems still need further detection, and a guided probe is involved to reproduce packet behaviors to obtain more information.

NetPoirot (Arzani et al., 2016) captures TCP statistics collected at virtual machines (VMs) to identify whether there are failures and to locate whether the failures are in the network, on server-side, or on client-side. A classification algorithm based on decision tree is designed to solve the problem and explain which feature affects the classification most.

Trumpet (Moshref et al., 2016) is a system for host-based eventing, which allows users to define events by two elements (namely, a packet filter and a predicate) and use triggers to detect them. The system can monitor every packet and report events in milliseconds. Accompanied by hardware development such as SmartNIC, Trumpet may become a preferred solution for network telemetry and root cause analysis.

CorrOpt (Zhuo et al., 2017) is a system that concentrates on packet corruption. Packet loss may be caused by not only network congestion, but also packet corruption. In CorrOpt, the root cause of packet corruption can be connector contamination, damaged or bent fiber, decaying transmitters, bad or loose transceivers, or shared-component failures. Once a packet corruption is obtained by CorrOpt, it will first disable corrupting links under the premise of ensuring network availability. Then, CorrOpt will figure out the root cause to help repair corruptions and release disabled links.

deTector (Peng et al., 2017) is a system that locates packet loss failures in data center networks. It will first calculate a probing plan according to the data center topology and its server state for the following probing. The probing plan is obtained by a greedy algorithm. Then, to find the smallest set of faulty links which can best explain the probing results, a packet loss localization algorithm based on the Tomo algorithm (Dhamdhere et al., 2007) is applied.

The system in Roy et al. (2017) concentrates on partial and intermittent faults that are not easy to troubleshoot in traditional networks. First, specific rules are installed in switches to mark the actual links of a flow's entire path, and then each link's data is gained. According to this work, links in data centers can be divided into several groups, within which the links have several similar functions and are supposed to perform equivalently. Therefore, the system can identify faulty links by comparing TCP statistics through each group of links.

007 (Arzani et al., 2018) is a lightweight packet drop diagnostic tool. It actively sends packets of interest and records whether it is dropped. With the correct network topology, based on the random routing policy, we can vote for each link that dropped packets may pass through and obtain the possibility of dropping packets for each link. The link with the largest possibility is most likely to blame.

Deepview (Zhang et al., 2018) was designed to localize VHD failures which happen in IaaS. Instead of diagnosing individual components, Deepview uses a global view to consider the edges between compute clusters, network devices, and storage clusters, and simplify the Clos network as a tree. A new inference algorithm, combining with lasso regression (Tibshirani, 1996) and hypothesis testing (Casella and Berger, 2002), was designed to decide who is to

blame. It has been deployed in Azure's production network, and high accuracy with operating efficiency was achieved.

NetBouncer (Tan et al., 2019) is a failure localization system based on IP-in-IP and packet bouncing technique in data center networks. It can generate a probing plan according to the topology of the data center network and send IP-in-IP packets to detect the packet loss rate of paths. Then, a well-designed algorithm against real-world data inconsistency is applied to find out faulty links or devices. The system has been running well in a production cloud network. However, it requires the switches in data centers to support the IP-in-IP function, which hinders its promotion.

SIMON (Geng et al., 2019) is a network tomography technology that tries to reconstruct network state variables, such as queueing time. It selects proper reconstruction interval according to the highest line rate to filter jitter and uses a mesh of probes to obtain network information. Its reconstruction algorithm is designed based on lasso (Tibshirani, 1996) to calculate the queueing time of every link and reconstruct accurately the averaged queue or wait time processes. Neural networks or the hierarchical structure of data centers is used to speed up SIMON.

4.2 Internet

In this subsection, we introduce related works talking about diagnostic systems on the Internet. Compared to the scenario of Intra-DC, there are relatively few studies on this scenario. It is because cloud service providers care more about the network that they can completely control, such as Intra-DC networks. However, most parts of the network in public networks belong to ISPs. ISPs have dedicated a great deal of effort in investigating the problems in the public Internet. However, considering the topic of this review, we discuss only the systems and tools that are designed for cloud networks.

