Network State Estimation by Spectral Analysis of Passively Measured TCP Flows

Kenta Murayama
Graduate School of Informatics
Kyoto University
Kyoto, Japan
murayama@net.ist.i.kyoto-u.ac.jp

Yasuo Okabe
Academic Center for Computing
and Media Studies, Kyoto University
Kyoto, Japan
okabe@i.kyoto-u.ac.jp

***Abstract*—This paper presents a novel method for estimating the states of upstream networks by analyzing TCP flows at key traffic aggregation points in home and enterprise environments. Utilizing a machine learning framework that incorporates frequency domain features from RTT time series, we conducted a preliminary evaluation on a simulated virtual network. The results demonstrate the method's capability to accurately classify the network state of a single flow within a basic model. Our findings suggest the method's potential for broader application in network state estimation.**

***Index Terms*—tcp flow, passive monitoring, network state estimation, spectral analysis**

I. Introduction

In the current landscape of our fast-evolving information society, the Internet has become indispensable to daily operations. Consequently, the ability of network administrators to swiftly identify and address network failures is paramount. Yet, as network architectures grow in complexity, the challenge and responsibility of detecting faults have significantly intensified.

Since the Internet is composed of networks of organizations around the world, communication is rarely completed on a single network alone, and it is common for communications to pass through multiple networks. When a user perceives a degradation of network quality, the candidates for the cause are all the networks on the communication path. In a stub network such as a home or enterprise, when a quality degradation occurs, the administrators can use SNMP [6] or other means to check the status of the communication devices under their control, but it is not easy to obtain information about the network operated by another administrator. Therefore, it is not easy for the administrator to identify the cause of quality degradation. Thus, research is being conducted to estimate the network state by observing the state of protocols that manage end-to-end communication, such as TCP, instead of observing the state of communication devices.

There are two methods for measuring network traffic: active measurement, in which new traffic is generated for investigation, and passive measurement, in which existing traffic is observed. Active measurement is often used for end-to-end quality measurements, such as Iperf [5], but it can place a heavy load on the network as the measurement traffic increases. Therefore, research is underway to use passive measurements to analyze network traffic. Specifically, some studies have passively measured multiple TCP flows to capture traffic characteristics [4] or to identify bottlenecks [9]. Some focus on the behavior of a single TCP flow to estimate its TCP variant [3]. However, few studies have been found that estimate problems in the unmanaged network by passive measurement of the managed network.

It is very meaningful to be able to estimate the state of the upstream network by measuring the managed network. When the quality of communication deteriorates, analyzing each flow one by one only tells us that the cause is somewhere on the path of each flow. However, by analyzing flows that share a common path, if a degradation of communication quality is confirmed for those flows, it can be inferred that some failure has occurred on that common path. The status of each flow can be estimated with high accuracy by analyzing the time series of sequence number, acknowledgement number, and window size in the TCP header in detail [15], but it is difficult to perform detailed analysis for a huge number of flows.

Therefore, we propose a method to estimate the upstream network status by monitoring multiple TCP flows at the boundary router, which is the traffic aggregation point of the managed network, for home and enterprise networks. First, multiple TCP flows are passively measured, and the information stored in their headers is analyzed for each flow. By comprehensively evaluating this information, we attempt to identify the cause of performance degradation on the upstream network paths commonly traversed by the flows. Focusing on the periodicity of TCP congestion control, the time-series data obtained is dropped into frequency domain by performing a spectral analysis using the Lomb-Scargle method. Strong frequency components are extracted as features, and the state of the network is inferred by machine learning. However, it is not easy to prepare supervised data because it is rare to identify the cause of performance degradation in a real network. Then, we have used NS-3 to simulate various events in the network and have created the data sets necessary for machine learning.

In this paper, we focus on the flow-by-flow analysis part of the proposed method and evaluate our approach by creating a dataset with the simplest virtual network as a preliminary

This work was supported by JSPS KAKENHI Grant Number 19H04094.

experiment. Although there are numerous considerations to take into account, we have confirmed that it is possible to classify the state of the network with high accuracy in a simple model.

