KICS The Korean Institute of Communications and Information Sciences

Available online at www.sciencedirect.com

ScienceDirect

ICT Express 10 (2024) 77–82

www.elsevier.com/locate/icte

Utility-based sampling rate allocation in in-band network telemetry for high network visibility

Kyeongtak Lee[a], Heewon Kim[b], Subin Han[b], Sangheon Pack[b,*], Hyunkyung Yoo[c], Namseok Ko[c]

[a] *Korea University, Seoul, Republic of Korea*
[b] *School of Electrical Engineering, Korea University, Seoul, Republic of Korea*
[c] *Telecommunication Media Laboratory, Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea*

Received 9 March 2023; received in revised form 15 June 2023; accepted 28 June 2023
Available online 4 July 2023

Abstract

In-band network telemetry (INT) is one of the promising technologies for achieving fine-grained and real-time visibility into the network, however, it brings non-negligible traffic overhead due to the increased packet length. Several sampling methods have been introduced to reduce the INT overhead; although, the provision of satisfactory network visibility was not well investigated. In this paper, we introduce a utility-based INT sampling rate allocation scheme, namely uINT, that aims to reduce data redundancy while improving network visibility. We conducted experiments using the software switches and demonstrated that uINT reduces redundantly collected INT data by up to 15% compared to sINT.

Keywords: Network monitoring; Programmable data plane; In-band network telemetry

1. Introduction

Network monitoring is an essential part of network management enabling network operators to observe network states and troubleshoot problems. In traditional network monitoring (e.g., simple network management protocol (SNMP) [1]), network status is periodically collected from devices, which consumes significant computing resources of devices and increases query latency. Moreover, such periodical sampling results in coarse-grained network monitoring and has limitations for getting real-time network status data. Other network monitoring techniques [2,3] have similar problems. For instance, Netflow [2] captures information about flows passing through switches; however, it requires additional workload on a switch CPU. Meanwhile, sFlow [3] requires a smaller amount of CPU usage but suffers from low accuracy. To solve these problems, in-band network telemetry (INT) has been recently introduced.

By means of INT, internal states of the switch such as queue occupancy and switch ID can be queried while visiting the switches. The queried switch states are stacked along the routing path and reported to the sink at the end of the data plane pipeline. Such per-packet telemetry capabilities enable us to collect fine-grained and real-time network information for building network-wide traffic views. However, the microscopic view of INT comes with the cost of data plane overhead (e.g., extra bandwidth consumption [4] and decreased packet processing speed due to the additional telemetry instruction [5]). Specifically, an INT metadata header requires at least 12 bytes and takes up the space where the packet payload can be loaded [6]. Moreover, if the maximum transmission unit (MTU) is limited, a lesser payload can be carried owing to the enlarged INT header.

To address the issue of INT overhead, researchers have proposed various techniques. These include (1) selectively inserting INT fields in a certain proportion of the passing packets [5,7,8], (2) compressing telemetry data to reduce the length of INT fields [9], and (3) allowing switch devices to locally decide whether to attach INT data [10,11]. One particularly effective approach is the sampling-based INT, i.e., selectively inserting INT fields in packets based on a sampling rate. Network data collected from per-packet INT have

* Corresponding author.
E-mail addresses: kyeo95@korea.ac.kr (K. Lee), harry0475@korea.ac.kr (H. Kim), subin993@korea.ac.kr (S. Han), shpack@korea.ac.kr (S. Pack), hkyoo@etri.re.kr (H. Yoo), nsko@etri.re.kr (N. Ko).

Peer review under responsibility of The Korean Institute of Communications and Information Sciences (KICS).

https://doi.org/10.1016/j.icte.2023.06.008

excessively short intervals (i.e., 0.8192 μs in a data rate of 10 Gbps), while the network does not experience rapid changes within such short time periods. Thus, the sampling-based INT efficiently reduces overhead by increasing the INT probing interval. However, existing studies about sampling-based INT suffer from the limitation of not considering a network-wide view. This can lead to the occurrence of redundant INT data depending on the paths of INT flows and the types of collected INT items. For instance, if two flows overlap on the same switch, the same types of INT data will be collected, which leads to degradation of overall data quality [12].

