Locating Delay Fluctuation-Prone Links by Packet Arrival Intervals in OpenFlow Networks

Nguyen Minh Tri
Kyushu Institute of Technology
Fukuoka, Japan
tri.nguyen-minh414@mail.kyutech.jp

Syunya Nagata
Kyushu Institute of Technology
Fukuoka, Japan
p232206s@mail.kyutech.jp

Masato Tsuru
Kyushu Institute of Technology
Fukuoka, Japan
tsuru@cse.kyutech.ac.jp

***Abstract*—In cloud computing and content delivery networking, OpenFlow-based centrally managed networks to connect distributed servers are becoming popular these days. To maintain service quality and availability in such networks by flexible and dynamic traffic engineering, detecting and locating deteriorated (e.g., congested) links in an efficient manner is essential. Following our previous study that actively monitors packet loss rate to find deteriorated links, in this paper, we actively estimate packet delay variance on each link (note both up and down directions of each full-duplex link are distinguished) in an OpenFlow network. A notable feature is that packet delay variance is estimated based on monitoring arrival time intervals of probe packets without directly measuring packet delay time over a link. In the proposed scheme, a series of probe packets is launched from a measurement host and traverses each direction of each link once and only once by multicasting, while arrival time intervals of those packets at each input port of OpenFlow switches are monitored. Then the OpenFlow controller collects the arrival time interval statistics from those switches to locate delay fluctuation-prone links, i.e., links with a high packet delay variance, which are likely congested or physically unstable. In addition, to minimize the necessary number of accesses to switch ports, an appropriate order of collecting statistics from switches is dynamically controlled. The results of numerical simulation on large-scale network topologies demonstrate the effectiveness of our proposed scheme. A prototype implementation which requires an extension of OpenFlow is also presented on Mininet.**

***Index Terms*—active measurement, multicast probing, delay variance, OpenFlow network**

I. Introduction

The Software Defined Networking (SDN) technology in general and OpenFlow technology in particular have been introduced to realize dynamic and reliable networking and proliferated not only in data centers but also in wide area networks (WAN), so called SD-WAN (Software Defined-WAN), and also wireless networks. In particular, the ongoing penetration of cloud computing and contents delivery networking requires a flexible traffic engineering on a network connecting globally-distributed datacenters, which is often centrally managed by OpenFlow [1], [2].

One of the most important tasks in networking is network status monitoring. Operators need to know the network status information in a real-time manner to make decisions about trouble-shooting, dynamic routing, load balancing, Service Level Agreement (SLA) management, and so on. In general, there are two kinds of measurement approaches: passive and active. The passive approach is used to monitor link traffic state by using the statistics information requested from switches or the operating messages of OpenFlow standard. In general, there is a trade-off between the measurement accuracy and the load incurred on switches and the control network. There are some works about this issue. In [3], the authors introduced a dynamic algorithm to balance the request frequency and the accuracy. The impact of queried switch selection on the accuracy is discussed in [4]. With no additional load, [5] can calculate network utilization by only using FlowRemoved and PacketIn messages of OpenFlow standard.

On the other hand, the active approach sends and receives probe packets to measure the packet loss, delay, the round-trip-time (RTT), and so on. With the developing of the edge-cloud computing for emerging IoT technologies, it is required reliable networks among a large number of heterogeneous sites over geographically-wider locations. In such networks, a "link" between two nodes is not always physical but virtual (e.g., tunneling). So an active measurement by probing packets is essential to monitor entire network information. However, probing at a high sending rate for a long duration can cause more load incurred on switches and the data network. Therefore, there are some studies to reduce such load but still retain the reliability and precision. Authors in [6] proposed a infrastructure to monitor RTT; it focuses on reducing the flow entries and the number of probe packets. In [7], a measurement scheme that can cover all links in both directions with minimizing flow entries on switches is presented. To reduce unnecessary load on the data plane in incurred by probe packets and unnecessary load on the control plane in the OFC and OFSs, we proposed a framework of monitoring in OpenFlow-based networks to locate high-loss links [8].

