Network troubleshooting: Survey, Taxonomy and Challenges

Van TONG*, Hai Anh TRAN†, Sami SOUIHI* and Abdelhamid MELLOUK*
*LISSI-TincNET Research Team
University of Paris-Est Creteil (UPEC), France
Email: van.tong@tincnet.fr, sami.souihi@u-pec.fr, mellouk@u-pec.fr
†Bach Khoa Cybersecurity Centre
Ha Noi University of Science and Technology, Hanoi, Vietnam
Email: anhth@soict.hust.edu.vn

***Abstract*—With the robust improvement of the Internet, there are lots of network issues existing in the network systems. Many research works proposed different approaches to resolve the network issues in order to enhance the performance of network management and guarantee the user's QoE (Quality of Experience). However, it is not easy to have the global views about the network issues and the troubleshooting approaches. This paper presents a survey on troubleshooting with a special concentration on network issues. The main contribution of this survey is a detailed analysis of state-of-the-art related to network issues troubleshooting which evaluates their benefits and drawbacks. Moreover, these research work are classified into some categories to provide the useful guideline for troubleshooting research. Open issues and challenges in this field are also discussed. To the best of my knowledge, this is the first survey on troubleshooting of network issues.**

***Index Terms*—Network issues, Congestion, Troubleshooting, Network management**

I. Introduction

Nowadays, the number of users using the Internet and their services are increasing robustly. To maintain the quality of service and improve the user's QoE, many service providers utilize the cloud service of the third party to store and process the user's data. If the failures happen or there are any errors related to cloud service providers, this will lead to the reduction of their incomes. Cerin et al. [1] present a survey on downtime statistics of some service providers. In this work, it is reported that downtime in YouTube, CloudFlare, Paypal, Facebook are worth approximately $200,000, $336,000, $225,000, and $200,000 per hour, respectively. When the issues exist in the systems, the IT administrators have to do some methods to find out and troubleshoot the network issues as soon as possible. The longer they resolve the problems, the more serious the damage becomes.

In the traditional approach, the IT administrators use some probing tools such as *ping, traceroute, netperf, tcpdump*, etc. In [2], Zeng conducted the survey on 58 tools of different kinds of networks including small, medium, large and very large networks. In this report, *ping, traceroute* are two tools which are more popular than others. Moreover, it is reported that there are approximately 40, 30, 23 percent of IT administrators who spend from 5 to 30 minutes, from 30 to 60 minutes, and from 60 to 300 minutes per day to troubleshoot the network issues, respectively. Besides, there is 35 percent of administrator generating over 100 tickets per months. According to this report, the necessity of network troubleshooting is extremely large, and it plays an essential role in network diagnoses and management. The development of distributed systems emerge some challenges for the traditional probing tools. The failures location in these systems [3] is not easy because these systems span into many locations, caching servers, CDN (Content Delivery Network), etc. This paper presents some issues that frequently exist in the network systems, the troubleshooting in SDN and the overview of troubleshooting techniques to resolve these troubles.

The remainder of the paper is as follows. Section II describes the issues in the network. Section III categories and evaluates some network troubleshooting methods. Section IV presents some drawbacks and challenges of current network troubleshooting methods. The paper concludes with Section V which highlights the conclusion and future works.

II. Network Troubleshooting

This section provides an overview of some popular issues. The literature on network troubleshooting is extensive so that the preliminary overview of issues go far beyond this section. When an issue appears, network administrators do not know which kinds of issues they are and where they locate. The issues can result from network service providers (NSP) or application service providers (ASP). In this section, we classify the issues into network issues which caused by NSP and application issues which resulted from ASP. The detail is described in Fig. 1.