4.2.1 Traffic between public network and the cloud

Odin (Calder et al., 2018) is embedded in a number of Microsoft applications to measure Microsoft's CDN performance, such as latency and availability through active detection on the client-side. It uses user-side, application-layer measurement of client connections, collecting much specific information different from that collected by traditional diagnostic systems. Odin is claimed to have a high coverage rate of different paths. Also, it is sensitive and quick to Internet events.

BlameIt (Jin et al., 2019) is a tool validated in production at Azure to localize the cause of latency degradation, which has a two-level blame assignment design. The first level is designed to detect faults with round trip time (RTT) data that coarsely determines whether the fault happens in the cloud, client, or public network. If the public network is to blame, a group of on-demand traceroutes from cloud to client will be sent to measure RTT through different hops to identify the faulty availability zone. With BlameIt, cloud service providers can quickly confirm the faulty location. If a third-party ISP's network is in trouble, cloud service providers can send the problem to the corresponding ISP. Moreover, the structure of BlameIt is similar to that of the Netprofiler (Padmanabhan et al., 2005).

4.2.2 Traffic among data centers

Pythia (Kanuparthy and Dovrolis, 2014) is a distributed system for end-to-end diagnosis of ISP network performance. It can discover network performance problems through end-to-end probing information and diagnose its pathological causes. It employs a well-designed diagnostic forest algorithm to perform analysis with low resource consumption (e.g., CPU and memory). The definition of pathological cause comes from the operator's knowledge.

NetSonar (Zeng et al., 2015) is a gray box detection method for locating network faulty links, which performs better than SNMP. Gray box indicates that only available paths between destination and source are used. Also, it combines the information of low-frequency traceroutes and high-frequency pings to locate failures.

4.3 Multi-scenario

In addition to the above systems, there are several systems or tools which can be used in multiple scenarios rather than only in one scenario, such as both Intra-DC and part of the Internet according to its design details. In this subsection, we describe the full details of this kind of system or tool.

Pingmesh (Guo et al., 2015) is an always-on tool to detect latency and packet drops between two

servers in large-scale data center networks. To reduce network overhead, the Pingmesh controller generates a ping-list and sends it to the Pingmesh agent in every server. It can identify whether the problem is caused by the network, and solve almost all types of network issues such as silent packet drop problems. However, it is because of the lightweight purpose that its detection is intermittent such that it cannot monitor network all the time and there are blank periods that Pingmesh cannot reach.

dShark (Yu D et al., 2019) is a tool used to trace network packets, which is a vital point in many cloud network systems. In the cloud network, headers of data packets may be transformed because of the complexity. Also, there is noise in collecting packets. dShark provides a method that traces packets in nearly real time. It can be used whether in or outside the cloud.

Gandalf (Li et al., 2020) is a system that focuses on failures caused by software changes. It extracts system data such as service logs and performance counters, and then ingests deployment events. By its specially designed correlation model, Gandalf can associate a system-level failure with a certain software change that causes it. Also, Gandalf uses the Lambda architecture (http://lambda-architecture.net/) to handle data, which makes it the ability to process real-time and long-term failures.

VTrace (Fang et al., 2020) desires to automatically diagnose the root cause of persistent packet loss in the cloud-scale overlay network. To achieve it, VTrace uses the "fast path-slow path" structure of virtual forwarding devices (VFDs) and installs several "coloring, matching, and logging" rules in VFDs, so that it can mark the packets of interest to track. Therefore, it obtains the packet information of each hop for further inspection.

5 Differences between the cloud and traditional network diagnostics

The development of the cloud network is based on traditional networks. Similarly, cloud network diagnostic systems and tools described in Section 4 are usually designed based on or inspired by the ones in traditional networks. However, due to the characteristics of the cloud network, the design of cloud diagnostic systems needs to be changed accordingly, to solve new problems generated in the cloud. In this section, we discuss the characteristics of cloud networks and challenges they bring accordingly.

5.1 Virtualization

Virtualization is a key technology of the cloud network. However, it causes a lot of challenges to cloud diagnostics. First, the cloud network is rapidly elastic and scalable. Resources in the cloud can be easily provisioned and released according to tenants' constantly changing business demand in any quantity at any time as long as within the capability of cloud service providers. It implies that the cloud network is quite unpredictable and complex. Besides, virtualization will change the technical realization of sending, transmitting, and receiving packets, causing the lapse of traditional technologies. For example, the header of one packet in the cloud network transforms more frequently than traditional networks when going through different components, resulting in tracing a packet header. dShark (Yu D et al., 2019) is a tool that captures packets and identifies packets' pipe by its programming model. Besides, multi-header will also bring problems when analyzing TCP statistics of five-tuple. Because no matter how accurate the diagnosis is, it can tell only what has happened to the outer packet header. If the information to be analyzed is hidden in the inner header, the diagnostic system will be powerless.