In this paper, based on those existing works, we present a method to estimate the packet delay variance from the arrival time interval of packets and locate delay fluctuation-prone links. Packet delay variance on a link or on a end-to-end path is one of important metrics of the network performance, which is sometimes related with jitter [9] and sometimes defined by slightly different ways. Here, we focus on packet delay variance on a directional link or a directional segment that represents the variance of packet delay time between two ports

(e.g., upper and lower ports of a link) over a series of packets traversing the link. Since delay fluctuation-prone links, i.e., links with a high packet delay variance, are likely congested or physically unstable, it is of importance to monitor and locate them in network performance management.

The next section overviews the system model and route scheme design. Estimating packet delay variance and locating delay fluctuation-prone links are presented in Sections III and IV, respectively. The simulation results are provided in Section V. The last section is discussion and concluding remarks.

II. System Model

The proposed system model is based on and similar to that we previously proposed to monitor and locate high-loss links using multicast probing on OpenFlow networks [8]. It assumes the OpenFlow-based networks comprising OpenFlow controller (OFC) and OpenFlow switches (OFS), i.e., the target networks include per-flow flexible routing/multicasting and per-flow monitoring of network statistics in a centralized manner. Note that, while the previous system in [8] performs within the minimal standard functions of OpenFlow, the present system requires an extension of flow entry and Flow-Stats Reply message to monitor the statistics of packet arrival time intervals on a specific flow, see Sections III and VI. The process begins when the measurement host (MH) sends a measurement request to the OFC, Fig. 1. Then, the OFC obtains network topology, calculates probe packet routes, and installs them to OFSs.

Following that, a series of probe packets is launched by a single MH. The switch port connected to the MH is the root port. Here, each probe packet (or a copy) passes through each link once and only once (in each direction of a full-duplex link separately) and is discarded at a leaf port on the last OFS along the measurement path. The arrival time intervals of those probe packets at each input port of OFSs are monitored and their statistics are recorded at each OFS. Note that we do not measure a delay time of each probe packet between two passing ports; instead we measure an arrival time interval of two adjacent probe packets. Then, the OFC collects the arrival time interval statistics from OFSs and estimates the **packet delay variance** on a **link** (or a **segment**, i.e., a sequence of links) between two switch ports based on the collected statistics, until locating all delay fluctuation-prone links, i.e., links with a high packet delay variance; the details are presented in the following sections.

To reduce the number of accesses (queries) to OFSs required to locate all delay fluctuation-prone links and reduce unnecessary load on the control plane in the OFC and OFSs, a sequential retrieval order of the necessary statistics in flow entries on the required OFSs is important. Our scheme dynamically determines an appropriate order of collecting statistics from OFSs by narrowing the search space (i.e., candidates of high packet delay links and segments)

In our previous work [8], to design an appropriate measurement paths to cover all links in both up and down directions on

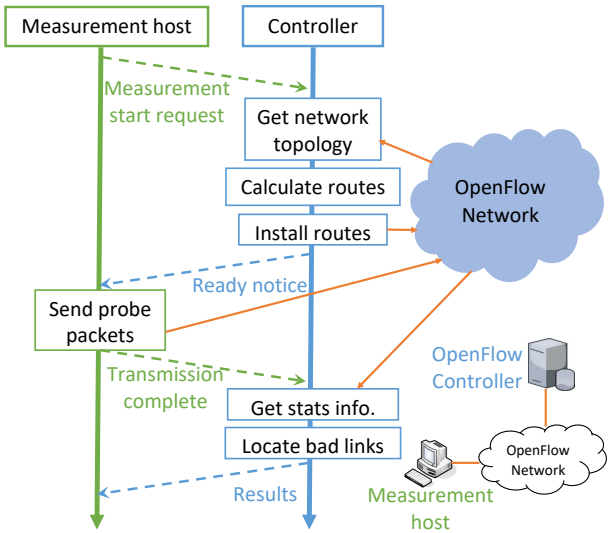


Fig. 1: Measurement process to locate bad links [8]

an OpenFlow network, we proposed two routing schemes that can be computed in a computationally lightweight manner.