The remainder of this paper is structured as follows. Section II describe the technology behind this paper and related research. In Section III, we describes the proposed method, and Section IV explains the structure of the preliminary experiments and their evaluation. Finally, Section V summarizes the paper.

II. Related Work

We will discuss studies that use passive measurements of TCP, where the observation point is an intermediate node rather than an end host. There are two main types of research on passive measurement for intermediate nodes. One is to measure a single flow and analyze the behavior of that flow [3], [16], and the other is to measure multiple flows and analyze the state of the entire network [4], [8], [9]. We provide a brief introduction to some related studies.

Estimation of congestion window size: They employed Byte-in-Flight, calculated from the TCP headers, as a feature and attempted congestion window size estimation using ensemble machine learning algorithms [3]. The results showed that the predictions were highly accurate for multiple types of TCP variants.

Estimating RTT using unidirectional packet traces.: Since network operators are not always able to observe traffic in both directions, a method of estimating RTT using only traffic in one direction was studied [16]. Among the methods for estimating RTT, it is shown that the method of spectral analysis of packet arrival intervals using Lomb periodgram is more advantageous than the method using autocorrelation. However, the method using the Lomb periodgram also only roughly estimated the RTT, and it was confirmed that it is difficult to accurately estimate the RTT.

Analysis of mobile network traffic characteristics: A study [4] analyzed the ToD (Time of Day) effect by performing passive measurements on a mobile network. The number of sessions, session size, RTT, etc. were extracted from multiple TCP flows and the relationship with ToD was shown. For periodicity analysis, Lomb-Scargle periodograms, which are used for spectral analysis in the field of astronomy, were used to confirm that there is a strong periodicity in a day.

Identification of wireless LAN bottlenecks by analyzing TCP at traffic aggregation points: One study [9] estimates bottlenecks caused by wireless LANs by passive measurement on the backbone router, which is the aggregation point of the network. If a bottleneck exists in the management network, the round-trip delay at the observation point depends on the bottleneck delay, and the round-trip delay is expected to increase. By assuming this property, the existence of bottlenecks in the managed network was estimated. The location of the bottleneck is also determined by integrating the information from the wireless APs.

Estimation of bottleneck links using per-flow packet loss rate and their paths: A study [8] estimated bottleneck links using per-flow loss rates and their routes. In addition to identifying the bottleneck link, they also attempted to identify the cause of the bottleneck by linking TCP retransmissions and duplicate acknowledgements.

III. Proposed Method

This section describes a method for estimating the state of upstream networks by monitoring multiple TCP flows in communication devices that serve as traffic aggregation points connecting to upstream networks, targeting home and enterprise networks. The main idea and flow of the proposed method is described and detailed in subsections.

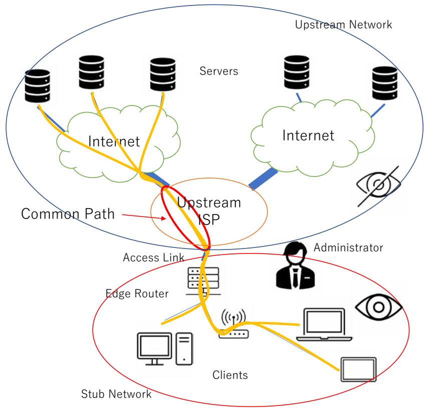


Fig. 1: Overview of the proposed method

Figure 1 is a overview diagram of the Internet including stub networks. Consider a client in the stub network connecting to a server in the upstream network using TCP. When they are connected by a yellow route as shown in Figure 1, the three routes will have a common route on the upstream ISP. If ”*event*” occurs on the common path, its impact occurs on all three communications, causing degradation of the network’s communication performance. However, ”*event*” is defined as the event that causes the performance degradation of the network, and this term is used in the discussion for simplicity. Using this causality, when a degradation of network performance is detected in each communication, it is judged that ”*event*” has occurred in the common path, and the state of the upstream network is estimated.

The flowchart of the proposed method is as follows, and is summarized as a figure in Figure 2.

1) Reproduce the situation of the network on the simulator.
2) Collect all packets by packet capture at traffic aggregation points in the simulator.
3) Select flows from the collected packets that are suitable for inference.
4) Collect data for each feature from the selected flows.
5) Apply the necessary preprocessing to the features.
6) Train machine learning models.
7) Infer the state of the network for each flow using the learned model.

