Dynamic optimization of multicast active probing path to locate lossy links for OpenFlow networks

doi: http://dx.doi.org/10.1109/ICOIN48656.2020.9016438

Dynamic optimization of multicast active probing path to locate lossy links for OpenFlow networks

Suguru Goto*, Masahiro Shibata†, and Masato Tsuru†
Graduate school of Computer Science and Systems Engineering, Kyushu Institute of Technology, Fukuoka, Japan
Email: *gotos@infonet.cse.kyutech.ac.jp, †{shibata, tsuru}@cse.kyutech.ac.jp

***Abstract*—To maintain a high quality of service in managed networks, detecting and locating high loss-rate links (i.e., lossy links that are likely congested or physically unstable) in a fast and efficient manner is required. In our previous work, we proposed a centrally-managed network-assisted framework of locating lossy links on OpenFlow networks. In the framework, the OpenFlow controller builds a multicast measurement route; a measurement host launches a series of multicast probe packets traversing all full-duplex links along the measurement route; and then the controller collects statistical information (flow-stats) on the arrival of those probe packets at different input ports on selected switches and compares them to narrow down and identify the locations of high loss-rate links. The number of accesses to switches in collecting the flow-stats until locating all lossy links should be as small as possible for fast and efficient measurement. However, it strongly depends on not only the collection order of the flow-stats but also the topological locations of lossy links in the multicast measurement route; the former one was investigated in the previous work but the latter has not been well explored. Therefore, in this paper, we develop a new dynamic scheme of building the multicast measurement route and controlling the collection order of flow-stats from switches, which leverages lossy link locations obtained in the recent past measurements in a repeated-measurement setting. The results of numerical simulation on real-world large-scale network topologies suggest the effectiveness and also the issues of the proposed lossy link location scheme.**

I. Introduction

In recent years, Software Defined Network (SDN) in general and OpenFlow in particular are attracting much attention and being deployed mainly due to its centrally-managed style and dynamic per-flow routing capability to support cloud computing and service virtualization. As globally distributed datacenters and content delivery networks become popular, OpenFlow-based networking is being applied to not only datacenters but also enterprise networks and wide area networks, so called SD-WAN. The emerging edge-cloud computing for the IoT era will further accelerate the demand for reliable as well as flexible networking among a large number of geographically-distributed heterogeneous sites connected by a centrally-managed virtual network based on SDN technology. Therefore, it is demanded to promptly detect, locate, and fix or avoid highly lossy links (i.e., links with packet loss rates exceeding a threshold; which are likely congested or physically unstable). However, in such networks, a "link" between two nodes is not always physical but sometimes virtual one (e.g., tunneling); in addition, a link that currently accommodates no traffic should also be monitored for its future use. Accordingly, to maintain a high quality of service of large-scale OpenFlow networks, flexibly and timely monitoring all links is hard to implement solely by a "passive" measurement-based approach and thus it requires an "active" measurement-based approach by sending probe packets to actively and periodically measure the performance and/or status of all full-duplex links in both directions.

In our previous study, by leveraging the per-flow routing/monitoring capability of OpenFlow, we proposed a centrally-managed network-assisted framework to fast and efficiently monitor, detect, and locate all lossy links that can distinguish both directions of each link. In the framework, the OpenFlow controller (OFC) builds a specially designed measurement route (i.e., a multicast measurement path tree) for probe packets in response to a request from a measurement host (MH). Then a series of probe packets are launched from the MH along the route. The number of probe packets that arrived at an individual input port (i.e., the flow-stats on the arrival of those probe packets) is recorded at each OpenFlow switch (OFS) as flow-stats for the probe flow. Then OFC collects the number of arrived probes at different input ports on selected OFSes, and calculates the packet loss rate on the segment between those two input ports based on a difference of those numbers of arriving packets. By repeating this procedure (the collection of flow-stats and the calculation of link loss rate on some segment) in an appropriate order, OFC can narrow down the suspected segments with a high loss rate and finally locate all high loss-rate links. In our framework, the number of accesses to switches should be as small as possible. However, it strongly depends on not only the collection order of flow-stats but also the topological locations of lossy links in the measurement path tree. Hence, a dynamic optimization of multicast measurement path tree is essential to improve our measurement framework.