An example routing scheme is shown in Fig. 2. Here, the root port is a switch port connected to the MH, and the leaf port is a switch port that discards the probe packet. A route of the probe packets (i.e., the measurement flow) from the root port to leaf port is referred to as a "terminal path". The number of links between the root port to the leaf port is considered the path length. The routing scheme designs an appropriate multicast measurement path tree with a number of terminal paths.

In this paper, we use the base-line routing scheme (named Model 1 in [8]) that minimizes the length of each terminal path. Note that, there are different possible multicast measurement routes (including a single unicursal unicast measurement route over all links as an extreme case); however, measurement robustness and accuracy are strongly affected by the path length of each terminal path. For example, when a large number of probe packets is lost on a given link, all succeeding downstream links on that terminal path may not be monitored accurately due to a reduced number of probe packets passing through those links.

As shown in Fig. 2, the proposed routing design involves three steps as follows.

- Generate the shortest path tree in the downward direction from the root (blue dashed lines in Fig. 2).
- Complement unused links not on the shortest path tree (green dotted lines in Fig. 2).
- Add return links in the upward direction bound for the root (red lines in Fig. 2).

A. Generate shortest path tree (Step 1)

We use Dijkstra's shortest path algorithm to build a path tree from the MH, on which probe packets flow and reach all OFSs.

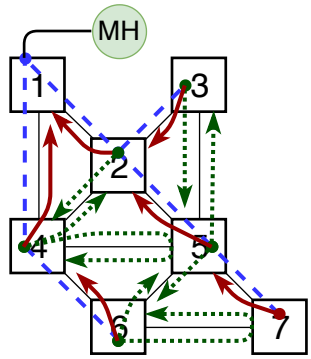


Fig. 2: Route scheme design [8]

B. Complement unused links (Step 2)

We build routes covering the links that are not included in the shortest path tree in Step 1 by extending the tree. Here, there are two cases. If two OFSs connected by an unused link, e.g., OFSs 2 and 4 in Fig. 2, are positioned at equal distance from the root (Case 1), the probe packets are routed to each other. If those two OFSs, e.g., OSFs 4 and 5 in Fig. 2, are at different distances from the root (Case 2), the probe packets are routed from the OFS nearer the root to the other one and back. In both cases, the flow (i.e., the terminal path) stops and the probe packets are discarded here.

C. Add return links (Step 3)

Each OFS on the shortest path tree in Step 1 forwards the probe packets back to its parent OFS to cover the upward link and the flow stops, which minimizes the length of each terminal path traversing the upward direction.

III. Estimating Packet Delay Variance from Arrival Intervals

A direct and simple way to estimate packet delay variance is measuring packet delay times of samples (i.e., probe packets in our case) and computing their unbiased variance. However, the delay time measurement requires matching and subtracting arrival times of a same packet monitored at two different points (i.e., OFSs in our case). Thus, the arrival time information of each packet should be moved from a place to another; inducing a more load on the control and/or data planes. We discuss this issue in Sec. VI.