8) Estimate the state of the upstream network from the observation point by examining common network conditions for multiple flows

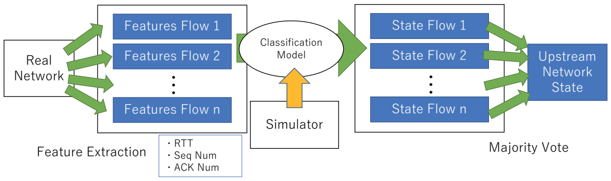


Fig. 2: Flowchart of the proposed method

In Section III-A, we delve into the underlying causes of network performance degradation and outline the scope of our analysis on the resulting phenomena. Moving on to Section III-B, we shift our focus to the creation of datasets for training machine learning models. Section III-C is dedicated to elucidating the targets of passive measurement, while Section III-D provides insights into the criteria guiding our selection of flows for analysis from the collected traffic. Within Section III-E, we outline the process of feature extraction from the measured data, and Section III-F takes you through the methodology we employ to classify network states using machine learning techniques.

A. Assumption of Factors that Degrade Network Performance and Scope of its Detection

The target scope of detection is shown in Figure 3. Since the state is estimated using the common paths of TCP flows, the target scope is the paths that a certain number of flows pass through in common. Therefore, the scope is from the connection point with the upstream network to the entrance of each Internet. The root causes of communication quality degradation could be network congestion, communication equipment failure, or external factors. These phenomena cause temporary increases and fluctuations in packet loss and delay, which degrade the quality of communication.

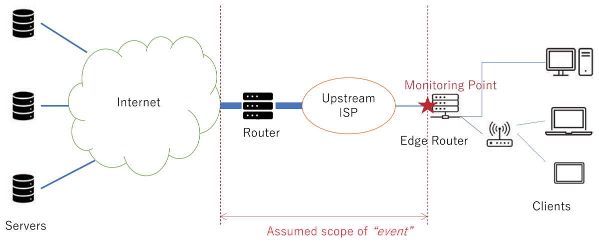


Fig. 3: Assumed scope of ”*event*”

B. Dataset

While it is possible to generate unsupervised data by collecting traffic in real networks, it is extremely difficult to collect supervised data that reflects actual network conditions. Therefore, supervised data is collected by modeling and simulating the real network. When constructing a virtual network, it is necessary to consider bandwidth, propagation time, and packet loss rate for lines, packet queue size and Active Queue Management (AQM) for routers, TCP congestion control algorithms and applications using TCP for end hosts. The above parameters are appropriately selected to simulate the situation of the target network.

C. Target for Measurement

To obtain information on the upstream network, it is preferable to measure routers that are connected to the upstream network and aggregate traffic. Therefore, we target broadband routers in home networks and backbone routers in enterprise networks. The target port for packet capture is the port connected to the upstream network. In addition, the protocols to be measured are only TCP, which has congestion control and whose behavior can be observed from its header information.

Among the fields of the TCP header, the sequence number, the acknowledgment number, the window size, and the value of the Timestamp option are useful for analyzing the network state. RTT and Byte-in-Flight, which can be inferred from these values, are also included in the analysis.

D. Flow Selection

It is difficult to estimate the state of a flow that has experienced little or no communication performance degradation. On the other hand, flows that send out data to fill the bandwidth as much as possible, such as bulk transfer, are considered to be strongly affected by network conditions. Therefore, we consider flows that perform bulk transmission as candidates for analysis.

E. Feature Extraction

Since time-series data is unwieldy for machine learning that is not deep learning, it must be converted to features with a small dimensionality. Since it is known that TCP during steady state has a periodicity, it is converted from the time domain to the frequency domain. Since the time-series data that can be obtained from the TCP header is unequally spaced, the Lomb-Scargle method [7] is used to convert it to the frequency domain. By using values with large amplitude in the frequency domain as feature values, we can extract strong periodicity in the time series data as representative.

For measurement data that cannot be meaningfully converted to the frequency domain, the maximum value, average value, minimum value, etc. are used as feature values. Measured values such as segment counts are also considered as feature values.