II. Related work

Network 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 either by active measurements that incur additional traffic on data plane, by passive measurements that incur additional load on control plane, or by both. In particular, to manage OpenFlow networks, real-time, light-weight, and precise network status monitoring schemes are strongly demanded. The passive approach is used to monitor link traffic status by querying and collecting the statistical information (e.g., flow-stats) from switches (through the OpenFlow monitoring messages or SNMP) or by using the OpenFlow operating messages themselves. In general, there is a trade-off between the measurement accuracy and the load incurred on switches and the control network. There are some challenges about this issue. In [1], the authors introduced a dynamic algorithm to balance the query frequency and the measurement accuracy. The impact of queried switch selection on the accuracy is discussed in [2]. With no additional load, [3] 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/estimate the packet loss, delay, the round-trip-time (RTT), and so on. 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 [4] proposed a infrastructure to monitor RTT; it focuses on reducing the flow entries and the number of probe packets. In [5], a measurement scheme that can cover all links in both directions with minimizing flow entries on switches is presented.

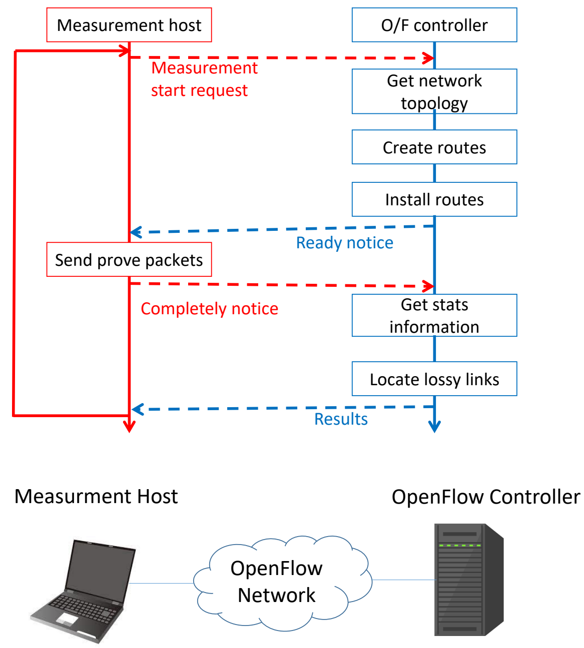


Fig. 1. Measurement system flow

III. Measurement system framework

Basically, as an active measurement approach, the packet loss rate of a link or a segment (a series of links) is measured by comparing the numbers of probe packets observed at the both ends of the target. Packet loss rate (**PLR**) from port i of some OpenFlow switch (OFS) to port j of an OFS can be expressed by the following equation (1).

$$PLR_{i,j} = 1 - \frac{r_j}{r_i} \quad (1)$$

where r_i is the number of packets arriving at port i. A threshold h is set and a link whose PLR exceeds h is defined as *lossy link*.

Figure 1 illustrates the measurement system framework. When a measurement host (MH) sends a measurement start request to the OpenFlow controller (OFC), the OFC grasps the network topology, determines the measurement path tree of the probe packets, and installs the path routes to the flow entry of each OFS. Then the MH launches a series of multicast probe packets to the network. To minimize the load incurred by probe packets on the data plane, each probe packet is multicasted along the measurement path tree so as to cover the whole full-duplex network, and each copy of the packet is discarded at some port so as not to pass through the same link (in the same direction) more than once [6]. The first port directly connected to MH is called the **root port** that is the starting point of the path tree. The ports at which the probe packets are discarded is called the **leaf port** that is each end of the path tree. A route from the root port to a leaf port is called a **terminal path**.

After all the probe packets were sent, the OFC starts obtaining the flow-stats information (i.e., the number of packets arrived at an input port in our case) from some OFSes. To locate lossy links fast without unnecessary load on the control plane in the OFC and OFSes, the order of flow-stats collection on selected OFSes is dynamically determined using a simple Binary network tomography with binary search to reduce the number of accesses to OFSes to retrieve the flow-stats until locating all lossy links and verifying that no other link is lossy [6]. To further reduce the necessary number of accesses to OFSes, we also developed a different location scheme with an appropriate collection order using a Bayesian-based network tomography to refine the candidates for lossy links based on the results in past measurements [7]. However, this scheme uses a per-packet correlation of lossy segments and requires an extension of the OpenFlow standard. More importantly, the necessary number of accesses to OFSes strongly depends on not only the collection order but also the topological locations of lossy links in the measurement path tree. In addition, if a link near to the MH is highly lossy, the number of probe packets after that link may be reduced to too small to guarantee a reliable loss rate measurement on succeeding downstream links along a long terminal path.

