The Joint Optimization of Online Traffic Matrix Measurement and Traffic Engineering For Software-Defined Networks

Xiong Wang, *Member, IEEE,* Qi Deng, Jing Ren, Mehdi Malboubi, Sheng Wang, Shizhong Xu, and Chen-Nee Chuah, *Fellow, IEEE*

***Abstract*—Software-Defined Networking (SDN) provides programmable, flexible and fine-grained traffic control capability, which paves the way for realizing dynamic and high-performance traffic measurement and traffic engineering. In the SDN paradigm, the traffic forwarding and measurement strategies are realized through flow tables stored in the Tenantry Content Addressable Memories (TCAM) of SDN switches. However, the number of TCAM entries in SDN switches is limited. In this paper, we aim to jointly optimize the Traffic Matrix Measurement (TMM) and Traffic Engineering (TE) process under the TCAM capacity and flow aggregation constraints in software-defined networks. We first formulate the joint optimization problem as a Mixed Integer Linear Programming (MILP) model. Then to get an initial traffic matrix for the joint optimization problem, we propose a simple flow rule generation strategy named Maximum Load Rule First (MLRF) to efficiently generate feasible flow rules, which are used to provide direct measurements for the traffic matrix measurement problem. At last, to solve the joint optimization efficiently, we propose two efficient heuristic algorithms named Traffic Matrix Measurement First (TMMF) and Traffic Engineering First (TEF), respectively. TMMF and TEF can generate feasible flow rules for realizing TMM and TE strategies. Our evaluations on real network topologies and traffic traces verify that by jointly optimizing the TMM and TE strategies, both TMMF and TEF can significantly improve TMM accuracy and TE objective (i.e., load balancing) with limited TCAM resource.**

***Index Terms*—Traffic matrix, traffic matrix measurement, traffic engineering, software-defined networking, flow rule.**

I. Introduction

TRaffic Engineering (TE) is to put the traffic where the network bandwidth available. TE is an efficient way to improve network performance and guarantee the QoS requirements of network users. With the rapid growth of Internet traffic, the TE problem has attracted extensive attention during the past decades [1]. To achieve good traffic optimization performance, TE schemes require an accurate and timely measurement of traffic volumes exchanged between the source and destination nodes/IP-prefixes pairs in a network.

Traffic volumes exchanged between node pairs in a network can be summarized in the form of a 2-dimensional matrix, which is usually called as Traffic Matrix (TM). The Traffic Matrix Measurement (TMM) also has attracted extensive attention from the research community in the past decades [2]. However, it is still challenging to accurately measure TMs for practical networks. First, direct measurement of TMs on large-scale networks is challenging due to the hard constraints of network measurement resources (e.g., TCAM entries, memory capacity, and processing power). In fact, well-known solutions such as NetFlow and sFlow may consume a large amount of computation and storage resource of network devices [3], and thus may have an impact on the forwarding performance of network devices as the traffic volume continues to rise. Second, indirect methods, which estimate TMs by solving an under-determined problem where the number of measurements is far less than the number of flows, may suffer from high estimation error [2]. Therefore, in order to improve the estimation accuracy, more side information (e.g., the sizes of some large flows [5] or linearly independent link-load measurements [23].) must be incorporated into the problem formulation. However, this is hard to achieve since existing networks lack flexible routing control and fine-grained traffic measurement capability.

In another hand, the advent of Software Defined Networking (SDN) [4] separates the logically centralized control plane from the underlying data plane, which brings potential benefits for TMM and TE. First, the centralized control plane provides a global view of network resources and enables programmable traffic measurement and routing. Moreover, in the data plane, each SDN switch provides several counters for each flow rule in the flow table, and each flow can be forwarded to any port by executing the actions of the corresponding flow rule. Therefore, the SDN networks have the capability of improving the performance of traffic measurement and traffic engineering. The studies in [5], [6] reveal that SDN networks can achieve accurate and timely traffic measurement by carefully designing the flow rules. Moreover, the experiment results in the SDN-enabled networks of Microsoft [7] and Google [8] verify that the SDN networks can achieve near-optimal performance in terms of throughput and link utilization by implementing effective traffic engineering techniques.

However, as it happens with most emerging network ar-

Manuscript received November 20, 2017; revised October 3, 2018 and September 18, 2019 ; accepted November 19, 2019. This work is partially supported by National Key R&D Program of China (2018YFB2100100, 2018YFC0831002) and National Natural Science Foundation of China (61671130, 61701058, 61271165). *(Corresponding author: Xiong Wang.)*

X. Wang, Q. Deng, J. Ren, S. Wang, and S. Xu are with the Key Lab of Optical Fiber Sensing & Communications, School of Information & Communication Engineering, University of Electronic Science & Technology of China, Chengdu 611731, China (Email: wangxiong@uestc.edu.cn).

M. Malboubi was with the Department of Electrical & Computer Engineering, University of California, Davis 95616, USA.

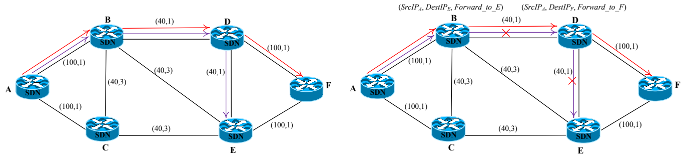
C. Chuah is with the Department of Electrical & Computer Engineering, University of California, Davis 95616, USA.

chitectures and protocols, migration to SDN will not happen overnight. The reason is that upgrading all existing legacy devices to SDN-enabled ones poses the high budget and operational burden, and also raises performance and security risks [9]. Thus, large network providers usually choose to incrementally deploy SDN devices in their existing networks [10], [11]. As a result, hybrid SDN architecture is likely to be a long-term solution for the real operational networks. Therefore, we also consider hybrid SDN networks in this paper.

In SDN networks, both TMM and TE tasks need to use flow rules. Specifically, TMM tasks use flow rules to pick flows for direct measurement, while TE tasks use flow rules to control the forwarding paths for flows. To achieve high-speed forwarding, the flow rules are usually stored in Ternary Content Addressable Memory (TCAM) of SDN switches. However, since the TCAMs are expensive and power hungry, the capacity of TCAMs in an SDN switch is very limited (e.g., commodity SDN switches generally have hundreds to thousands TCAM entries [12]). In contrast, an SDN network may have a huge number of flows. Accordingly, to increase the capacity of network devices while supporting the highspeed forwarding of packets, the latest programmable switching chips (e.g., Trident 4 and Tofino) use the hybrid architecture where flow tables are implemented using both Static Random-Access Memory (SRAM) and TCAM technologies. In these chips, the TCAM allows ternary match type tables, while SRAM flow tables support exact match. However, these chips also have restricted processing ability and storage space. For example, Tofino [14] can process packets up to line rate of 6.5Tbps. But it contains 12 physical stages, and each stage only possesses 1.28MB SRAM+ 67.6KB TCAM. These constraints limit the number of flows rules used to measure and control flows. Therefore, it is still meaningful to optimally use all available resources and ensure the most efficient utilization of TCAM entries.

In this paper, we provide a practical and efficient solution to carefully design the flow rules under the TCAM capacity constraint by jointly considering the TMM and TE objectives, and we aim to propose efficient, feasible and scalable TMM and TE optimization strategies. Here, we say TMM and TE strategies are feasible if they satisfy resource and flow aggregation constraints. We assume that to save TCAM entries, the flow rules in each SDN switch are initially aggregated. In theory, the TM can be estimated based on the statistics of these aggregated rules, and the traffic routing also can be adjusted by modifying the forwarding actions of the aggregated rules. However, to improve the performance of TMM and TE, we generate new rules by disaggregating the aggregated rules and install the new rules in available TCAM entries of each SDN switch. The controller collects the measurement statistics of TCAM entries periodically, estimates TM based on these statistics, and design flow rules according to the estimated TM. To the best of our knowledge, we make the first attempt to jointly optimize TMM and TE performance in the SDN paradigm. In this paper, we tackle the problem and make the following contributions.

1) We formulate the joint optimization problem of TMM and TE as a Mixed Integer Linear Programming (MILP) problem.



(a) (b)

Fig. 1: Example for the joint optimization of TMM and TE

2) We propose a simple flow rule generation algorithm named Maximum Load Rule First (MLRF) [21] to efficiently generate feasible flow rules, which are used to provide direct measurements for the TM estimation in the initial stage.

3) To solve the joint optimization problem efficiently, we respectively propose two efficient algorithms named TMM First (TMMF) and TE First (TEF) to design flow rules for TMM and TE tasks. TMMF initially generates flow rules to directly measure as many large flows as possible, and then it determines the forwarding actions of the rules for the directly measured large flows by considering the TE objective. While the TEF first generates flow rules to adjust the forwarding paths of some flows such that the TE objective is optimized, and then it uses the rest of the available TCAM entries to measure the large flows that are not directly measured.

4) We evaluate the performance of TMMF and TEF using traffic traces from real ISP networks. The results verify that both TMMF and TEF can achieve good performance for TMM and TE.

II. Motivations and Related Works

A. Motivations

In SDN networks, TMM task uses flow rules to pick flows for direct measurement, while TE task uses flow rules to adjust the forwarding ports of the flows by executing different forwarding actions. Let us first consider an example in Fig. 1(a). In Fig. 1(a), the numbers on the links denote the link capacities and link weights, respectively. We assume that to save TCAM entries, the routing rules for the flows are aggregated based on the destination IP prefixes. The red and purple dashed lines represent the default routes (shortest paths) for flows f_{AF} and f_{AE}, respectively. In Fig. 1(a), if SDN switches have sufficient TCAM entries, the two flows f_{AF} and f_{AE} can be directly measured at any of the switches traversed by them, and the forwarding ports of the two flows can also be changed at the switches A, B and D. However, the TCAM entries are scare resource in SDN switches, and each SDN switch has a very limited number of TCAM entries. Therefore, under the TCAM capacity constraint, we need to optimize the TCAM usage by allocating the right TCAM entries to TMM and TE tasks. In other words, we need to jointly optimize the flow rules design for TMM and TE tasks.

