Rhaban Hark, Mohamed Ghanmi, Sounak Kar, Nils Richerzhagen, Amr Rizk, and Ralf Steinmetz. *Representative Measurement Point Selection to Monitor Software-defined Networks (accepted for publication)* In: Proceedings of the 43nd IEEE Conference on Local Computer Networks (LCN), p: 1–8, IEEE, October 2018.

Representative Measurement Point Selection to Monitor Software-defined Networks

Rhaban Hark, Mohamed Ghanmi, Sounak Kar, Nils Richerzhagen, Amr Rizk, and Ralf Steinmetz
Multimedia Communications Lab
Technical University of Darmstadt, Germany
Email: {first.last}@kom.tu-darmstadt.de

***Abstract*—Network state monitoring is a fundamental task for network management. However, determining the full network state in Software defined Networks requires disproportionately too many resources. This stems from the discrepancy between the established methods used for state monitoring compared to the varying contribution in terms of information obtained from every additionally monitored network node. This relationship may even become more complicated depending on the network state information of interest. One solution to overcome bottlenecks by reducing the overall monitoring footprint is the use of spatial sampling, which allows the estimation of the network state based a fraction of the overall state.**

In this work, we propose schemes to place a small number of measurement points in the SDN network to maximize the obtained network state information. Considering different conditions, we utilize routing information and graph theoretic centrality metrics, respectively, to estimate the amount of information a node provides. Based on this knowledge, we, furthermore, develop a mechanism to place multiple measurement points while avoiding redundant measurements. For demonstration purpose, we use the developed mechanisms to estimate the Flow Size Distribution in SDN environments. An emulative evaluation taking several known topologies shows the effectiveness of spatial sampling using the proposed scheme.

I. Introduction

In the last decade the Software-defined Networking (SDN) paradigm became very popular in industry as well was in academia. This is, among others, justified by its major advantage of broad support for highly flexible and dynamic network management resulting from logical control centralization [19]. The basis for flexible network configuration is the continuous observation of the underlying forwarding network. To observe the network status SDN eases monitoring providing new techniques, such as practical flow-level packet/byte/lifetime counter [17]. However, as most management applications heavily rely on monitoring information, they require a tremendous number of various measurements. Depending on the required accuracy, timeliness, and completeness, monitoring constitutes one of the major resource consumption aspects on the control-, but also the data-plane of SDNs [5].

Several works propose sophisticated approaches to optimize the statistic collection process in order to overcome this scalability issue using, e.g., adaptive granularity in time and space [4], [28], eliminating redundant measurements [11], adaptively selecting the best-suitable measurement technique [10], and further. In contrast, in this work, we propose to reduce monitoring costs using only a small set of measurement points instead of optimizing the collection process. Through spatial sampling, we measure the desired information only at a subset of data-plane elements to infer the entire network state. The superficial notion to achieve good estimations is our assumption that many paths go through *central* nodes, thus, that *central* nodes carry information with the most entropy and are representative of the overall traffic.

We investigate the selection of the most central node under different assumptions: *(i)* Either the operator knows which flows traverse the network and their paths, which means that the most central node is the one seeing most flows, or *(ii)* the operator has no information on the flows in the network and presumes the importance only based on the physical topology. The former case dynamically adapts to the current node load and calculates the centrality based on active flows. For the latter case, we rely on metrics from the graph theory to estimate different types of centrality only based on the rather rigid structure of the network.

Furthermore, despite using a single measurement point, we propose methods to allow multiple measurement points if more resources are available. For this, we require more elaborated centrality metrics supporting an intelligent selection of multiple nodes. Additionally, considering multiple nodes, we must convolve the monitoring information of all nodes into a single estimation.

In this work, we investigate the aforementioned approach in particular for the estimation of the Flow Size Distribution (FSD), which is an important monitoring metric [15] providing a good use case.

The remainder of the paper is structured as follows: The next section describes flows and the FSD, as multiple definitions exist in the literature. In Section III, we introduce our selection strategies for *(A)* the case with knowledge on flow information and *(B)* assuming absence of flow information. Subsequently, Section IV describes how we propose to perform the FSD measurement once monitoring locations are selected. In Section V, we provide an excessive evaluation where we investigate the achieved accuracy versus cost reduction. Before Section VII concludes the paper, Section VI outlines relevant related works.

II. Flows and the Flow Size Distribution

In this section, we clarify assumptions on our understanding of the *flow* term. Furthermore, the section describes rele-

Rhaban Hark, Mohamed Ghanmi, Sounak Kar, Nils Richerzhagen, Amr Rizk, and Ralf Steinmetz. *Representative Measurement Point Selection to Monitor Software-defined Networks (accepted for publication)* In: Proceedings of the 43nd IEEE Conference on Local Computer Networks (LCN), p: 1–8, IEEE, October 2018.

vant background on the Flow Size Distribution metric and exemplary uses.

A. Flow Model

As there is no broadly uniformed clear definition of flows, we give the definition we use in the scope of this work. Here, a flow is defined as a stream of packets that are processed by the same forwarding rules while traversing the network, thus, share equal header fields. In the context of SDN and in particular OpenFlow-enabled networking this results in having *one flow rule* processing *exactly one flow*. Hence, if a switch handles multiple logically independent flows of different applications or of different endpoints with the same rule, we define such flows to belong to the same *aggregate flow of interest*. Furthermore, the rule lifetime defines a flows end, so that a reinstalled flow rule is defined as a new and independent flow of the predecessor. In addition, subsequent sub-flows, managed with the same rule, which has not yet been removed in the meantime, are part of the same flow.