F. Machine Learning Classification

The machine learning model is trained using the extracted features to classify the state of the network for each flow. Inference of the upstream network state is performed by majority voting of the inference results of the selected flows.

IV. Preliminary Experiments and Discussion

This section describes the configuration of the preliminary experiments of the proposed method, and provides an evaluation and discussion of the results. We simulated the simplest network configuration, created a dataset, and comprehensively evaluated the inference accuracy and other aspects of the machine learning model for a single flow.

A. Modeling Networks in Preliminary Experiments

In this preliminary experiment, we assume a situation where large files are downloaded from an upstream network. The network can then be divided into an upstream network and a stub network, each of which is described below. The stub network consists of end hosts and communication devices that aggregate their communications, and can be modeled as shown on the right side of Figure 4. The upstream network consists of servers and the Internet, which is the path from the servers to the stub network. Therefore, it can be modeled as shown on the left side of Figure 4. For the following explanations, we define server-side delay as the delay between the server and the Internet in Figure 4.

Figure 5 shows the occurrence of ”*event*” that is the subject of this study in the modeled network. The assumed scope of the ”*event*” in Figure 3 is encompassed by the Internet in Figure 4. Therefore, ”*event*” in the modeled network can be considered to occur at a single node, the Internet. The passive measurement is performed on the router that connect to the Internet, as shown in Figure 5 .

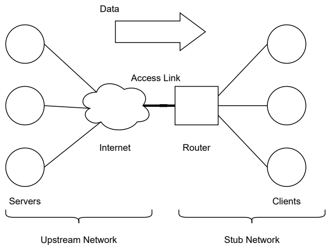


Fig. 4: Modeling the network

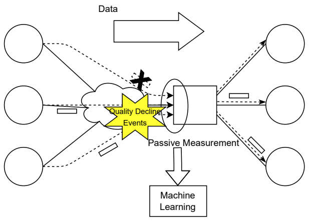


Fig. 5: Occurrence of ”*event*” in the model network

B. Testbeds in Preliminary Experiments

The network modeled in Section IV-A is simulated using ns-3 [2]. In this experiment, we reproduced normal network conditions, congestion in a network containing only TCP flows, and network failure due to link transmission errors. In each network condition, the simulator was run for 30 seconds and data was collected. The configuration of the network used in the simulation is shown in Figure 6, and the parameters of the link are shown in Table I. This configuration replaces the Internet portion of the network modeled in Figure 4 with a router. The above three network conditions are simulated by setting the parameters of this network configuration accordingly. Sections IV-B1 $\sim$ IV-B3 describe the specific settings of the above three models in the simulator.

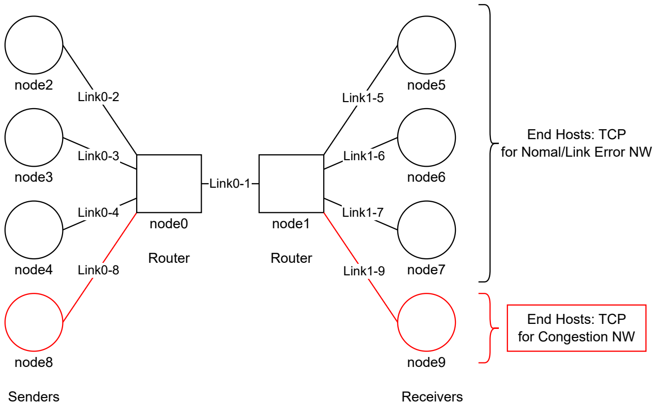


Fig. 6: Virtual Network Topology

TABLE I: Link settings

Link	delay time(ms)	bandwidth(Mbps)
0-1	1	30
0-2 $\sim$ 0-4	1 $\sim$ 50	10
1-5 $\sim$ 1-7	10	10
0-8,1-9	10	10

1) Normal Network Configuration: In the normal network, only node2 $\sim$ 7 are used, and TCP is used for communication between the left and right nodes. In this case, it can be seen that congestion does not occur on Link0-1 due to the link configuration. The delay time is set randomly from Link0-2 to Link0-4 to reproduce the physical distance of the network. However, it does not account for the variability in latency known as jitter. Other settings are shown below.