We illustrate the joint optimization of TMM and TE in Fig. 1(b). We assume that the bandwidth requirements of the two flows (f_{AD} and f_{AE}) are 30 and 20 units, respectively. The default routes for flow f_{AF} and f_{AD} are shown with red and purple dashed lines, respectively. We assume only switches B and D have one available TCAM entries, and other switches do not have available TCAM entries. In this example, if the TCAM entry of switch B is allocated to the TMM task for measuring flow f_{AF}, the link (B, D) will be congested. Because the TE task cannot redirect flow f_{AE} to path $A - B - E$ due to lack of TCAM entries at switch B. However, for this example, a better way is to allocate the available TCAM entry of switch B to flow f_{AE} and allocate the available TCAM entry of switch D to flow f_{AF}. In this case, the flow rules ($SrcIP_A$, $DestIP_E$, $Forward\_to\_E$) and ($SrcIP_A$, $DestIP_F$, $Forward\_to\_F$) are respectively generated and installed on switches B and D, where $SrcIP$ denotes the source IP and $DestIP$ denotes the destination IP. By using the two flow rules, flows f_{AE} and f_{AF} can be measured at nodes B and D, respectively, and the route of flow f_{AE} can also be adjusted to $A - B - E$.

B. Related Works

Recently, some research efforts have been made to develop traffic measurement frameworks in the SDN paradigm. To support various measurement tasks, OpenSketch [16] introduces a variety of hash-based sketches, and can configure the sketches dynamically. However, OpenSketch [16] assumes specialized hardware support on switches for traffic measurement. In order to avoid using custom hardware for traffic measurement, [17] and [18] propose practical traffic measurement solutions running on commodity SDN switches, and [6] extends the work in [17], [18] by enabling concurrent and dynamic instantiating traffic measurement tasks. However, all the solutions proposed in [6], [16]–[18] are targeted for measuring a specific set of flows (e.g, heavy hitters), and they are not suitable for TMM.

OpenTM [19], DCM [20], and iSTAMP [5] aim to measure TMs in SDN. OpenTM and DCM are per-flow based measurement solutions, which directly measure TM by keeping track of statistics for each flow. OpenTM and DCM are not scalable since the measurement resources (e.g., TCAM) are limited while the number of flows is large. To meet the constraints on measurement resources and improve measuring accuracy, iSTAMP infers TMs based on both aggregated and the k largest per-flow measurements. iSTAMP seems to make a good tradeoff between measurement resources and accuracy, but it also faces the following issues. First, iSTAMP omits the flow aggregation constraints, leading to infeasible aggregated measurements. Second, to find out the k largest flows, iSTAMP uses all of the TCAM entries to measure all individual flows over multiple time intervals, which will introduce non-negligible measurement cost. To overcome the drawbacks of iSTAMP, [21], [22] propose TM measurement schemes considered flow aggregation and TCAM capacity constraints for SDN networks. Recently, [23], [24] investigates the TM measurement problem in SDN capable data center networks. The infeasibility issue of traffic aggregation is considered in [23] based on the assumption that the traffic measurement is only taking place at the ToR SDN switches. The assumption makes the method proposed in [23] hard to apply in general networks. In addition, the complexity of choosing feasible aggregation paths in [23] is also high for large-scale networks. To reduce the high-speed memory consumption and avoid incurring high computation overhead in SDN switches, FlowRadar [24] encodes flows and their counters using hash-based approach and decodes the flow sizes by leveraging the computing power at the remote servers. However, FlowRadar also requires specialized hardware support on switches for traffic measurement.

The SDN based TE is first applied in data center networks. Hedera [25] and MicroTE [26] are SDN based TE approaches proposed for data center network. To efficiently utilize network resources, Hedera [25] uses Equal-Cost Muti-Path (ECMP) for short-lived flows but uses a centralized approach to explicitly route large flows. MicroTE [26] optimizes the traffic routing based on the short-term and partial predictability of the TM. Google and Microsoft implement SDN based TE approaches called as SWAN [7] and B4 [8] respectively for their Wide Area Networks (WAN). The experiments conducted by Google and Microsoft [7], [8] verify that the SDN based TE can achieve near-optimal performance in the aspects of throughput and link utilization.

The studies in [7], [8], [25], [26] assume that the networks are the full SDN network, where network nodes are SDN-enabled. The TE problem in hybrid SDN network attracts more attention [27]–[33] in recent years. Agarwal et al. [27] first study the TE problem in hybrid SDN networks, where they propose a polynomial time algorithm to optimize the traffic routing on admissible paths. In [28], they improve the TE performance of hybrid SDN networks by introducing an enhanced routing protocol in hybrid SDN networks. Guo et al. [29] optimize the OSPF weights and flow splitting ratio of the SDN nodes to achieve better TE performance. To avoid routing inconsistency and achieve high network utilization, Wang et al. [30] propose an efficient approach to construct forwarding graphs and optimize traffic routing on the forwarding graphs. In hybrid SDN networks, the placement of SDN switches significantly affect the TE performance. Caria et al. [31] propose an algorithm to optimize the sequence of nodes for SDN upgrade by considering the TE performance. To improve the utilization of SDN devices, Xu et al. [32] study the joint optimization of incremental SDN placement and flow routing decisions on the SDN switches. In addition, Zhao et al. [33] design TE approach in the hierarchical control plane for multi-domain and multi-layer networks. Although the TE in SDN networks attracts a lot of attention, the existing studies do not consider the TCAM capacity constraint when implementing TE in SDN networks.

In summary, the TMM and TE of SDN networks have attracted much research interest in recent years. However, most of the existing solutions have shortcomings in the aspects of feasibility and scalability. Most importantly, none of the previous studies has considered the joint optimization of TMM and TM under the TCAM capacity constraint.

III. The System Model and Problem Formulation

A. System Model and Assumptions

Since deploying SDN devices incrementally is a natural choice for network providers [9], we also consider the hybrid SDN networks, where only a subset of the nodes are SDN switches and the rest of the nodes are traditional routers. We assume the set of nodes deploying with SDN switches are given. The rationality behind this assumption is twofold: 1) Joint optimization of SDN device deployment strategy and traffic management strategy is so difficult and complicated that network operators prefer to consider the two problems separately [10], [11], [27]; 2) The SDN deployment decision is usually made based on the predicted traffic patterns, and thus to achieve desirable network performance, the traffic management strategy should be optimized according to the real traffic pattern and the SDN deployment solution [11], [27]; 3) The existing studies [10], [11] on SDN device deployment problem usually maximize programmable traffic or TE flexibility based on predicted traffic patterns, and how to optimize the traffic management policy is not mentioned in these studies.

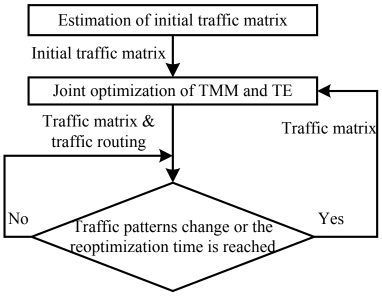
We assume that the network operators will assign a set of IP prefixes to each node, and this mapping is known a priori. For simplicity, we assume that a flow is indicated by a source and destination IP prefixes pair (src_prefix, dst_prefix), where src_prefix/dst_prefix is one of the prefixes assigned to source-node/destination-node. However, the approaches proposed in this paper can also be used to the scenarios, where the flows are flexibly defined by 12-tuple of packet headers supported by OpenFlow specification. The joint optimization system for TMM and TE contains two parts. In the data plane, the TCAMs in SDN switches match, count, and forward packets with wildcard rules. In the control plane, the controller: 1) fetches flow statistics (TCAM counters and link loads), 2) estimates TM based on the statistics, 3) dynamically designs flow rules based on the estimated TM and TE objective, and 4) installs the new rules in the SDN switches. Since TCAMs are expensive and power hungry, the SDN switches have a limited number of TCAM entries. We assume that part of the TCAM entries in each SDN switch are used to implement default routing for flows. To save TCAM entries, the default routing rules are aggregated based on the destination IP prefixes. Without loss of generality, we assume that the flows are routed along shortest path in default.

B. Problem Formulation

To achieve optimal network performance, the TMM and TE strategies should be adjusted according to the current traffic patterns and traffic distribution. Therefore, the current TM is a necessary input for the joint optimization of TMM and TE. Fig. 2 shows the framework for the joint optimization of TMM and TE. As shown in Fig. 2, an initial TM is first estimated and fed to the joint optimization algorithm, which allocates TCAM resources and generates flow rules for TMM and TE tasks by jointly considering their objectives. The network controller will repeatedly invoke the joint optimization algorithm for a fixed time period or when the traffic patterns change.

(1) Traffic Matrix Estimation

We can model the network as a directed graph $G = (V, L)$, where V and L are the sets of nodes and links, respectively. Each link $l \in L$ is associated with a capacity c_l and a routing weight w_l. Let $V_{SDN} \subseteq V$ denote the set of SDN nodes and $V_{NSDN} = V \backslash V_{SDN}$ denote the non-SDN nodes. Let n_i and m_i be the total number of TCAM entries and the number of available (i.e. unused or reserved) TCAM entries in SDN node i ($i \in V_{SDN}$), respectively. Let R_i be the set of flow rules of SDN node i. Y_S denotes the vector of TCAM statistics, and Y_L denotes the vector of link loads. For ease of formulation, we use a vector $X \in R^N$ to represent the vector representation of TM, where N is the number of flows. Y_S and Y_L have the following relationship with X



Estimation of initial traffic matrix
Initial traffic matrix
Joint optimization of TMM and TE
Traffic matrix & traffic routing
Traffic matrix
No
Traffic patterns change or the reoptimization time is reached
Yes

Fig. 2: The framework for the joint optimization of TMM and TE

$$Y_S = A_S X \quad and \quad Y_L = A_L X, \tag{1}$$

where $A_S = (A_S^{ij})$ and $A_L = (A_L^{ij})$ are binary aggregation matrices. The element $A_S^{ij} \in \{0, 1\}$ indicates whether flow j is forwarded by rule i, and the element $A_L^{ij} \in \{0, 1\}$ indicates whether flow j is going through link i. A_L is given and it is fixed while A_S is determined by the flow rules designed by the controller. Having measurements Y_S and Y_L as well as aggregation matrices A_S and A_L, the TM X can be estimated using the following optimization formulation (2), which is a convex optimization problem that is effective for estimating highly fluctuating network flows [5].

$$\begin{aligned} \hat{X} = &\underset{X}{\text{minimize}} \, \|Y_L - A_L X\|_2^2 + \|Y_s - A_s X\|_2^2 + \lambda \, \|X\|_1 \\ \text{s.t.} \quad & \qquad X \geq 0, \end{aligned} \tag{2}$$

where $\hat{X}$ is the estimated TM, and λ is the weighting factor for $\|X\|_1$. Considering the optimization formulation (2), we can improve the estimation accuracy by generating a more informative Y_s via designing a better aggregation matrix A_S. Since A_S is determined by the measurement rules installed in the SDN switches, a better A_S can be realized by installing traffic measurement rules on the available TCAM entries. To this end, we generate measurement rules by disaggregating the default routing rules (i.e., use some rules with longer prefixes to offload the traffic flows from the rules with shorter prefixes), and install the newly generated measurement rules in the available TCAM entries.