A. Network Issues

According to [2], **network reachability** is the network state which illustrates that the clients are unable to connect to the servers. This is the most popular issues in the network systems. There are two kinds of reachability issues [4] containing transient and non-transient. The transient reachability issues can be caused by some transient events (e.g. link flaps). The non-transient reachability issues are more difficult to be resolved from the failures of physic links, router misconfigurations,

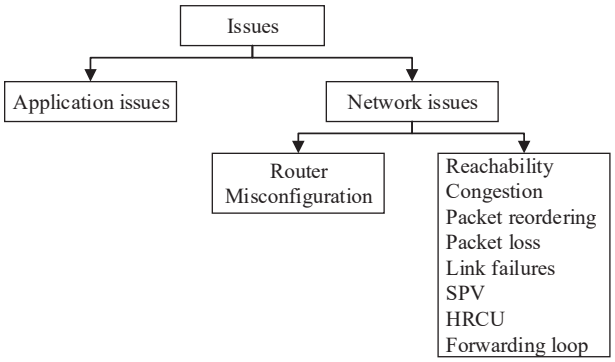

Fig. 1: Taxonomy of issues in the network.

etc. So many research works focus on this kind of issue. Fonseca et al. [5] proposed X-trace, a tracing framework that provides the global overview through reconstructing the task tree of all sub-operations making up the tasks. This can hep to understand the causal paths in the network protocols to define the location of the issues and make suitable decisions for network troubleshooting. Dhamdhere et al. [4] proposed NetDiagnoser, a troubleshooting method to define the location of errors in an internetwork environment and address the network unreachability. The authors extend the tomography approach [6] with some improvements and additional information including routing information, routing messages at ISP (Internet Service Provider), and end-to-end probing data to decrease the number of false positives in failure identifying.

Congestion is the result of network routes becoming too full when there are many requests on these network routes. Anand et al. [7] proposed *Net-Replay*, a method re-transmits the packets and receives the feedback of the network elements related to the congestion and the location of issues. When the receivers inform the senders that some arrival packets are in high delay, the receivers then replay the delayed packets, and the senders will parse the annotations in these packets to identify the location of issues. Similarly, Zeng et al. [8] also proposed a method using the packet re-transmission to locate the congestion in the network, called Automatic Test Packet Generation (ATPG). If the congestion exists, ATPG generates the congestion test to evaluate the latency between the pair packets in order to locate the location of congestion. In another approach, Chandra et al. [9] presented WiFiProfiler, a system in the wireless network, to diagnose and troubleshoot the network issues including network congestion. This system consists of three main components containing the sensing component to passively monitor the connectivity and configurations, the communication component to gather the information and the diagnosis component to identify the issues. In [10], Traverso uses the active probes and the packet level trace in the network flows to detect the congestion in the network. Kumar et al. [11] use the ICMP (Internet Control Message Protocol) packets to identify the congestion.

In [7], Net-Replay can point out the position of **packet re-ordering** and **packet loss** in the network. Packet re-ordering is the network behavior when the order of some packets in the flows is confused during the packet transmission in the network. Leung et al. [12] provide the comprehensive survey on packet re-ordering in TCP. In the packet re-ordering issues, the router where the change takes place will be remarked in the replayed packets. Packet loss is the issue that occurs when the packet transmission across the network fail to reach their destinations. For the packet loss issues, Net-Replay uses the *NotFoundAt* field in the replayed packet to inform the receivers about the location of packet loss. Besides, there are some research works concentrating on packet loss issues. In [11], the authors also present the proposed ICMP packets to support for troubleshooting the packet loss issues in the network. Mahimkar et al. [13] proposed a statistical correlations-based approach to troubleshoot the network issues. They use the local spatial proximity at the router level using the information extracted from syslogs and SNMP (Simple Network Management Protocol) measurements to detect the packet loss in the network. In [14], the authors proposed a tool, called BADABING, that overcomes the standard Poisson-based methods [15] to evaluate the end-to-end packet loss characteristics.

Link failure is the issue which caused by many factors such as a cable being unplugged, misconfiguration or denial of service attack, etc. Duffield et al. [6] [16] presented an approach using smallest consistent failure set (SCFS) rule to detect the failed links. Padmanabhan et al. [17] developed and evaluated three approaches based on passive network tomography using random sampling, linear optimization, and Bayesian inference using Gibbs sampling. The input of this method contains the header of TCP packets and corresponding ACKs. Kandula et al. [18] present Shrink, a tool using the Bayesian network and the SRLG (Shared Risk Link Groups) description and marginal probability of SRLG failures. In [19], Nguyen proposed a technique based on Maximum likelihood and Bayesian principles using the packet loss and end-to-end data transmission in the network to estimate the link loss rate and identify the link loss.