- Congestion control algorithm: NewReno [10]
- Queuing Algorithm : Droptail
- Queue size : 40 packets
- MTU : 1500byte
- Applications running on TCP : Bulk transfer

2) Congestion Network Configuration: In the congestion network, all nodes from node2 $\sim$ 9 are used, and TCP is used for communication between the left and right nodes. In this case, Link0-1 becomes the bottleneck link due to the link configuration, and congestion occurs at the router of node0.

3) Link Error Network Configuration: We reproduced a failure in which packet loss occurs with a certain probability on Link0-1 in Figure 6 of the Normal network. However, packet loss occurs in two-way communication. The packet loss rate was set to 0.001.

4) Data collected: The data subjected to analysis in this preliminary experiment were RTT between intermediate node and sender, Byte-in-Flight, the number of segments in the duplicate acknowledgement.

The RTT estimation method is supplemented with Figure 7. We look at the Timestamp Value (TSval) of the ACK segment sent by the receiving node and the segment with the corresponding Timestamp Echo Reply (TSEcr). The difference in the time taken for each segment to reach the intermediate node corresponds to the RTT. However, instead of using the raw RTT values for analysis, smoothed values are used.

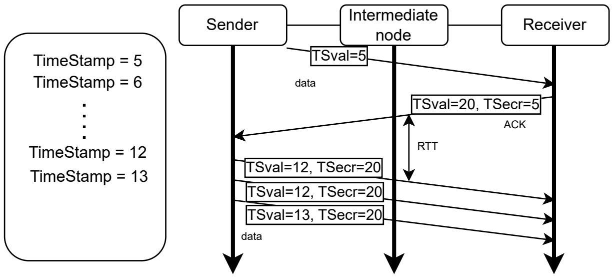


Fig. 7: Estimation of RTT between intermediate and sending nodes

C. Analysis of RTT Time Series Data

We set the delay time of Link0-2 $\sim$ Link0-4 in Figure 6 to 1, 5, and 25 ms, respectively, and took measurements. The results are shown in Figures 8. Representative RTT time series data for each network situation are shown in Figures (a), (c) and (e), and periodograms transformed by the Lomb-Scargle method are shown in Figures (b), (d) and (f). In the periodograms, the top three plots with the largest amplitudes are marked with red dots. In all networks, the slow-start ends about 5 seconds after the start, during which time a rapid increase and decrease in RTT can be observed. The subsequent behavior is described for each network.

1) Normal Network: As a whole, it repeats relatively clear sawtooth waves. Figure (b) shows that the peaks stand out clearly due to its periodicity.

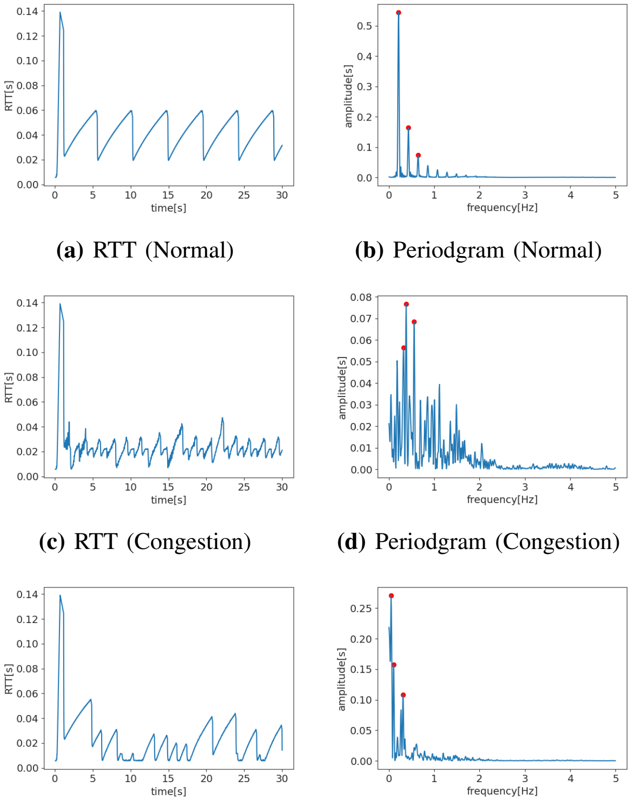
2) Congestion Network: As a whole, it oscillates with repeating small and large peaks. In Figure (d), it can be seen that the oscillations have various frequency components without large peaks.