On the other hand, it has been shown that in real networks, a small number of large flows may account for more than 80% the traffic volume [34]. The previous studies have revealed that accurately measuring the large flows yields the best improvement of overall TMM performance [5], and optimizing the forwarding paths for large flows can achieve near-optimal TE performance [25]. Therefore, accurately measuring the size of large flows is essential for both TMM and TE tasks. Accordingly, we first need to identify large flows.

F	The set of flows
P_i	The set of candidate paths for flow f_i, and we assume that P_i is pre-computed for each flow f_i.
$\|f_i\|$	The size of flow f_i.
t	A decision variable denotes the maximum link utilization of the network.
u_i	A binary decision variable denotes whether flow f_i is directly measured.
ρ_{ij}	A binary decision variable indicates whether flow f_i goes through SDN node j.
ω_{ij}	A binary decision variable indicates whether flow f_i is directly measured at node j.
θ_{ij}	A binary decision variable indicates whether flow f_i is forwarded by a dedicated flow rule at node j.
δ_{ip}	A binary decision variable indicates whether flow f_i chooses the pth candidate path from its candidate path set P_i.
y_l	A decision variable denotes the load of link l.
C	A large constant.
ξ_{ip}^j	A binary constant indicates whether the pth candidate path of flow f_i goes through node j.
γ_{ip}^j	A binary constant indicates whether the pth candidate path of flow f_i needs to use a flow rule at SDN node j.
η_{ip}^l	A binary constant indicates whether the pth candidate path of flow f_i goes through link l.
α	A weighting factor for the optimization objective of TMM.
β	A weighting factor for the optimization objective of TE.
λ	the weighting factor for $\|X\|_1$.

TABLE I: The notations used in the formulation

(2) The Joint Optimization of TMM And TE

As presented in the previous subsection, the performance of TMM and TE can be improved significantly by accurately measuring the sizes of large flows and optimizing their forwarding paths. Therefore, the joint optimization of TMM and TE is trying to directly measure more large flows and improve the TE objective by adjusting the forwarding paths of the large flows under the TCAM capacity constraint. Jointly optimizing TMM and TE requires the TM to be known. However, the TM is not known at the initial stage. For ease of description, we first assume that the initial TM is given when we formulate the problem, then we will present how to estimate the TM at the initial stage in the next section.

The notations used in the formulation are summarized in Table I. To improve the TM estimation accuracy, the optimization objective of TMM is maximizing the total volume of the directly measured flows. To avoid network congestion, the optimization objective of TE is to minimize the maximum link utilization. Thus, the objective for the joint optimization problem can be formulated as:

$$maximize \quad \alpha \cdot \sum_{f_i \in F} u_i \cdot |f_i| - \beta \cdot t \tag{3}$$

To accommodate the flows in the network, each flow f_i must be carried on one of its candidate paths:

$$\sum_{p \in P_i} \delta_{ip} = 1 \quad \forall f_i \in F \tag{4}$$

If flow f_i goes through SDN node j, ρ_{ij} equals 1, otherwise, and it equals 0. So we have the following constraints:

$$\sum_{p \in P_i} \delta_{ip} \xi_{ip}^j = \rho_{ij} \quad \forall f_i \in F, j \in V_{SDN} \tag{5}$$

A flow f_i can be directly measured at SDN node j only if it goes though SDN node j.

$$\rho_{ij} \geq \omega_{ij} \quad \forall f_i \in F, j \in V_{SDN} \tag{6}$$

If flow f_i is directly measured at any of the SDN nodes, u_i must be equal to 1, otherwise, it must be equal to 0. This is expressed as:

$$\sum_{j \in V_{SDN}} \omega_{ij} \leq C \cdot u_i \quad \forall f_i \in F \tag{7}$$

$$\sum_{j \in V_{SDN}} \omega_{ij} \geq u_i \quad \forall f_i \in F \tag{8}$$

Moreover, a flow rule is required at SDN node j if a flow f_i will be forwarded on a link (j, h) that is not on the default route (e.g., shortest path) of the flow f_i. In this case, a flow rule in SDN node j is used to adjust the forwarding path for flow f_i, and θ_{ij} must be equal to 1.

$$\sum_{p \in P_i} \gamma_{ip}^j \delta_{ip} \leq \theta_{ij} \quad \forall f_i \in F, j \in V_{SDN} \tag{9}$$

To ensure that all the flow rules used for TMM and TE can be realized, the following TCAM capacity for each SDN node must be satisfied.

$$\sum_{f_i \in F} (\omega_{ij} + \theta_{ij}) \leq m_j \quad \forall j \in V_{SDN} \tag{10}$$

At last, we have the link utilization constraint:

$$\sum_{f_i \in F} \sum_{p \in P_i} \eta_{ip}^l \delta_{ip} \leq c_l \cdot t \quad \forall l \in L \tag{11}$$

The above joint optimization problem can be easily proved to be NP-hard. Given that $\alpha = 0$, $V_{SDN} = V$ and $m_i = |F|$ (for $\forall i \in V$), the joint optimization problem is a multi-commodity flow problem with non-bifurcation constraint, which has been proved to be NP-hard [35]. Hence, to efficiently solve the joint optimization problem in the large-scale network, we propose two two-phase heuristic algorithms in Section V.

IV. Estimation of Initial Traffic Matrix

As introduced in Section III. B, the joint optimization of TMM and TE relies on an initial TM as the critical input. Initially, the flow rules installed in the TACM of each SDN switch are used to realize default routing for the flows. In practical networks, the rules used for routing are usually aggregated to save TCAM space (e.g., the rules for routing the flows to the same prefix can be aggregated into one rule). In theory, the TM can be estimated based on the statistics of these aggregated default routing rules. However, due to the under-determined nature of TM inference problems [5], the direct estimation of initial TM based on the statistics of those aggregated routing rules may suffer from significant estimation errors. Hence, in order to improve the estimation accuracy of initial TM, we can generate additional rules to measure the flows under the TCAM capacity and flow aggregation constraints. In this section, we will present the proposed traffic measurement rule generation strategy called Maximum Load Rule First (MLRF).

For a flow (defined by a source and destination prefixes pair) going through SDN switch u, the controller can easily find out the flow rule matching the flow in SDN switch u by simply checking each rule installed in SDN switch u. Since the prefixes owned by a node are known (see Section III.A), the network operators can get the set of flows in their networks (there is a flow between each pair of prefixes). Thus, given the set of flows and the default routes of the flows, the number of flows matching each rule in an SDN switch can be easily computed. We assume that the total rate of the flows hitting a flow rule is proportional to the number of flows hitting the flow rule. We define the load of a rule in an SDN switch as the number of flows matching the rule at the SDN switch. Thus, two flow rules, which matches the same number of flows, are assumed to have the same loads.

The detailed procedures of MLRF are described in **Algorithm 1**. The basic idea of MLRF is trying to generate a new flow measurement rule that can offload half the load from the rule with the maximum load in an SDN switch in each step. MLRF first greedily selects the rule with the maximum load in an SDN switch, and then based on the selected rule (we call it old rule below), it generates a new rule with a higher priority and a longer source IP prefix. It is notable that except the priority and the source IP prefix fields, all other fields for the new rule are the same as the old rule (lines 7, 8, 21 and 25 in **Algorithm 1**). Evidently, if the new rule is added into the SDN switch, some of the flows matching the old rule will be offloaded to the new rule. The load of the new rule is determined by its source IP prefix. MLRF tries to choose a source IP prefix for the new rule such that the load of the new rule and the old rule are balanced. To do that, MLRF searches the prefix trie of source IPs using width first strategy (lines 12 - 33 in **Algorithm 1**).

Algorithm 1 The Maximum Load Rule First Measurement Rule Generation Strategy

```
Input: Network topology $G(V, L)$.
Output: The rule sets $R$ for the SDN switches.
 1: $R \leftarrow \emptyset$
 2: for each node $s \in V_{SDN}$ do
 3:     add the routing rules in node $s$ to set $R_s$
 4:     compute the load of each rule $r_s \in R_s$ and the set of flows
        matching the rule $r_s$
 5:     while $|R_s| < n_s + m_s$ do
 6:         $r_{old} \leftarrow$ the rule with the maximum load in $R_s$
 7:         $r_{new} \leftarrow r_{old}$
 8:         $r_{new}.priority \leftarrow r_{old}.priority + 1$
 9:         $l_{old} \leftarrow load(r_{old})$   //$load(r)$ denotes the load of rule $r$
10:         $\Delta_{min} \leftarrow \frac{1}{2} \cdot l_{old}$   // $\Delta_{min} = 0$ represents that the loads
            of $r_{new}$ and $r_{old}$ are balanced
11:         $r_{temp} \leftarrow r_{new}$
12:         while $load(r_{temp}) > \frac{1}{2} \cdot l_{old}$ do
13:             $pre_{src} \leftarrow r_{temp}.src\_prefix$
14:             $pre^{L}_{src} \leftarrow$ left child of $pre_{src}$ on the prefix trie
15:             $pre^{R}_{src} \leftarrow$ right child of $pre_{src}$ on the prefix trie
16:             $r^{L} \leftarrow r_{new}$
17:             $r^{R} \leftarrow r_{new}$
18:             $r^{L}.src\_prefix \leftarrow pre^{L}_{src}$
19:             $r^{R}.src\_prefix \leftarrow pre^{R}_{src}$
20:             if $\Delta_{min} > |load(r^{L}) - \frac{1}{2} \cdot l_{old}|$ then
21:                 $r_{new}.src\_prefix \leftarrow pre^{L}_{src}$
22:                 $\Delta_{min} \leftarrow |load(r^{L}) - \frac{1}{2} \cdot l_{old}|$
23:             end if
24:             if $\Delta_{min} > |load(r^{R}) - \frac{1}{2} \cdot l_{old}|$ then
25:                 $r_{new}.src\_prefix \leftarrow pre^{R}_{src}$
26:                 $\Delta_{min} \leftarrow |load(r^{R}) - \frac{1}{2} \cdot l_{old}|$
27:             end if
28:             if $load(r^{L}) > load(r^{R})$ then
29:                 $r_{temp} = r^{L}$
30:             else
31:                 $r_{temp} = r^{R}$
32:             end if
33:         end while
34:         $R_s \leftarrow R_s \cup r_{new}$
35:         update the loads of the rules $r_{old}$ and $r_{new}$, and update the
            sets of flows matching the rules $r_{old}$ and $r_{new}$.
36:     end while
37:     $R \leftarrow R_s \cup R$
38: end for
39: return $R$
```

V. The Joint Optimization Algorithms

In this section, we propose two heuristic algorithms to efficiently tackle the joint optimization problem of TMM and TE. The two algorithms are called TMM First (TMMF) and TE First (TEF), respectively.