B. Flow Size Distributions

The flow size is the number of packets contained in a flow[1]. Hence, the Flow Size Distribution (FSD) of a network is the distribution of all flow sizes in the network. The FSD gives valuable knowledge about the traffic traversing the network. We highlight exemplary usages afterwards.

Following [25], we assume N_f as the number of flows with m_i being the size of the i'th flow. $W \in \mathbb{N}^*$ is the finite maximum flow size: $W = \max_i \{m_i\}$. Thus, $1 \leq m_i \leq W$. Furthermore, we denote M_j as the number of flows having size j. The total number of flows N_f can be written as $N_f = \sum_{i=0}^{W} M_i$. Hence, the distribution of flow sizes, denoted FSD, is the set $\theta = \{\theta_1, \ldots, \theta_W\}$ containing the flow occurrence ratios of all possible sizes:

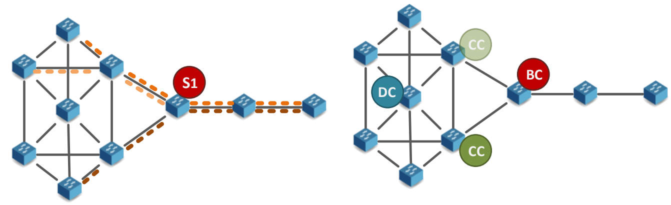
$$\theta_j = \frac{M_j}{N_f} = \frac{M_j}{\sum_{i=0}^{W} M_i}. \tag{1}$$

The FSD allows to **infer the type of traffic** traversing a network [15]. Applications, such as voice, music and video streaming show deterministic packet size and flow length characteristics, allowing a reconstruction of their bandwidth portion. Knowledge about the active traffic type allows optimization regarding the utilization of network resources and meeting performance requirements.

Using the FSD, a network operator can **detect abnormal traffic and faults** in a network [13]. As an example, he can detect DDoS attacks by examining the number of small flows. DDoS attacks flood a target with service requests using, e.g. Pings or TCP SYNs. Such attacks produce large numbers of minimal size flows which can be detected easily using the FSD compared to resource-consuming intrusion detection systems.

Another example is the detection of simple worms in a network, which result in a large number of flows having the same size [13].

[1]Note, that other definitions take the number of bytes or the like into account. In this work we focus on the number of packets per flow.



(a) With routing knowledge: the node with most traversing flows is considered as *most informative*.

(b) Without flow knowledge: centrality metrics based on the physical topology give an estimate on the *most informative* node.

Fig. 1: Switch selection strategies based on *(a)* the assumption that full knowledge on flows/routing is available and *(b)* the assumptions that there is no knowledge on flows.

III. Measurement Point Selection

In this section, we first describe the measurement point selection based on the assumption that routing information is accessible. Afterwards, we describe the second method assuming that there is no information given on flows.

A. Measurement Point Selection Considering Flow Information

Given knowledge on flows from, e.g., the routing application, we can count the number of flows that traverse a single switch. Considering such information, we base the centrality on active flows. Subsequently, we consider the most informative measurement point (switch) to be the switch with the highest number of flows traversing the switch.

Formally spoken, the network is a collection $G(V, E, F)$ of nodes/vertices $V = \{1, 2, \ldots, N\}$ (*here:* switches), edges $E = \{(v_i, v_j) \mid i, j \in V, i \neq j\}$ as a set of node pairs (*here:* links), and flows $F = \{f_1, \ldots, f_{N_f}\}$ (while N_f is the total number of flows). A single flow can be expressed as $f_l = \{i_1, i_2, \ldots, i_k\}$ with $\{i_1, i_2, \ldots, i_k\} \subseteq V$. The number of flows traversing through a node i is then $C(i, F) = \sum_{f_l \in F} \mathbb{1}(i \in f_l)$ for $i \in V$. The first selected node s_1 is

$$s_1 = \arg\max_{i \in V} C(i, F). \tag{2}$$

Consider the network shown in Figure 1a. The figure shows active flows as dashed lines with shades of orange. As the switch marked with S1 sees three flows, while all other switches see fewer flows, it will be selected first.

For the selection of further measurement points, we do not any further consider the flows that are already taken into account. Hence, the same method is used to select the next measurement point s_2 ignoring all flows traversing s_1. Given the set of flows traversing s_i with $F_{s_i} = \{f_l \in F : s_i \in f_l\}$, we select s_{i+1} with

$$s_{i+1} = \arg\max_{j \in V} C(j, F \setminus \cup_{k=1}^{i} F_k). \tag{3}$$

Thus, we select all further points s_{i+1} analogously ignoring all flows which are already captured at $\{s_1, s_2, \ldots, s_i\}$ assuming we are able to identify flows uniquely.

Rhaban Hark, Mohamed Ghanmi, Sounak Kar, Nils Richerzhagen, Amr Rizk, and Ralf Steinmetz. *Representative Measurement Point Selection to Monitor Software-defined Networks (accepted for publication)* In: Proceedings of the 43nd IEEE Conference on Local Computer Networks (LCN), p: 1–8, IEEE, October 2018.

B. Measurement Point Placement without Knowledge on Flows

As monitoring applications might not have access to flow information in some scenarios due to privacy concerns or technical limitations, we investigate a second approach to select the most informative measurement point in this section. Without knowledge on traffic information we rely only on the physical topology of the network $G(V, E)$. Predominantly, the social network research community considers different metrics to measure the importance of nodes in a graph using primarily *centrality* measures [2], [8]. In the following we briefly introduce a subset of graph centrality metrics we consider to select measurement points.