3) Link Error Network: Figure (e) confirm that there is no characteristic period and that there are large peaks in rare cases. Figure (f) confirm that the amplitude of the frequency component is large, as seen in the relatively clean sawtooth wave.

D. Classification Performance with Selected Features

The following three features were collected in this preliminary experiment.

- Frequency domain of RTT time series data: The frequency component and amplitude of the point of maximum value in the frequency domain are used as the feature values. In addition, the number of selected peaks is added as the feature values in order of increasing amplitude.
- Duplicate ACK (DUP): The number of segments with the same acknowledgment number during the measurement period is a feature.
- Byte-in-Flight (Byte): The maximum value of Byte-in-Flight during the measurement period is taken as the feature value.



(a) RTT (Normal) **(b)** Periodgram (Normal)

(c) RTT (Congestion) **(d)** Periodgram (Congestion)

(e) RTT (Link Error) **(f)** Periodgram (Link Error)

Fig. 8: RTT time series of TCP flow with server-side delay of 1 ms and their periodograms

We then show how much each feature contributed to the performance of the classification. A random forest was used as the classifier. The settings were those described in Section IV-B, and the comparisons were made based on accuracy. Since we could not prepare a sufficient number of training data, we trained 10 times, using 70% as training data and 30% as evaluation data, and evaluated the results based on the average accuracy.

Table II shows the experimental results. The baseline is the one in which only the frequency domain is selected as a feature, and is compared with those to which other features are added. The number of peaks taken from the frequency space corresponds to the columns. From this result, it can be considered that Byte-in-Flight and Duplicate ACK are not very effective features.

In a decision tree-based ensemble classifier, feature importances can be calculated. Therefore, Table III shows the feature importance of each when Byte-in-Flight, Duplicate ACK, and frequency components up to the third peak are added to the features. This shows that Byte-in-Flight does not contribute much to classification in any of the conditions. It can also be confirmed that both the amplitude and frequency of the first peak have a high contribution to classification in all conditions.

E. Evaluation by Confusion Matrix

From Section IV-D, it was confirmed that there was no significant change in classification accuracy even when only features in the frequency domain were used without adding other features. Therefore, we evaluated the classifier using up to the third peak in the frequency domain as the feature. To simplify the table, the labels of Normal/Congestion/Link Error are denoted as N/C/L.

The results are shown in Table IV. As a whole, there were many cases of misclassification between Link Error and Congestion. When the maximum server-side delay is 50 ms, the cases of misclassification of Link Error and Congestion are smaller than other cases, and when the maximum server-side delay is 100 ms, the cases of misclassification of Normal and LinkError are also more frequent.

TABLE II: Accuracy for each selected feature

Feature\#Peak	1	2	3
Baseline	0.818	0.859	0.925
+BiF	0.800	0.914	0.903
+DUP	0.837	0.859	0.851
+BiF and DUP	0.800	0.877	0.892

TABLE III: Feature Importance

Feature\#Peak	1	2	3
amplitude_1	0.399	0.311	0.27
frequency_1	0.365	0.231	0.226
amplitude_2	x	0.129	0.094
frequency_2	x	0.146	0.074
amplitude_3	x	x	0.11
frequency_3	x	x	0.067
DUP	0.216	0.149	0.138
BiF	0.02	0.033	0.02

Feature labels
amplitude_i : Amplitude of the i-th peak
frequency_i : Frequency component of the i-th peak
DUP : Duplicate ACK
BiF : Byte-in-Flight

TABLE IV: Confusion matrix based on server-side delay

Delay time	10ms			50ms			100ms		
Correct \Prediction	N	C	L	N	C	L	N	C	L
N	90	0	0	87	0	3	80	0	10
C	0	79	11	0	87	3	0	66	24
L	2	20	68	1	7	82	8	15	67

V. Conclusion

This study targets home and enterprise networks and examines a method for estimating the state of upstream networks by monitoring multiple TCP flows at communication devices that serve as traffic aggregation points connecting to the upstream network.