A. The TMMF Algorithm

It has been shown that in real networks, a small number of large flows may account for more than 80% the traffic volume [34]. Accurately measuring the large flows can yield the best improvement of TM estimation performance [5]. Therefore, TMMF first tries to directly measure the maximum total volume of flows by using available TCAM entries (traffic measurement optimization), then it adjusts the forwarding actions of the rules for the large flows by considering the TE objective (flow routing optimization). It is notable that different from heavy hitter detection algorithms, TMMF maximizes the total volume of the directly measured flows rather than tracking the top-k heavy hitters lows under the TCAM capacity constraint.

Thus, how to find out the expected large flows is important for TMM. To solve this problem, iSTAMP [5] uses a two-phase approach. In the first phase, iSTAMP sequentially measures the initial sizes of individual flows over multiple time slots, i.e., only a portion of flows are measured in each time slot due to the TCAM capacity constraint. In the second phase, iSTAMP measures the k largest flows and estimates TM based on the large flow and aggregation flow measurements. In iSTAMP, measuring the per-flow sizes in the first stage is costly and time-consuming, especially when the available TCAM entries are limited and the number of flows is large. In order to mitigate the overheads, TMMF estimates the per-flow sizes based on the statistics of the rules generated by MLRF. Although the estimated per-flow sizes may not accurate, they

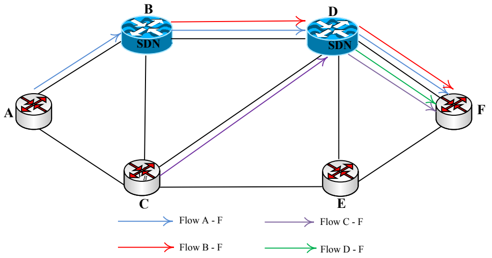
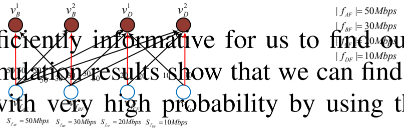


Fig. 3: Illustration of flow measurement allocation



are sufficiently informative for us to find out the large flows. The simulation results show that we can find out the real large flows with very high probability by using the estimated per-flow sizes.

(1) Traffic Measurement Optimization

In SDN networks, a flow may go through several SDN switches. Thus, allocating which SDN switch to measure an interested large flow is also an important problem, which is called Flow Measurement Allocation (FMA) in this paper. The solution of FMA has an impact on the measurement results. Let us consider the example in Fig. 3. There are four flows: $A-F$, $B-F$, $C-F$, and $D-F$. The routes of the flows are indicated by lines with different colors. We assume that both SDN switches B and D have two available TCAM entries. So if flows $A-F$ and $B-F$ are allocated to be measured at SDN switch D, the flow $C-F$ and $D-F$ cannot be measured. Nevertheless, we can measure flows $A-F$ and $B-F$ at SDN switch B and measure flows $C-F$ and $D-F$ at SDN switch D. In order to achieve the best improvement of overall estimation accuracy, we need to get an optimal solution of FMA. The FMA problem can be formulated to a Mixed Integer Linear Program (MILP) [36]. However, to make our solution more scalable, we proposed an efficient algorithm for solving the FMA problem. For facilitating the discussion of how to find an optimal solution of FMA, we first give the definitions for the feasible solutions and optimal solutions of FMA.

Definition 1 (Feasible solutions of FMA): Given the set of flows $F=\{f_1,f_2,\cdots,f_m\}$ and the set of SDN switches $V_{SDN}=\{v_1,v_2,\cdots,v_k\}$, a solution of FMA is denoted as $\Psi=\{\psi_{f_1}^{v_1},\psi_{f1}^{v_2},\cdots,\psi_{f_i}^{v_j},\cdots,\psi_{f_m}^{v_k}\}$ where $\psi_{f_i}^{v_j}=1$ if flow f_i is allocated to be measured at SDN switch v_j, and $\psi_{f_i}^{v_j}=0$ otherwise. We say an allocation solution is feasible if it satisfies the following constraints.

c1) If $\psi_{f_i}^{v_j}=1$, flow f_i must go through SDN switch v_j.

c2) For $\forall v_j\in V_{SDN}$, $\sum_{f_i\in F}\psi_{f_i}^{v_j}\le m_{v_j}$, where m_{v_j} is the number of available TCAM entries in SDN switch v_j.

c3) For $\forall f_i\in F$, $\sum_{v_j\in V_{SDN}}\psi_{f_i}^{v_j}\le 1$.

Definition 2 (The utility of a feasible solution): The utility of a feasible solution Ψ is denoted by $f(\Psi)$, which is defined as:

$$f(\Psi)=\sum_{v_j\in V_{SDN}}\sum_{f_i\in F}\psi_{f_i}^{v_j}\cdot|f_i|$$

Definition 3 (The optimal solution of FMA): A feasible solution Ψ^* is optimal if it meets the following condition:

For any feasible solution Ψ, $f(\Psi^*)\ge f(\Psi)$.

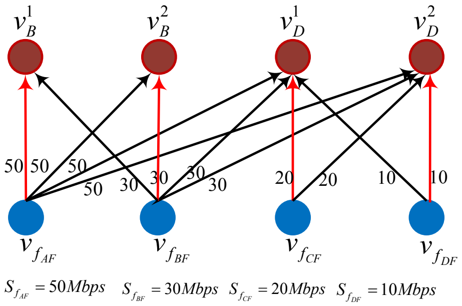


Fig. 4: The auxiliary bipartite graph and a maximum weight matching denoted by red dashed lines.

In order to represent the relationship between flows and SDN switches, we construct an auxiliary bipartite graph. We denote the auxiliary bipartite graph as $G_A(V_A=V_F\cup V_S,L_A)$, where V_A represents the node set and L_A is the link set. Each node $v_{f_i}\in V_F$ corresponds to a flow $f_i\in F$, and each node $v_s^j\in V_s$ corresponds to an available TCAM entry j in SDN switch $s\in V_{SDN}$. If a flow $f_i\in F$ goes through a SDN switch $s\in V_{SDN}$, there is a directed link (v_{f_i},v_s^j) from node v_{f_i} to each node v_s^j $(j\le m_s)$. The weight of the link (v_{f_i},v_s^j) is set to the estimated size of flow f_i. The auxiliary bipartite graph of the example in Fig. 3 is illustrated in Fig. 4.

Theorem 1: *A maximum weight matching of the auxiliary bipartite graph is an optimal solution of the FMA problem.*

Proof: It can be easily verified that a matching of the auxiliary bipartite graph corresponds to a feasible solution of the FMA problem, i.e., the matching satisfies constraints c1) to c3) in ***Definition 1***, and the weight of the matching equals to the utility of the feasible solution. Conversely, a feasible solution of the FMA problem may correspond to a set of matchings of the auxiliary bipartite graph, whose weights equal to the utility of the feasible solution. Because the flows selected by a feasible solution of the FMA problem can be measured by using different TCAM entries (i.e., corresponding to different matchings). For example, in Fig. 4 the matchings denoted by red lines and blue lines respectively correspond to the same feasible solution of FMA. Thus, a maximum weight matching of the auxiliary bipartite graph corresponds to an optimal solution of the FMA problem

Based on the discussions above, the detailed procedures of TMMF for selecting large flows to take direct measurement are shown in Algorithm 2. Since a maximum weight matching of the constructed auxiliary bipartite graph is an optimal flow measurement allocation solution, TMMF will select the flows covered by the maximum weight matching for direct measurement (lines 5-7). The red dashed lines in Fig. 4 denote a maximum weight matching of the auxiliary bipartite graph. In the example, two rules will be generated and installed in node B to measure flow f_{AF} and flow f_{BF}, and two rules will be generated and installed in node D to measure flow f_{CF} and flow f_{DF}.

(2) Traffic routing optimization

In the above TMM optimization problem, the set of directly measured flows is determined. In this step, TMMF optimizes the traffic distribution by adjusting the paths for the flows in

Algorithm 2 Selecting Flows For Direct Measurement in TMMF

```
Input: Network topology $G(V, L)$ and the set $F$ of flows.
Output: The set $F_m$ of flows for direct measurement.
$F_m \leftarrow \emptyset$
estimate the flow sizes based on the statistics of the rules generated by MLRF strategy (Algorithm 1)
sort the flows in $F$ according to their estimated sizes in decreasing order
construct the auxiliary bipartite graph $G_A(V_A = V_F \cup V_S, L_A)$, based on the estimated flow sizes and the routes of the flows
find a maximum weight matching $M$ on $G_A(V_A = V_F \cup V_S, L_A)$
for each link $(v_{f_i}, v_s^j) \in M$ do
    add flow $f_i$ to $F_m$.
end for
return $F_m$
```

F_m under the TCAM capacity constraint. From the Section III. A, we know that all the flows are routed along the shortest paths in default. We assume that the source and destination nodes of flow f_i are s and d, respectively, and sp_{sd} is the shortest path between nodes s and d. If a SDN node u is on the shortest path sp_{sd}, the flow f_i can be forwarded to any feasible neighbour of node u by adding a dedicated flow rule for flow f_i in node u. A neighbor v of SDN node u is feasible for flow f_i if it satisfies the following constraint:

c4) $sp_{su} \cap sp_{vd} = \Phi$ (i.e,. sp_{su} and sp_{vd} do not include the same nodes), where sp_{su} and sp_{vd} are the shortest paths.