Betweenness Centrality [9]: The betweenness centrality rates the relative number of shortest paths in the network that traverse through the investigated node. This metric is particularly interesting as shortest paths are often preferred for routing between end-points in communication networks and, thus, are potentially good candidates for measurement points with major relevance. If $\rho_{i,j}$ is the number of shortest path between node $i \in V$ and node $j \in V$ and if $\rho_{i,j}(k)$ is the number of shortest path between both nodes traversing through $k \in V$, we compute the betweenness centrality BC_k of node k with

$$BC(k) = \frac{1}{(N-1)(N-2)} \sum_{\substack{i,j \in V \\ i \neq j \neq k}} \frac{\rho_{i,j}(k)}{\rho_{i,j}}. \tag{4}$$

For the network shown in the example of Figure 1b, BC indicates the highest value.

Closeness Centrality [9]: The closeness centrality measures the inverse distance between an investigated node and all other nodes in the graph. Hence, it spreads information efficiently in the network. It can directly be captured using the inverse average path length to all other nodes: Consider $d(i, j)$ as the distance between node i and j. Then the closeness centrality of node k is given with

$$CC(k) = \frac{n-1}{\sum_{i \in V, i \neq k} d(i,k)}. \tag{5}$$

In the example of Figure 1b, two nodes have the highest closeness centrality score. They are marked with CC and we randomly pick one candidate.

Degree Measure: The degree of a node represents the number of edges connecting it to other nodes. Its use as node with high representativeness strongly depends on the graph structure. Assume $a(i, j)$ equals 1 if a connection between node i and j exists, otherwise it equals 0 (adjacency matrix). For a node k, the degree is given in Equation (6). It is marked with DC in Figure 1b.

$$DC(k) = \sum_{i \in V} a(i,k). \tag{6}$$

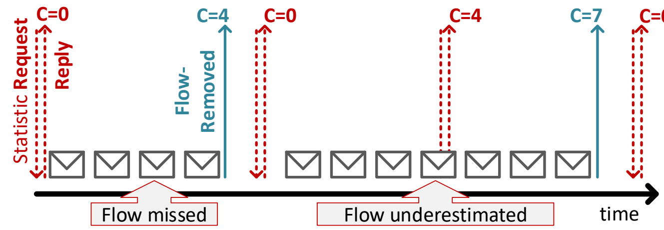


Fig. 2: Illustration of problems connected to periodic polling *(red, dashed)* versus leveraging OpenFlow FlowRemoved control messages *(blue, solid)*.

Multiple Measurement Points: Extended Centrality: Intelligently selecting additional measurement points is a challenging task as the intuitive approach of selecting the *second most central* node is not reasonable in most cases. In most centrality metrics the score of two neighboring nodes can be assumed alike, hence, one would select a neighboring node s_{i+1} of the previously selected node s_i with higher probability. Subsequently, again with higher probability, the measurements at s_{i+1} include a huge subset of the flows already captured at s_i. To avoid this problem, we propose to take shortest paths into account: Say $P_{j,k}(s_i)$ is the set of shortest paths between j and k that pass s_i. Then s_i captures the set of shortest path $P(s_i) = \bigcup_{j,k \in V} P_{j,k}(s_i)$. In order to decide if we take $s_{next} = s_{i+1}$, with the second highest centrality as next measurement point, we check if Inequality (7) is fulfilled. Otherwise we iteratively check $s_{next} = s_{i+2}, s_{i+3}, \ldots$

$$\frac{|P(s_{next}) \setminus P(s_i))|}{|P(s_i)|} \geq \delta_{min}. \tag{7}$$

Intuitively, this makes sure that the portion of shortest paths traversing through the next measurement point exceeds a certain threshold. By default we require the next measurement point to have at least 50% different shortest paths, thus $\delta_{min} = 1/2$. If none can be found the node with the highest ratio of Eq. (7) is preferred.

IV. Estimating the Flow Size Distribution

Given a measurement point we use SDN mechanisms, e.g. available in the popular SDN protocol OpenFlow [17], to estimate the FSD. To do so, an intuitive possibility is to fetch statistics of all flows periodically (statistic polling) and count the number of packets (and bytes, respectively) for each flow. However, this concept has two major problems: Using the information on the number of packets of incomplete flows falsify the distribution as the real length is not yet known and smaller flows become overrepresented. Furthermore, as soon as a flow expires, the monitoring application cannot fetch statistics of this particular flow anymore so that the true length is almost never included in the FSD. The application might miss flows with a smaller lifetime than the polling period completely. Figure 2 sketches both problems.

To avoid this, in analogy to [27], we make use of optionally activatable control messages that are dispatched whenever a

Rhaban Hark, Mohamed Ghanmi, Sounak Kar, Nils Richerzhagen, Amr Rizk, and Ralf Steinmetz. *Representative Measurement Point Selection to Monitor Software-defined Networks (accepted for publication)* In: Proceedings of the 43nd IEEE Conference on Local Computer Networks (LCN), p: 1–8, IEEE, October 2018.

flow times out (cf. Figure 2). Such messages, in the case of OpenFlow denoted *FlowRemoved* (cf. [20]), contain statistical information on the expired flow including the number of packets and bytes. Based on the packet number delivered whenever a flow ends, we can update the FSD and add the size of the expired flow. Despite the ability to capture all flow length correctly, this process makes active requests of statistics obsolete decreasing the costs for statistic retrieval.