Security policy violation (SPV) is one of the most important issues in the network systems. This can be caused by many factors including malware, botnet or DDoS attack. These are the implicit threats causing to other network issues including congestion, packet loss, unreachability that can reduce the quality of service (QoS) and quality of experience (QoE). Karimazad et al. [20] proposed a method DDoS attack detection using Radial Basis Function (RBF) neural networks and features extracted from network traffic including average packet size, number of packets, time interval variance, etc. Nowadays, the DDoS attack can result from malware and botnet, so many research works concentrate on these fields. Wang et al. [21] presented a malware traffic classification using raw network traffic and convolutional neural network (CNN). Tran et al. [22] proposed an LSTM-based framework to detect the DGA botnet and their command and control servers.

Beyond the above issues, there are another network issues including **high router CPU utilization (HRCU)**, and

forwarding loop. According to troubleshooting technotes of Cisco [23], the high router CPU utilization can result from different reasons such as interrupts, processes, software encryption, fragmentation, etc. In [13], the proposed method collects CPU utilization measurements of the routers every 5 minutes via SNMP to detect and troubleshoot this issue. For the forwarding loop, these errors occur in the routing algorithm which lead to the looping in the path to the destinations. Whitaker et al. [24] proposed forwarding loop framework through inserting the small Bloom filter to the packet headers. Kazemian et al. [25] proposed a framework, called Header Space Analysis (HSA), that can detect the forwarding loop. If there are any packets across a port twice, this framework will implement some tests to find the location of issues.

All the above issues relate to the network symptoms while there is some issues caused by administrators such as **router misconfiguration**. In [26], Feamster proposed a router configuration checker that can detect the errors in BGP configuration using static analysis. This tool can detect two kinds of faults including route validity faults, and path visibility faults. Route validity faults are the issues where the routers learn the route that is not in usable paths while path visibility faults are the issues where the routers can fail to learn existing routes in the network. Le et al. [27] proposed a method applying association rules mining to configuration files of routers to address the problem of router misconfigurations. In [28], Aggarwal presents the method for the misconfiguration in the home network using machine learning algorithm (Decision Tree C4.5) and some network features including features related to flags of TCP, UDP packet, etc.

Similarly to traditional network systems, there are some issues in SDN-based network systems. Consequently, some researchers focus on **SDN troubleshooting**. Muthumanikandan et al. [29] present a survey on link failures on SDN. In [30], ONOS (Open Network Operating System) is an open-source project that can monitor the whole network to detect the forwarding loop in SDN in real time. Dethise et al. [31] proposed Prelude, a system for forwarding loop detection in SDN using Secure Multi-Party Computation (SMPC) techniques. Network Debugger (NDB) [32] implements breakpoints and packet backtraces to detect the correctness of forwarding such as logical errors, switch implementation errors, etc. In [33], NetSight uses the packet histories to troubleshoot some issues including packet drop, forwarding loop, congestions, etc. Gheorghe et al. [3] proposed SDN-RADAR that incorporates SDN features and distributed monitoring of network traffic to help administrators resolve some issues such as fault link, congestion, and packet loss.

B. Application Issues

When the errors appear, this can relate to the application faults. For example, the poor quality of some video call services do not result from the network issues, but it can be caused by the faults in the server of these services. Consequent, some researchers focus on the failure diagnose in applications. Zhuang et al. [34] designed a tool, called NetCheck, that can diagnose the problems in applications. NetCheck collects the sequence of system call generated by the applications to generate the network model and make the diagnoses. This tool can detect the faults in some applications such as FTP, Pidgin, Skype, and VirtualBox. Wang et al. [35] proposed PeerPressure that uses statistics and Window registry-based features to diagnose the application issues. PeerPressure can detect 20 real-world issues including slowing in MSN Instant Messenger, setting fails in IE Proxy, etc. Su et al. [36] and Zhang et al. [37] proposed the methods using configuration settings and the executing environment to troubleshoot the issues related to MySQL, PHP and Apache.