Therefore, in our method proposed in this paper, each OFS monitors the arrival time intervals of two adjacent packets (per input port) in a series of probe packets, and records their statistics, which can be performed within each OFS independently and does not require a movement of per-packet information between OFSs or the OFC. Note that possible holes in a series of probe packets due to packet losses are considered and removed in the process of monitoring the arrival time intervals. After the above measurement of probe packets is finished, the OFC collects the arrival time interval statistics at each input port of OFSs and estimates the packet delay variance between two ports using the collected statistics

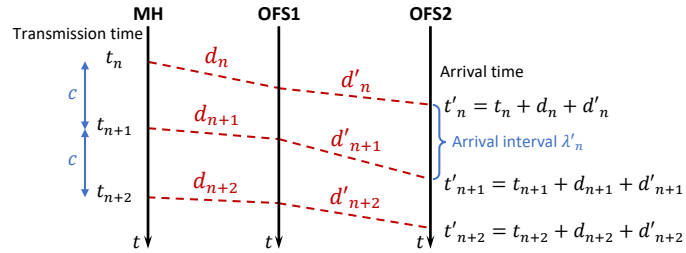


Fig. 3: Sequence diagram

in an appropriate retrieval order of necessary statistics on OFSs.

The sequence diagram of probe packets is shown in Fig. 3. Assuming that t_n, t_{n+1}, t_{n+2} are the transmission times of probe packets at the MH, and t'_n, t'_{n+1}, t'_{n+2} are the arrival times at OFS2, respectively. If the transmission time interval is a fixed value c, we have

$$t_{n+1} = t_n + c \tag{1}$$

The delay time of probe packets between the MH and the OFS 1 are d_n, d_{n+1}, d_{n+2} and the delay time between OFS1 and OFS2 are d'_n, d'_{n+1}, d'_{n+2}, respectively. The arrival times at OFS2 are expressed as follows

$$t'_n = t_n + d_n + d'_n \tag{2}$$

$$t'_{n+1} = t_{n+1} + d_{n+1} + d'_{n+1} \tag{3}$$

The arrival time interval of packets at OFS2, λ'_n, is

$$\begin{aligned} \lambda'_n &= t'_{n+1} - t'_n \\ &= (t_{n+1} + d_{n+1} + d'_{n+1}) - (t_n + d_n + d'_n) \\ &= c + (d_{n+1} + d'_{n+1}) - (d_n + d'_n) \end{aligned} \tag{4}$$

The following preconditions are defined to estimate the delay variance from the arrival time intervals.

- d_n and d_{n+1} are independent and identically distributed.
- d'_n and d'_{n+1} are independent and identically distributed.
- d and d' are independent.

Therefore, the variance of the arrival interval is expressed as follows

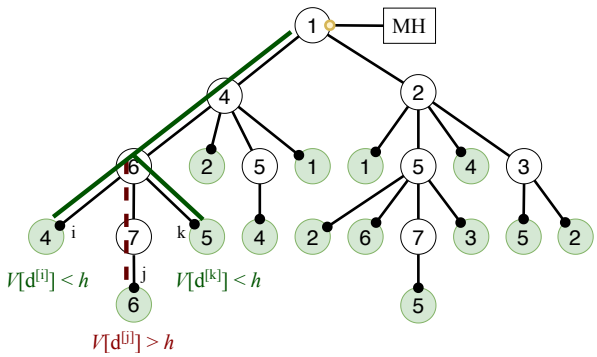
$$\begin{aligned} V[\lambda'] &= V[c + (d_{n+1} + d'_{n+1}) - (d_n + d'_n)] \\ &\cong 2(V[d + d']) \end{aligned} \tag{5}$$

Equation (5) can be rewritten as

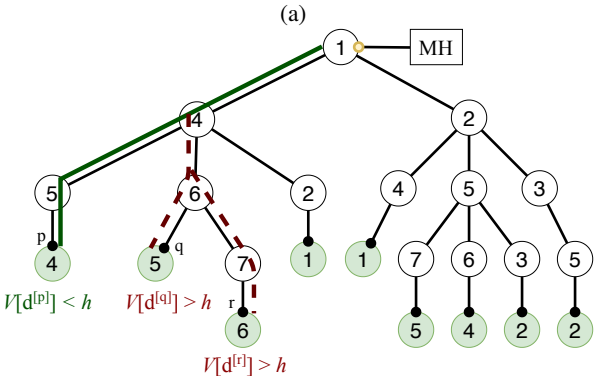
$$V[d + d'] \cong \frac{V[\lambda']}{2} \tag{6}$$