Note that we defined a flow's lifetime equal to the lifetime of its corresponding rule in Section II. Hence, two logically different subsequent flows processed by the same flow rule are assumed to be the same flow, e.g. if the flow rule idle timeout is not reached in between.

Aggregation of Multiple Measurement Points

In analogy to the measurement point selection (cf. Sections III-A and III-B), we use two different methods to convolve the measurements of multiple measurement points into a single distribution based on the assumptions.

On the one hand, assuming there is knowledge on flow information, we can easily identify the flows within the application using, e.g., flow cookies (known from OpenFlow). With the knowledge on the flow identifiers, the FSD measurement application calculates the union of both flow size distributions while it removes duplicates from their intersection.

On the other hand, considering a situation without knowledge on flows, the statistic messages return information on the flows such as the match. In this case, we identify the flows based on the exact matches and calculate the union while removing duplicates.

Both methods are limited to scenarios without rule optimization. If the routing application aggregates multiple rules into a single rule, the identification and picking out of duplicates is not possible anymore in many cases: Consider measurement points A and B. Flow $F1$ is mapped to exactly one rule in A, but convolved with another flow $F2$, that does not traverse A to a common rule in B. As the routing application knows about this situation, the first methods allows to identify $F1$, receive its packet count and calculate the packet count for $F2$ as well. However, without knowledge on the rule installation in the network the application cannot infer the packet counter for $F1$ and $F2$ precisely. This problem becomes even worse if the routing adds a third flow to the aggregated rule in B or in, e.g., scenarios where A aggregates $F1$ with another flow $F3$ in while B aggregates $F1$ with $F2$. In such cases, also the first method fails and a mathematical convolution of both distribution is of choice, which is left for future work.

V. Evaluation Methodology and Results

This section covers an emulative evaluation we conduct to show the performance of the selection methods.

A. Evaluation Environment

We use mininet [16] to set up virtual networks with different topologies that is managed by a RYU[2] OpenFlow controller. The controller performs the measurement point selection locally and collects FSD information. It serves the network with shortest-path routing.

We generate traffic using RUDE/CRUDE[3]. If not stated differently, we trigger 100 flows per emulation from a randomly chosen source host to a randomly chosen destination. Hosts are connected to each switch by default (with exception of the data-center topology, which has only hosts at the leaf/rack switches). The flows are sent with exponentially distributed inter arrival times, hence, as Poisson process. Furthermore, we use constant bit rate flows and model the flow sizes using Zipf's distribution with parameter 1.6. Thus, we mainly have very short flows (*mice flows*), while a small number of large flows occur (*elephant flows*) (cf. Figure 5c).

We perform the measurements on various topologies from the Internet Topology Zoo [14] and a classical data-center topology with a tree-like structure (cf. [22]).

B. Evaluation Metrics: Accuracy and Costs

We evaluate the system regarding its performance in terms of accuracy and cost reduction. In this work, the *Bhattacharyya distance* [1] is of choice to evaluate the **accuracy**. It provides a measure for the similarity of distributions. It is defined as

$$B(f,g) = -\ln(\rho(f,g)) \tag{8}$$

where f and g are two probability mass functions and $\rho(f,g)$ is the *Bhattacharyya coefficient* given with

$$\rho(f,g) = \sum_{x \in X} \sqrt{(f(x) \cdot g(x))} \tag{9}$$

while $0 \leq \rho \leq 1$, thus $0 \leq B \leq \infty$. The closer B is to 0, the lower is the relative distance of the two taken distribution samples.

To measure the **costs** required to determine the FSD, we use the number of bytes required for statistic collection. As we designed the system in a way, that it does not require statistic requests, this metric includes only the bytes for FlowRemoved messages. Such optional messages are only triggered if the operator tells the switches to do so.

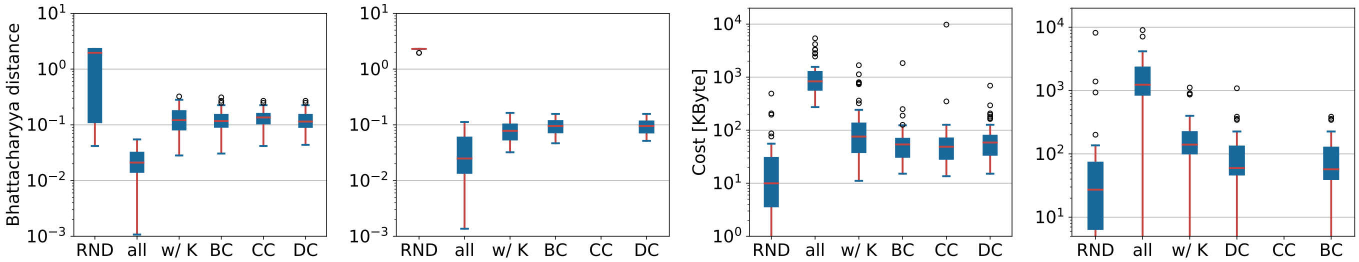
C. Selected Results

In the following, we show results with regard to *(i)* the accuracy and costs using only a single measurement point that is selected using the proposed methods compared to naive selections and the baseline of using all measurement points; *(ii)* the influence on the accuracy and costs when changing the number of available measurement points; *(iii)* the performance of selection methods within different topologies; and *(iv)* the trade-off between accuracy and costs.