5.2 Multi-service

Based on the gradual maturity of the cloud industry, the cloud network can provide abundant cloud applications such as cross-border communication, video stream, and CDN service. In general, every application is composed of multiple services (e.g., gateway and database). It is not easy to design a universal diagnostic system for these complex services, so that it is challenge to diagnose failures in a industry cloud network. Take the cross-border communication as an example. A packet for cross-border communication may traverse DC overlay networks, DC physical networks, and several ISP networks in the global cloud network. Naturally, it becomes a challenge to distinguish where faults actually happen. Faults may locate in ISPs' networks, DC overlay networks, or DC physical networks, which are supported by different teams and cannot access easily to the performance logs of each other (Arzani

et al., 2016). For example, the global accelerator service of Alibaba (https://www.alibabacloud.com/product/ga) that provides network acceleration service for tenants' Internet-facing application globally with guaranteed bandwidth and high reliability would go through a long path from the source to the destination VM. It passes through global public network access points, cross-border links leased from different ISPs, local virtual gateways, physical networks, and virtual switches, and finally to the destination VM. Such a complicated path rarely appears in traditional networks. Therefore, it is common that the diagnosis is kicked back and forth between different cloud service teams when there are failures, which leads to a decrease of efficiency.

5.3 Multi-tenant

It is a fundamental feature that there are millions of tenants in major production cloud networks such as Google, and that computing resources are shared with hundreds of thousands of tenants. Despite the underlying logic of the cloud network, the mutual influence between tenants cannot be avoided. For example, if one tenant's traffic is abnormally large, then a single core's processing capacity is insufficient. Therefore, other traffic assigned to this core is very likely to be affected. In traditional networks, it is easy to understand the behavioral patterns of users and recognize whether large traffic is an attack or a failure, because the users may be of the same type. However, cloud tenants have different behavioral patterns and business characteristics. Capturing cloud tenants' features for analysis is fairly troublesome. Thus, analyzing tenants' data to identify tenants' anomalies becomes much harder, and data-driven methods to predict and diagnose failures can hardly be used in the cloud environment. Large traffic may just mean that tenants have deployed new business or pressure test. The impact of cloud tenants on different business is very complex. Thus, it becomes more difficult to find the root cause of cloud network failures. Other than that, one single failure is likely to be perceived differently by different tenants in actual production networks. It is because each tenant may have a unique demand on the network (Huang et al., 2017). Stream media business may be sensitive to latency, while cloud CDNs have fewer stringent requirements for latency. The same degradation in the network performance may be a failure for the former but not for the latter. Thus, cloud service providers must provide an accurate diagnosis of failures without sufficient knowledge of tenant business.

5.4 Elastic service

We note that one of the great benefits of the cloud network is that tenants can use network resources according to their continuous changing demands. However, problems arise when it comes to cloud diagnostics. The traffic models vary from time to time and are hard to predict. Therefore, many traditional network diagnostic methods based on traffic model prediction will be less effective when transplanted to the cloud network. In the cloud network, it is normal for a cluster to generate dozens of single-user traffic spikes in a day. Consequently, few articles mentioned in Section 4 have analyzed traffic data (e.g., bits per second).

5.5 Centralized control

In the software defined network (SDN), its control plane and data plane are separated so that SDN can realize logical centralization of network control (Bannour et al., 2018). To properly manage the network, cloud computing extensively applies the SDN technology, which brings advantages that traditional network diagnostics cannot achieve. On one hand, deploying a cloud diagnostic system will be much more convenient, no matter in its data collection or data processing stage. Most deployment works can be conducted on the software, with more convenient modification than in the hardware. On the other hand, we can collect more data in different dimensions or granularities and modify the type of data collected at any time as needed.

6 Cloud diagnostics: design requirements

When deploying a cloud network diagnostic system in the production cloud network, many other requirements need to be considered in addition to its help in actual diagnostic work by cloud operation engineers during the designing process. This is to ensure that it can work efficiently and effectively in the production network. In this section, we introduce the properties that operation engineers

consider when designing a diagnostic system or tool for production demand.