The constraint ***c4)*** guarantees that the path $p_i = sp_{su} \cup (u, v) \cup sp_{vd}$ is loop-free. If node v is a feasible neighbor of measurement node u, the path $p_i = sp_{su} \cup (u, v) \cup sp_{vd}$ is called a feasible path for flow f_i. Since the flow f_i can be directly measured at SDN node u if path p_i is select to carry flow f_i, node u is called the measurement node of path p_i. For each directly measured flow $f_i \in F_m$, we add all its feasible paths to the set of candidate paths P_i. Then we select a path to carry each flow $f_i \in F_m$ from the feasible path set P_i such that the maximum link utilization of the links is minimized and the TCAM capacity constraints of the SDN switches are obeyed. The traffic engineering problem can be formulated as follows:

$$Minimize \quad t \tag{12}$$

$$\sum_{p \in P_i} \delta_{ip} = 1 \quad \forall f_i \in F_m \tag{13}$$

$$\sum_{f_i \in F_m} \sum_{p \in FP_i} \gamma_{ip}^{v} \delta_{ip} \leq m_v \quad \forall v \in V_{SDN} \tag{14}$$

$$\frac{\sum_{f_i \in F_m} \sum_{p \in P_i} \eta_{ip}^{l} \delta_{ip} + r_l}{c_l} \leq t \quad \forall l \in L \tag{15}$$

Same as the problem in Section III, constraints (13), (14), and (15) are the demand constraints, TCAM capacity constraints, and link utilization constraints, respectively. In constraints (15), r_l denotes the total volume of flows on link l, which are not directly measured. Although the above formulation is also a MILP problem, the number of variables and constraints is far less than that of the MILP problem in Section III. B. So the above MILP problem can be solved in a short time (just several seconds in our simulations).

By solving the MILP problem, we can obtain the path for each flow in set F_m. To realize the selected paths for the flows in set F_m, we need to install the flow rules generated by Algorithm 3 in the SDN switches. It is notable that the flows in F_i can also be directly measured by the flow rules.

B. The TEF Algorithm

Unlike TMMF, TEF first uses TCAM entries to adjust the routes of some flows such that the TE performance is optimized, and then it uses the rest of the available TCAM entries to measure the large flows that are not measured.

(1) Traffic Routing Optimization

The traffic routing optimization problem in hybrid SDN networks is studied in [27]. In [27], the authors propose a polynomial time algorithm to find admissible paths for flows such that the TE performance is optimized. However, the algorithm cannot be used in the SDN networks with TCAM capacity constraint. To cope with the TCAM capacity constraint, TEF assumes that the feasible candidate paths for each flow are pre-computed. In hybrid SDN networks, a path p from source node s to destination node d is termed feasible if it satisfies the following two constraints:

c5) For each non-SDN node $u \in p$, link (u, v) is on the shortest path from node u to node d, where v is the next node of u on path p.

c6) The path p is loop-free.

We note that constraint ***c5)*** ensures that the next hop to a given destination node at a non-SDN node follows the shortest path routing paradigm. Let P_i denote the set of feasible paths for flow f_i. Given the sets of feasible paths for the flows, the traffic routing optimization algorithm selects a path from the feasible path set to carry each flow such that the TE performance is optimized. However, the complexity of the traffic routing optimization problem increases exponentially with the number of feasible paths. To reduce the complexity, we find at most K feasible paths for each flow. We use an algorithm modified from the K-shortest path algorithm [37] to find at most K feasible paths for each flow. The K-shortest path algorithm generates candidate paths from each node on an existing path. However, in our problem, a non-SDN node cannot forward a flow to a neighbor that is not on the shortest path from the non-SDN node to the destination node of the flow. Thus, the modified algorithm only generates candidate paths from the SDN nodes on an existing path. To ensure that the returned paths are feasible, infeasible paths generated by the modified algorithm will be ignored.

Given the feasible path sets for flows, the traffic routing optimization problem can also be formulated to a MILP problem as follows.

$$Minimize \quad t \tag{16}$$

$$\sum_{p \in P_i} \delta_{ip} = 1 \quad \forall f_i \in F \tag{17}$$

Algorithm 3 The Flow Measurement Rule Generation Strategy of TMMF

```
Input: Flow set $F_m$ and path set $P$.
Output: The sets of rules $\bar{R}$ for the SDN switches.
for each node $s \in V_{SDN}$ do
    add the default routing rules in node $s$ to $R_s$
end for
for each flow $f_i \in F_m$ do
    $p \leftarrow$ the path $p$ for carrying flow $f_i$.
    $u \leftarrow$ the measurement node of path $p$.
    $v \leftarrow$ the next node of node $u$ on path $p$.
    $r_{old} \leftarrow$ the rule matching flow $f_i$ in set $R_s$
    $r_{new} \leftarrow r_{old}$
    $r_{new}.priority \leftarrow r_{new}.priority + 1$
    $r_{new}.src\_prefix \leftarrow f_i.src\_prefix$
    $r_{new}.action \leftarrow$ forward_to_node_$v$.
    $R_u \leftarrow R_u \cup r_{new}$
end for
for each node $s \in V_{SDN}$ do
    $\bar{R} \leftarrow \bar{R} \cup R_s$
end for
return $\bar{R}$
```

$$\sum_{f_i \in F_i} \sum_{p \in P_i} \gamma_{ip}^{v} \delta_{ip} \leq m_v \quad \forall v \in V_{SDN} \tag{18}$$

$$\frac{\sum_{f_i \in F} \sum_{p \in FP_i} \eta_{ip}^{l} \delta_{ip}}{c_l} \leq t \quad \forall l \in L \tag{19}$$

Algorithm 4 The GA Based Traffic Routing optimization Algorithm

```
Input: The network topology $G(V, L)$, flow set $F$, and candidate
    path sets for the flows.
Output: The set of selected paths $\bar{P}$ for the flows.
randomly generate $M_p$ chromosomes stratifying the TCAM
    capacity constraint, and add the chromosomes to vector $Pop$.
$iter_{num} \leftarrow 0$.
$i_A \leftarrow \mu_A \times M_p$, $i_B \leftarrow (\mu_A + \mu_B) \times M_p$ .
while $iter_{num} \leq I_{max}$ do
    evaluate the fitness of the chromosomes in $Pop$.
    sort the chromosomes in $Pop$ in increasing order of their
    fitness value.
    class $A \leftarrow Pop[1, i_A]$.    // $Pop[1, i_A]$ denotes the elements
    of vector $Pop$ index from 1 to $i_A$.
    class $B \leftarrow Pop[i_A + 1, i_B]$.
    class $C \leftarrow Pop[i_B + 1, M_p]$.
    $k \leftarrow 0$.
    while $k \leq (M_p - i_B)$ do
        select a parent $p1$ from class $A$.
        select a parent $p2$ from class $A \cup B$ ($p2 \neq p1$).
        add the parents pair $(p1, p2)$ to set $Par$.
        $k++$.
    end while
    for each pair of parents $(p1, p2) \in Par$ do
        generate a child $c$ by performing crossover and mutation
        operations (Algorithm 5).
        add $c$ to set $O$.
    end for
    replace chromosomes of $Pop$ in class $C$ by the children in
    $O$.
    $iter_{num}++$
end while
$c_{best} \leftarrow$ the best chromosome in $Pop$.
add paths represented by the chromosome $c_{best}$ to set $\bar{P}$.
return $\bar{P}$
```

Comparing with the MILP model used in TMMF, the MILP model used in the TEF is more complex since it has much more variables and constraints. The MILP model used in TEF needs to decide the paths for all flows, while the MILP model used in the TMMF only need to select paths for large flows in F_m, which are determined by **Algorithm 2**. In real networks, the number of flows in F is much larger than the number of large flows in F_m [34]. Furthermore, the number of feasible paths for the MILP model used in TEF is also larger than that for the MILP model used in TMMF. We cannot get optimal solutions for the MILP model used in TEF within an acceptable time in our simulations. Therefore, to efficiently solve the problem, TEF uses the Genetic Algorithm (GA) [38] to search a good solution for the traffic routing optimization problem.

The GA derives from the principles of natural selection and evolutionary theory. The GA mainly involves the following steps:

1) Represent a solution as a chromosome

2) Randomly generate an initial population of solutions.

3) Evaluate the fitness of the solutions and select a portion of solutions called parents to breed a new generation.

4) Generate children from selected parents by crossover and mutation operations.

The GA will return the best solution when the termination criterion is reached. Based on the above description, the GA based traffic routing optimization algorithm is given in Algorithm 4. For the traffic routing optimization problem, a chromosome is represented as a vector $c = [c_1, c_2, \cdots, c_i, \cdots c_N]$, where gene c_i ($c_i \leq K$) is an integer to denote that flow f_i selects the c_ith candidate path from P_i. Namely, a chromosome in the GA algorithm represents a routing strategy for all flows. A chromosome is feasible if the routing strategy represented by the chromosome satisfies the TCAM capacity constraints of SDN switches. Initially, we randomly generate M_p feasible chromosomes. To eliminate the infeasible chromosomes and keep good feasible chromosomes in the evolution process, the fitness of a chromosome c is evaluated using the cost function defined as follows.

$$f(c) = t(c) + max\{\sum_{v \in V_{SDN}} (\bar{m}_j - m_j), 0\}, \tag{20}$$

where $t(c)$ and $\bar{m}_j$ are the maximum link utilization and the number of TCAM entries required for realizing the routing strategy represented by the chromosomes c, respectively. We note that if $\sum_{v \in V_{SDN}} (\bar{m}_j - m_j) > 0$, the chromosome is not feasible (violate the TCAM capacity constraint). The fitness value of a feasible chromosome is the maximum link utilization implementing the routing strategy represented by the chromosome, and the fitness value of an infeasible chromosome is the summation of the maximum link utilization and the total number of TCAM entries still required for realizing the routing strategy represented by the chromosome. Clearly, the infeasible chromosomes have higher fitness values than the feasible chromosomes.

To inherit good chromosomes, we first sort the chromosomes in increasing order of their fitness values, the population

Algorithm 5 The Crossover and Mutation Operations

```
Input: The parents $p1$ and $p2$, $\tau$, and $p_m$.
Output: A child $c$ of $p1$ and $p2$.
for each gene $g = 1, 2, \cdots, N$ do
    generate a random number $r_m$ between 0 and 1.
    generate a random number $r_c$ between 0 and 1.
    if $r_m \leq p_m$ then
        generate a random number $k$ between 0 and $K$.
        $c[g] = k$.
    else if $r_c \leq \tau$ then
        $c[g] = p1[g]$
    else
        $c[g] = p2[g]$
    end if
end for
return $c$
```

is divided into classes: the first $\mu_A \times M_p$ chromosomes (class A), the next $\mu_B \times M_p$ chromosomes (class B), and the remaining chromosomes (class C). μ_A and μ_B represent the proportions of class A and class B chromosomes in the population, respectively, and they are respectively set to 0.2 and 0.4 in this paper. For the parent selection, one parent is chosen form class A, and the other parent is selected from class $A \cup B$. In each generation, we choose $(1-\mu_A-\mu_B)\times M_p$ pairs of parents, and each pair of parents generates a child. To create the next generation, we directly promote all chromosomes in classes A and B, and replaces all chromosomes in class C by the children generated by the selected parents. With these design principles, the genes of chromosomes with lower fitness values have the higher probability of being inherited to the next generation, and the good chromosomes are retained.