Here, by setting the packet transmission interval constant, the delay variance can be estimated with only the arrival interval variance. In addition, the delay variance between the MH and OFS2 $V[d + d']$ is the sum of the delay variance of MH-OFS1, $V[d]$, and the delay variance of OFS1-OFS2, $V[d']$,

$$V[d + d'] \cong V[d] + V[d'] \tag{7}$$



(a)



(b)

Fig. 4: Multicast route tree example

or the delay variance of OFS1-OFS2 is

$$\begin{aligned} V[d'] &\cong V[d+d'] - V[d] \\ &\cong \frac{V[\lambda'] - V[\lambda]}{2} \end{aligned} \tag{8}$$

where λ represents the arrival time interval of packets at OFS1. This means that, in general, the delay variance of a specific link or segment between two OFSs can be estimated from the arrival interval variances of those OFSs. Note that the arrival interval variance is simply computed by

$$V[\lambda] = E[\lambda^2] - (E[\lambda])^2 \tag{9}$$

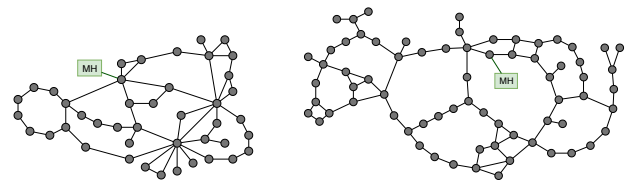
If the $(n+1)$-th prove packet is lost somewhere and an OFS receives the n-th and $(n+2)$-th packets but not receive the $(n+1)$-th packet, that OFS discards the arrival time interval between n-th and $(n+2)$-th packets and does not count it in the statistics. To detect such a hole by lost packets at OFS, the MH embeds a sequence number into ID field of IP header of each probe packet.

IV. Locating Delay Fluctuation-Prone Links

The delay fluctuation-prone link identification method is based on the idea that if the delay variance at a special OFS is less than a threshold value h, the segment from the MH to it does not include any delay fluctuation-prone link. Otherwise, there may be one or more delay fluctuation-prone links in this segment. Here, h is a design parameter that represents the target delay variance quality of links to maintain, which depends on the target applications. The detailed algorithm is as follows. Figures 4a and 4b show examples in which $d^{[x]}$ represents the delay time of probe packets between the MH and the port x on some OFSs.

First, the OFC queries OFSs that have leaf ports to collect the information on arrival time intervals at those ports and estimates the delay variance of each terminal path using the information at the leaf ports by (6). If the delay variances of all terminal paths are less than h, that means the network do not include any delay fluctuation-prone link. If the delay variance of a terminal path exceeds h, this terminal path is likely to include one or more delay fluctuation-prone links. Then, by considering the correlation among terminal paths in terms of delay variance, OFC can narrow the search scope, i.e., the expected locations of delay fluctuation-prone links. For example, if a terminal path is delay fluctuation-prone and no other terminal paths are delay fluctuation-prone, the delay fluctuation-prone links are located within a segment between the leaf port and the nearest multicast parent port on that delay fluctuation-prone terminal path. The dashed line in Fig. 4a shows an example of this case. Here, to locate delay fluctuation-prone links, the ports along this segment should be queried by OFC in a binary-search manner. Eventually, the delay variance of each delay fluctuation-prone link is measured based on the difference between the delay variance at the link's upper and lower ports by (8).

If there are multiple terminal paths whose delay variance values exceed threshold h, the port that is most commonly shared by those paths and nearest to the root among them is queried first to collect the delay variance of probe packets at that port. By considering the sub-trees separated by that port, the same procedure can be performed on each sub-tree recursively. The dashed lines in Fig. 4b illustrate this case. Here, the next queried port is the OFS4's received port.