III. Network Troubleshooting Techniques

In this section, we present the overview of some troubleshooting techniques and categorize these methods according to their main algorithms including active methods, passive methods and hybrid methods. The active methods use probing packets into network systems to analyze and monitor the system. The passive methods just monitor the network traffic to detect and locate the issues. This approach does not increase the overhead of the system. The hybrid methods use both active probes and network traffic analysis to detect and locate the issues. The detail is described in Fig. 2. The characteristic of some reviewed research works is shown in Tab. I.

A. Active methods

1) Traditional Tool-based Methods: When the network systems have some abnormal behaviors, the IT administrators utilize some troubleshooting tools such as *ping, traceroute, tcpdump,* or *nmap* to define the network issues. It is simple to use these tools and requires low time processing. However, these tools cannot identify the location of the network issues. For instances, *ping* can provide the round trip time and packet loss, check the reachability of a host to another hosts or targets, but it cannot detect the root causes when the requests timed out.

2) Methods without Traditional Tool Utilization: This method uses probing packets through the network systems to get the information and evaluate the measures to locate the network issues. In [7], the authors replay the test packets to detect the location of network faults such as packet loss, packet re-ordering, and congestion. Zeng et al. [8] use the router configuration and test packet selection algorithms to find the test packets necessary. This method can detect and locate the network issues such as incorrect firewall rule, forward rule, congestion, strict priorities, etc. However, the probing methods need to send the test packets to analysis and evaluate. It is ineffective when test packets are lost due to some reasons. Besides, it also increases the link capacity when it transmits the test packets in the network systems. Moreover, some methods [7] need to modify the test packets to insert some useful feedback for network troubleshooting.

B. Passive methods

1) Rule-based Methods: The rule-based method uses the rule matching to interpret information in a useful way. In

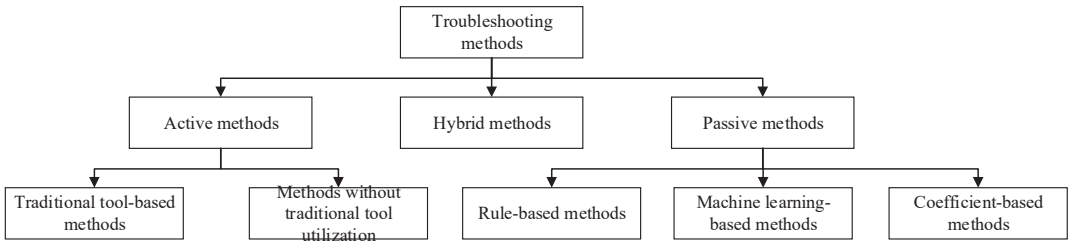

Fig. 2: Classification of troubleshooting approaches.

[27], Le uses router configuration and association rule mining, which learns the relationship of input data and identifies the strong rules using some measures of interestingness, to address the router misconfiguration. Moreover, Chen et al. [38] implement the association rule mining to detect the eBay's failures. Rule-based method does not require the test packets as in [7] and [8], but it has the essential drawbacks in the trade-off between time usage and accuracy. When the dataset of rules is huge, the accuracy will be high, but it requires more time processing. Therefore, it is not easy to implement in real-time.

2) Machine Learning-based Methods: Similarly, machine learning also uses statistical techniques to learn with data, and make the decision. In [28] and [38], they implement the decision tree (machine learning algorithm) to troubleshoot the misconfiguration in the home network, and eBay's failures, respectively. Kalibhat et al. [39] evaluate the performance of SVM (Support Vector Machine), ANN (Artificial Neural Network) and LR (Logistic Regression) in the troubleshooting application issues due to misconfiguration. Jayaraj et al. [40] classify the loss into congestion loss and contention loss using uses both a supervised learning technique (hidden Markov model) and an unsupervised learning technique (Expectation Maximization clustering). With the improvement of SDN, Mestres et al. [41] proposed a new paradigm, called Knowledge-Defined Networking that can improve the routing in an overlay network and resource management in an NFV (Network Function Virtualization) scenario using ANN (Artificial Neural Network). Nguyen et al. [19] use the Bayesian inference technique to troubleshoot the link loss in the sensor network. These methods can implement in real-time due to small time processing, but it is not easy to detect and locate new network and application issues. A noticeable features of the rule and machine learning-based methods are easy to expand in order to trouble new network issues because these methods only modify the data input while others have to change the core algorithms.