The crossover operations are done on the selected parents. Let parameter τ be a real number between 0.5 and 1, which determines whether a gene of a child is inherited from parent $p1$ (selected from class A) or parent $p2$ (selected from class $A \cup B$). To avoid falling into the local optimal solution and diversify the solutions, a mutation operation is performed on each child. The mutation operation simply modifies the value of a gene to a random integer between 1 and K. Each gene of a chromosome is mutated with probability p_m, which is set to 0.01 in our simulations. The details of crossover and mutation operations are shown in Algorithm 5. In our implementation, the generational process of GA is repeated over N_i generations.

(2) Traffic Measurement Optimization

In the above traffic routing optimization problem, TEF selects a feasible path for each flow. Let p_i be the path selected for flow f_i, v_i be a SDN node on path p_i, and $NH(v_i, p_i)$ be the next hop of node v_i on path p_i. We know that if link $(v_i, NH(v_i, p_i))$ is not on the shortest path from node v_i to the destination node of flow f_i, a new rule is required at node v_i, and thus the flow f_i can be directly measured at node v_i. Note that there may still have available TCAM entries at SDN switches after implementing the traffic routing optimization. For this case, the available TCAM entries can be used to measure large flows that are not directly measured. Algorithm 6 shows the details of flow rule generation strategy of TEF.

Algorithm 6 The Flow Measurement Rule Generation Strategy of TEF

```
Input: Network topology $G(V, L)$ and the set of flows $F$.
Output: The rule sets $R$ for the SDN switches.
$R \leftarrow \emptyset$
for each node $s \in V_{SDN}$ do
    add the routing rules in node $s$ to $R_s$
end for
estimate the flow sizes based on the statistics of the rules generated by MLRF strategy (Algorithm 1)
find the feasible candidate path sets $P$ for the flows.
select a path $p_i$ for each flow $f_i \in F$ using Algorithm 4
for each flow $f_i \in F$ do
    $d \leftarrow$ destination node of flow $f_i$.
    for each node $v_i \in p_i$ do
        if node $v_i$ is a SDN node and link $(v_i, NH(v_i, p_i))$ is not on the shortest path from node $v_i$ to node $d$ then
            $r_{old} \leftarrow$ the rule matching flow $f_i$ in set $R_s$
            $r_{new} \leftarrow r_{old}$
            $r_{new}.priority \leftarrow r_{new}.priority + 1$
            $r_{new}.src\_prefix \leftarrow f_i.src\_prefix$
            $R_s \leftarrow R_s \cup r_{new}$
            $F_m \leftarrow F_m \cup f_i$
        end if
    end for
end for
if there still have available TCAM entries in SDN switches then
    $F \leftarrow F/F_m$
    sort the flows in set $F$ according to their estimated sizes in decreasing order
    construct the auxiliary bipartite graph $G_A(V_A = V_F \cup V_S, L_A)$, based on the estimated flow sizes and the routes of the flows
    find a maximum weight matching $M$ on $G_A(V_A = V_F \cup V_S, L_A)$
    for each link $(v_i, v_s^j) \in M$ do
        $r_{old} \leftarrow$ the rule matching flow $f_i$ in set $R_s$
        $r_{new} \leftarrow r_{old}$
        $r_{new}.priority \leftarrow r_{new}.priority + 1$
        $r_{new}.src\_prefix \leftarrow f_i.src\_prefix$
        $R_s \leftarrow R_s \cup r_{new}$
    end for
end if
for each node $s \in V_{SDN}$ do
    $R \leftarrow R \cup R_s$
end for
return $R$
```

VI. Performance Evaluation

A. Simulation Setup

Network topologies and dataset: We use two well known real network topologies: Geant [40] (23 nodes and 37 links) and Abilene [39] (12 nodes and 15 links). We assume only a subset of nodes are deployed with SDN switches. The nodes with the higher degree have higher priority to deploy as SDN switches. If there is a tie, the nodes are ordered arbitrarily. Unless specified, the number of SDN switches in Geant and Abilene is set to 6 (6/23 ≈ 24%) and 4 (4/12 ≈ 33%), respectively. We assume that the number of available TCAM entries (m) is the same for all of the SDN switches. Since the IP prefixes assigned to each node are unknown, we randomly select a set of IP prefixes from IP prefixes owned by China Telecom for each node. The number of prefixes assigned to each node is uniformly distributed in the range [2, 5]. The traffic matrices of Geant and Abilene for a specific time period are publicly available. We randomly choose 100 TMs from the dataset, and we use X^i to denote the ith TM. The TMs provide the traffic sizes between nodes in the networks. However, in our simulation, we need fine-grained TMs, which provide the traffic sizes between the prefixes. To get the fine-grained TMs, we use the following equation:

$$|f_i| = |f_{sd}| \cdot \frac{len(f_i.src\_prefix)}{\sum\limits_{pref \in Pre_s} len(pre)} \cdot \frac{len(f_i.dst\_prefix)}{\sum\limits_{pre \in Pre_d} len(pref)}, \quad (21)$$

where $|f_{sd}|$ denotes the size of aggregated flow between nodes s and d (given in the dataset), $len(\cdot)$ operator returns the length of an IP prefix, and Pre_s and Pre_d denote the set of prefixes owned by nodes s and d, respectively. In the simulations, we use r to represent the flow aggregation ratio, which is defined as the ratio between the number of total available TCAM entries and the number of flows, i.e., $r = m \cdot \frac{|V_{SDN}|}{N}$. Since many TCAM entries are used to configure the flow rules for a variety of network management and operation tasks, a limited number of available TCAM rules in each switch are used to measure flows. Thus, the flow aggregation ratio is low (varies from 0.1 to 0.2) in the simulations. The MILP model used in TMMF is solved by CPLEX.

Performance Metrics: The metrics used in our performance evaluation are defined in equation (22). Normalized Mean Absolute Error (NMAE) is widely used performance metric for measuring the accuracy of TM estimation. In addition, TMMF and TEF can also be used for Heavy Hitter (HH) detection. So to evaluate the effectiveness of using TMMF and TEF for HH detection, we use the average probability of detection (P^d_{HH}) and average probability of false alarm (P^{fa}_{HH}) defined in equation (4), where θ is a pre-determined threshold. To evaluate the TE performance of the joint optimization algorithms, we use maximum link utilization (MLU).

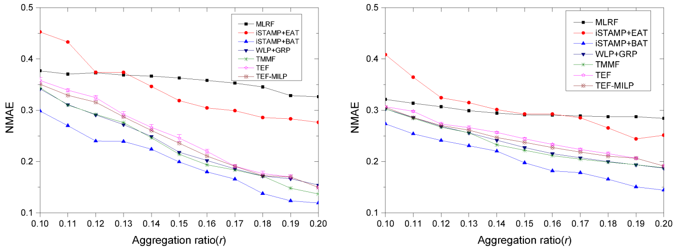
$$\begin{aligned}
NMAE &= \frac{1}{M}\sum_{i=1}^{M}\frac{|X^i - \hat{X}^i|}{|X^i|} \\
P^d_{HH} &= \frac{1}{M}\sum_{i=1}^{M} pr(\hat{X}^i \geq \theta | X^i \geq \theta) \\
P^{fa}_{HH} &= \frac{1}{M}\sum_{i=1}^{M} pr(\hat{X}^i \geq \theta | X^i < \theta) \\
MLU &= arg\max_{l \in L}\{c_l | l \in L\}
\end{aligned} \quad (22)$$

where M is the number of evaluated traffic matrices.

B. Simulation Results

(1) The TMM Performance

In the simulations, we compare TMMF and TEF with iSTAMP+EAT (iSTAMP with EAT) [5], iSTAMP+BAT (iSTAMP with BAT) [5], and WLP+GRP [22], where EAT (Exponential Aggregation Technique) and BAT (Block Aggregation Technique) are two different aggregation matrix design strategies used in iSTAMP [5]. In BAT, each TCAM entry aggregates an equal number of flows. While in EAT, more TCAM entries are allocated to larger flows by adjusting parameters ρ and σ [5]. We use the same setting for parameters ρ and σ as [5], i.e., $\rho = 1$ and $\sigma = 5$. Although the traffic measurement strategies generated by EAT and BAT may be infeasible for implementation in practice, EAT and BAT can also be viewed as performance benchmarks for our proposed algorithms in the aspect of TMM. To improve TMM accuracy under TCAM capacity constraint, WLP-GRP also selects large flow to take the direct measurement. However, different from TMMF and TEF, WLP-GRP uses weighted linear prediction method to predict the flows that need to be directly measured. Moreover, to show the efficiency of TMMF and TEF, we also compare TMMF and TEF with TEF-MILP , which also uses the TE first strategy, but gets the optimal routing solution by solving the MILP model (Eq.(16)-Eq.(19)) instead of using GA. In the TEF algorithm, the initial chromosomes are randomly generated. Thus, to reduce the impact of randomness on the simulation results, we show the average results of 100 runs for each simulation of TEF algorithm. In order to show the stability of the TEF algorithm, we also show the 95% confidence intervals for the results of TEF algorithm.