In this paper, we propose a utility-based INT sampling rate allocation scheme, namely uINT, that determines the sampling rate of each flow to maximize the network visibility under a given budget on the INT overhead. In uINT, we introduce a novel redundancy metric that jointly considers the collected INT items and the underlying network topology. After that, we leverage a utility maximization framework with a concave utility function on the redundancy metric subject to the INT overhead threshold. We carried out extensive simulations using a well-known SDN controller (i.e., ONOS) and software switches (i.e. bmv2). Simulation results demonstrate that uINT outperforms [7] in terms of network visibility and monitoring accuracy compared with the existing one.

The remainder of this paper is organized as follows. Section 2 presents the system model and Section 3 proposes the utility-based sampling rate allocation scheme. Section 4 shows the simulation results and Section 5 concludes this paper.

2. System model

The overall system is modeled as an undirected graph $G(V, E)$, in which V and E represent the sets of network nodes and links, respectively. Each node represents an INT-capable switch that can work as either an INT source or an INT sink. The flow set in the networks is represented by F and each flow f has the path P_f, which is pre-determined as the shortest path to the destination. Note that, P_f is the set of nodes v that is on the path of flow f. Each switch is able to embed network INT items $t \in T$ into the packets of flow f where T is the set of INT items.

Fig. 1 illustrates a motivating example consisting of two flows: (1) $f1$ (h1 $\rightarrow$ h3) and (2) $f2$ (h2 $\rightarrow$ h4). The network controller assigns INT sampling rates to each flow, and INT items are collected depending on the sampling rates of its visiting flow. Hence, increasing the sampling rate improves the reliability and accuracy of INT-based network monitoring applications. However, excessive redundancy imposes significant INT overhead. To allocate appropriate sampling rates for flows and to achieve better network visibility, two types of redundancy on the collected INT item need to be considered: (1) spatial redundancy and (2) temporal redundancy.

Typically, INT packets follow the shortest path to the destination, and thus central nodes (e.g., SW1 in Fig. 1) linked with a number of neighbor nodes are frequently visited by different flows [13]. Under this situation, INT items from SW1 will be

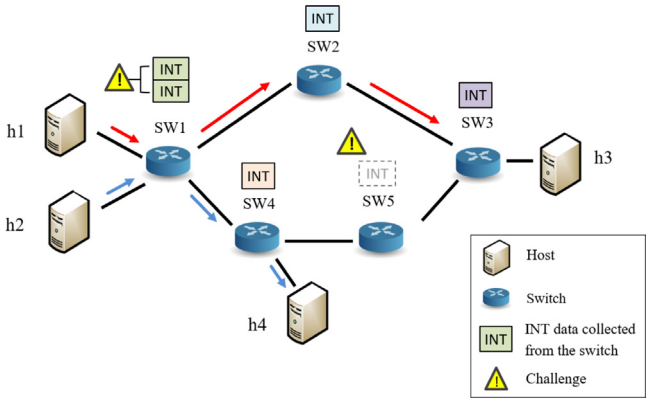

Fig. 1. Example scenario.

redundantly collected, i.e., spatial redundancy. Meanwhile, the flows will rarely visit the nodes (e.g., SW5 in Fig. 1) placed on the peripherals of the network. Thus, sufficient network-wide visibility can be provisioned only when higher sampling rates are assigned to such unpopular nodes. Specifically, this spatial redundancy can be quantitated by node centrality, which is one of the centrality indices to measure how many shortest paths go through the node in the network topology. The node centrality of node v is calculated as [14],

$$c_v = \sum_{s \in V} \sum_{d \neq s \in V} \frac{sp(s, d; v)}{sp(s, d)}, \tag{1}$$

where $sp(s, d)$ refers to the number of the shortest paths between source node s and destination node d, and $sp(s, d; v)$ represents the number of the shortest paths that include node v.