[2]https://osrg.github.io/ryu/, accessed 15 Dec. 2017

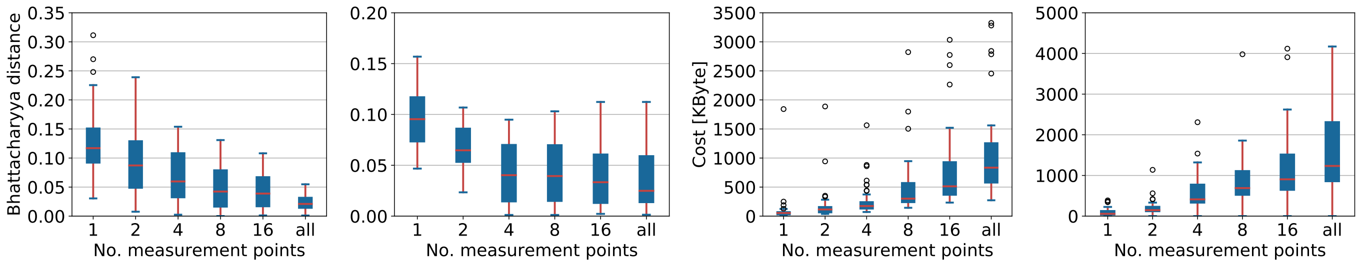
[3]http://rude.sourceforge.net, accessed 16 April 2017

Rhaban Hark, Mohamed Ghanmi, Sounak Kar, Nils Richerzhagen, Amr Rizk, and Ralf Steinmetz. *Representative Measurement Point Selection to Monitor Software-defined Networks (accepted for publication)* In: Proceedings of the 43nd IEEE Conference on Local Computer Networks (LCN), p: 1–8, IEEE, October 2018.



(a) Surfnet topology (b) DC topology (c) Surfnet topology (d) DC topology

Fig. 3: Accuracy and cost of estimating the flow size distribution using representative measurement points with different placement methods for the Surfnet topology given from the Internet Topology Zoo and a data-center topology. The accuracy is given in terms of relative distance between the estimation and the ground truth distribution. Costs are given with bytes needed for statistic exchange for the additionally required and optionally activated FlowRemoved messages.



(a) Surfnet topology (b) DC topology (c) Surfnet topology (d) DC topology

Fig. 4: Accuracy and cost of estimating the flow size distribution with changing number of measurement points. The information contribution of additional points clearly depends on the topology. Topologies and metrics are equivalent to Figure 3.

1) Qualify of Estimation Using a Minimal Number of Representative Measurement Points: In this investigation, we first show how using a minimal number of measurement points to estimate the FSD of a network influences the measurement quality. For this, we consider two exemplary topologies to indicate the quality using the presented placement strategies.

Figures 3a and 3b show the accuracy using the Bhattacharyya distance on a logarithmic axis for only a single measurement point that is selected randomly (RND), using all available measurement points (all), using the switch that sees the highest number of flows, thus, assuming knowledge on flows (w/ K), and using a single measurement point placed using the presented centrality score functions (DC, CC, BC).

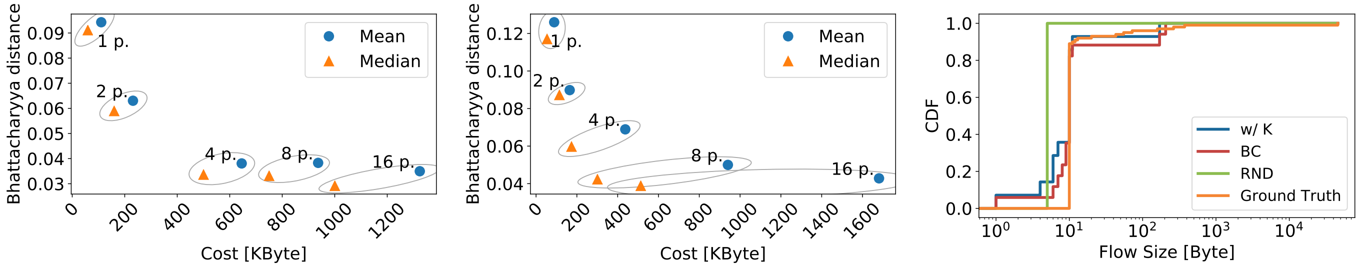
In both figure, it is observable, that using one of the proposed strategies provides better estimations of the FSD compared to a random placement among all possible points. Using all measurement points leads, of course, to the best estimation result. It is notable that we could not find large differences between placement strategies based on graph centrality metrics. This holds true for almost all topologies we investigated. We have observed, that commonly all placement strategies pick the same point in most topologies. However, in Figure 3b, we see that using closeness centrality does not deliver any results at all. As we use shortest-path routing within our evaluation and the structure of the topology allows routing without using the root switches, we conclude that in some rare situations a flow-agnostic measurement point selection leads to a poor representation of the network state. During our evaluation this was the only case of such behavior.

Despite that rare case, comparing centrality-based placement strategies with a random placement and the placement based on knowledge on flow paths, shows that centrality measures give good estimations of the representativeness of measurement points. There is no improvement compared to the placement strategy with knowledge on flows, while it is much better than a random placement.

If we, furthermore, consider the costs (cf. Figures 3c and 3d) the number of bytes required for the measurement is the highest if we activate the statistics for all flows on all switches. Using only a single measurement point lowers these costs significantly. A random selection shows the lowest number of bytes required for the, optionally activated, FlowRemoved messages containing the processed statistics as they cover the least number of flows. In addition, the flow-aware placement takes slightly more resources than the all centrality-based placement strategies. Taking this slightly higher resource consumption and the unvaried accuracy into account, we argue to use centrality-based placement although knowledge on flows is available.