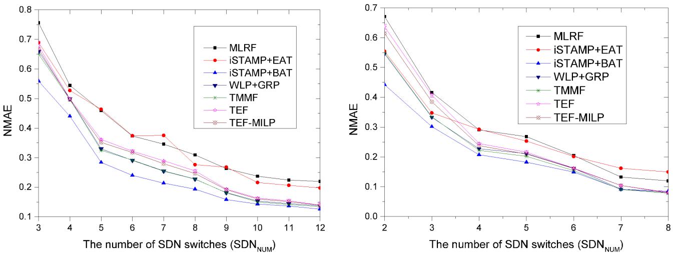


(a) Geant Network (b) Abilene Network

Fig. 5: NMAE in Geant and Abilene topologies when r varies

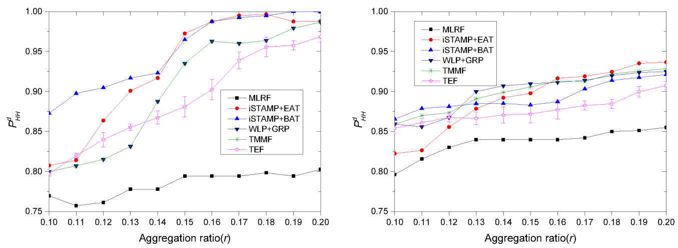
Fig. 5 shows the NMAEs of MLRF, iSTAMP+EAT, iSTAMP+BAT, WLP+GRP, TMMF, TEF, and TEF-MILP under different aggregation ratios. From Fig. 5, we can observe that as expected, the NMAEs of the seven methods decrease with the increase of the flow aggregation ratio r. Most importantly, we can observe that the NMAEs of TMMF and TEF are much better than iSTAMP+EAT and are very close to those of iSATMP+BAT (the differences are within 0.05). These results demonstrate that TMMF and TEF can generate feasible traffic measurement rules that can achieve high TM estimation accuracy. We note that in all cases, the NMAEs of WLP+GRP and TMMF are slightly lower than the NMAEs of TEF. This is because WLP+GRP and TMMF directly measure more large flows than TEF under the TCAM capacity constraint. Moreover, we also can see that WLP+GRP and TMMF almost have the same performance in terms of NMAE, but WLP+GRP does not consider the TE objective. This verifies that TMMF has good TM measurement performance. Furthermore, we also can see that MLRF has much higher NMAE than TMMF and TEF. However, MRLF is a simple algorithm with low computational complexity and it can provide useful information as the first-stage estimator for TMMF and TEF. Although, TEF-MILP performs slightly better than TEF in terms of NMAE, however, the average running time of TEF-MILP is 10 times longer than that of TEF.

To evaluate the impact of the number of deployed SDN switches (SDN_{Num}) on the traffic matrix measurement, we conduct simulations under different number of SDN switches. Since the capacity of TCAM is very limited, the flow ag-



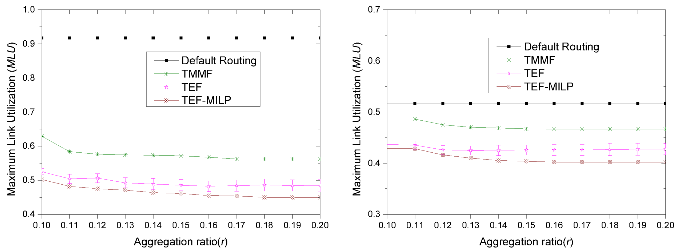
(a) Geant Network (b) Abilene Network

Fig. 6: NMAE in Geant and Abilene networks when SDN_{Num} varies



(a) Geant Network (b) Abilene Network

Fig. 7: P^d_{HH} in Geant and Abilene networks when r varies



(a) Geant Network (b) Abilene Network

Fig. 8: MLU in Geant and Abilene networks when r varies

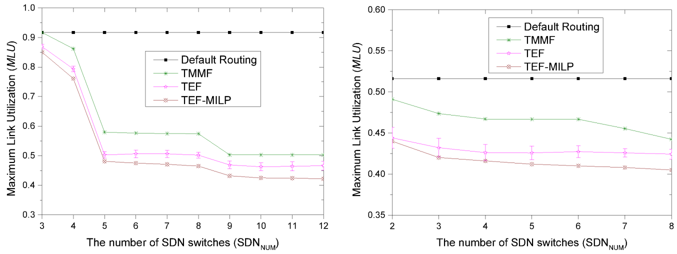
gregation ratio is low in real networks. In order to evaluate the performance of our proposed approaches under low flow aggregation ratio, the number of TCAM entries in each SDN switch of GEANT and Abilene networks is set to 110 and 75, respectively. Under this setting, the flow aggregation ratio of Geant network is about 0.2 when 50% of the nodes are SDN-capable. In Fig. 6, the NMAEs of all the algorithms decrease quickly with the increasing number of deployed SDN switches. When 50% of the nodes are SDN-capable ($r = 0.2$), the NMAEs of TMMF and TEF are about 0.1 and 0.15 for both Geant and Abilene, respectively. This demonstrates that even if a small number of SDN switches are deployed in the network, the TM estimation accuracy can be significantly improved. The reason is that both the number of flows going through SDN nodes and the flow aggregation ratio increase with the number of deployed SDN nodes. This implies more large flows can be directly measured as the number of SDN nodes increases. As shown in Fig. 6, we also can observe that the NMAEs of TMMF and TEF are very close to the NMAEs of iSATMP+BAT, which demonstrates TMMF and TEF can achieve satisfactory TM measurement performance under the flow aggregation constraint. In addition, similar to the results in Fig. 5, TMMF and TEF (ILP) also perform slightly better than TEF in term of NMAE under most cases.

HH detection is important for traffic engineering and network security. The proposed MLRF, TMMF, and TEF can also be applied for HH detection. Fig. 7 presents the effectiveness of MLRF, iSTAMP+EAT, iSTAMP+BAT, TEF and TMMF for detecting HHs in Geant and Abilene networks. In the simulations, the threshold θ defined in Eq. (22) is set as 15% of the size of the largest flow in a TM. From Fig. 7, we can see that both TEF and TMMF can achieve very high probability of detection. For example, even when the flow aggregation ratio r is 0.1, the P^d_{HH} of TEF and TMMF is higher than 0.85, and when the flow aggregation ratio is 0.2, the P^d_{HH} of TEF and TMMF can be higher than 0.95 in Geant network. Note that the P^d_{HH} of MLRF is high (higher than 0.75) even when the flow aggregation ratio is low (e.g., $r = 0.1$). This implies that we can identify large flows for TEF and TMMF using the rules generated by MLRF. We also have the probability of false alarms P^{false}_{HH} for TMMF and TEF, and the P^d_{HH} for TEF and LFF are negligible (less than 0.001). To save space, we do not show the results here.

(2)The TE Performance

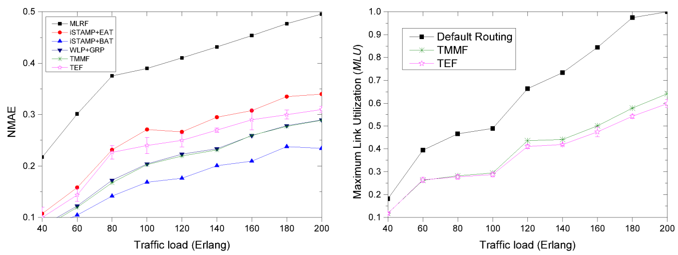
The TMMF and TEF optimize the TE objective by adjusting the routes for some large flows with the available TCAM entries of SDN switches. The TE objective considered in this paper is the maximum link utilization MLU. We assume that the flows follow the default routing if their routes are not adjusted by TEF and TMMF. Fig. 8 shows the MLU of default routing, TMMF, TEF and TEF-MILP in Geant and Abilene networks. As shown in Fig. 8, the MLU of TMMF, TEF and TEF-MILP decrease with the increase of flow aggregation ratio. This is because higher flow aggregation ratio means more TCAM entries can be used to adjust the routes of large flows. Most importantly, we observe that the MLU of TMMF, TEF and TEF-MILP is much lower than that of default routing, which indicates that TMMF, TEF and TEF-MILP can significantly improve the TE performance. In all cases, TEF can achieve lower MLU than TMMF. The results are determined by the different strategies used by TEF and TMMF. In TEF, the available TCAM entries are first used to improve the TE objective by adjusting the routing of some larges flow. Accordingly, TEF can achieve better TE performance than TMMF. As expected, TEF-MILP performs better than TEF in terms of MLU, and the performance gap between TEF-MILP and TEF is less than 5%. However, the running time of TEF-MILP is at least 10 times longer than that of TEF and TMMF. This demonstrate that TEF and TMFF can efficiently find good solutions. The simulation results in Abilene network also follow the similar trend as in Geant network.

Fig. 9 shows the MLU of the algorithms under the different number of SDN switches. In these simulations, the number of TCAM entries in each SDN switch for the GEANT and Abilene networks are set to 110 and 75, respectively. This figure shows that by increasing the number of SDN switches (SDN_{Num}), the MLU of TEF and TMM is reduced significantly. For instance, in Geant network, the MLU of TEF and TMM is higher than 0.8 when $SDN_{Num} = 3$, while the MLU of TEF and TMM are lower than 0.6 when $SDN_{Num} = 5$.



(a) Geant Network (b) Abilene Network

Fig. 9: MLU in Geant and Abilene networks when SDN_{NUM} varies



(a) Geant Network (b) Abilene Network

Fig. 10: NMAE in Geant and Abilene networks when SDN_{Num} varies

The reason is that by increasing SDN_{Num}, both the available TCAM resource and the number of large flows going through SDN nodes increase, indicating that the routes of more large flows can be optimized. We can see that similar to the results in Fig. 8, TEF also performs better than TMMF under all cases since TEF uses a prior load-balancing with available TCAM resource. Since TEF-MILP can obtain optimal routing solution by solving a MILP model, TEF-MILP performances better than TEF. However, the performance gap between TEF-MILP and TEF is less than 5% in all cases.

TEF and TMMF jointly optimize the traffic routing and traffic measurement strategies based on the estimated TMs. However, the estimated TMs inevitably have errors (see Fig. 5 and Fig. 6). To evaluate the impact of traffic matrix estimation error on the TE optimization performance, we use Mean Relative Error (MRE) of MLU as the performance metric. Let MLU_i and $M\hat{L}U_i$ denote the maximum link utilization of the algorithms by using real TM i and estimated TM i as inputs, respectively.The MRE of MLU is defined as:

$$MRE_{MLU} = \frac{1}{S}\sum_{i=1}^{S}\frac{|MLU_i - M\hat{L}U_i|}{MLU_i} \times 100\%, \qquad (23)$$

In all cases, the $M\hat{R}E_{MLU}$ of TEF and TMMF are very low (lower than 4%). The results reveal that the measured traffic matrices are sufficient for TEF and TMMF to implement routing optimization.

(3) Dynamic Traffic Scenario

At last, we compare the performance of the algorithms under dynamic traffic scenario, where flows randomly arrive and depart the network. We assume that the flows arrive in a Poisson process with the mean rate of λ_a, and the flow durations follow the negative exponential distribution with mean rate $\frac{1}{\mu_d}$. Thus, the traffic load is $\frac{\lambda_a}{\mu_a}$. The flow sizes are randomly generated according to equation (21), and the source and destination prefixes are also randomly selected. Fig. 10 plots the NMAE and MLU of the algorithms under different traffic load. We know that as the traffic load increases, more flows will share the limited number of TCAM entries. Thus, in Fig. 10, we can see that the NMAE and MLU of the algorithms increase with traffic load. Similar to the results in Fig. 5 and Fig. 6, TMME and TEF have almost the same performance in terms of NMAE in all cases, and TMMF has better TMM performance than TEF, while TEF achieves better TE performance than TMMF. Furthermore, we also can observe that as the traffic load increases, the TMM performance gap between TMM and TEF reduces, and the TE performance gap between TMM and TEF increases. This implies that we can use TMMF and TEF in low and high traffic load scenarios, respectively.