6.1 Timeliness

Once there is a failure in cloud networks, operation engineers need to figure out the problem and solve it as soon as possible. Thus, cloud network diagnostic systems need to be responsive, or even in (nearly) real time. It is challenging due to the sheer scale, complex structure, and large data volume of the cloud. There are many ways to decrease diagnostic time. First, we can control the amount of data that needs to be processed. Both 007 (Arzani et al., 2018) and NetBouncer (Tan et al., 2019) design a suitable detection scheme that collects data as little as possible to identify problems. The system in Roy et al. (2017) develops a lightweight packet marking technology to filter required packets instead of mirroring all traffics. Second, some advanced algorithms are deployed to speed up data processing. SIMON (Geng et al., 2019) adopts multi-layer neural networks and GPUs. Consequently, it has achieved a 5000x–10 000x acceleration performance compared with the one without multi-layer neural networks and GPUs. Third, some stream processing techniques, such as Flink (https://flink.apache.org/) and Storm (http://storm.apache.org/), can also be applied to the large-scale network data processing to ensure timeliness.

Considering the systems mentioned in Section 4, NetPilot (Wu et al., 2012) and Everflow (Zhu et al., 2015) use data at the scale of data center networks, and the systems in Herodotou et al. (2014) and Gandalf (Li et al., 2020) are deployed in Azure. These systems can obtain diagnostic results within several minutes. Furthermore, NetBouncer (Tan et al., 2019), which has been deployed in Azure, can achieve second-level diagnosis. Also, Trumpet (Moshref et al., 2016) can diagnose failures within several milliseconds, while SIMON (Geng et al., 2019) obtains results in the microsecond-level. The former is tested in Mbps-level link speed, and the latter one is tested in Gbps-level link speed.

6.2 Influence on the production network

If operation engineers deploy a cloud network diagnostic system in the production cloud network to help maintain the stability of the cloud network, the impact of the system on the actual production network must be considered. Different from traditional networks, the characteristics of the cloud network may pose a more stringent requirement for the impact of the designed diagnostic system on the production network. For example, due to the multi-tenant nature of cloud networks, the designed system may affect many tenants, even if it is caused by only one tenant indeed. Thus, the minimum requirement of a cloud network diagnostic system is that it should not interfere with the normal traffic of cloud network services. Several aspects need to be considered to guarantee it. Here, we summarize three points that should be concerned according to the literature review.

First, it is a major concern whether the data collection process goes through the control plane or data plane. If there is only the data plane involved and the control plane does not intervene, the collection speed will be faster and the impact on the cloud network will be smaller. It is because the data plane does not occupy the CPU resources and the flows on it compete for fewer with other flows. For instance, to avoid passing through the control plane, Everflow (Zhu et al., 2015) installs certain rules on switches in advance and uses the data premature mirror to filter packets. It successfully guarantees that the system is not too heavy-weight. Additionally, the emerging technology of in-band network telemetry (INT) (Kim et al., 2015) is out of similar considerations, which can obtain various data of network state by only the data plane.

Another influential factor is where to obtain data, in the bypass system or the main system. Several diagnostic tools, such as Pingmesh (Guo et al., 2015) and NetFlow (Claise et al., 2004), are installed directly on network devices such as routers and switches. Therefore, they may affect the network. Besides, some systems are deployed in a set of equipment that is independent of the production network. The required information is usually mirrored to these systems for analysis. For instance, fiber splitters are applied to copy signals to the bypass system (Duffield et al., 2009) in an optic fiber network.

Also, the frequency of data collection is vital. In the cloud network, data collection requires reading data from memory, while the network components themselves are writing data. These two actions are

mutually exclusive. Therefore, the higher the frequency of the data collection, the larger the influence of the diagnostic system. Thus, diagnostic systems have to limit the sampling frequency under the premise that the accuracy of the results can be guaranteed. We observe that the probing window of NetPoirot (Arzani et al., 2016) is limited to 30 s, and that SIMON (Geng et al., 2019) also tries to find a proper frequency to collect data in different networks through several experiments.