In this paper, therefore, we develop a new dynamic scheme of building the multicast measurement path tree and of controlling the collection order of flow-stats from switches to minimize the necessary number of accesses and to avoid unreliable measurement results. The idea is to assign a cost that reflects the degree of possibility of being a lossy link to each link by using lossy link information obtained in past recent measurement cycles in a repeated-measurement scenario.

IV. Proposed method

A. Overview

In a periodically-repeated measurement setting, one measurement cycle consists of the following four steps.

1) Building a multicast measurement tree by using link costs so that the links that were lossy in the recent past measurements are likely placed at the leaves of the terminal paths.
2) Probing and monitoring the measurement packets along the measurement tree.
3) Narrowing down lossy segments and locating lossy links by considering link costs so that the links that were lossy in the recent past measurements are checked preferentially.
4) Updating the link costs so that the links that are detected as lossy in the current measurement have higher link costs.

The simple idea behind this approach is that the links with higher loss rates in recent past are more likely to be lossy than those with lower loss rates. The link cost here is defined to indicate the degree of loss-proneness of link. However, loss-prone links are not always lossy. Thus, the adaptability to such deviations is required and will be discussed in simulation evaluation in Sections 5 and 6 later.

B. Assigning a cost to each link

The cost of each link is updated by the following Exponential Moving Average with smoothing factor α after each measurement cycle.

$$\begin{aligned} w_n &= \alpha w_{n-1} + (1-\alpha) w_{now} \\ w_{now} &= \max(1, \lfloor \text{PLR} \times \text{MaxCost} \rfloor) \end{aligned} \quad (2)$$

where w_n represents the cost of a link in the n-th cycle; if **PLR** of a link in the cycle is unavailable (i.e., not measured), $w_{now} = 1$. **MaxCost** is the upper bound to control the granularity of cost. The initial value w_1 is set to 1.

As explained in the following subsections, based on the assigned cost of each link, a shortest cost tree (rather than shortest hop tree) is used in building the measurement path tree, and eventually the previously lossy links are likely located downward on terminal paths. Then an weighted (i.e., cost-aware) binary search is used in controlling the collection order of flow-stats from OFSes.

C. Building the multicast measurement path tree

In the previous scheme, the measurement path tree was built based on the number of hops. In the proposed scheme, a cost is assigned to each link, and the route is built using the sum of cost values instead of the number of hops. Figure 2 compares the paths build in the previous scheme and the proposed scheme assuming that the cost of one link ($2 \rightarrow 5$) is 100 (1 for the other links), and indicates how the position of this link in the tree is changed by the proposed method. The measurement path tree is built in the following three steps.

- Generate the shortest cost path tree
  1) The shortest cost path tree (by considering single direction only) from the root port is generated using the standard Dijkstra's method.
- Complement unused links
  1) Links included in neither upward nor downward of the above initial shortest path tree, for example link of between Switches 2 and 4, are called "unused" links.
  2) For each unused link, there are two end ports, say port a on OFS A and port b on OFS B. We compare the total cost of the route from the root port to a via b with that of the route from the root to b via a, and select one of the two route with the lower cost, and add the route on the unused link to extend the path, which covers one direction on the unused link. The OFS that forwards the selected route in step 2 is called "the primary-extension OFS". For example, if the root to a via b has a lower cost and thus is selected, OFS "B" is the primary-extension OFS for this unused link.
  3) To cover the remaining (opposite) direction from the non primary-extension OFS to the primary-extension OFS of the unused link, we compare the total cost of the further extension of the route that is extended in step 2 with that of the extension of the route from the root to the non primary-extension OFS, select one of two routes with the lower cost, and add the route.
- Add return links
  In this step, the remaining upward links are added to the multicast path tree.
  1) A switch that extends a route on an unused link to the other switch in complementing the unused link (e.g., switch 2): it extends (returns) the route on the shortest cost path to that switch to its parent switch.
  2) A switch that is at a leaf of the shortest cost path (e.g., switch 7): it extends (returns) the route on the shortest cost path to that switch to its parent switch.
  3) A switch that does not match the above conditions: it selects one of its child switches with the lowest cost from the leaf switch to it, and extends the route from the selected child switch to the parent switch.