3) Coefficient-based Methods: Apart from the above methods, there are other approaches including consistent coefficient [6] or correlation coefficient [42] to detect and resolve link failures. Moreover, Mahimkar et al. [13] apply the correlation coefficient to address the packet loss, router CPU utilization and link loss. However, these methods are not easy to customize to detect the new kinds of network issues. Besides, the time processing of these methods is necessary to carefully evaluate.

C. Hybrid Methods

In [10], Traverso proposed a hybrid method that uses active probes, the packet level trace in the network, and implements correlation analysis to troubleshoot the congestion in the network. The active probes are implemented via data transmission to the server and log the application layer throughput. Zhuang et al. [34] use system call in the applications and the rule matching to identify the application issues. Besides, this method also supports traces generated by *strace* on Linux. The hybrid methods can combine the advantages of active methods and passive methods to improve the network performance.

IV. Challenges

In this section, we wrap up our survey with a qualitative look at the reviewed research works in the area of network troubleshooting.

A. Time Usage

Most of the reviewed works have not concentrated on the runtime of their methods. They only focus on finding the solutions to deal with network troubleshooting [3]–[5].

In [8], the authors pointed out that the time processing of the troubleshooting systems need to be less than the life's time of the faults. Besides, some methods estimate the runtime, but their research works have not evaluated whether the runtime is less than the life's time of the faults or not. In this paper, the authors also estimate the runtime of APTG algorithm, and propose the real-time version of APTG algorithm. Others [34] proposed the troubleshooting method using trace-ordering algorithm which has the best-case linear running time (less than 1 seconds). With the large trace (1GB trace), the time processing is less than 2 minutes.

B. Network Influencing

Troubleshooting without influencing the network systems need to be carefully concerned. Some research works [8] [43] focus on active methods which generate the test packets to the network systems in order to evaluate and troubleshoot the network issues. However, in these methods, they illustrated that the test packets consume approximately 1 percent of link capacity.

TABLE I: A summary of research viewed in section II.

Work	Method	Input	Traditional tools	Scalability	Issues
Anand et al. [7]	Active	No	No	Easy	Packet loss, packet reordering, congestion
Zeng et al. [8]	Active	Router configuration	No	Easy	Incorrect firewall rule, forward rule, congestion, strict priorities, etc.
Le et al. [27]	Passive	Router configuration	No	Easy	Router misconfiguration
Agarwal et al. [28]	Passive	Network traffic-based features	No	Easy	Application issues
Mestres et al. [41]	Passive	Network traffic-based features	No	Easy	Routing in an Overlay Network, and Resource Management in an NFV
Chen et al. [38]	Passive	6 features of basic trace	No	Easy	13 eBays failures
Jayaraj et al. [40]	Passive	Number of bursts between failures	No	Easy	Congestion or contention losses
Kalibhat et al. [39]	Passive	Heap, CPU, thread of master and slaves nodes	No	Easy	Application misconfigurations
Mahimkar et al. [13]	Passive	Layer-1 Alarms, SNMP, Router Syslogs, etc	No	Medium	Packet loss, router CPU utilization
Duffield et al. [6]	Passive	Status of links	No	Medium	Link failures
Kompella et al. [42]	Passive	SNMP MIB, router syslog, SONET PM data, router config	No	Medium	Link failures
Nguyen et al. [19]	Passive	End-to-end application traffic	No	Medium	Link failures
Traverso et al. [10]	Hybrid	Network traffic-based features	Yes	Medium	Congestion
Zhuang et al. [34]	Hybrid	System calls	Yes	Medium	Application issues