(a) Topology 1 (b) Topology 2

Fig. 5: Simulation network topology

TABLE I: Network topology parameters

	Topology 1	Topology 2
Number of OFSs	43	70
Number of links (in both directions)	112	170
Number of terminal paths	29	36
Average length of terminal paths	5.7	9.5

V. Simulation Results

We evaluate the search performance of our proposal by numerical simulation on two real-world network topologies

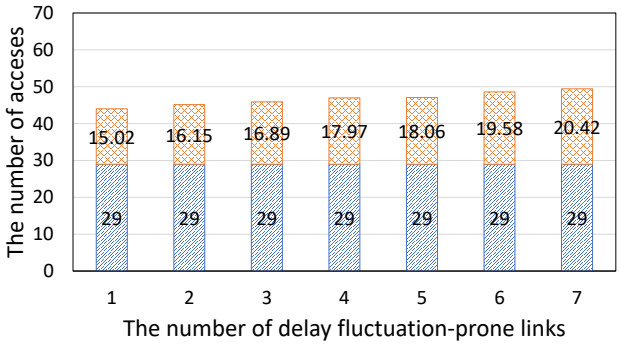


Fig. 6: Number of required accesses to locate delay fluctuation-prone links on Topology 1

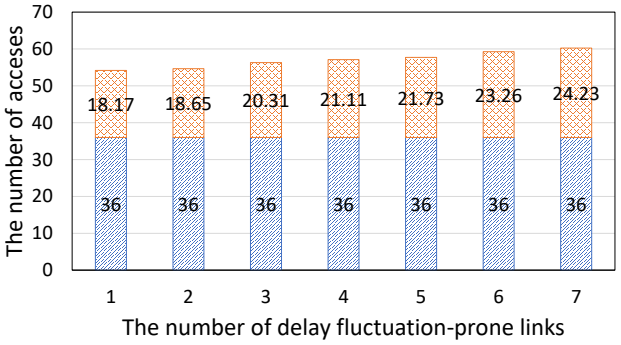


Fig. 7: Number of required accesses to locate delay fluctuation-prone links on Topology 2

in a topology database [10]. They are illustrated in Fig. 5 with parameters in Table I. Topology 1 is used to simulate a medium-scale network environment. It is based on the RENATER, the national research and education network in France. Topology 2 is a large-scale topology based on the Columbus network in Latin America.

In the simulation, packet transmission delay is set on each link as follows. A baseline static delay time of a link is set to a randomly selected fixed value from a range of $[10.0, 20.0]$ (ms). An additional dynamic delay time of a link is a random variable with a exponential distribution that is independent of each other. The mean value (the expectation) of this random variable of dynamic delay is selected from a range of

- $[5.0, 10.0]$ (ms) for each of a specific number of high delay variance links,
- $[2.0, 4.0]$ (ms) for each of 10% moderate delay variance links,
- $[0.002, 1.0]$ (ms) for each of other little delay variance links.

Fig. 6 and Fig. 7 illustrate the simulation results on Topology 1 and Topology 2, respectively. The number of delay fluctuation-prone links is considered from 1 to 7. In each column, the lower part is the number of accesses of the first request process, on which the OFC accesses to all leaf ports. It equals the number of terminal paths. The higher part