The final measurement path tree is built by the above algorithm, and probe packets are sent according to the tree.

As shown in Fig. 2, while in the previous method, a high loss link ($2 \rightarrow 5$) is placed at upstream of the measurement path and may affect the measurement reliability of the downstream links, in the proposed method, that link is placed at the leaf.

D. Controlling the collection order of flow-stats from switches

After sending the prove packets, the Openflow controller accesses some Openflow switches in the appropriate order and collects flow-stats on some input ports. The algorithm to control the collection order is as follows.

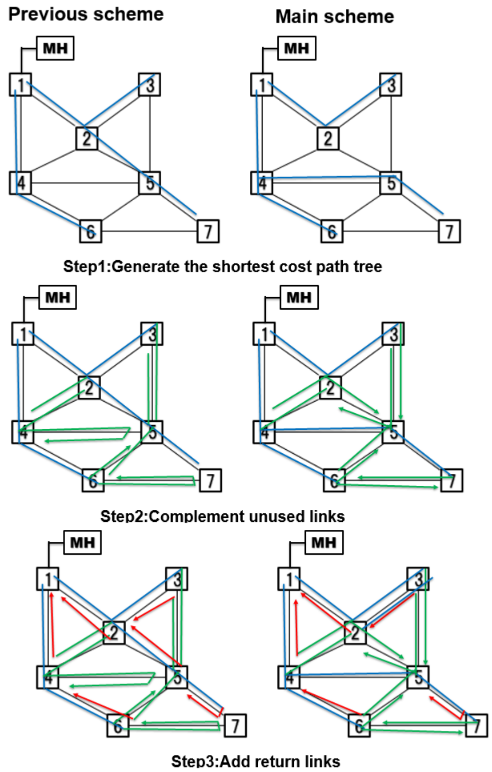


Fig. 2. Comparison of the path between the conventional method (left) and the proposed method (right)

First, the OFC collects flow-stats of the root port and all the leaf ports, and calculates the PLR for each terminal path. If the PLR of a terminal path does not exceed the threshold h, there is no lossy link in the terminal path. On the other hand, if the PLR exceeds threshold h, there is a possibility that lossy links exist. In addition, the search range can be further narrowed by using the relationship with other terminal paths. If several terminal paths share some common links and the PLR of at least one terminal path does not exceed h, those common links can be removed from a search candidate. An example is shown in Fig. 3. The numbers in circle in Fig. 3 are port numbers (not switch numbers), and it can be determined that the link between ports 1 and 2 is not a lossy link. After that, the next port to be accessed to collect the flow-stats is a port that is included in multiple terminal paths because such ports are likely to have an impact on the PLR of multiple terminal paths. When the flow stats of ports other than the root and leaves is collected, a subtree is generated, and PLR calculation is performed again. By doing this procedure recursively, it is possible to finally determine the status of all links. When there is no common port, for example, when considering the subtree in the dotted line in Fig. 3 (the number next to the link indicates the cost), the next port to be accessed to collect the flow-stats is a port that halves the sum of the cost of the entire segment. This is a weighted binary search. In this case, the flow-stats of port 6 is collected.