We also evaluated the algorithms on a larger synthetic topology (50 nodes and 80 links) generated using Barabási-Albert model. The aggregated TM between network nodes is generated using the gravity model, and the fine-grained TM (about 20,000 flows) is also generated using Eq. (21). The simulation results in the synthetic topology also show the similar trend as in Geant and Abilene ISP topologies. However, due to space limitation, we do not show the detailed results in this paper.

In summary, by jointly optimizing the traffic routing and traffic measurement strategies, TEF and TMMF can improve the TMM accuracy and the TE objective by efficiently utilizing the available TCAM entries in each SDN node. The simulation results verify that TEF and TMMF are promising approaches for realizing online TMM and TE.

VII. Conclusion

In this paper, we studied how to jointly optimize the TMM and TE strategies jointly under the TCAM capacity and flow aggregation constraints in SDN networks. To describe and solve the joint optimization problem, we first formulated this problem as a MILP model. Then to provide an accurate initial TM for the joint optimization problem, we proposed an efficient traffic measurement rule generation strategy called MLRF. MLRF can generate more flow rules to provide informative measurements for the TM estimation problem. Also, to efficiently solve the joint optimization problem in large-scale networks, we proposed two heuristic algorithms, namely, TMMF and TEF. TMMF and TEF allocate TCAM entries and generate flow rules for TMM and TE tasks by jointly considering the TMM and TE optimization objectives. At last, extensive simulation results on real network topologies and traffic traces verified that both TMMF and TEF can significantly improve the performance of TMM and TE under very limited TCAM resource. The simulation results also show the effectiveness of MLRF.

References

[1] X. Wang, Q. Zhang, J. Ren, S. Xu, S. Wang, and S. Yu, Toward efficient parallel routing optimization for large-scale SDN networks using GPGPU, vol. 113, pp.1-13, 2018.

[2] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu. Spatio-temporal compressive sensing and Internet traffic matrices (extended version), *IEEE/ACM Transactions on Networking*, vol. 20, no. 3. pp. 662-676, 2012.

[3] A. D'Alconzo, I. Drago, A. Morichetaa, M. Mellia, and P. Casas, Survey on big data for network traffic monitoring and analysis, *IEEE Transactions on Network and Service Management*, Early Access, 2019.

[4] 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 Communicaiton Review*, vol. 38, no. 2, pp. 69-74, 2008.

[5] M. Malboubi, L. Wang, C. Chuah, and P. Sharma, Intelligent SDN based traffic (de)aggregation and measurement paradigm (iSTAMP), *in proceedings of IEEE INFOCOM*, 2014.

[6] M. Moshref, M. Yu, R. Govindan, and A. Vahdat, DREAM: dynamic resource allocation for software-defined measurement, *in proceedings of ACM SIGCOMM*, 2014.

[7] C. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer. Achieving high utilization with software-driven WAN, *in proceedings of ACM SIGCOMM*, 2013.

[8] S. Jain, A. Kuma, S. Mandal, et al. B4: Experience with a globally-deployed software defined WAN, *in proceedings of ACM SIGCOMM*, 2013.

[9] S. Vissicchio, L. Vanberer, O. Bonaventure, Opportunities and research challenges of hybrid software defined networks, ACM CCR, vol. 44, no. 2, 2014.

[10] D. Levin, M. Canini, S. Schmid, F. Schaffert, A. Feldmann, Panopticon: reaping the benefits of incremental SDN deployment in enterprise networks, *USENIX ATC*, 2014.

[11] K. Poularakis, G. Iosifidis, G. Smaragdakis, and L. Tassiulas, One step at a time: optimizing SDN upgrades in ISP networks, *IEEE INFOCOM*, 2017.

[12] S. Yeganeh, A. Tootoonchian, and Y. Ganjali, On scalability of software-defined networking, *IEEE Communication. Magazine*, vol. 51, pp. 136 141, 2013.

[13] Z. Ullah, M. K. Jaiswal, and R. C. C. Cheung, Z-TCAM: an SRAM-based architecture for TCAM, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 2, pp.402-406.

[14] Barefoot. 2018. https://www.barefootnetworks.com/.

[15] X. Jin, H. H. Liu, R. Gandhi, S. Kandula, R. Mahajan, M. Zhang, J. Rexford, and R. Wattenhofer, Dynamic scheduling of network updates, ACM SIGCOMM, 2014.

[16] M. Yu, L. Jose, and R. Miao. Software defined traffic measurement with OpenSketch, *in proceedings of NSDI*, 2013.

[17] M. Moshref, M. Yu, and R. Govindan. Resource/accuracy tradeoffs in software-defined measurement. *in proceedings of ACM HotSDN*, 2013.

[18] L. Jose, M. Yu, and J. Rexford, Online measurement of large traffic aggregates on commodity switches, *in proceedings of USENIX Hot-ICE*, 2011.

[19] A. Tootoonchian, M. Ghobadi, and Y. Ganjali, OpenTM: traffic matrix estimator for OpenFlow networks, *in proceedings of PAM*, 2010.

[20] Y. Yu, C. Qian, and X. Li, Distributed and collaborative traffic monitoring in software defined networks, *in proceedings of HotSDN*, 2014.

[21] Y. Gong, X. Wang, M. Malboubi, S. Wang, S. Xu, and C. Chuah, Towards Accurate Online Traffic Matrix Estimation in Software-Defined Networks, *in proceedings of ACM SOSR*, 2015.

[22] C. Liu, M. Malboubi, and C. N. Chuah, OpenMeasure: adaptive flow measurement & inference with online learning in SDN, *INFOCOM Workshop*, 2016.

[23] Zhiming Hu and Jun Luo, Cracking network monitoring in DCNs with SDN, *in proceedings of IEEE INFOCOM*, 2015.

[24] Y. Li, R. Mao, C. Kim, and M. Yu. FlowRadar: A better netFlow for data centers, *in proceedings of NSDI*, 2016.

[25] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, A. Vahdat. Hedera: dynamic flow scheduling for data center networks, *in proceedings of NSDI*, 2010.

[26] T. Benson, A. Anand, A. Akella, and M. Zhang. MicroTE: fine grained traffic engineering for data centers, *in proceedings of ACM CoNEXT*, 2011.

[27] S. Agarwa, M. Kodialam, and T. V. Lakshman. Traffic engineering in software defined networks, *in proceedings of IEEE INFOCOM*, 2013.

[28] J. He and W. Song. Achieving near-optimal traffic engineering in hybrid software defined networks, *in proceedings of IFIP Networking*, 2015.

[29] Y. Guo, Z. Wang, X. Yin, X. Shi, and J. Wu, Traffic engineering in SDN/OSPF hybrid network, *in proceedings of IEEE ICNP*, 2014.

[30] W. Wang, W. He, and J. Su. Enhancing the effectiveness of traffic engineering in hybrid SDN, *in proceedings of IEEE ICC*, 2017.

[31] M. Caria, A. Jukan, and M. Hoffmann. A performance study of network migration to SDN-enabled traffic engineering, *in proceedings of IEEE Globecom*, 2013.

[32] H. Xu, J. Fan, J. Wu, C. Qiao, and L. Huang, Joint deployment and routing in hybrid SDNs, *in proceedings of IEEE/ACM IWQoS*, 2017.

[33] L. Zhao, J. Hua, X. Ge, and S. Zhong. Traffic engineering in hierarchical SDN control plane, *in proceedings of IEEE/ACM IWQoS*, 2017.

[34] L. Yuan, C. Chuah, and P. Mohapatra. ProgME: towards programmable network measurement, *IEEE/ACM Transactions on Networking*, vol.19, no. 1, pp.115-128, 2011.

[35] Y. Wang and Z. Wang, Explicit routing algorithms for Internet traffic engineering, *in ICCCN*, 1999.

[36] N. P. Katta, J. Rexford, and D. Walker, Incremental consistent updates, *ACM HotSDN*, 2013.

[37] J. Y. Yen, Finding the K Shortest Loopless Paths in a Network, Management Science, vol. 17, no. 11, pp. 712-716.

[38] D. Whitley, A genetic algorithm tutorial, *Statistics and Computing*, vol. 4. pp. 65-85, 1994.

[39] Abilene traffic, http://www.cs.utexas.edu/ yzhang/research/AbileneTM.

[40] Geant network, http://totem.info.ucl.ac.be/dataset.html.

Xiong Wang is an associate professor in the school of information and communication at the University of Electronic and Science of China, Chengdu, China. His research interests include network measurement, modeling and optimization, algorithm analysis and design, network management in communication networks.

Qi Deng is a master student in the school of information and communication at the University of Electronic and Science of China, Chengdu, China. His research interests include network monitoring and data mining.

MehdiMalboubi received the M.Sc. degree in computer science and the Ph.D. degree in electrical and computer engineering from the University of California at Davis. His main research interests include networking, communications, signal/image processing, machine learning, deep-learning, artificial intelligence, applied mathematics and statistics, and data analysis/mining.

Jing Ren is a lecturer in the school of information and communication at the University of Electronic and Science of China, Chengdu, China. Her research interests include network architecture and protocol design, information-centric networking, and software-defined networking.

Sheng Wang is a professor in the school of information and communication at the University of Electronic Science and Technology of China, Chengdu, China. His research interests include planning and optimization of wire and wireless networks, next generation of internet, and next-generation optical networks.

Shizhong Xu is a professor in the school of information and communication at the University of Electronic Science and Technology of China, Chengdu, China. His research interests include Internet of Things, next-generation network, and network science.

Chen-Nee Chuah Chen-Nee Chuah is a Professor in Electrical and Computer Engineering at the University of California, Davis. She received her B.S. in Electrical Engineering from Rutgers University, and her M. S. and Ph.D. in Electrical Engineering and Computer Sciences from the University of California, Berkeley. Her research interests include Internet measurements, network management, and applying data and network science techniques to online social networks, security detection, digital healthcare, and intelligent transportation systems. She was a recipient of the NSF CAREER Award and was named a Chancellors Fellow of UC Davis in 2008. She has served as an Associate Editor for IEEE/ACM Transactions on Networking and IEEE Transactions on Mobile Computing. Chuah is a Fellow of the IEEE and an ACM Distinguished Scientist.