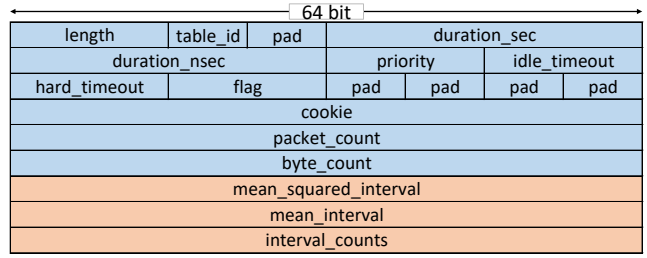


Fig. 8: Extension of FlowStats Reply message

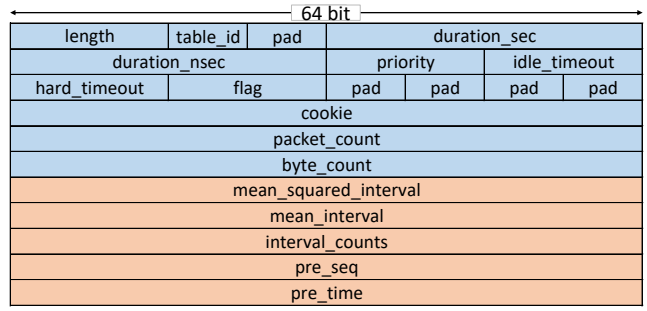


Fig. 9: FlowStats in FlowEntry in Lagopus OFS

presents the more required accesses to locate delay fluctuation-prone links. In this phase, by considering calculated results and the relationship of terminal paths, we can reduce the number of requested ports significantly. The results show that although the number of required accesses tends to increase when the number of delay fluctuation-prone links increases, all problem links can be detected with the number of accesses less than a half of the number of total links. It demonstrates the effectiveness of the proposed method.

VI. Discussion and Concluding Remarks

Based on our previously proposed framework for monitoring and locating links with a high packet loss rate, in this paper, we have proposed a scheme to actively monitor and locate delay fluctuation-prone links, i.e., links with a high packet delay variance, in an OpenFlow network. A notable feature is that packet delay variance is estimated based on monitoring arrival time intervals of probe packets without directly measuring packet delay time over a link or a segment. However, our proposal and its evaluation are still preliminary and several questions may arise as follows.

- Implementation feasibility and efficiency
- Assumption on independency of packet delays in time and space
- Arrival time interval monitoring versus time-stamped probing packets
- Accuracy and speed in locating delay fluctuation-prone links

On implementation feasibility and efficiency, since our scheme needs a new flow entry and Flowstats function to monitor the statistics of packet arrival time intervals on a specific flow, we design an extension of OpenFlow both on

switch and controller as illustrated in Fig. 8 and Fig. 9, where $mean\_squared\_interval$ is the average of squared arrival time intervals, $mean\_interval$ is the average of arrival time intervals, $interval\_counts$ is the number of counted intervals, pre_seq is a sequence number of the last arrived packet, which is used to detect packet losses, pre_time is a arrival time of the last arrived packet, which is used to compute an arrival time interval. The additional field is small, and the additional computation is light-weight. This is because the variance of arrival time interval at an input port is estimated from two sample means for $E[\lambda^2]$ and $E[\lambda]$, and they can be computed in an incremental manner by a current sample mean and a new λ, i.e., we do not record a sequence of past λ values. The proposed extension was already implemented on Lagopus software switch [11] and Ryu controller [12] for OpenFlow version 1.3, and tested on Mininet emulator [13].

The assumption (in Sec. III) on the independency of packet delays in time and space is not strictly held in general, and the degree of dependency matters. On the dependency between d_n and d_{n+1}, i.e., delay times of two adjacent probe packets, it will happen more or less in congestion by a long lasting queue along a path (a segment). At least, we should take care that any succeeding packet does not overtake its preceding packet. A large initial transmission time interval at the MH mitigates the dependency, although it will prolong the measurement duration and be harmful for timeliness. In addition, if a number of lost packets happen, the accuracy of monitoring on succeeding links will be harmed due to a reduced number of probe packets. On the dependency between d_n and d'_n, i.e., delay times on two adjacent links or segments along a terminal path, they are correlated positively in some cases and negatively in some other cases. However, the degree is expected not very high if a number of heterogeneous application flows coexists in the network. In addition, this type of dependency is affected by the measurement route; see the last item. Therefore, in order to clarify the applicable conditions of our proposal, we should investigate the degree of delay time dependency in real networks and the impact of real dependency on accuracy of the delay time variance estimated by the statistical information on arrival time intervals.