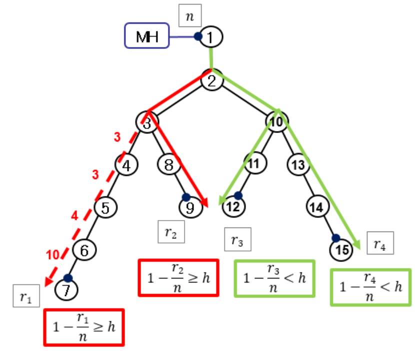


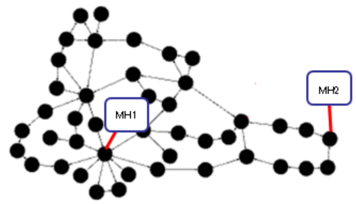
Fig. 3. Flow-stats collection example

V. Simulation

The simulation uses two types of real-world large-scale network topologies illustrated in Fig. 4, which are provided by a topology database [8]. One is topology 1 with 43 switches and 112 links, and the other is topology 2 with 70 switches and 170 links. The following values are set as parameters. In one simulation, we continuously perform 8 measurement cycles. At the beginning of each (except for the first cycle), the link costs are updated as shown in Section IV-B. For 5% of the links in the entire topology, the packet loss rate is always high and exceeds the threshold h in each cycle. For 10% of the links, the packet loss rate unlikely exceeds h; but the accumulated loss rate of those links along a terminal path may exceed h. For 5% of the links, the packet loss rate may or may not exceed the threshold for each cycle, and the number of degraded links varies between cycles. This setting is intended to reflect the deviations of lossy links in real networks.

- Number of prove packets : $n = 1000$
- Number of measurement cycles : 8
- Packet loss rate of each link
  - Lossy (5% of the all links) : 0.05˜0.1
  - Non-lossy (10% of all links) : 0.01˜0.02
  - Uncertain (medium) (5% of all links): 0.01˜0.05
  - Other almost loss-free links: 0.001
- Lossy link threshold : $h = 0.03$
- Cost parameters
  - Initial cost for each link : 1
  - Cost update coefficient: $\alpha = 0.4$
  - MaxCost: 1000.

For these two types of topologies, the simulation is performed for a total of four patterns when the measurement terminal is located at MH1 and MH2, respectively. The MH1 is located at an OFS that is connected to many other OFSes, while the MH2 is located at an OFS that is connected to few other OFSes. In each pattern, the position of the more or less lossy links is randomly changed in 20 times and the average performance values are evaluated. In one simulation consisting of 8 continuous measurement cycles, the position of links is not changed but the loss rate set to each link is randomly changed. The performance metric in each cycle is the average



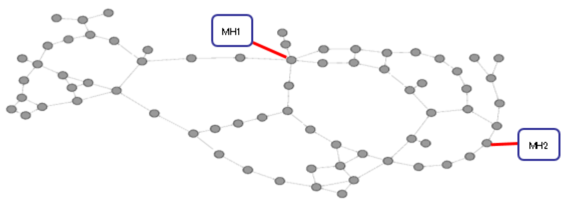


Fig. 4. Used Topology 1(top); Used Topology 2(bottom)

number of accesses for the flow-stats collection until locating all lossy links and verifying that no other link is lossy; that is averaged over 20 simulation instances. To investigate the effects of the proposed scheme, we also check the locations of lossy links on the measurement path tree (i.e., how the lossy links are located at leaves of the terminal paths).

VI. Simulation Result

Figure 5 shows the number of accesses for flow-stats collection of Topologies 1 and 2 with MH1 and MH2. The bottom bar (blue-colored) in each cycle indicates the number of accesses necessary for flow-stats collection from the root port and all the leaf ports. The second (orange) is for the number of accesses until all lossy links are located. The last (yellow) is for the number of accesses until confirming no other link is lossy (i.e., when the cycle finishes). This part is necessary if there is a segment that has a loss rate exceeding the threshold but does not include actual lossy links; that is, the cause of the high loss rate of the segment is an accumulation of mid-level loss rates of links in the segment. We can observe a typical change of performance as the measurement cycle progresses. Generally it can be seen that the number of accesses to switches for flow-stats collection in the first cycle is larger than that in each of the succeeding cycles. This result indicates the expected benefit of the proposed scheme. In the first cycle, all links have the same cost, and thus the scheme is the same as the previously proposed basic one. On the other hand, in the succeeding cycles, links on which many packet losses are measured in the past cycles have larger costs. Thus, the newly proposed scheme dynamically changes the measurement path tree so that (likely) lossy links are located downward on terminal paths. This change from the 1st cycle, with the weighted binary search, can reduce the number of accesses for collection of flow-stats. However, the reduction may be less than what we expected. From the figure, the blue part (the number of accesses necessary for flow-stats collection from the root port and all the leaf ports) is increased from the 1st cycle. This comes from an increase of the number of terminal paths compared from the 1st cycle. Suppose a link at the middle of a terminal path is detected as lossy link in the 1st cycle. Then, in the 2nd cycle, the measurement path is recomputed so that the detected lossy link is placed at the end of a terminal path, and the links in downstream from the lossy link on the original terminal path are placed on other terminal paths; this separation of the original terminal path is likely to create a new branched terminal. Therefore the number of terminal paths should not be increased in changing the measurement path tree by costs. In general, since the proposed scheme always needs to access the root port and all leaf ports first, which is major part of the number of access for flow-stats collection, it is necessary to reduce the number of terminal paths while considering that the lossy links are placed downstream.