2) Influence of the Number of Measurement Points On the Estimation Performance: Figure 4 depicts again the Bhattacharyya distance and costs in terms of bytes for statistics

Rhaban Hark, Mohamed Ghanmi, Sounak Kar, Nils Richerzhagen, Amr Rizk, and Ralf Steinmetz. *Representative Measurement Point Selection to Monitor Software-defined Networks (accepted for publication)* In: Proceedings of the 43nd IEEE Conference on Local Computer Networks (LCN), p: 1–8, IEEE, October 2018.



(a) DC topology (b) Surfnet topology (c) FSD estimate for the Surfnet topology

Fig. 5: The two leftmost plots show the trade-off between the accuracy and cost of estimating the flow size distribution using representative measurement points with different numbers of measurement points. Topologies and depicted metrics equivalent to Figure 3. The rightmost plot shows an exemplary FSD estimation. It shows the ground truth, a single randomly selected measurement point (RND), a single measurement point selected with maximum betweenness centrality score (BC), and a selection based on the number of flows a switch sees, thus, with flow knowledge (w/ K).

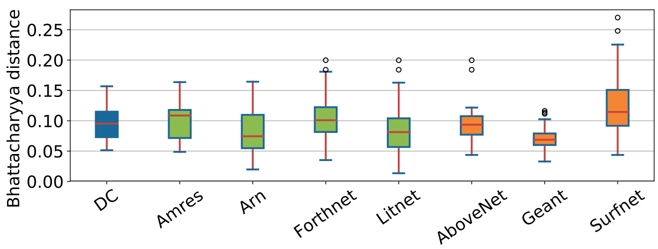


Fig. 6: Accuracy measured using the *Bhattacharyya distance* between the ground truth FSD and the estimated FSD with representative measurement point selection within different topologies taken from the Internet Topology Zoo.

delivery within FlowRemoved messages for both, the Surfnet Topology and a data-center topology with changing numbers of available measurement points. The two leftmost figures show the distance to the true distribution. In case of the Surfnet topology we observe a performance enhancement with more measurement points until all points are used as the structure of the topology is rather meshed, thus, the importance of single nodes is rather uniformly distributed compared to, e.g., a tree-like data-center topology. Figure 4b shows the latter case. It can be observed that there is no improvement after exceeding the number of four measurement points. Hence, the majority of flows are already captured quickly using the centrality metric (in this case betweenness centrality). However, the costs still increase, as depicted in Figure 4d, as additional measurement points redundantly report statistics that are already captured with previously selected measurement points.

This observation particularly shows the strength of performing smart spatial sampling and not gathering the full state from all possible switches.

3) Influence of Network Topologies Structure on the Estimation Performance: Figure 6 represents how the network topology structure influences the centrality metric score. We investigated different types of network structures: *(i)* a data-center topology, marked with blue; *(ii)* three tree-like or star-like structured topologies from the Internet Topology Zoo [14], marked with green; and *(iii)* three rather meshed topologies form the Topology Zoo, marked in orange.

Although we expect the centrality scores to select different measurement points and, thus, provide different estimation qualities, we found that the scores rate similar points as highest in most cases for the investigated topologies. Subsequently, as the figure shows, no significant change in the quality is observable. This was neither the case for the betweenness centrality (cf. Figure 6), nor the closeness centrality or degree centrality. However, as shown in the previous evaluation parts, in exceptional situations (such as using the degree centrality, shortest-path routing and a data-center like topology) some centralities must be preferred or avoided.

4) Discussion of the Trade-Off Between Measurement Accuracy and Costs: In Figures 5a and 5b, we show the direct dependency of both accuracy and costs. Both figures indicate that the relation between accuracy and costs is non linear. First, by increasing the number of measurement points, the number of bytes required for statistic transmission increases quickly before slowing down when taking more measurement points into account. This is due to the lower number of flows covered by additionally selected measurement points. We observe slight differences between the different topology structures: The observed effect is stronger on data-center topologies compared to a meshed topology (Surfnet), where the number of seen flows distributes nearly uniformly among the switches.

Furthermore, the Bhattacharyya distance between the estimation and the true distribution reduces faster in the beginning, i.e. taking a second and forth measurement point into account. For both cases the mean and median distance, respectively, decreases slower considering higher numbers of measurement points. Here again, the effect is stronger for the data-center topology, where the information obtained from a *central* node is higher than in the meshed Surfnet topology.

Given Figures 5a and 5b, the preferred number of measurement points can be either directly selected based on the allowed distance (accuracy) or optimally chosen to have

Rhaban Hark, Mohamed Ghanmi, Sounak Kar, Nils Richerzhagen, Amr Rizk, and Ralf Steinmetz. *Representative Measurement Point Selection to Monitor Software-defined Networks (accepted for publication)* In: Proceedings of the 43nd IEEE Conference on Local Computer Networks (LCN), p: 1–8, IEEE, October 2018.

minimal costs with almost negligible accuracy, e.g., using four measurement points in the exemplary data-center topology.

5) Evaluation Results Summary: In the presented evaluation we have shown that an intelligent selection of measurement points allows good estimates of the network state (*here:* of the Flow Size Distribution). In detail, we observed that the usage of flow-agnostic centrality measures for measurement point placement provides as good results as using methods with knowledge on the number of flows passing switches. We found that, in most cases, the difference between placement strategies based on different centrality measures is negligible. Furthermore, the three investigated centrality measures perform equally good in real-world topologies given from the Internet Topology Zoo. Finally, our evaluation indicates that the number of efficient measurement points for meshed networks flattens in terms of efficiency.