To avoid direct measurement of packet delay time that requires matching and subtracting arrival times of a same packet monitored at two different OFSs, we introduce the arrival time interval monitoring. A use of time-stamped probing packets in direct measurement of packet delay time can be considered and popular in some systems, in which the sending time is embedded in each probe packet at an MH and that time is referred at each OFS to subtract it from the arrival time. The time-stamped probing packets can be used without a strict clock synchronization among OFSs. However, the time-stamp requires at least a 32-bit additional field in each packet. It cannot be placed in IP or TCP header so it must be in application payload, which is not efficiently handled in each OFS and may sometimes be difficult. In our scheme, we only use a sequential number (id) of probe packet embedded in ID field of IP packet, in order to detect lost ones in a series of probe packets.

Finally, although we use a baseline routing scheme and a simple collection order scheme in this paper, the accuracy and speed are expected to be improved by developing a more suitable routing scheme of probe packets including MH placement in conjunction with a more efficient retrieval order of necessary statistics on OFSs collected by the OFC. One promising direction is the use of information on past measurement results. Based on such "prediction", we may build an appropriate measurement route that is likely to avoid a high dependency between delays on adjacent links or segments and also likely to reduce the number of accesses to OFSs required to locate all delay fluctuation-prone links.

Acknowledgment

The research results have been achieved by the "Resilient Edge Cloud Designed Network (19304)," NICT, and by JSPS KAKENHI JP16K00130, Japan. We thank Mr. Suguru Goto for assistance.

References

[1] S. Jain, A. Kumar, et al., "B4: Experience with a globally-deployed software defined WAN," Proc. ACM SIGCOMM'13, pp. 3–14, 2013.
[2] C-Y. Hong, S. Kandula, et al., "Achieving high utilization with software-driven WAN," Proc. ACM SIGCOMM'13, pp. 15–26, 2013.
[3] S. R. Chowdhury, M. F. Bari, R. Ahmed, and R. Boutaba, "PayLess: A low cost network monitoring framework for software defined networks," Proc. IEEE Network Operations and Management Symposium (NOMS), pp. 1–9, 2014.
[4] A. Tootoonchian, M. Ghobadi, and Y. Ganjali, "OpenTM: Traffic matrix estimator for OpenFlow networks," Passive and Active Measurement, Lecture Notes in Computer Science, vol. 6032, pp. 201–210, 2010.
[5] C. Yu, C. Lumezanu, Y. Zhang, V. Singh, G. Jiang, and H.V. Madhyastha, "FlowSense: Monitoring network utilization with zero measurement cost," Passive and Active Measurement, Lecture Notes in Computer Science, vol. 7799, pp. 31–41, 2013.
[6] A. Atary and A. Bremler-Barr, "Efficient round-trip time monitoring in OpenFlow networks," Proc. IEEE INFOCOM, pp. 1-9, 2016.
[7] M. Shibuya, A. Tachibana, and T. Hasegawa, "Efficient active measurement for monitoring link-by-link performance in OpenFlow networks," IEICE Trans. Commun., vol. E99B, no. 5, pp. 1032–1040, 2016.
[8] N. M. Tri and M. Tsuru, "Locating deteriorated links by network-assisted multicast proving on OpenFlow networks," in IEEE Symposium on Computers and Communications (ISCC), 2019, unpublished, to appear in July 2019.
[9] C. Demichelis and P. Chimento, IP packet delay variation metric for IP performance metrics (IPPM), The Internet Engineering Task Force, IETF-RFC, Nov. 2002.
[10] The Internet Topology Zoo, http://www.topology-zoo.org/ accessed Jan. 20, 2019.
[11] Lagopus Software Switch, https://github.com/lagopus/lagopus, accessed Feb. 15, 2019.
[12] Ryu SDN Framework, https://osrg.github.io/ryu/, accessed Feb. 15, 2019.
[13] Mininet, http://mininet.org/, accessed Feb. 15, 2019.