Next, the detailed information is shown in Table 1: the average number of hops of terminal path, the number of lossy links, the number of lossy links located at leaves, the average number of hops from leaves to lossy links, and the number of switch accesses until all lossy links are located and confirming no other link for each measurement cycle. Each of those values are averaged 20 simulation instances. The topological locations of lossy links are clearly changed (at the 2nd cycle) near or at leaves of the terminal paths because the proposed scheme forces the locations of previously lossy links downward on terminal paths. For example, the 2nd cycle of Topology 1 with MH1, the number of lossy links located at leaves is 4.85 in 7.1 of all loosy links; and the average number of hops from leaves to lossy links is 0.78 (less than one). The reduction of the number of accesses to switches is about 3 from 52.55 in the 1st cycle to 49.65 in the 2nd cycle. As cycles proceed, those values are changed more or less but stable. At least, the number of accesses in each succeeding cycle is less than that in the 1st cycle.

Finally, we check the difference depending on the location of the measurement host (MH) from Fig. 5 and Table 1. There was no significant difference in the number of accesses between the two types of location of MH. However, if there are few adjacent switches of the root switch and the link near to the root switch is a lossy link, it is difficult to force the location of the lossy link in downstream of a terminal path and the link may be located in upstream regardless of the scheme of building measurement path tree. In such cases, the links in downstream of the terminal path have been affected. Therefore, it should also be considered to dynamically change the location of MH depending on the location of likely lossy links in future work.

VII. Concluding Remarks

We have been studying a centrally-managed network-assisted framework of locating lossy links on OpenFlow networks. In this paper, on the top of the framework, we focus

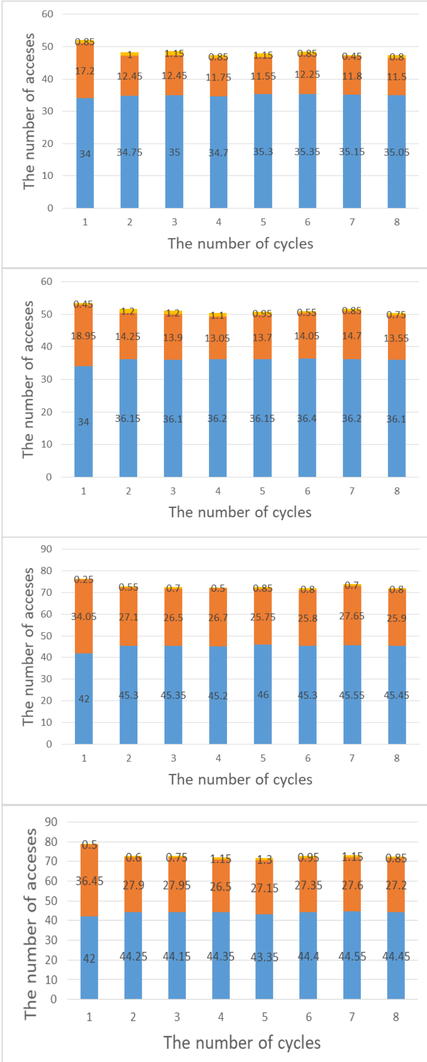


Fig. 5. Results of Topology 1 with MH1 (top) and with MH2 (2nd); Results for Topology 2 with MH1 (3rd) and with MH2 (bottom)

on a dynamic optimization of multicast measurement path tree by leveraging the past measurement results, which has not been well explored in the previous work. Our contribution in this paper is a reduction of the number of accesses to switches for flow-stats collection by introducing the following schemes.