6.3 Portability

In addition to these two issues mentioned above, operation engineers need to consider the portability in some cases. Portability means that whether a cloud diagnostic system or tool can be transplanted to other devices and networks. It is known that the cloud network is heterogeneous in terms of device, network structure, and so on. We note that physical devices may come from different vendors or have distinct software versions, and that different parts of the network could be structured in diverse ways. For example, there are old four-post cluster design and next-generation technology in Facebook's data centers (Roy et al., 2015). In such scenarios, operation engineers expect that the designed system should be applicable across these inconsistent parts (e.g., devices or network structure), namely, being portable. Although traditional networks face the same problem as well, the cloud network is more complex and changeable. Thus, challenges brought by the portability are more worth watching. Considering that whether a system is portable depends mainly on where and for what it will be implemented, it is hard to conclude the portability for all works listed in Table 1. Hence, we summarize the portability of the system that has been discussed in the literature. Pingmesh is a very successful diagnostic tool in portability. It was first proposed by Microsoft, which now has been reproduced by many other commercial companies and can adapt to different network situations. However, several systems are more or less limited in being ported to other networks or application environment due to the specific characteristics of certain networks. For instance, Odin (Calder et al., 2018) is a typical hard-to-port system since it requires a high authority to obtain network measurement data. It is deployed on the CDN client of Microsoft, which is hard to achieve if the CDN does not belong to the cloud service providers. Furthermore, sometimes the network hardware needs to have certain functions, or the existing hardware needs to be modified, which leads to extra deployment costs. The faulty location system of Facebook (Roy et al., 2017) requires that the switches can mark suspicious packets, while NetBouncer (Tan et al., 2019) requires that switches have the IP-in-IP function. Similarly, Everflow (Zhu et al., 2015) and deTector (Peng et al., 2017) need special hardware to implement them. Besides, 007 cannot be transplanted to large-scale networks (Jin et al., 2019), but can diagnose other types of problems by modifying the detection index (Arzani et al., 2018).

7 Challenges and future development of cloud diagnostics

Nowadays, the structure of the cloud network is more and more mature, and researchers have gradually concentrated on cloud diagnostics. However, there are still a lot of challenges in cloud diagnostics research. Many recent studies are based on traditional network diagnostic tools and the characteristics of the cloud network have not been considered in depth. In this section, we conclude some challenges and future development directions of cloud diagnostics.

1. The cloud diagnostics is preferred to be in-band and in data plane. The data path of in-band detection can be the same as the path of real user data. So, diagnosis with in-band technologies captures cloud network faults more effectively. Thus, it can figure out problems that cannot be diagnosed by active probing diagnostic methods. It is because there is a small part of packet paths different from the path of real user data in active probing. For the data plane, it will not consume CPU resources, and the diagnostic performance will be better. INT (Kim et al., 2015) is a very promising technology in this area. It can collect network state without the intervention of the control plane. When a data packet marked by INT in advance passes INT traffic sources (applications, end-host networking stacks, hypervisors, NICs, send-side ToRs, etc.), the data required by the INT would be put into the packet in the form of packet headers. Therefore, the extra burden brought by INT is also a concern for deploying the system. Besides, to achieve INT, switches in

the network must stand by the Openflow protocol. Thus, the technology is currently not widely used due to hardware limitations. However, in the near future, with the promotion of Openflow switches, the application of INT can tell us the specific location of packet loss, the actual path packet passing by, and other information that is difficult to obtain before.

2. Besides, the state-of-the-art technologies should be deployed such as stream computing, big data, and artificial intelligence (AI). In Section 6, we have concluded that cloud diagnostics will face challenges of a large amount of data and changeable data characteristics due to the features of cloud networks. It is not advisable to use traditional methods to solve these problems. Therefore, some cutting-edge technologies are needed. Stream computing can better deal with the massive data generated by the cloud network in real time with a lower latency. AI algorithms can provide more ways to improve detection accuracy. However, because of the multi-tenant characteristics of the cloud, it is hard to distribute a unified model of network traffic for the cloud. Thus, most algorithms cannot give good results if used directly. In the future, traffic may be classified by the types of tenants or other properties, and corresponding well-designed algorithms will be used for different situations of traffic.