The contributions of our proposed method are the following three points. First, by passively measuring multiple TCP flows and analyzing the information in their headers, we attempted to identify the events that cause performance degradation in the upstream network paths commonly traversed by the flows. Second, focusing on the periodicity of TCP congestion control, we performed spectral analysis using the Lomb-Scargle method on time-series data obtained from headers, and showed that using strong frequency components in frequency space as features for machine learning may be effective in classifying network states. Third, we proposed a method to reproduce various network conditions using a simulator to generate supervised datasets for real network classification.

While this paper focuses on the simplest of the factors to be considered in Section III-B and only conducts preliminary experiments, we are currently collecting real-world network traffic and are investigating the differences in traffic characteristics between real and virtual networks.

Acknowledgment

The authors are deeply grateful to Dr. Daisuke Kotani, Kyoto University, for his discussion and valuable comments.

References

[1] Borman, D. A., Braden, R. T. and Jacobson, V.: TCP Extensions for High Performance, RFC 1323 (1992).

[2] ns-3 — a discrete-event network simulator for Internet systems, https://www.nsnam.org/. (Accessed on 01/25/2022).

[3] Hagos, D. H., Engelstad, P. E., Yazidi, A. and Kure, c.: A machine learning approach to TCP state monitoring from passive measurements, 2018 Wireless Days (WD), pp. 164171 (2018).

[4] Garcia, J., Alfredsson, S. and Brunstrom, A.: Examining TCP Short Flow Performance in Cellular Networks Through Active and Passive Measurements, Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges, AllThingsCellular '15, New York, NY, USA, Association for Computing Machinery, p. 712 (2015).

[5] iperf3 - iperf3 3.10.1 documentation, https://software.es.net/iperf/. (Accessed on 01/30/2022).

[6] Wijnen, B., Harrington, D. and Presuhn, R.: An Architecture for Describing SNMP Management Frameworks, RFC 2571 (1999).

[7] Scargle, J. D.: Studies in astronomical time series analysis. II. Statistical aspects of spectral analysis of unevenly spaced data., The Astrophysical Journal, Vol. 263, pp. 835853 (1982).

[8] Brosh, E., Lubetzky-Sharon, G. and Shavitt, Y.: Spatial-temporal analysis of passive TCP measurements, Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies., Vol. 2, pp. 949-959 vol. 2 (2005).

[9] Sumiyo, O., Chunghan, L. and Tomohiro, I.: A Measurement Method for Wireless Bottlenecks on Enterprise Network, IEICE Technical Report IA2018-75 , Vol. IEICE-118, pp. 279285 (2019).

[10] Gurtov, A., Henderson, T., Floyd, S. and Nishida, Y.: The NewReno Modification to TCP's Fast Recovery Algorithm, RFC 6582 (2012).

[11] Ha, S., Rhee, I. and Xu, L.: CUBIC: A New TCP-Friendly High-Speed TCP Variant, SIGOPS Oper. Syst. Rev., Vol. 42, No. 5, p. 6474 (2008).

[12] Brakmo, L. and Peterson, L.: TCP Vegas: end to end congestion avoidance on a global Internet, IEEE Journal on Selected Areas in Communications, Vol. 13, No. 8, pp. 1465-1480 (1995).

[13] Cardwell, N., Cheng, Y., Gunn, C. S., Yeganeh, S. H. and Jacobson, V.: BBR: Congestion-Based Congestion Control, ACM Queue, Vol. 14, September-October, pp. 20-53 (2016).

[14] Tan, K., Song, J., Zhang, Q. and Sridharan, M.: A Compound TCP Approach for High-Speed and Long Distance Networks, Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications, pp. 1-12 (2006).

[15] Tlaiss, Z.: Anomaly root cause diagnosis from active and passive measurement analysis, 2021 33th International Teletraffic Congress (ITC-33), pp. 1-3 (2021).

[16] Kato, T., Yan, X., Yamamoto, R. and Ohzahata, S.: A Study on Round-trip Time Estimation from Unidirectional Packet Traces Using Different TCP Congestion Control Algorithms, International Journal on Advances in Networks and Services Volume 12, Number 1 & 2, 2019 (2019).