Thus, by calculating c_v, the proportion of the shortest paths passing certain node v can be measured. Then, the spatial redundancy of flow f, denoted by SR_f, can be defined as

$$SR_f = \frac{1}{\max_{v \in V}\{c_v\}} \frac{1}{n(P_f)} \sum_{v \in P_f} c_v, \tag{2}$$

where $n(P_f)$ is the total number of nodes in path P_f, which is used for computing the average node centrality of nodes in P_f. Note that SR_f is normalized by the maximum node centrality value (i.e., $\max_{v \in V}\{c_v\}$) and thus its range is [0, 1].

On the other hand, the temporal redundancy is caused by collecting INT items more than once at the specified time epoch. Obviously, the temporal redundancy is affected by the sampling rates of other flows that are overlapped at the same node. Let $O_{f,v,t}$ be the sum of the sampling rates of overlapped flows with flow f at node v on INT item t. Then, $O_{f,v,t}$ can be defined as

$$O_{f,v,t} = \sum_{t \in T} \sum_{v \in P_f} \sum_{k \neq f \in F} \delta^f_{v,t,k} x_k \tag{3}$$

where $\delta^f_{v,t,k}$ is a binary variable to indicate whether flow k is overlapped with flow f at node v on INT item t and x_k is the sampling rate of flow k.

Note that the type of INT items also affects temporal redundancy. For example, static INT items (e.g., switch ID)

will experience higher temporal redundancy than dynamic INT items (e.g., hop latency and queue length), since static network state only needs to be collected once. In order to consider the sensitivity variation among INT items, we define a parameter S_t as the sensitivity of redundancy on INT item t. Specifically, static INT items have values of S_t close to 1, while dynamic INT items have values close to 0. To summarize, the overall temporal redundancy of flow f considers all types of INT items and all traversing nodes, and thus it is given by

$$TR_f = \frac{1}{TR_{max}} \sum_{v \in P_f} \sum_{t \in T} S_t O_{f,v,t}, \tag{4}$$

where TR_{max} represents $\max_{f \in F}\{TR_f\}$, which is used for normalization. Therefore, similar to the range of SR_f, TR_f ranges between 0 and 1.

3. Utility-based sampling rate allocation

In this section, we present a utility optimization problem where the utility function is characterized as the level of satisfaction of flow f when network visibility is achieved by extracting INT items from INT packets. Specifically, the utility function is a monotone-increasing function since a higher sampling rate guarantees higher satisfaction (or network visibility). Let $U(\cdot)$ be concave and increasing utility function that models the "law of diminishing returns". The utility function can be defined to be strictly concave, such as a logarithmic function [15] and thus we define $U(x_f) = ln(x_f)$. In addition, we introduce the weight for flow f, denoted by w_f, which represents the network monitoring quality of flow f. As analyzed in Section 2, the spatial redundancy and temporal redundancy are jointly considered to derive w_f, which is given by

$$w_f = \alpha(1 - SR_f) + (1 - \alpha)(1 - TR_f) \quad (\forall f \in F), \tag{5}$$

where the spatial redundancy and temporal redundancy are prioritized according to the parameter α. By (5), a higher weight can be assigned to a flow with low levels of spatial and temporal redundancy.

Since the goal of the utility maximization framework is to maximize network visibility (i.e., quantitated by utility) by allocating appropriate sampling rates for flows, the optimization problem can be defined as

$$\max_{x} \quad \sum_{f \in F} w_f \cdot U(x_f) \tag{6a}$$

$$\text{s.t.} \quad 0 \le x_f \le Z_e, \quad (\forall f \in F, e \in E) \tag{6b}$$

$$\sum_{f \in F} x_f \le \pi \tag{6c}$$

(6b) represents the upper bound and lower bound of the sampling rate, i.e., Z_e refers to the capacity of link e that can be used for sampling of flow f. Meanwhile, (6c) represents the total INT sampling rate constraint, i.e., the total INT sampling rate should be bound to the upper limit π.