3. Finally, analyzing only traffic data is proved to be not enough in Section 6. The information of the transport layer should be given more attention, because it will show the quality of traffic. Generally speaking, the transport layer protocol is TCP/IP and we call the data from the transport layer the TCP statistics. Moreover, some emerging transmission protocols are worth studying, such as quick UDP internet connection (QUIC) (Roskind, 2013). Related diagnostic tools should be designed based on the characteristics of the corresponding protocol. Shortly, P4 switches supporting the Openflow protocol will be more and more popular, which means that we can obtain more information on traffic quality. However, the more information we obtain, the more impact the network system will suffer from. Therefore, we must try to design an economical information collection method for diagnosis.

8 Conclusions

Diagnostics in the cloud is an area that is not fully matured but vital in future cloud network development. In this paper, we analyzed the definition of cloud diagnostics, and introduced and classified several state-of-the-art cloud network diagnostic systems. By understanding the design of these systems, we presented the differences between the cloud network diagnostics and traditional ones. Also, we proposed the requirements for designing cloud network diagnostic systems and tools. Thus, we obtained some key development directions for the cloud network diagnostics. We hope that our review of cloud network diagnostics can stimulate researchers' interest in this field and provide help for researchers' follow-up research.

Contributors

Yining QI and Chongrong FANG investigated and summarized the literature. Daxiang KANG and Biao LYU provided the insights to the cloud network deployed in the industry. Yining QI, Chongrong FANG, and Haoyu LIU drafted the manuscript. Yining QI, Chongrong FANG, Haoyu LIU, Peng CHENG, and Jiming CHEN revised and finalized the paper.

Compliance with ethics guidelines

Yining QI, Chongrong FANG, Haoyu LIU, Daxiang KANG, Biao LYU, Peng CHENG, and Jiming CHEN declare that they have no conflict of interest.

References

Aceto G, Botta A, de Donato W, et al., 2013. Cloud monitoring: a survey. *Comput Netw*, 57(9):2093-2115. https://doi.org/10.1016/j.comnet.2013.04.001

Andreyev A, 2014. Introducing Data Center Fabric, the Next-Generation Facebook Data Center Network. https://engineering.fb.com/2014/11/14/production-engineering/introducing-data-center-fabric-the-next-generation-facebook-data-center-network/

Armbrust M, Fox A, Griffith R, et al., 2010. A view of cloud computing. *Commun ACM*, 53(4):50-58. https://doi.org/10.1145/1721654.1721672

Arzani B, Ciraci S, Loo BT, et al., 2016. Taking the blame game out of data centers operations with NetPoirot. Proc ACM SIGCOMM Conf, p.440-453. https://doi.org/10.1145/2934872.2934884

Arzani B, Ciraci S, Chamon L, et al., 2018. 007: democratically finding the cause of packet drops. Proc 15th USENIX Conf on Networked Systems Design and Implementation, p.419-435.

Bahl P, Chandra R, Greenberg A, et al., 2007. Towards highly reliable enterprise network services via inference

of multi-level dependencies. Proc Conf on Applications, Technologies, Architectures, and Protocols for Computer Communications, p.13-24. https://doi.org/10.1145/1282380.1282383

Bannour F, Souihi S, Mellouk A, 2018. Distributed SDN control: survey, taxonomy, and challenges. *IEEE Commun Surv Tutor*, 20(1):333-354. https://doi.org/10.1109/COMST.2017.2782482

Calder M, Schröder M, Gao R, et al., 2018. Odin: Microsoft's scalable fault-tolerant CDN measurement system. Proc 15th USENIX Conf on Networked Systems Design and Implementation, p.501-517.

Casella G, Berger RL, 2002. Statistical Inference (2nd Ed.). Duxbury Press, Pacific Grove, USA.

Claise B, Sadasivan G, Valluri V, et al., 2004. RFC 3954: Cisco Systems NetFlow Services Export Version 9. https://www.hjp.at/doc/rfc/rfc3954.html

Dhamdhere A, Teixeira R, Dovrolis C, et al., 2007. NetDiagnoser: troubleshooting network unreachabilities using end-to-end probes and routing data. Proc ACM CoNEXT Conf, p.1-12. https://doi.org/10.1145/1364654.1364677

Duffield N, Haffner P, Krishnamurthy B, et al., 2009. Rule-based anomaly detection on IP flows. IEEE INFOCOM, p.424-432. https://doi.org/10.1109/INFCOM.2009.5061947

Fang CR, Liu HY, Miao M, et al., 2020. VTrace: automatic diagnostic system for persistent packet loss in cloud-scale overlay network. Proc Annual Conf of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication, p.31-43. https://doi.org/10.1145/3387514.3405851

Ganguli S, Corbett T, 2019. Gartner Magic Quadrant for Network Performance Monitoring and Diagnostics.