C. Accuracy

Network troubleshooting systems are to detect and locate the issues in the network. However, some methods can provide the false alarms which make the IT administrators difficult to manage the network systems. There are few research works concentrating on this issue. Padmanabhan et al. [17] proposed the method based on Random Sampling, Linear Optimization, and Bayesian Inference using Gibbs Sampling to troubleshoot the link loss with the false positive rate under 5 percent. Mahimkar et al. [13] presented NICE that can detect and define the location of packet loss for 98 percent of loss events. Prelude [31] is forwarding loop detection system that can achieve 0.3 percent of false positives. With machine learning or rule-based technique [27], the network troubleshooting systems can easily evaluate and estimate the accuracy. However, these approaches need to define the trade-off between troubleshooting performance and the resource consumption.

D. Scalability

Scalability is a factor which evaluates the expanding of troubleshooting systems when the IT administrators want to troubleshoot and locate new issues because one system only focuses on some defined network troubles. None of the reviewed research works seriously consider this issue. The machine learning and rule-based methods have the potential to easily expand and detect new kinds of networks issues. These methods only modify the data input to construct another model or dataset of the rule to detect other network issues. However, this issue is not investigated in the most of these works. With coefficient-based methods, it is not easy to modify the core algorithms which are compatible with some defined issues, to troubleshoot new network issues.

E. Knowledge Extraction

Currently, most of the reviewed research works are only responsible for network troubleshooting. Over time, the network systems not only resolve the current network issues but also it need to anticipate the implicit issues in the future to decide appropriate traffic policies [41]. It is necessary to analysis the log events to discover new knowledge. For instances, node X is usually congested at Y pm, some abnormal traffic sent to node Z every morning, etc. As for the long-term planning, network systems can use the historical data to forecast and estimate the network capacity (e.g. high traffic load, high packet loss, etc). For example, if we can estimate the traffic load, we can accurately estimate when the network system needs to be upgraded.

V. Conclusion

Network troubleshooting plays an essential role in network management and diagnoses. The novel aspect of this survey is special concentration on the issues in the network. These issues are categorized into application issues and network issues. Besides, we thoroughly investigate some troubleshooting approaches to provide the extensive understanding of their characteristics and the useful guidelines for network troubleshooting research. Moreover, we present some open issues with some insights into the future trends.

With the improvement of SDN, the necessary for SDN troubleshooting is very large. However, the research works on SDN troubleshooting are in early stage, so the implementation of network troubleshooting in ONOS (Open Network Operating System) [30] also is the necessary issue.

References

[1] C. Cérin, C. Coti, P. Delort, F. Diaz, M. Gagnaire, Q. Gaumer, N. Guillaume, J. Lous, S. Lubiarz, J. Raffaelli *et al.*, "Downtime statistics

of current cloud solutions," *International Working Group on Cloud Computing Resiliency, Tech. Rep*, 2013.

[2] H. Zeng, P. Kazemian, G. Varghese, and N. McKeown, "A survey on network troubleshooting," *Technical Report Stanford/TR12-HPNG-061012, Stanford University, Tech. Rep.*, 2012.

[3] G. Gheorghe, T. Avanesov, M.-R. Palattella, T. Engel, and C. Popoviciu, "Sdn-radar: Network troubleshooting combining user experience and sdn capabilities," in *Network Softwarization (NetSoft), 2015 1st IEEE Conference on*. IEEE, 2015, pp. 1–5.

[4] A. Dhamdhere, R. Teixeira, C. Dovrolis, and C. Diot, "Netdiagnoser: Troubleshooting network unreachabilities using end-to-end probes and routing data," in *Proceedings of the 2007 ACM CoNEXT conference*. ACM, 2007, p. 18.

[5] R. Fonseca, G. Porter, R. H. Katz, S. Shenker, and I. Stoica, "X-trace: A pervasive network tracing framework," in *Proceedings of the 4th USENIX conference on Networked systems design & implementation*. USENIX Association, 2007, pp. 20–20.