Since the problem (6a) is a convex optimization problem due to the concave utility function, it can be solved by the Lagrange method and Karush–Kuhn–Tucker (KKT) condition. The Lagrangian of the problem (6a) is defined as

$$\mathcal{L}(x_f, \mu_f^U, \mu_f^L, \lambda) = \sum_{f \in F} w_f \cdot U(x_f) - \sum_{f \in F} \mu_f^U (x_f - Z_e) + \sum_{f \in F} \mu_f^L (x_f) - \lambda \left(\sum_{f \in F} x_f - \pi\right), \tag{7}$$

where μ_f^U, μ_f^L and λ are the Lagrangian multipliers associated with the constraints on the link capacity (6b) and total sampling rate (6c). Then the KKT condition-based approach can be used to find the optimal sampling rate that meets the primal–dual optimality. To satisfy the necessary and sufficient condition of the KKT condition, the optimal sampling rate is derived as

$$x_f^* = \frac{w_f}{\lambda} \quad (0 \le x_f \le Z_e). \tag{8}$$

To obtain the Lagrangian multiplier λ, we apply the subgradient method [16]. We define a recursive equation as

$$\lambda^{t+1} = max(0, \lambda^t + \epsilon \times (\sum (x_f^* - \pi))), \tag{9}$$

where ϵ is the step size. In [16], it is proven that if we choose an appropriate step size, λ converges to the optimal variable λ^*, thus an explicit form of (8) can be used to obtain the optimal sampling rate x_f^*.

Note that the sampling rates of other flows should be given to derive the temporal redundancy TR_f, and TR_f is used for deriving the weight in the utility maximization problem. Therefore, we cannot obtain any closed form on the optimal x_f and thus an iterative approach is required. In other words, the initial values on x_f are used to obtain TR_f, which is then used for solving the utility maximization problem. After that, the obtained x_f is fed to TR_f for the next iteration. This iteration is repeated until converged values are derived. To this end, we devise an iteration algorithm in Algorithm 1. First of all, the sampling rates are randomly set under the constraints, and SR_f and TR_f are obtained by (2) and (4) (line 3). Given the obtained redundant metrics, the weight of each flow, w_f, is calculated and an inner iteration is implemented to solve the utility optimization problem (6a) (line 4). Under the condition of ϵ satisfying the *diminishing step size rule* [17], the dual variable λ converges, and thus the optimal sampling rate can be obtained by an explicit form of (8) (lines 8–14). Here, TR_f has been obtained from an arbitrary flow sampling rate, and therefore it should be through the optimal sampling rate obtained from the inner iteration. As a result, an outer iteration is performed until the required iteration steps are done or the variance of TR_f becomes equal to or smaller than a small value (i.e., $\triangle$) (lines 5–7).

4. Performance evaluation

In this section, we evaluate the performance of uINT. The goal of our evaluation is to compare network monitoring quality between uINT and other sampling-based INT scheme, i.e., sINT [7].

Algorithm 1 Iterative Algorithm

input ϵ: Step size; T, R: Total number of iteration rounds; $\triangle$: Minimum gap between two iterations
output x_f^* : Optimal sampling rates for each flow
Choose an initial flow sampling rate x_f^0 and compute SR_f^0 and TR_f^0
while $r < R$ or $TR_f^{r+1} - TR_f^r > \triangle$ **do**
  Update TR_f^r by (4) and configure w_f^r based on (5)
  while $t < T$ or $x_f^{(t)} - x_f^{(t+1)} > \triangle$ **do**
    Initialize $\lambda^{(0)}$
    $x_f^{(t)} = \frac{w_f^r}{\lambda^{(t)}}$
    $\lambda^{(t+1)} = max(0, \lambda^{(t)} + \epsilon(\sum x_f^{(t)} - \pi))$
    Update step size ϵ
    Update iteration round $t = t + 1$
  end while
  Update x_f^{r+1} with converged sampling rate
  Update iteration round $r = r + 1$
end while