Garfinkel SL, 1999. Architects of the Information Society: Thirty-Five Years of the Laboratory for Computer Science at MIT. The MIT Press, Cambridge, USA.

Geng YL, Liu SY, Yin Z, et al., 2019. SIMON: a simple and scalable method for sensing, inference and measurement in data center networks. Proc 16th USENIX Conf on Networked Systems Design and Implementation, p.549-564.

Gong CY, Liu J, Zhang Q, et al., 2010. The characteristics of cloud computing. Proc 39th Int Conf on Parallel Processing Workshops, p.275-279. https://doi.org/10.1109/ICPPW.2010.45

Guo CX, Yuan LH, Xiang D, et al., 2015. Pingmesh: a large-scale system for data center network latency measurement and analysis. Proc ACM Conf on Special Interest Group on Data Communication, p.139-152. https://doi.org/10.1145/2785956.2787496

Herodotou H, Ding BL, Balakrishnan S, et al., 2014. Scalable near real-time failure localization of data center networks. Proc 20th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining, p.1689-1698. https://doi.org/10.1145/2623330.2623365

Huang P, Guo CX, Zhou LD, et al., 2017. Gray failure: the Achilles' heel of cloud-scale systems. Proc 16th Workshop on Hot Topics in Operating Systems, p.150-155. https://doi.org/10.1145/3102980.3103005

Jin YC, Renganathan S, Ananthanarayanan G, et al., 2019. Zooming in on wide-area latencies to a global cloud provider. Proc ACM Conf on Special Interest Group on Data Communication, p.104-116. https://doi.org/10.1145/3341302.3342073

Kanuparthy P, Dovrolis C, 2014. Pythia: diagnosing performance problems in wide area providers. Proc USENIX Conf on USENIX Annual Technical Conference, p.371-382.

Kim C, Bhide P, Doe E, et al., 2015. In-Band Network Telemetry via Programmable Dataplanes. Technical Specification P, 4:2015.

Li Z, Cheng Q, Hsieh K, et al., 2020. Gandalf: an intelligent, end-to-end analytics service for safe deployment in large-scale cloud infrastructure. Proc 17th USENIX Symp on Networked Systems Design and Implementation, p.389-402.

Marston S, Li Z, Bandyopadhyay S, et al., 2011. Cloud computing—the business perspective. *Dec Support Syst*, 51(1):176-189. https://doi.org/10.1016/j.dss.2010.12.006

Mell P, Grance T, 2011. The NIST Definition of Cloud Computing. Gaithersburg: Computer Security Division, Information Technology Laboratory.

Moshref M, Yu ML, Govindan R, et al., 2016. Trumpet: timely and precise triggers in data centers. Proc ACM SIGCOMM Conf, p.129-143. https://doi.org/10.1145/2934872.2934879

Padmanabhan VN, Ramabhadran S, Padhye J, 2005. NetProfiler: profiling wide-area networks using peer cooperation. Proc 4th Int Conf on Peer-to-Peer Systems, p.80-92. https://doi.org/10.1007/11558989_8

Peng YH, Yang J, Wu C, et al., 2017. deTector: a topology-aware monitoring system for data center networks. Proc USENIX Conf on Usenix Annual Technical Conf, p.55-68.

Roskind J, 2013. Quick UDP Internet Connections: Multiplexed Stream Transport over UDP. https://docs.google.com/document/d/1RNHkx_VvKWyWg6Lr8SZ-saqsQx7rFV-ev2jRFUoVD34/

Roy A, Zeng HY, Bagga J, et al., 2015. Inside the social network's (datacenter) network. Proc ACM Conf on Special Interest Group on Data Communication, p.123-137. https://doi.org/10.1145/2785956.2787472

Roy A, Zeng HY, Bagga J, et al., 2017. Passive realtime datacenter fault detection and localization. Proc 14th USENIX Symp on Networked Systems Design and Implementation, p.595-612.