[6] N. Duffield, "Network tomography of binary network performance characteristics," *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5373–5388, 2006.

[7] A. Anand and A. Akella, "Netreplay: a new network primitive," *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, no. 3, pp. 14–19, 2010.

[8] H. Zeng, P. Kazemian, G. Varghese, and N. McKeown, "Automatic test packet generation," in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*. ACM, 2012, pp. 241–252.

[9] R. Chandra, V. N. Padmanabhan, and M. Zhang, "Wifiprofiler: cooperative diagnosis in wireless lans," in *Proceedings of the 4th international conference on Mobile systems, applications and services*. ACM, 2006, pp. 205–219.

[10] S. Traverso, E. Tego, E. Kowallik, S. Raffaglio, A. Fregosi, M. Mellia, and F. Matera, "Exploiting hybrid measurements for network troubleshooting," in *Telecommunications Network Strategy and Planning Symposium (Networks), 2014 16th International*. IEEE, 2014, pp. 1–6.

[11] M. A. Kumar, K. Appathurai, and P. Nagarajan, "Troubleshooting networks using internet control message protocol," *CiiT International Journal of Networking and Communication Engineering*, 2009.

[12] K.-C. Leung, V. O. Li, and D. Yang, "An overview of packet reordering in transmission control protocol (tcp): problems, solutions, and challenges," *IEEE transactions on parallel and distributed systems*, vol. 18, no. 4, pp. 522–535, 2007.

[13] A. Mahimkar, J. Yates, Y. Zhang, A. Shaikh, J. Wang, Z. Ge, and C. T. Ee, "Troubleshooting chronic conditions in large ip networks," in *Proceedings of the 2008 ACM CoNEXT Conference*. ACM, 2008, p. 2.

[14] J. Sommers, P. Barford, N. Duffield, and A. Ron, "Improving accuracy in end-to-end packet loss measurement," in *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4. ACM, 2005, pp. 157–168.

[15] R. W. Wolff, "Poisson arrivals see time averages," *Operations Research*, vol. 30, no. 2, pp. 223–231, 1982.

[16] N. Duffield, "Simple network performance tomography," in *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*. ACM, 2003, pp. 210–215.

[17] V. N. Padmanabhan, L. Qiu, and H. J. Wang, "Server-based inference of internet link lossiness," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 1. IEEE, 2003, pp. 145–155.

[18] S. Kandula, D. Katabi, and J.-P. Vasseur, "Shrink: A tool for failure diagnosis in ip networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data*. ACM, 2005, pp. 173–178.

[19] H. X. Nguyen and P. Thiran, "Using end-to-end data to infer lossy links in sensor networks," in *IEEE Infocom 2006*, no. CONF, 2006.

[20] R. Karimazad and A. Faraahi, "An anomaly-based method for ddos attacks detection using rbf neural networks," in *Proceedings of the International Conference on Network and Electronics Engineering*, vol. 11, 2011, pp. 44–48.

[21] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *Information Networking (ICOIN), 2017 International Conference on*. IEEE, 2017, pp. 712–717.

[22] D. Tran, H. Mac, V. Tong, H. A. Tran, and L. G. Nguyen, "A lstm based framework for handling multiclass imbalance in dga botnet detection," *Neurocomputing*, vol. 275, pp. 2401–2413, 2018.

[23] Cisco, "Technical notes of ciso," in *https://www.cisco.com/c/en/us/support/docs/routers/10000-series-routers/15095-highcpu.html*, 2016.

[24] A. Whitaker and D. Wetherall, "Forwarding without loops in icarus," in *Open Architectures and Network Programming Proceedings, 2002 IEEE*. IEEE, 2002, pp. 63–75.

[25] P. Kazemian, G. Varghese, and N. McKeown, "Header space analysis," Ph.D. dissertation, Stanford University, 2013.

[26] N. Feamster and H. Balakrishnan, "Detecting bgp configuration faults with static analysis," in *Proceedings of the 2Nd Conference on Symposium on Networked Systems Design & Implementation-Volume 2*. USENIX Association, 2005, pp. 43–56.