- A scheme to dynamically assign an appropriate cost to each link is proposed; the cost represents some degree of how the link was lossy in the recent past.
- A scheme to build the multicast measurement path tree is proposed. The scheme forces the locations of the links that were lossy in the recent past measurement cycles to downstream of terminal paths.
- A scheme to control the collection order of flow-stats from switches is proposed to efficiently narrow down and identify the locations of all lossy links (the links with packet loss rates exceeding a threshold). The scheme uses a weighted binary search.

TABLE I
POSITION OF THE LOSSY LINK IN THE TERMINATION PATH

	Number of cycles	1	2	3	4	5	6	7	8
Topology 1									
MH1	Average length (hops) of paths	4.73	4.88	4.86	4.89	4.84	4.84	4.89	4.89
	Number of lossy links	7.20	6.95	6.80	7.10	6.75	7.20	6.80	6.80
	Number of lossy links located at leaves	2.25	4.50	4.65	4.75	4.60	4.65	4.85	4.75
	Average number of hops from leaves to lossy links	2.16	0.76	0.67	0.80	0.66	0.76	0.64	0.71
	Number of accesses	52.05	48.20	48.60	47.30	48.00	48.45	47.40	47.35
MH2	Average length (hops) of paths	9.03	9.57	9.57	9.70	9.66	9.44	9.50	9.64
	Number of lossy links	7.50	7.50	7.15	7.25	7.50	7.65	7.45	7.35
	Number of lossy links located at leaves	2.40	5.35	5.40	5.20	5.60	5.20	5.50	5.45
	Average number of hops from leaves to lossy links	3.25	0.72	1.31	0.49	1.07	1.51	0.76	0.74
	Number of accesses	53.40	51.60	51.20	50.35	50.80	51.00	51.75	50.40
Topology2									
MH1	Average length (hops) of paths	8.68	9.68	9.90	9.88	9.89	9.88	9.85	9.83
	Number of lossy links	11.80	12.15	11.95	12.2	12.65	11.75	12.05	11.85
	Number of lossy links located at leaves	2.85	7.40	7.50	7.40	7.55	7.15	6.80	6.95
	Average number of hops from leaves to lossy links	3.57	1.94	1.47	1.38	1.31	1.21	1.13	1.12
	Number of accesses	76.30	72.95	72.55	72.40	72.60	71.90	73.90	72.15
MH2	Average length (hops) of paths	12.00	13.33	13.43	13.09	13.50	13.65	13.31	13.23
	Number of lossy links	11.75	11.60	11.05	11.65	11.45	11.60	11.55	11.85
	Number of lossy links located at leaves	2.70	6.80	6.75	6.85	6.90	6.80	6.90	7.00
	Average number of hops from leaves to lossy links	5.91	2.04	2.00	1.59	1.74	1.01	1.54	1.53
	Number of accesses	78.95	72.75	72.85	72.00	71.80	72.70	73.30	72.50

The research results have been achieved by the Resilient Edge Cloud Designed Network (19304), NICT, and by JSPS KAKENHI JP17K00135, Japan.

REFERENCES

[1] 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), 2014.
[2] A. Tootoonchian, M. Ghobadi, and Y. Ganjali, “OpenTM: Traffic matrix estimator for OpenFlow networks,” Passive and Active Measurement, LNCS vol.6032, 2010.
[3] 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, LNCS vol.7799, 2013.
[4] A. Atary and A. Bremler-Barr, “Efficient round-trip time monitoring in OpenFlow networks,” Proc. IEEE INFOCOM, pp.1–9, 2016.
[5] M. Shibuya, A. Tachibana, and T. Hasegawa, “Efficient active measurement for monitoring link-by-link performance in OpenFlow networks,” IEICE Trans. Commun., E99B(5):1032–1040, 2016.
[6] N. M. Tri and M. Tsuru, “Locating deteriorated links by network-assisted multicast proving on OpenFlow networks,” Proc. IEEE ISCC2019, 2019.
[7] T. Nakamura, M. Shibata, M. Tsuru, “On Retrieval Order of Statistics Information from OpenFlow Switches to Locate Lossy Links by Network Tomographic Refinement,” Proc. INCoS 2019, AISC vol.1035, 2019.
[8] The Internet Topology Zoo, http://www.topology-zoo.org/ accessed Aug., 2019.