Tan C, Jin Z, Guo CX, et al., 2019. NetBouncer: active device and link failure localization in data center networks. Proc 16th USENIX Conf on Networked Systems Design and Implementation, p.599-614.

Tibshirani R, 1996. Regression shrinkage and selection via the lasso. *J R Stat Soc Ser B*, 58(1):267-288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Veloso B, Malheiro B, Burguillo JC, et al., 2020. Impact of trust and reputation based brokerage on the CloudAnchor platform. Int Conf on Practical Applications of Agents and Multi-agent Systems, p.303-314.

Wang M, Li BC, Li ZP, 2004. sFlow: towards resource-efficient and agile service federation in service overlay networks. Proc 24th Int Conf on Distributed Computing Systems, p.628-635. https://doi.org/10.1109/ICDCS.2004.1281630

Wang T, Zhang WB, Ye CY, et al., 2016. FD4C: automatic fault diagnosis framework for web applications in cloud computing. *IEEE Trans Syst Man Cybern Syst*, 46(1):61-75. https://doi.org/10.1109/TSMC.2015.2430834

Widanapathirana C, Li J, Sekercioglu YA, et al., 2011. Intelligent automated diagnosis of client device bottlenecks in private clouds. Proc 4th IEEE Int Conf on Utility and Cloud Computing, p.261-266. https://doi.org/10.1109/UCC.2011.42

Wu X, Turner D, Chen CC, et al., 2012. NetPilot: automating datacenter network failure mitigation. Proc Conf on Applications, Technologies, Architectures, and Protocols for Computer Communication, p.419-430. https://doi.org/10.1145/2342356.2342438

Yu D, Zhu YB, Arzani B, et al., 2019. dShark: a general, easy to program and scalable framework for analyzing in-network packet traces. Proc 16th USENIX Conf on Networked Systems Design and Implementation, p.207-220.

Yu ML, Greenberg A, Maltz D, et al., 2011. Profiling network performance for multi-tier data center applications. Proc 8th USENIX Conf on Networked Systems Design and Implementation, p.57-70.

Zeng HY, Mahajan R, McKeown N, et al., 2015. Measuring and Troubleshooting Large Operational Multipath Networks with Gray Box Testing. Technical Report MSR-TR-2015-55 (Microsoft Research).

Zhang Q, Yu G, Guo CX, et al., 2018. Deepview: virtual disk failure diagnosis and pattern detection for Azure. Proc 15th USENIX Conf on Networked Systems Design and Implementation, p.519-532.

Zhu YB, Kang NX, Cao JX, et al., 2015. Packet-level telemetry in large datacenter networks. *ACM SIGCOMM Comput Commun Rev*, p.479-491. https://doi.org/10.1145/2829988.2787483

Zhuo DY, Ghobadi M, Mahajan R, et al., 2017. Understanding and mitigating packet corruption in data center networks. Proc ACM Conf on Special Interest Group on Data Communication, p.362-375. https://doi.org/10.1145/3098822.3098849

Yining QI received her BS degree in Control Science and Engineering at Zhejiang University, Hangzhou, China, in 2019. She is currently an MS candidate in Control Science and Engineering at Zhejiang University. Her research interests are cloud network, network diagnostics, and network measurement.

Peng CHENG received his BS degree in Automation and his PhD degree in Control Science and Engineering at Zhejiang University, in 2004 and 2009, respectively. From 2012 to 2013, he worked as a research fellow in Information System Technology and Design Pillar at Singapore University of Technology and Design, Singapore. He is currently a professor at the College of Control Science and Engineering, Zhejiang University. He is a corresponding expert of *Front Inform Technol Electron Eng*. His research interests include networked sensing and control, cyber-physical systems, and control system security.

Jiming CHEN received his BS and PhD degrees in Control Science and Engineering at Zhejiang University, in 2000 and 2005, respectively. He was a visiting researcher at the University of Waterloo, Canada, from 2008 to 2010. He is the Deputy Director of the State Key Laboratory of Industrial Control Technology and a member of the Academic Committee at Zhejiang University. He is now serving as an editor of *Front Inform Technol Electron Eng*. His research interests include the Internet of Things, sensor networks, networked control, and control system security. He was a recipient of the Fok Ying Tung Young Teacher Award of the Ministry of Education and the IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award. He is an IEEE Vehicular Technology Society distinguished lecturer and an IEEE fellow.