[27] F. Le, S. Lee, T. Wong, H. S. Kim, and D. Newcomb, "Detecting network-wide and router-specific misconfigurations through data mining," *IEEE/ACM transactions on networking*, vol. 17, no. 1, pp. 66–79, 2009.

[28] B. Agarwal, R. Bhagwan, T. Das, S. Eswaran, V. N. Padmanabhan, and G. M. Voelker, "Netprints: Diagnosing home network misconfigurations using shared knowledge." in *NSDI*, vol. 9, 2009, pp. 349–364.

[29] V. Muthumanikandan and C. Valliyammai, "A survey on link failures in software defined networks," in *Advanced Computing (ICoAC), 2015 Seventh International Conference on*. IEEE, 2015, pp. 1–5.

[30] ONOS, "Open network operating system," in *https://wiki.onosproject.org/display/ONOS/Wiki+Home*, 2017.

[31] A. Dethise, M. Chiesa, and M. Canini, "Prelude: Ensuring inter-domain loop-freedom in˜ sdn-enabled networks," *arXiv preprint arXiv:1806.09566*, 2018.

[32] N. Handigol, B. Heller, V. Jeyakumar, D. Maziéres, and N. McKeown, "Where is the debugger for my software-defined network?" in *Proceedings of the first workshop on Hot topics in software defined networks*. ACM, 2012, pp. 55–60.

[33] N. Handigol, B. Heller, V. Jeyakumar, D. Mazières, and N. McKeown, "I know what your packet did last hop: Using packet histories to troubleshoot networks." in *NSDI*, vol. 14, 2014, pp. 71–85.

[34] Y. Zhuang, E. Gessiou, S. Portzer, F. Fund, M. Muhammad, I. Beschastnikh, and J. Cappos, "Netcheck: Network diagnoses from blackbox traces." in *NSDI*, 2014, pp. 115–128.

[35] H. J. Wang, J. C. Platt, Y. Chen, R. Zhang, and Y.-M. Wang, "Automatic misconfiguration troubleshooting with peerpressure." in *OSDI*, vol. 4, 2004, pp. 245–257.

[36] Y.-Y. Su and J. Flinn, "Automatically generating predicates and solutions for configuration troubleshooting." in *USENIX Annual Technical Conference*, 2009.

[37] J. Zhang, L. Renganarayana, X. Zhang, N. Ge, V. Bala, T. Xu, and Y. Zhou, "Encore: Exploiting system environment and correlation information for misconfiguration detection," *ACM SIGARCH Computer Architecture News*, vol. 42, no. 1, pp. 687–700, 2014.

[38] M. Chen, A. X. Zheng, J. Lloyd, M. I. Jordan, and E. Brewer, "Failure diagnosis using decision trees," in *null*. IEEE, 2004, pp. 36–43.

[39] N. M. Kalibhat, S. Varshini, C. Kollengode, D. Sitaram, and S. Kalambur, "Software troubleshooting using machine learning," in *2017 IEEE 24th International Conference on High Performance Computing Workshops (HiPCW)*. IEEE, 2017, pp. 3–10.

[40] A. Jayaraj, T. Venkatesh, and C. S. R. Murthy, "Loss classification in optical burst switching networks using machine learning techniques: improving the performance of tcp," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 6, 2008.

[41] A. Mestres, A. Rodriguez-Natal, J. Carner, P. Barlet-Ros, E. Alarcón, M. Solé, V. Muntés-Mulero, D. Meyer, S. Barkai, M. J. Hibbett *et al.*, "Knowledge-defined networking," *ACM SIGCOMM Computer Communication Review*, vol. 47, no. 3, pp. 2–10, 2017.

[42] R. R. Kompella, J. Yates, A. Greenberg, and A. C. Snoeren, "Ip fault localization via risk modeling," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*. USENIX Association, 2005, pp. 57–70.

[43] R. C. N. D. J. Horowitz and D. T. T. Bu, "Multicast-based inference of network-internal characteristics: accuracy of packet loss estimation," in *Proc. IEEE Infocom*, 1999.