4.1. Setup

All experiments were conducted on a machine with Intel i7-9700K processors and 32 GB of RAM, using the Ubuntu 22.04 operating system. We consider a simple tree topology shown in Fig. 2, which has been implemented on Mininet. All switches are bmv2 software switches and they are connected to one ONOS controller. Hosts are either INT sink or source nodes. Under the given tree topology, different values of node centrality can be observed. For example, the node centrality of SW 2 and SW 3 is 1.5 times larger than that of SW 1. For the workload, we used the packet capture files from UNIV1 trace [18]. We distributed the packet capture files of UNIV1 to the Mininet hosts and replayed them using the *tcpreplay* tool. The source and destination nodes of each flow are randomly selected and the routing paths of the flows are determined by the shortest path algorithm. We used the ONOS controller for the path setup and implemented segment routing (SR) to steer the packets. For instance, when h1 sends packets to h4, the ONOS controller calculates the shortest path to the destination (i.e., SW4 → SW2 → SW3) and includes the corresponding segments header in the packet header to instruct the path. Then, the switch checks the packet header and routes to the switch instructed in the segment header. We assume that each flow can collect two INT items: switch ID and hop latency. The sensitivity parameters S_t for the switch ID and hop latency are set to 1 and 0.6, respectively. This value is based on the fact that static telemetry items (e.g., switch ID) rarely change compared to dynamic ones (e.g., hop latency). On each link e, the INT sampling capacity Z_e is randomly selected between 50% and 80%. Since the available INT overhead depends on the link capacity. Also, we have $\alpha = 0.5$, and $\pi = 25\% \times N_f$ where N_f is the number of flows. For comparative study, we consider sINT [7] whereby static and equal sample rates are set for flows. Sampling rates of flows in sINT are set as 25% to ensure both INT schemes have the same INT overhead.

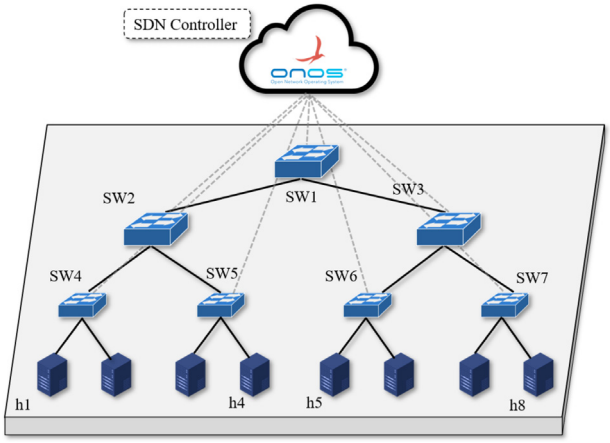

Fig. 2. Experimental setup.

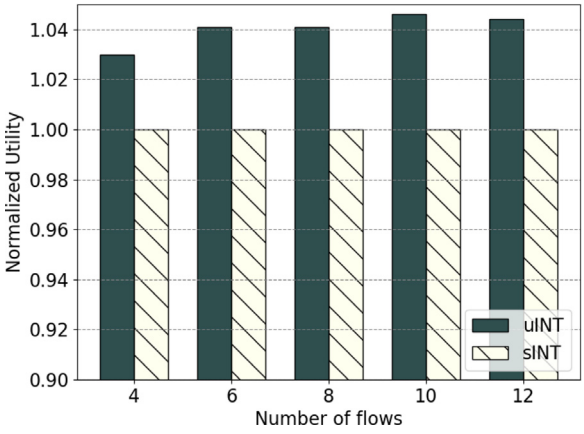

Fig. 3. Normalized utility comparison with sINT.

4.2. Experimental result

Fig. 3 shows the weighted utility of uINT normalized by that of sINT. It can be seen that uINT always exhibits higher utility than sINT. By its definition in (5) and (6a), the weighted utility is inversely proportional to the spatial and temporal redundancy. Therefore, these results indicate that uINT can provide improved satisfaction on network visibility and effectively reduce the redundancy (or INT overhead) compared with sINT. Moreover, it can be found that the weighted utility of uINT increases as the number of flows increases. This is because the impacts of the spatial and temporal redundancy become apparent as more flows are overlapped at spatial and temporal domains.