VI. Related Work

The field of monitoring in SDNs in general offers a variety of works regarding different approaches to reduce statistic collection costs through adaptive granularity and timeliness [4], [10], [28], optimized statistic storing [18], and many more [23]. In the following we focus in particular on the selection of measurement points as investigated in this paper and the measurement of the FSD.

A. Measurement Point Placement

In the context of SDN Tootoonchian et al. [24] included a discussion of different measurement point placements already in 2010. In their work, they estimate a traffic matrix using SDN techniques. In order to fetch statistics of each flow, they propose different strategies to place the measurement point: To optimize the accuracy the last switch on the path can be used as all losses are included in the measurement. A uniform random selection of a switch on the path and a round-robin approach allow equally distributed load among the switches. Furthermore, non-uniformly selections allow selecting a switch that is rather at the end of the path, thus, has better accuracy. At last, they propose to use the least-loaded switch in the network. The authors assume that the correct value is the value seen at the egress switch, which is arguable.

Yoon et al. [26] propose a comparable approach to ours. They use centrality metrics from graph theory to determine the relative importance of nodes. In addition, with packet sampling based on a probabilistic rate, they improve the efficiency for an intrusion detection system measuring statistics at *more important* switches. They argue that the traffic at the edges of the network might have different characteristics than in the networks core, which the centrality metrics that reflect only the physical topology rather than the actual traffic do not cover.

Other network management fields also consider the centrality of switches to improve the performance: In [3], Challa et al. propose a routing algorithm that weights nodes and edges based on their temporal betweenness centrality to balance load better in the network and optimize the utilization. The field of wireless sensor networks provides another example for load balancing based on centrality metrics. Cuzzocrea et al. [6] propose an algorithm to enrich the QoS throughput using the edge betweenness centrality.

B. Flow Size Distribution Measurement Approaches

The exact determination of the Flow Size Distribution (FSD) is straightforward using traffic mirroring. However, covering the whole traffic requires excessive resources, thus, is not reasonable. Consequently, a number of works propose FSD estimation methods based on packet/flow sampling.

Based on sampled packets flows length are roughly reconstructed, thus, the FSD is estimated. However, Duffield et al. [7] show how that utilizing packet sampling leads to inaccuracies, in particular as sampling does not cover small flows in many cases. They propose to consider also packet information, such as TCP Flags (e.g. SYN), which leads to better results. Hohn et al. [12] further investigated flow sampling, thus, sampling whole flows instead of independent packets, which leads to further improvements regarding the FSD estimation. Nevertheless, the feasibility of such solutions is strictly limited due to resource exhaustive lookups required in the forwarding devices. Taking also the sequence number into account to guess the flow length, Ribeiro et al. [21] also show improvements in contrast to pure packet sampling, however, did not compare their solution to flow sampling. Tune et al. [25] show how these concepts can be combined and which different sampling methods exist with their advantages and disadvantages with regard to the estimation of the FSD.

With the advent of SDN, new techniques with reasonable costs to collect flow length information occurred making packet or flow sampling obsolete. SDN-based flow counter techniques allow a trivial calculation of the FSD assuming a direct correlation between rules and flows (cf. Section II).

VII. Conclusion

This work targets to overcome disproportional resource consumption in order to monitor the full network state in SDNs. We propose mechanisms for spatial sampling, in particular placing a small number of measurement points in the network instead of measuring on every network switch. Doing so, we argue that the estimation of the full network state is significantly simplified if the selection is based on the nodes *importance* and *informativeness*, respectively. Depending on different network knowledge, we propose two schemes for this: *(i)* taking, if available, flow knowledge into account to maximize the number of covered flows with the available number of measurement points; or *(ii)* use centrality metrics known from graph theory to estimate the relevance of the nodes and, thus, estimate the information contribution the different nodes can provide. We demonstrate the effectiveness of such a selection with a handy mechanism to estimate the Flow Size Distribution (FSD) within a network leveraging OpenFlow's FlowRemoved messages.

Rhaban Hark, Mohamed Ghanmi, Sounak Kar, Nils Richerzhagen, Amr Rizk, and Ralf Steinmetz. *Representative Measurement Point Selection to Monitor Software-defined Networks (accepted for publication)* In: Proceedings of the 43nd IEEE Conference on Local Computer Networks (LCN), p: 1–8, IEEE, October 2018.

Using network emulations, we demonstrate how to gather the network state effectively using spatial sampling. We measure the estimation accuracy using the relative distance between the estimated and the true FSD using the *Bhattacharyya distance* [1]. We find that the centrality metrics provide good estimates, especially compared to the assumption of available routing information, thus, when we can directly maximize the number of covered flows. Furthermore, we find only small differences between the selection methods within different topologies that correspond to real-world topologies taken from the Internet Topology Zoo [14]. Nevertheless, it is observable that the accuracy does not linearly relate to the measurement costs when increasing the number of placed measurement points.

The current OpenFlow 1.5 protocol, as well as the promising P4 language provide further technologies, to sample signal packages such as TCP SYN and ACK. Relying on such mechanisms, we will investigate the detection of sub-flows within aggregating flow rules in follow-up works.

References

[1] A. Bhattacharyya, "On a Measure of Divergence Between two Multinomial Populations," *Sankhyā: The Indian Journal of Statistics*, vol. 7, no. 4, pp. 401–406, 1946.

[2] S. P. Borgatti, "Identifying sets of key players in a social network," *Computational & Mathematical Organization Theory*, vol. 12, no. 1, pp. 21–34, 2006.