In Fig. 4, we demonstrate the impact of redundancy in INT data on the accuracy. To measure the accuracy of the data collected, we first compute the root mean squared error (RMSE) of the hop latency using INT reports collected through uINT and SINT. In addition, the ground truth value, which is used to measure deviation error, is obtained by collecting per-packet INT reports on the INT sink node. Then we normalize RMSE by its range and represent it as the normalized root mean

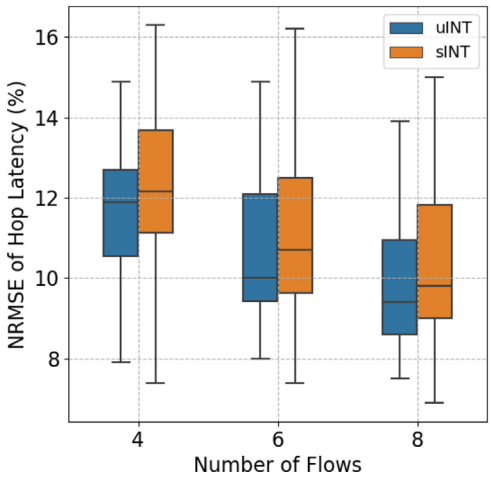

Fig. 4. Measurement accuracy of hop latency.

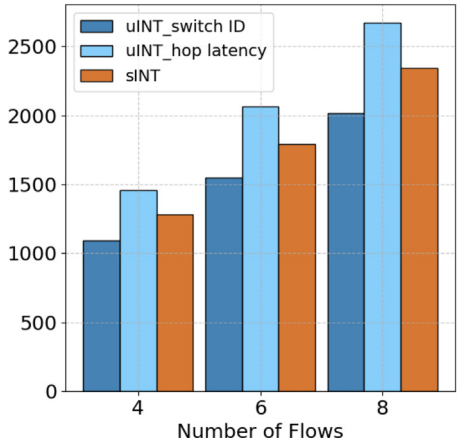

Fig. 5. INT items collected.

squared error (NRMSE) depicted in Fig. 4. Since sINT maintains constant sampling rates regardless of the redundancy, it results in high redundancy in some nodes. For example, when INT flows overlap on a switch, redundant INT data are collected at the switch. In the case of dynamic telemetry items, this can lead to a minor increase in accuracy at the node level; however, it results in a decrease in accuracy from a network-wide perspective, since other nodes in the network experience INT data starvation. In Fig. 4, it can be found that the NRMSE of uINT is lower than that of sINT, which indicates that uINT can increase network-wide visibility by efficiently handling data redundancy even in scenarios where the number of flows increases. Moreover, the minimum and maximum values of NRMSE in sINT are larger than those of uINT, respectively. This is because there is a significant variation in the data accuracy measured at each node when using sINT, thereby adversely affecting balanced network monitoring across the entire network.

Fig. 5 shows the effect of sensitivity of redundancy, S_t, on the INT item collection. We consider two telemetry items (i.e., switch ID and hop latency) with two INT schemes. From Fig. 5, uINT collects different amounts of INT items with the consideration of their sensitivity. In particular, it can be found that uINT collects more INT items on the hop latency compared with sINT. Since the hop latency is dynamic, its more collections guarantee improved accuracy of INT measurement and network visibility. Meanwhile, it can be seen that sINT does not differentiate INT items and therefore the same amounts of INT items are collected regardless of INT items. In this situation, most of the INT items including the switch ID are redundant and do not contribute to improving the network visibility.

5. Conclusion

In this paper, we proposed uINT, a utility-based sampling rate allocation scheme for INT. In the utility maximization framework, novel spatial and temporal redundancy metrics have been introduced, which contribute to improving the network visibility and reducing monitoring errors compared with the existing sampling-based INT. Our simulation results demonstrate that uINT effectively reduces redundant data by 15% compared to sINT while increasing network monitoring quality under the same INT overhead. In our future work, we will investigate the information theory to measure data redundancy and design a redundancy-aware adaptive INT collection scheme.

CRediT authorship contribution statement

Kyeongtak Lee: Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing – original draft. **Heewon Kim:** Conceptualization, Writing – review & editing. **Subin Han:** Software, Writing – review & editing. **Sangheon Pack:** Methodology, Writing - review & editing, Supervision, Funding acquisition. **Hyunkyung Yoo:** Supervision, Funding acquisition. **Namseok Ko:** Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by ETRI grant funded by the Korean government (23ZR1110, A Study of Hyper-Connected Thinking Internet Technology by Autonomous Connecting, Controlling, and Evolving Ways).

References

[1] J. Case, M. Fedor, M. Schoffstall, J. Davin, Simple Network Management Protocol (SNMP), tech. rep., 1989, [Online]. Available : https://www.rfc-editor.org/info/rfc1157.

[2] B. Claise, Cisco Systems Netflow Services Export Version 9, tech. rep., 2004, [Online]. Available: https://www.rfc-editor.org/rfc/rfc3954.

[3] M. Wang, B. Li, Z. Li, Flow: Towards resource-efficient and agile service federation in service overlay networks, in: Proc. IEEE ICDCS 2004, 2004.

[4] J. Marques, M. Luizelli, I. Tavares da Costa Filho, L. Gaspary, An optimization-based approach for efficient network monitoring using in-band network telemetry, J. Internet Serv. Appl. 10 (2019) 1–20.
[5] S. Tang, D. Li, B. Niu, J. Peng, Z. Zhu, Sel-int: A runtime-programmable selective in-band network telemetry system, IEEE Trans. Netw. Serv. Manag. 17 (2) (2019) 708–721.
[6] In-band network telemetry (INT) dataplane specification v2.1 [Online]. Available : https://github.com/p4lang/p4-applications/blob/master/docs.
[7] Y. Kim, D. Suh, S. Pack, Selective in-band network telemetry for overhead reduction, in: Proc. IEEE CloudNet 2018, 2018.
[8] D. Suh, S. Jang, S. Han, S. Pack, X. Wang, Flexible sampling-based in-band network telemetry in programmable data plane, ICT Express 6 (1) (2020) 62–65.
[9] R. Basat, S. Ramanathan, Y. Li, G. Antichi, M. Yu, M. Mitzenmacher, PINT: Probabilistic in-band network telemetry, in: Proc. ACM SIGCOMM 2020, 2020.
[10] Z. Xu, S. Tang, Z. Zhu, Entropy-driven adaptive int and its applications in network automation of ip-over-eons, IEEE J. Sel. Top. Quantum Electron. 28 (4) (2022) 1–13.
[11] S. Sheng, Q. Huang, P. Lee, A general delta-based in-band network telemetry framework with extremely low bandwidth overhead, Comput. Netw. 223 (109573) (2023).
[12] Y. Li, X. Chao, S. Ercisli, Disturbed-entropy: A simple data quality assessment approach, ICT Express 8 (3) (2022) 309–312.
[13] S. Borgatti, Centrality and network flow, Social Networks 27 (1) (2005) 55–71.
[14] D. Arrowsmith, R. Mondrag, M. Woolf, Data traffic, topology and congestion, Complex Dyn. Commun. Netw. (2005) 127–157.
[15] S. Shakkottai R. Srikant, Network optimization and control, Found. Trends Netw. 2 (3) (2007) 271–379.
[16] F. Kelly, A. Maulloo, D. Tan, Rate control for communication networks: shadow prices, proportional fairness, and stability, J. Oper. Res. Soc. 49 (1998) 237–252.
[17] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[18] T. Benson, A. Akella, D. Maltz, Network traffic characteristics of data centers in the wild, in: Proc. ACM SIGCOMM 2010, 2010.