[3] R. Challa, S. Jeon, D. S. Kim, and H. Choo, "CentFlow: Centrality-Based Flow Balancing and Traffic Distribution for Higher Network Utilization," *IEEE Access*, vol. 5, pp. 17 045–17 058, 2017.

[4] S. R. Chowdhury, M. F. Bari, R. Ahmed, and R. Boutaba, "PayLess: A Low Cost Network Monitoring Framework for Software Defined Networks," in *Proceedings of the IEEE Network Operations and Management Symposium*, 2014, pp. 1–9.

[5] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, "DevoFlow: Scaling Flow Management for High-performance Networks," in *Proceedings of the ACM SIGCOMM Conference*, 2011, pp. 254–265.

[6] A. Cuzzocrea, A. Papadimitriou, D. Katsaros, and Y. Manolopoulos, "Edge betweenness centrality: A novel algorithm for QoS-based topology control over wireless sensor networks," *Journal of Network and Computer Applications*, vol. 35, no. 4, pp. 1210–1217, 2012.

[7] N. Duffield, C. Lund, and M. Thorup, "Estimating Flow Distributions from Sampled Flow Statistics," *IEEE/ACM Transactions on Networking*, vol. 13, no. 5, pp. 933–946, 2005.

[8] M. G. Everett and S. P. Borgatti, "The Centrality of Groups and Classes," *The Journal of Mathematical Sociology*, vol. 23, no. 3, pp. 181–201, 1999.

[9] L. C. Freeman, "Centrality in Social Networks Conceptual Clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1978.

[10] R. Hark, N. Richerzhagen, B. Richerzhagen, A. Rizk, and R. Steinmetz, "Towards an Adaptive Selection of Loss Estimation Techniques in Software-defined Networks," in *Proceedings of the IFIP Networking Conference*, 2017, pp. 1–9.

[11] R. Hark, D. Stingl, N. Richerzhagen, K. Nahrstedt, and R. Steinmetz, "DistTM: Collaborative Traffic Matrix Estimation in Distributed SDN Control Planes," in *Proceedings of the IFIP Networking Conference*, 2016, pp. 82–90.

[12] N. Hohn and D. Veitch, "Inverting Sampled Traffic," in *Proceedings of the ACM SIGCOMM Conference on Internet Measurement*, 2003, pp. 222–233.

[13] M.-S. Kim, H.-J. Kong, S.-C. Hong, S.-H. Chung, and J. W. Hong, "A Flow-based Method for Abnormal Network Traffic Detection," in *2004 IEEE/IFIP Network Operations and Management Symposium*, vol. 1, 2004, pp. 599–612.

[14] S. Knight, H. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The Internet Topology Zoo," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1765 – 1775, 2011.

[15] A. Kumar, M. Sung, J. J. Xu, and J. Wang, "Data Streaming Algorithms for Efficient and Accurate Estimation of Flow Size Distribution," *SIGMETRICS Performance Evaluation Review*, vol. 32, no. 1, pp. 177–188, 2004.

[16] B. Lantz, B. Heller, and N. McKeown, "A Network in a Laptop: Rapid Prototyping for Software-defined Networks," in *Proceedings of the ACM SIGCOMM Workshop on Hot Topics in Networks*, 2010, pp. 19:1–19:6.

[17] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[18] M. Moshref, M. Yu, R. Govindan, and A. Vahdat, "SCREAM: Sketch Resource Allocation for Software-defined Measurement," in *Proceedings of the ACM Conference on Emerging Networking Experiments and Technologies*, 2015, pp. 14:1–14:13.

[19] Open Networking Foundation, "Software-defined networking: The new norm for networks," *ONF White Paper*, vol. 2, pp. 2–6, 2012.

[20] B. Pfaff, B. Lantz, and B. Heller, *OpenFlow Switch Specification, Version 1.3.0*, Open Networking Foundation, 2012.

[21] B. Ribeiro, D. Towsley, T. Ye, and J. C. Bolot, "Fisher Information of Sampled Packets: An Application to Flow Size Estimation," in *Proceedings of the ACM SIGCOMM Conference on Internet Measurement*, 2006, pp. 15–26.

[22] A. Roy, H. Zeng, J. Bagga, G. Porter, and A. C. Snoeren, "Inside the Social Network's (Datacenter) Network," in *Proceedings of the ACM SIGCOMM*, 2015, pp. 123–137.

[23] S. Scott-Hayward, S. Natarajan, and S. Sezer, "A Survey of Security in Software Defined Networks," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 623–654, 2016.

[24] A. Tootoonchian, M. Ghobadi, and Y. Ganjali, "OpenTM: Traffic Matrix Estimator for OpenFlow Networks," in *Proceedings of the Springer Conference on Passive and Active Measurement*, 2010, pp. 201–210.

[25] P. Tune and D. Veitch, "Fisher Information in Flow Size Distribution Estimation," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 7011–7035, 2011.

[26] S. Yoon, T. Ha, S. Kim, and H. Lim, "Scalable traffic sampling using centrality measure on software-defined networks," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 43–49, 2017.

[27] C. Yu, C. Lumezanu, Y. Zhang, V. Singh, G. Jiang, and H. V. Madhyastha, "FlowSense: Monitoring Network Utilization with Zero Measurement Cost," in *Proceedings of the Passive and Active Measurement Conference*, 2013, pp. 31–41.

[28] Y. Zhang, "An Adaptive Flow Counting Method for Anomaly Detection in SDN," in *Proceedings of the ACM Conference on Emerging Networking Experiments and Technologies*, 2013, pp. 25–30.